Logic Engineering

The Case of Description and Hybrid Logics

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-5256051
fax: +31-20-5255101
e-mail: illc@science.uva.nl

Logic Engineering

The Case of Description and Hybrid Logics

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof.dr. J.J.M. Franse
ten overstaan van een door het college voor promoties ingestelde
commissie in het openbaar te verdedigen in de
Aula der Universiteit
op donderdag 12 oktober 2000, te 10.00 uur

door

Carlos Eduardo Areces

geboren te Buenos Aires, Argentinië.

Promotor: Prof.dr. J.F.A.K. van Benthem
Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

Co-promotor: dr. M. de Rijke
Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

Institute for Logic, Language and Computation
Faculty of Science
Universiteit van Amsterdam

`e-mail:carlos@wins.uva.nl`
`http://www.illc.uva.nl/~carlos`

Cover design by Carlos Areces.
Printed and bound by Print Partners Ipskamp.

ISBN: 90–5776–053–3

Contents

Acknowledgments

'How do you know I'm mad?' said Alice.
'You must be,' said the Cat,
'or you wouldn't have come here.'

from "Alice's Adventures in Wonderland," Lewis Carroll

Writing the words I'm writing now, is the end of a long trip, long in different ways. Obviously, it is long when measured in *time*, starting on the 25th of December of 1996, the day I took the plane from Buenos Aires and ending on the 15th of July of 2000, the day I submitted this thesis to the graduation committee. And even longer than it seems when measured in *kilometers*, as it involves a one year stop at the University of Warwick, England where I started my doctoral work with a grant from the British Council. Only after the first year, we (my supervisor Maarten de Rijke and me) moved to the University of Amsterdam, The Netherlands, where I completed my thesis with a PhD position at the Institute of Logic, Language and Computation.

But most important of all, the trip is long when we measure it in *changes*: the new things I've learned and the new people I've met. I take these changes as a treasure of experiences which I will always carry with me, marking everything I will do from now on. My PhD gave me the opportunity to meet different people, visit new places and do things I've never done before. And many of the people that accompanied me in this trip deserve a place here, in the first pages of my thesis.

I want to thank many, for many reasons. As I said, the most important reason of all is the time they shared with me, in person and in thought, and the strength they lent me with this shared time. But there are also many other reasons, too many to list all of them below. For each of them I will mention at least some of the reasons, some of the marks and memories they have left with me.

I start by dedicating this thesis to my family. I thank my parents Aníbal and Esther, for teaching me, without me even realizing, how to do things by myself and how to care for people, how to be happy, and how we can be far away and still together; for the many mails and the many letters; and for all the time we missed each other. I thank Aita for years of tales, sweets, kisses and gifts; for teaching me how to swim, how to ride a bike, and how to skate even though she didn't know how to do any of the three; and for thinking all the time the best of her three grandchildren and always being proud of us. This thesis is for Pablo, for being the best friend in addition to be the best brother. For Alejandra for being the mother of Felipe. For Felipe, for bringing so much happiness, in such a small package. And I thank Graciela for an example of how to enjoy life, and Alejandro for loving Gra.

This thesis is also for my family here, Maarten, Zsofie and Daan. We adopted each other a long time ago, when they visited Argentina. I carried Daan in my arms through the streets of Buenos Aires, and Maarten carried me in his, through the streets of Logic. Zsofie gave me and my family all her love, which is huge. Maarten taught me how to think, how to struggle with a proof and how to write a paper. Daan taught me how to read a Dutch poem. They have been with me during the year in England and during the time in Amsterdam, and I know I will miss them immensely the day we will not be close anymore. But as with my family in Argentina, I also know that we will always be able to be together even if far away.

Another family has played an important role in this thesis. I've forgoten already when I first met Patrick Blackburn, nowadays it is like if I knew him and his family for life. I want to thank Patrick for his examples of how to do Logic and how to teach Logic. Thanks for transmitting to me a spark of his burning passion on Hybrid Logics. Thanks to the whole Blackburn family for making me feel really at home while visiting them. Thanks to Claire for introducing me to new, interesting connections between my work and Linguistics. Thanks to Jennifer and Gabrielle for playing with me. My joint work with Maarten and Patrick is among the most happy times in these doctoral years.

My friends in Argentina have been always with me through these years, bearing without complain (well, not many complains and well deserved in any case), the long periods of silence on my side, when I was too deep in some project to have time for answering their letters. Thanks for receiving me each time I was back in Buenos Aires with a display of parties, dinners and excursions to which I was invariably late. Thanks for making me feel as if I never left. I thank Analía, for a month that will last a life; Gabriela, for telling me off when I stayed too late working in the office; Fer, for so many years of company; Verónica, for teaching me and learning from me and always, always being there; Natalia, for a sun; Bárbara, for a poem and more; Pablo, for a mail each Friday; Edu and Angelita, for a visit and the fun; Ricardo, for a visit and a talk; Gustavo, for a visit and a CD; Dan, for walking with me the first steps; Adrio, for the patience with my always delayed answers to her mails; Vale, for joining the group; Juan Martín, for coming over; Alejandro and Iris, for a spot in the Alps ($\ddot\smile$); Roberto, for a nickname; and GrAcE, for a surprise.

This thesis is dedicated also to my friends everywhere else. To Raffaella, for making me happy so many times and sad some; to Smaro, for her love; to Aleka, for the way she laughs, her voice in the phone and the rides back home with a smile in my face; to Els, for a house and her friendship; to Karianne, for sharing some happy moments with me; to Marco A., for the Italian movies; to Renata, for many parties and a question; to Eva, for a tango lesson and a question; to Nikos, Joost, Jon, Christof, Shai, Cigal, Alex, Maria, Paola, Richard, Willemijn, Rosella, Mónica, Rosalie, Alessandra and Riccardo, Caterina, Christina, Elena and Jelle, for the company along the way (and I hope we will keep walking together), to Rafael, Catarina, Hanno and Benjamin, for letting me learn from teaching (Borges said once that "No se puede enseñar algo sin aprender mucho," so thanks for the opportunity); to Marco V., for the role playing; to Gert-Jan, for a Christmas card; to Carsten and Stephan, for the discussions; and to Ingrid, Marjan, Marco and Peter for all the help.

A special thanks goes to Eva, who wanted to read my thesis; to Wiet Bouma, for introducing me to an interesting problem; and to Valentin Goranko, for the possibility to discover a great person, a beautiful family and a wonderful continent. I hope we will keep our work together for many years to come.

I should thank immensely my professors, the people who at one time or another took me under their wing and were interested in my work and ideas. Besides the many things about Logic proper I learned from them, they have taught me that the field is full of brilliant people which in addition have a wish to share knowledge, discover new ideas and learn more. Learn in a symmetric way, where the students learn from the teacher, and the teacher from the students. For this I deeply thank Carlos Alchourrón, Verónica Becher, Patrick Blackburn, Roberto Cignoli, Dick de Jongh, Maarten de Rijke, David Makinson, Maarten Marx, Francisco Naishtat, Hans Rott, and Johan van Benthem.

Finally, I thank Maarten de Rijke, my supervisor, for all the things he taught me, not all of them concerning Logic. Thanks for putting me on a path and letting me walk, deciding by myself where I really wanted to go, but always being there if I needed directions. Thanks for the many corrections to the many things we wrote together. Thanks for always been ready with advice on any possible problem (from house hunting to references on modal interpolation). Thanks for reading carefully through the thesis, and for not getting mad at my dislike of conclusions.

Carlos Eduardo Areces
Amsterdam, The Netherlands
July, 2000

Abstract

As the title indicates, there are two levels involved in the research carried out in this thesis: the general issue of understanding (and promoting) Logic Engineering, together with a detailed study of its particular instantiation for Description and Hybrid Languages.

For some years now, a trend has been developing in the field of computational logic: given the wide diversity of applications the field has advanced into (theorem proving, software and hardware verification, computational linguistics, knowledge representation, etc.), a multiplicity of formal languages has been developed, offering a wealth of alternatives to classical languages. With the advantages of the diversity of choice, comes its complexity. How do we decide what the best formalism is for a given reasoning or modeling task? Or even more, what are the important rules to take into account when designing yet another formal language? How do we compare, how do we measure, how do we test? These are the questions that the young field of Logic Engineering is supposed to investigate and, if possible, answer.

What we know about Logic Engineering is still not a lot, and as yet there are no general answers to these questions. Don't expect to find a list of "recipes" of how things should be done here. But much can be learned from analyzing in detail a particularly interesting case. This will be the main thrust of the work carried out in the thesis.

Description logics are a family of formal languages used for structured knowledge representation. They have been designed as a tool to describe information in terms of concepts and their interrelation (definitions), together with means to specify that certain elements of the domain actually fit such definitions (assertions). In addition, they provide a formal notion of inference in terms of this structured knowledge. Description logics constitute the best example we are aware of, of a broad, homogeneous collection of formal languages with a clearly specified semantics (in terms of first-order models) devised to deal with particular applications. They offer an assortment of specialized inference mechanisms to handle tasks like knowledge classification, structuring, etc. The complexity of reasoning in the different languages of this family has been widely investigated, theorem provers effectively deciding some of the most expressive languages have been implemented (and they are among the fastest provers for non-classical languages available), and these languages have been successfully applied in many realistic problems, even at an industrial level. Connections between description languages and modal

logics have been investigated, but a unifying logical background theory explaining their expressive power and logical characteristics was largely missing. This is the role to be played by hybrid logics.

Hybrid languages are modal languages extended with the ability to explicitly refer to elements in the domain of a model. They were first introduced in the mid 1960s, in the field of temporal logic, and were subsequently developed mainly in a purely theoretical environment. The work in the field focused on investigating complete axiomatizations for these languages, characterizing their meta-logical properties and understanding their semantic and proof-theoretical behavior.

Hybrid languages provide the exact kind of expressive power required to match description languages. Having been optimized for applications, description logics are difficult to handle with classical model- and proof-theoretical tools, but given the close match between description and hybrid logics we will be able to apply these techniques to the hybrid logic counterpart of description logics instead. Going in the other direction, description logics provide hybrid logics with extensively tested examples of useful languages, knowledge management lore, and implementations. In this thesis we will draw these two complementary fields together and investigate in detail what each of them has to offer to the other. Given that the two areas have developed different techniques and evolved in divergent directions, “trading” between them will be especially fruitful. Description logics can export reasoning methods, complexity results and application opportunities; while hybrid logics have their model-theoretical tools, axiomatizations and analyses of expressive power to offer.

The particular aim of this thesis is, then, to explore and exploit the connections between description and hybrid logic, their similarities and differences. The main results we will present specifically concern this issue. But we hope to take the first steps in setting and discussing this work in the wider perspective of logic engineering, and provide a small contribution to the general issue of better understanding the rules behind the good design of new formal languages.

The thesis is organized in four parts. In the first, containing Chapter 1, we discuss different ways of identifying interesting fragments (and fragments of extensions) of first-order logic. We argue that traditional methods, like prenex normal form and finite variable fragments, are not completely satisfactory. We propose, instead, to capture relevant fragments *via translations*. The semantics of many formal languages (including modal, description and hybrid languages) can be given in terms of classical logics, and as such they can be considered fragments of classical languages. But now, these fragments come together with an extremely simple presentation — modal languages, for example, are usually introduced as *extensions of propositional logic* — and with novel and powerful proof- and model-theoretical tools (simple tableaux systems, elegant axiomatizations, fine-grained notions of equivalence between models, new model-theoretical constructions, etc). Modal-like logics in general, and description and hybrid logics in particular, will be presented as examples of useful fragments identified in such a way.

Part II introduces both description and hybrid logics (in Chapters 2 and 3 respectively) providing the necessary background and the basic notions which will be used in the rest of the thesis. The chapters can be read independently and serve as introductions to the kinds of methods and results which have been developed in these areas. They also provide a detailed guide to the literature. As we make clear in our presentation, description and hybrid logics are closely related, and their connections are spelled out in Chapter 4. We start by presenting already known embeddings of description languages into converse propositional dynamic logics, and discussing why they provide a less satisfactory match than the one obtained through hybrid languages. In particular, we highlight that two ingredients are needed for a successful embedding: the ability to refer to elements in the domain of a model, and the ability to make statements about the whole model from a local point. The first ingredient is needed to account for assertions, the second to account for definitions. Both are provided, in an elegant and direct way, by hybrid languages in the form of nominals, the satisfiability operator and the existential modality. We also clarify the relation between local and global notions of consequence, the first being the standard notion of consequence for hybrid (and in general modal) languages while the second is predominant in the description logic community.

After providing two-way satisfaction preserving translations between description and hybrid logics (Theorems 4.5 and 4.7), we explore the transfer of results. We show how the embedding into hybrid languages provides sharp upper and lower complexity bounds (Theorems 4.8 and 4.9), separations in terms of expressive power and characterizations (Theorem 4.14), and meta-logical properties like interpolation and Beth definability (Theorem 4.15). Concerning interpolation and Beth definability, to the best of our knowledge this is the first time that such results have been investigated in connection with description languages. Many of these results are obtained from the general theorems we will prove in Part III. We also discuss how results from description logics can fill important gaps which have not yet received attention in the hybrid logic community. Some examples are the known complexity bounds concerning description logics with counting operators, or the PSpace results when certain syntactic restrictions are imposed on the existential modality.

Part III of the thesis contains the core technical work. In Chapter 5 we show how ideas from description and hybrid logics can be put to work with benefit even when the subject is purely modal. In particular, aided by the notions of nominal/individual, we define well behaved direct resolution methods for modal languages. This example shows how the additional flexibility provided by the ability to name states can be used to greatly simplify reasoning methods. We proceed to build over the basic resolution method and obtain extensions for description and hybrid languages. In Chapters 6 and 7 we take a hybrid logic perspective as we dive into model-theoretical issues. But we have already demonstrated in Chapter 4 how hybrid logic results shed their light on description languages.

In Chapter 6 we turn to expressive power. We start by considering $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$, a very expressive hybrid language. The two main results concerning this language are Theorems 6.10 and 6.27. The first theorem provides a five fold characterization of the first-order formulas equivalent to the translation of a formula in $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. In particular, it identifies this fragment as the set of formulas which are invariant for generated submodels. Theorem 6.27 shows that the arrow interpolation property not only holds in this language, but also for any system obtained from $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ by the addition of pure axioms. In a more general perspective, the results in Chapter 6 show that $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is surprisingly well behaved in model-theoretical terms. As we discuss in this chapter, it can be characterized in many different and natural ways, it responds with ease to both modal and first-order techniques, and possess one of the strongest versions of the interpolation and Beth properties we are aware of for modal languages. For these reasons, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ can be used as a "logical laboratory:" what we learn from it using the plethora of techniques it offers, can provide us, in many cases, with intuitions on restrictions and extensions. We see this process in action throughout the chapter, as we are able to transfer certain results from $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ to extensions and sublanguages.

In Chapter 7 we discuss complexity. We start with an excursion into undecidability and we prove that a small fragment of $\mathcal{H}_{\mathsf{S}}(\downarrow)$ already has an undecidable local satisfiability problem (Theorem 7.1). This is a hint that only very severe restrictions on the $\downarrow$ binder will bring us back into decidability. We show in Theorem 7.10 that if we restrict ourselves to sentences of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$, where $\downarrow$ appears non-nested, decidability is regained. In Chapter 4 we have already shown that even this restricted use of binding proves interesting from a description logic perspective. We then turn to weaker languages (without binders) which remain closer to standard description languages. In Theorem 7.15 we prove that the addition of nominals and the satisfiability operator to the basic modal language $\mathbf{K}$ does not modify its complexity, while it greatly increases its expressive power. Interestingly, the same is not true when we extend the basic temporal language $\mathbf{K}_t$: the addition of just one nominal increases the complexity of the local satisfiability problem to ExpTime (Theorem 7.18), when the class of all models is considered. But usually temporal languages are interpreted on models where the accessibility relation is forced to adopt a "time-like" structure, the two best known cases being strict linear orders (linear time) and transitive trees (branching time). We prove in Theorems 7.27 and 7.29 that over these classes of models, complexity is tamed and again coincides with the complexity of the basic temporal language.

Part IV contains our conclusions and directions for further research. Here we highlight some of the lessons we have learned during the research presented in this thesis. As we said, we cannot hope yet for general answers concerning logic engineering, but we can proceed by analogy: the same questions we posed and answered for description and hybrid logics can be tested on other formal languages, and we have presented tools and methodologies (bisimulations, model construction and comparison games, translations, etc.) which are powerful and versatile enough to be useful in many diverse situations.

Samenvatting

Zoals de titel al aangeeft, zijn er twee niveaus te onderscheiden in het onderzoek dat in deze dissertatie wordt uitgevoerd: het betreft de algemene kwestie aangaande het begrijpen (en bevorderen) van Logic Engineering, tesamen met een gedetailleerde uitwerking hiervan voor het concrete geval van Descriptieve en Hybride Talen.

Over de afgelopen jaren tekent zich de volgende trend af op het gebied van de computationele logica: dankzij de diversiteit van de verschillende toepassingen waar dit gebied zich mee bezig houdt (automatische stelling bewijzers, software en hardware verificatie, computationele linguistiek, kennis representatie, etc.) is een verscheidenheid aan formele talen ontwikkeld die een schat aan alternatieven bieden voor klassieke logica. Maar zoals altijd, de voordelen van het kunnen kiezen komen samen met de moeilijkheid te moeten kiezen. Hoe nu kunnen we beslissen wat het beste formalisme is voor een gegeven redeneer- of modeleertaak? Of sterker nog, door welke regels zouden we ons het beste kunnen laten leiden bij het ontwerpen van een nieuwe formele taal? Hoe kunnen we vergelijken, hoe kunnen we meten, hoe kunnen we testen? Dit zijn het soort vragen dat het pasontgonnen gebied van Logic Engineering onderzoekt en, waar mogelijk, beantwoordt.

Op dit moment is er nog niet zo veel bekend over Logic Engineering, en er zijn dan ook nog geen algemene antwoorden op bovenstaande vragen. Om de verwachtingen enigzins te temperen: ook hier zal geen lijst van recepten worden gegeven van hoe de zaken dienen te worden aangepakt. Maar we kunnen wel veel leren door één interessant geval grondig te analyseren. Zo een analyse vormt het hoofdbestanddeel van deze dissertatie.

Descriptieve logicas zijn een familie van formele talen die gebruikt worden voor gestructureerde kennis representatie. Ze zijn ontwikkeld als een hulpmiddel om informatie te beschrijven in termen van concepten en hun onderlinge verbanden (definitions) samen met middelen om te specificeren dat bepaalde elementen van een domein feitelijk aan die definities voldoen (assertions). Verder leveren descriptieve logicas een formele notie van afleiding in termen van deze gestructureerde kennis. Zo ver als ons bekend, vormen descriptieve logicas het beste voorbeeld van een brede, homogene collectie van formele talen met een duidelijk gespecificeerde semantiek (in termen van eerste orde modellen) die daarnaast ontwikkeld zijn met het oog op een bepaalde toepassing; d.w.z, elke descriptieve logica bestaat uit een gespecialiseerd afleidingsmechanisme waarmee een specifieke taak vervuld kan worden. Hieronder vallen taken als de classificering van kennis, of de structurering van kennis. De complexiteit van het redeneren in de verschillende talen binnen deze familie is grondig onderzocht en er zijn stelling bewijzers geïmplementeerd die een aantal van de meest expressieve talen op een effectieve manier kunnen beslissen (deze behoren tot de snelste stelling bewijzers die er op dit moment zijn voor niet-klassieke talen). Descriptieve talen zijn toegepast op vele realistische problemen, zelfs in de industrie, en met succes. Er is onderzoek gedaan naar verbanden tussen descriptieve talen en modale logicas, maar wat ontbreekt is een unificerende logische theorie die het verschil in uitdrukkingskracht en het verschil in logische kenmerken verklaart. Deze rol zal worden vervuld door hybride logicas.

Hybride talen zijn modale talen die verder nog de mogelijkheid hebben expliciet te refereren naar elementen van het domein van een model. Deze talen zijn halverwege de jaren zestig voor het eerst geïntroduceerd binnen het gebied van temporele logicas, en zijn nadien voornamelijk ontwikkeld in deze theoretische omgeving. Hier werd de aandacht gericht op het onderzoeken van volledige axiomatiseringen voor deze talen, het karakteriseren van hun meta-logische eigenschappen, en het begrijpen van hun semantisch en bewijstheoretisch gedrag.

Hybride talen bieden precies het soort uitdrukkingskracht dat nodig is om een interessante tegenhanger te vormen voor descriptieve talen. Hiermee bedoelen we dat er niet alleen een sterke overeenkomst bestaat tussen hybride talen en descriptieve talen, maar, en dat is het interessante hieraan, dat op een bepaalde manier deze twee gebieden elkaars tegenhanger vormen; ze lopen in verschilende richtingen uiteen en hebben verschillende technieken ontwikkeld. Zo zijn descriptieve talen, doordat ze zozeer geoptimaliseerd zijn met het oog op een specifieke toepassing, moeilijk te bestuderen met klassieke model- en bewijstheoretische hulpmiddelen. Hybride talen echter lenen zich daar uitstekend toe. Anderzijds, descriptieve talen zijn, anders dan hybride talen, uitgebreid getest op hun bruikbaarheid, ze zijn in de praktijk ingezet bij kennis management, en geïmplementeerd. Juist daarom kan een "ruilhandel" tussen hen buitengewoon vruchtbaar zijn. Om in de handel-metafoor te blijven (we zijn niet voor niets in Amsterdam), descriptieve logicas hebben redeneer methoden te verhandelen, als ook complexiteits resultaten en toepassings mogelijkheden. Exportprodukten van hybride logicas zijn hun modeltheoretische hulpmiddelen, hun axiomatiseringen en hun analyses van uitdrukkingskracht.

Het specifieke doel van deze dissertatie is nu om in detail de verbanden tussen descriptieve en hybride logica te bestuderen en te benutten; zowel hun verschillen als de overeenkomsten. De belangrijkste resultaten die we zulen presenteren betreffen specifiek dit onderwerp. Behalve dat deze resultaten op zichzelf genomen interessant genoeg zijn, dragen ze ook bij aan de algemene kwestie die een logica engineer bezig houdt: ze geven een beter begrip van de richtlijnen volgens welke een formele taal ontworpen zou moeten worden.

De dissertatie is onderverdeeld in vieren. In het eerste deel, dat bestaat uit Hoofdstuk 1, bespreken we verschillende manieren waarop interessante fragmenten van eerste orde logica (en fragmenten van uitbreidingen van eerste orde logica) kunnen worden geïdentificeerd. We zullen beargumenteren dat traditionele methoden, zoals prenex normaal vorm en eindige variabelen fragmenten, niet geheel en al tot tevredenheid stemmen. In plaast daarvan stellen wij voor relevante fragmenten uit te houwen *via vertalingen.* Van veel formele talen (modale, descriptieve, en hybride talen incluis) kan de semantiek gegeven worden in termen van klassieke logicas, en als zodanig kunnen ze beschouwd worden als fragmenten van klassieke talen. Echter, door deze talen op te vatten als vertalingen, krijgen we een aantal extras: modale logicas bijv. kunnen nu simpel gepresenteerd worden (namelijk als eenvoudige uitbreidingen van *propositionele* logica), en krijgen nieuwe en krachtige bewijs- en model-theoretische hulpmiddelen (eenvoudige tableau systemen, elegante axiomatiseringen, verfijnde noties van equivalentie tussen modellen, nieuwe model-theoretische constructies, etc). We zullen laten zien dat logicas die, ruw gezegd, verwant zijn aan modale logicas, op deze wijze kunnen worden geïdentificeerd. Hieronder vallen descriptieve en hybride logicas.

Deel II introduceert descriptieve en hybride logicas (in de Hoofdstukken 2 and 3 respectievelijk), en levert hiermee de noodzakelijke achtergrond voor de rest van de dissertatie. Deze hoofdstukken kunnen onafhankelijk van elkaar gelezen worden, en dienen ter introductie van het soort methoden en resultaten die in deze gebieden zijn ontwikkeld. Tevens bevatten ze volop verwijzingen naar de literatuur. Zoals in onze presentatie naar voren zal komen, zijn descriptieve en hybride logica nauw verbonden, en deze verbanden zullen nauwkeurig worden omschreven in Hoofdstuk 4. We beginnen dit hoofdstuk met het beschrijven een aantal bekende inbeddingen van descriptieve logicas in converse propositionele dynamische logicas, en behandelen de vraag waarom deze dynamische logicas minder goed bij descriptieve logicas passen dan hybride talen. In het bijzonder zullen we er de aandacht op vestigen dat voor een succesvolle inbedding twee ingrediënten nodig zijn: de mogelijkheid te refereren naar de elementen van een model, en de mogelijkheid om vanuit een locaal punt beweringen te maken die het gehele model betreffen. Het eerste is nodig om *assertions* te verklaren, het tweede voor *definitions*. Beide bestanddelen worden op een elegante en directe manier geleverd door hybride talen in de vorm van nominals, de vervulbaarheids relatie, en de existentiële modaliteit. We zullen ook de relatie tussen de locale en globale notie van gevolgtrekking ophelderen. De eerste is de standaard notie voor hybride talen (en modale talen in het algemeen) terwijl de tweede gebruikelijk is binnen de descriptieve logica.

Nadat we vertalingen hebben gegeven tussen descriptieve en hybride logicas die de vervulbaarheidsrelatie in beide richtingen behouden (Theorems 4.5 en 4.7), onderzoeken we welke resultaten zich nu laten overdragen. Voor descriptieve logicas verkrijgen we op deze manier scherpe boven- en ondergrenzen aan de complexiteit (Theorems 4.8 en 4.9), karakteriseringen (Theorem 4.14), meta-logische eigenschappen waaronder interpolatie en Beth definieerbaarheid (Theorem 4.15), en scheidingen van descriptieve logicas in termen van uitdrukkingskracht. Wat betreft interpolatie en Beth definieerbaarheid merken we op dat dit, zover als ons bekend, de eerste keer is dat dit type vragen onderzocht is voor descriptieve logicas. Veel van bovengenoemde resultaten zijn verkregen via de algemene stellingen die in Deel III zullen worden bewezen. Aan de andere kant zullen we ook bespreken hoe resultaten die bekend zijn over descriptieve logicas belangrijke hiaten kunnen opvullen in onze kennis over hybride logicas aangaande kwesties die nog niet eerder ter discussie zijn gesteld in dit gebied. Om twee voorbeelden te noemen, de welbekende grenzen op de complexiteit van descriptieve logicas met een tellings-operator, en de PSpace resultaten die gelden als bepaalde syntactische restricties aan de existentiële modaliteit worden opgelegd.

Het noodzakelijke technische werk wordt in Deel III afgehandeld. In Hoofdstuk 5 laten we zien hoe zelfs in een zuiver modale context ideeën uit descriptieve logicas en hybride logicas van nut kunnen zijn. Zo definiëren we bijvoorbeeld directe resolutiemethoden voor modale talen met behulp van de noties nominals/individuals. Dit voorbeeld laat zien dat de flexibiliteit die verkregen is uit de mogelijkheid toestanden te benoemen, gebruikt kan worden om redeneer methoden danig te vereenvoudigen. We zullen later deze fundamentele resolutiemethode uitbreiden voor descriptieve en hybride talen. In Hoofdstukken 6 en 7 bekijken we modeltheoretische kwesties vanuit de hybride logica. We hebben dan al laten zien in Hoofdstuk 4 hoe resultaten over hybride logicas inzicht kunnen geven in descriptieve talen.

Hoofdstuk 6 behandelt uitdrukkingskracht. Eerst bestuderen we $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$, een zeer expressieve hybride taal. De twee belangrijkste resultaten over deze taal zijn Stellingen 6.10 en 6.27. De eerste stelling geeft een vijf-voudige karakterisering van de verzameling eerste orde formules die equivalent zijn aan een vertaling van een formule uit $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$. In het bijzonder wordt dit fragment herkend als de verzameling formules die invariant zijn voor gegenereerde submodellen. Stelling 6.27 laat zien dat de →-interpolatie eigenschap opgaat, niet alleen voor deze taal maar ook voor elk ander systeem dat vanuit $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ verkregen is door een willekeurig aantal axiomas toe te voegen. Meer algemeen blijkt dat $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ verrassend mooie model-theoretische eigenschappen heeft. Zoals we in dit hoofdstuk zullen bespreken, kan het gekarakteriseerd worden op vele verschillende en natuurlijke manieren, er kunnnen met gemak zowel eerste orde als modale technieken op worden toegepast, en het bezit een van de sterkste versies van de interpolatie en Beth eigenschappen die er bij ons weten voor modale talen bestudeerd zijn. Dit al was voor ons reden om $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ als een "logisch laboratorium" te beschouwen: wat we in dit laboratorium te weten komen, levert in veel gevallen intuïties over restricties en extensies van deze taal. Dit proces zien we aan het werk in dit hoofdstuk.

In Hoofdstuk 7 richten we ons op complexiteit. We beginnen met een uitstapje richting onbeslisbaarheid en bewijzen dat een klein fragment van $\mathcal{H}_{\mathsf{S}}(\downarrow)$ al een onbeslisbaar lokaal vervulbaarheidsprobleem heeft (Theorem 7.1). Dit wijst erop dat een fragment dat de $\downarrow$-binder bevat slechts beslisbaar is onder zeer sterke restricties. Zo een fragment is $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ waar $\downarrow$ enkel niet-genest voorkomt (Stelling 7.10). In Hoofdstuk 4 hebben we al laten zien dat zelfs dit beperkte gebruik van de binder interessant is vanuit een descriptieve logica perspectief. Vervolgens beschouwen we zwakkere systemen (zonder binder) die dichter tegen de standaard descriptieve talen aanliggen. In Stelling 7.15 wordt bewezen dat het toevoegen van de nominals en de vervulbaarheidsoperator aan de basis modale taal $\mathbf{K}$ de complexiteit van laatstgenoemde niet verhoogd, terwijl het zijn uitdrukkingskracht enorm vergroot. Interessant genoeg is dit niet het geval als we de basis tijdslogica $\mathbf{K}_t$ uitbreiden: de complexiteit van het locale vervulbaarheidsprobleem wordt al verhoogd tot ExpTime door het toevoegen van slechts één nominal (Theorem 7.18), als we de klasse van alle modellen beschouwen.

Gewoonlijk echter worden tijdslogicas geïnterpreteerd op modellen waar de toegankelijkheidsrelatie een structuur heeft die min of meer de tijdsloop zou kunnen verbeelden. De meest bekende zijn de strict lineare ordeningen (lineaire tijd) en de transitieve bomen (splitsende tijd). We bewijzen in Stellingen 7.27 en 7.29 dat als we deze klasse van modellen beschouwen de complexiteit niet uit de hand loopt maar integendeel overeenkomt met de complexiteit van de basis tijdslogica.

Deel IV bevat onze conclusies en geeft de richtingen aan voor verder onderzoek. Zoals gezegd, de tijd is nog niet daar om algemene richtlijnen te geven voor logic engineering, maar we kunnen wel via analogie te werk gaan; dezelfde vragen die wij gesteld en beantwoord hebben voor descriptieve en hybride logicas kunnen aan andere formele systemen worden voorgelegd en de hulpmiddelen en methodes (bisimulaties, spelen ter constructie van modellen en het vergelijken van uitdrukkingskracht, etc) die wij in deze dissertatie besproken hebben, bezitten voldoende potentie om in diverse situaties van nut te kunnen zijn.

Part I
Logic Engineering

ocurre en todos los casos
en que participa la verdad
que se transforma todo sentido aparente
espejo falso
en lugares ciertos entre la nada y el infinito

from "Guitarra Negra," Luis Alberto Spinetta

For many years "Logic" was "Classical Logic," mainly classical first-order logic, and there were good reasons for this. To mention some, first-order logic offers high expressive power, simplicity, good behavior (both syntactically and semantically), and a clean and well developed model theory.

This is just a complicated way of saying that first-order logic is beautiful. . . for many tasks. But when we think about applications requiring *effective inference*, first-order logic is simply *not* the choice: its satisfiability problem — i.e., the problem of determining whether there exists a model in which a given first-order formula is true — is not decidable. In addition, first-order logic sometimes does not measure up to the task at hand. It cannot, for example, capture the fact that one relation is the transitive closure of another one, and this might be crucial for a certain modeling task.

For these reasons mainly, first-order logic has been loosing its privileged position as a representation formalism in many areas where applications requiring effective inference methods are central, such as Artificial Intelligence, Knowledge Representation, Computational Linguistics, Software Design and Verification, or Databases. In these fields, the applications themselves have given rise to new formalisms, specially tailored for the problems to be addressed. In some cases, like in the early days of Artificial Intelligence, this growth has even been chaotic, with hundreds of new proposals, and very restricted means to evaluate them.

As an answer to this problem, a new field of Logic Engineering is starting to develop. To judge the appropriateness of the name, consider the definition of *engineer* [Davidson *et al.*, 1994]

> **engineer:** *n.* one who designs or makes, or puts to practical use [. . .]

In line with its name, Logic Engineering studies ways to construct new formalisms, with good properties like decidability, appropriate expressive power, effective reasoning methods, and good meta-logical characteristics (completeness, interpolation, etc.), for a given, particular need.

How do we design "made-to-fit" logics? That is the topic we will discuss in this part of the thesis.

Chapter 1

Cutting Out Fragments

The logic is invariant,...
but the data are different.
So the results are different.

from "Stranger in a Strange Land," Robert Heinlein

1.1 Looking for the Right Language

The rules of the game are set: we are searching for good formal languages for specific tasks, where by "formal languages" we mean *languages with a precise syntax and semantics*, and by "specific tasks" we mean *reasoning tasks.* In particular, we will focus on reasoning tasks involving *inference* and hence a further condition is required on our notion of formal language, namely it should provide a calculus defining some kind of consequence relation.

We might as well start by looking into why first-order logic (FO) is not necessarily the best choice in all cases. First, it might be too complex for the reasoning task we have to address.

Theorem 1.1. *The satisfiability problem for first-order logic is not decidable.*

If we are in search of effective inference mechanisms, Theorem 1.1 immediately disqualifies first-order logic. Of course, if what we require are languages for specifying properties to *verify* in a given model, i.e. if our problem requires only model checking, then FO enters the list of candidates again. But on the other hand, even with all its expressive power, it is possible that FO just does not provide exactly the needed expressivity. For example,

Theorem 1.2. *First-order logic cannot define the transitive closure of a relation.*

Theorems 1.1 and 1.2 show that there might be different reasons why we need to look for alternatives to FO, in addition they will also let us introduce a number of definitions and notions that will keep coming up as a central theme in the thesis, like the importance of formal syntax and semantics, the use of model-theoretical techniques like reduction arguments and games, comparisons of expressive power, etc. To make things crystal clear, let us start by formally introducing first-order logic.

Definition 1.3. [First-order language] Let $\mathsf{REL} = \{R_1, R_2, \ldots\}$ be a countable set of *relation symbols*, $\mathsf{FUN} = \{f_1, f_2, \ldots\}$ a countable set of *function symbols*, $\mathsf{CON} =$

$\{c_1, c_2, \ldots\}$ a countable set of *constant symbols* and $\mathsf{VAR} = \{x_1, x_2, \ldots\}$ a countable set of *variables*. We assume that REL, FUN, CON and VAR are pairwise disjoint. To each relation symbol $R_i \in \mathsf{REL}$ and each function symbol $f_i \in \mathsf{FUN}$ we associate an arity $n > 0$. We call $\mathcal{S} = \langle \mathsf{REL}, \mathsf{FUN}, \mathsf{CON}, \mathsf{VAR}\rangle$ a *signature*, and we will sometimes focus on *relational signatures* where $\mathsf{FUN} = \{\}$. This is usually not a restriction as we can represent functions as constrained relations.

The *well-formed terms* of the first-order language over the signature $\langle\mathsf{REL}$, FUN, $\mathsf{CON}, \mathsf{VAR}\rangle$ are

$$\mathsf{TERMS} := x_i \mid c_i \mid f_i(t_1, \ldots, t_n),$$

where, $x_i \in \mathsf{VAR}$, $c_i \in \mathsf{CON}$, $f_i \in \mathsf{FUN}$ of arity n and $t_1, \ldots, t_n \in \mathsf{TERMS}$. The *well-formed formulas* over the signature are

$$\mathsf{FORMS} := \top \mid t_1 = t_2 \mid R_i(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x_i.\varphi,$$

where $t_1, t_2, \ldots, t_n \in \mathsf{TERMS}$, $R_i \in \mathsf{REL}$ is an n-ary relation symbol, $\varphi, \varphi_1, \varphi_2 \in \mathsf{FORMS}$ and $x_i \in \mathsf{VAR}$. As usual, we take $\vee, \rightarrow, \leftrightarrow$ and $\forall$ as defined symbols.

Turning to semantics, first-order formulas are interpreted on first-order models.

Definition 1.4. [First-order models and satisfiability] A first-order *model* for a signature $\mathcal{S}$, is a structure $\mathcal{M} = \langle M, \cdot^{\mathcal{M}}\rangle$ where M is a non-empty set and $\cdot^{\mathcal{M}}$ is an *interpretation function* defined over $\mathsf{REL} \cup \mathsf{FUN} \cup \mathsf{CONS}$ such that $\cdot^{\mathcal{M}}$ assigns an n-ary relation over M to n-ary relation symbols in REL, an n-ary function ${}^{n}M \rightarrow M$ to n-ary function symbols in FUN, and an element in M to constant symbols in CONS. When the signature is small, we will simple write $\mathcal{M} = \langle M, \{R_i^{\mathcal{M}}\}, \{f_j^{\mathcal{M}}\}, \{c_k^{\mathcal{M}}\}\rangle$ instead of $\mathcal{M} = \langle M, \cdot^{\mathcal{M}}\rangle$.

An *assignment* g for $\mathcal{M}$ is a mapping $g : \mathsf{VAR} \rightarrow M$. Given an assignment g for $\mathcal{M}$, $x \in \mathsf{VAR}$ and $m \in M$, we define g^x_m (an *x-variant* of g) by $g^x_m(x) = m$ and $g^x_m(y) = g(y)$ for $x \neq y$. Given a model $\mathcal{M}$ and an assignment g for $\mathcal{M}$, the interpretation function $\cdot^{\mathcal{M}}$ can be extended to all elements in TERMS:

$$\begin{array}{rcl} x_i^{\mathcal{M}} & = & g(x_i) \\ f(t_1, \ldots, t_n)^{\mathcal{M}} & = & f^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}). \end{array}$$

Finally the *satisfiability relation* $\models$ is defined as

$$\begin{array}{rcl} \mathcal{M} \models \top[g] & & \text{always} \\ \mathcal{M} \models t_1 = t_2[g] & \text{iff} & t_1^{\mathcal{M}} = t_2^{\mathcal{M}} \\ \mathcal{M} \models R(t_1, \ldots, t_n)[g] & \text{iff} & R^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}}) \\ \mathcal{M} \models \neg\varphi[g] & \text{iff} & \mathcal{M} \not\models \varphi[g] \\ \mathcal{M} \models \varphi_1 \wedge \varphi_2[g] & \text{iff} & \mathcal{M} \models \varphi_1[g] \text{ and } \mathcal{M} \models \varphi_2[g] \\ \mathcal{M} \models \exists x_i.\varphi[g] & \text{iff} & \mathcal{M} \models \varphi[g^{x_i}_m] \text{ for some } m \in M. \end{array}$$

If a given formula φ is satisfied under every assignment for $\mathcal{M}$, we say that φ is *valid* in $\mathcal{M}$ and write $\mathcal{M} \models \varphi$. For a sentence φ, let $Mod(\varphi) = \{\mathcal{M} \mid \mathcal{M} \models \varphi\}$.

A piece of notation now, for a given language $\mathcal{L}$, let $Sat(\mathcal{L})$ denote the satisfiability problem for $\mathcal{L}$ (i.e., the problem of deciding wether a formula of $\mathcal{L}$ is satisfiable), and $Val(\mathcal{L})$ the dual validity problem.

We are now ready for the proof of Theorem 1.1. In 1936, Church put forward a bold thesis: every computable function from natural numbers to natural numbers is recursive in the sense of Herbrand-Gödel-Kleene. He then showed that no recursive function could decide the validity of first-order sentences, and concluded that there was no decision algorithm for $Sat(\mathsf{FO})$ [Church, 1936]. Independently, a different proof was published by Turing a year later [Turing, 1937], by formalizing Turing machines by means of first-order formulas, and reducing an undecidable class of particular word problems for Turing machines to the validity problem of the encoding. But a very simple proof can be given by means of an undecidable tiling or domino problem [Berger, 1966].

Definition 1.5. [The $\mathbb{N} \times \mathbb{N}$ tiling problem] A *tiling system* is a triple $\mathcal{T} = \langle T, H, V\rangle$ where T is a finite set (the set of tiles), and $H, V \subseteq T \times T$ are relations expressing horizontal and vertical compatibility constraints between the tiles.

We say that $\mathcal{T}$ *tiles* $\mathbb{N} \times \mathbb{N}$ if there exists a *tiling function* $t : \mathbb{N} \times \mathbb{N} \to T$ such that for all $(x, y) \in \mathbb{N} \times \mathbb{N}$, whenever $t(x, y) = t_1$, $t(x + 1, y) = t_2$ and $t(x, y + 1) = t_3$ then $H(t_1, t_2)$ and $V(t_1, t_3)$.

In [Berger, 1966] the undecidability of the $\mathbb{N} \times \mathbb{N}$ tiling problem is proved by reducing the halting problem for Turing machines to the problem of deciding whether a tiling function exists, for any given tiling system. Tiling problems are also very useful as a means to establish complexity lower bounds [Chlebus, 1986].

Proof of Theorem 1.1. Consider the following encoding into FO of a given tiling system $\mathcal{T} = \langle\{t_1, \ldots, t_n\}, H, V\rangle$ (from [Gurevich, 1976]). For $h, v, t_1, \ldots, t_n$ unary functions, let $\varphi(x)$ be the conjunction of the following formulas

$$\begin{gathered}
h(v(x)) = v(h(x)), \\
\bigvee\{t_i(x) = x \mid 1 \leq i \leq n\}, \\
\bigwedge\{t_i(x) = x \to \neg(t_j(x) = x) \mid 1 \leq i < j \leq n\}, \\
\bigvee\{t_i(x) = x \land t_j(h(x)) = h(x) \mid H(t_i, t_j)\}, \\
\bigvee\{t_i(x) = x \land t_j(v(x)) = v(x) \mid V(t_i, t_j)\}.
\end{gathered}$$

It is almost trivial to verify that $\forall x.\varphi(x)$ is satisfiable iff $\mathcal{T}$ tiles $\mathbb{N} \times \mathbb{N} \to T$. From right to left, let t be a tiling function, then $\mathcal{M} = \langle \mathbb{N} \times \mathbb{N}, \cdot^{\mathcal{M}}\rangle$, where

$$\begin{array}{rcll}
t_i^{\mathcal{M}}(a, b) & = & (a, b) & \text{if} \quad t(a, b) = t_i \\
t_i^{\mathcal{M}}(a, b) & = & (a + 1, b + 1) & \text{if} \quad t(a, b) \neq t_i \\
h^{\mathcal{M}}(a, b) & = & (a + 1, b) & \\
v^{\mathcal{M}}(a, b) & = & (a, b + 1) &
\end{array}$$

satisfies $\forall x.\varphi(x)$. For the other direction, assume $\mathcal{M} \models \forall x.\varphi(x)$ and let $a \in M$ be fixed but arbitrary, then the function

$$t(n, m) = t_i \quad \text{iff} \quad \mathcal{M} \models t_i(h^n(v^m(x))) = h^n(v^m(x))[a]$$

is a tiling function. QED

As we see from the proof, very little of first-order expressive power is needed to arrive at undecidability. If the language has a countable number of unary function symbols and equality then only one variable is needed, and the formulas are purely universal. As we already said, we can always trade the function symbols used in the encoding for relational symbols, and use the language itself to force them to behave as functions (enforcing totality $\forall x.\exists y.(R(x, y))$ and functionality $\forall x y_1 y_2.((R(x, y_1) \wedge R(x, y_2)) \rightarrow y_1 = y_2))$, but notice that we pay a price on more variables and a more complicated pattern of quantification.

Studying this kind of trade-off is a long standing project of classical logic, and attempts to map decidable and undecidable fragments of FO are varied [Ackermann, 1954; Lewis, 1979; Börger *et al.*, 1997]. We will elaborate on this point in Section 1.2, when we analyze different fragments of FO.

The discussion so far may be summarized by saying that FO is, in some cases, *too expressive*: it allows us to encode undecidable problems. At the same time it is often not expressive enough. A typical example when expressivity beyond first-order logic is called for, is the need to refer to the transitive closure of a relation in a model. As we stated in Theorem 1.2, first-order logic lacks the expressive power required to define the transitive closure of a relation. For the proof of the theorem we will introduce another important tool: a way to compare models.

Suppose we want to prove that a certain property P is not expressible in first-order logic. One way of showing this is to exhibit two first-order models, which agree on all first-order sentences, but such that P is true in one and false in the other. This argument can be refined further: suppose we are able to prove that for any $k \in \mathbb{N}$ there are two first-order models $\mathcal{M}_k$ and $\mathcal{N}_k$ such that they agree *on all first-order sentences up to quantifier rank k* but they disagree on the truth value of P. This is enough, as any first-order sentences defining P will have a certain fixed finite quantifier rank.

Ehrenfeucht [1961] established a strong correlation between the truth of sentences of quantifier rank k in a pair of models, and the rounds of a model comparison game. Let us define things formally. The notion of partial isomorphism is one of the crucial ingredients.

Definition 1.6. [Partial isomorphism] Let $\mathcal{M}$ and $\mathcal{N}$ be first-order models over a relational signature $\mathcal{S}$. Let $p : M \rightarrow N$ be a partial map. p is said to be a *partial isomorphism* from $\mathcal{M}$ to $\mathcal{N}$ iff

i. p is injective,
ii. $\forall c \in \mathsf{CONS}$, $c^{\mathcal{M}} \in dom(p)$ and $p(c^{\mathcal{M}}) = c^{\mathcal{N}}$,
iii. $\forall R \in \mathsf{REL}$, and $a_1, \ldots, a_n \in dom(p)$, $R^{\mathcal{M}}(a_1, \ldots, a_n)$ iff $R^{\mathcal{N}}(p(a_1), \ldots, p(a_n))$.

Notice that, in general, partial isomorphisms do not preserve the validity of formulas with quantifiers as isomorphisms do. But interestingly, they allow us to introduce a notion of *extensions* which will match the restriction to a fixed quantifier rank. This will become clear in Definition 1.8 below. First, we introduce the game-theoretical perspective, which is more intuitive in many cases.

Definition 1.7. [Ehrenfeucht games] Let $\mathcal{M}$ and $\mathcal{N}$ be first-order models over the same relational signature. The *Ehrenfeucht game* $G_k(\mathcal{M}, \mathcal{N})$ is played by two players called the *Spoiler* and the *Duplicator*. Each player has to make k moves in the course of a play. The players take turns. In his i-th move the Spoiler first selects a structure, $\mathcal{M}$ or $\mathcal{N}$, and an element in this structure. If the Spoiler chooses e_i in $\mathcal{M}$, then the Duplicator in his i-th move chooses an element f_i in $\mathcal{N}$. If the Spoiler chooses f_i in $\mathcal{N}$, then the Duplicator chooses an element e_i in $\mathcal{M}$. The Duplicator *wins* the game if $\bar{e} \mapsto \bar{f}$ is a partial isomorphism from $\mathcal{M}$ to $\mathcal{N}$. Otherwise the Spoiler wins. We say that one of the players has a *winning strategy* in $G_k(\mathcal{M},\mathcal{N})$, if it is possible for him to win each play whatever choices are made by the opponent.

Ehrenfeucht games provide a natural characterization of elementary equivalence up to rank k. If Duplicator has a winning strategy for the game $G_k(\mathcal{M},\mathcal{N})$, then $\mathcal{M}$ and $\mathcal{N}$ agree on all sentences up to quantifier rank k (notation $\mathcal{M} \equiv_k \mathcal{N}$). The advantage of Ehrenfeucht games is that the existence of a winning strategy for Duplicator is often *easy to grasp*. Their disadvantage is that sometimes, arguments proving this are *hard to describe.*

Instead, Fraïssé [1954] provided a purely algebraic characterization by means of k-back-and-forth systems.

Definition 1.8. [k-back-and-forth systems] Let $\mathcal{M}$ and $\mathcal{N}$ be two first-order models on a relational signature. A *k-back-and-forth system* is a sequence $(I_j)_{j\leq k}$ with the following properties:

i. Every I_j is a non-empty set of partial isomorphisms from $\mathcal{M}$ to $\mathcal{N}$.

ii. (Forth property) $\forall j < k$, $p \in I_{j+1}$, and $m \in M$ there is $q \in I_j$ extending p and $m \in dom(q)$.

iii. (Back property) $\forall j < k$, $p \in I_{j+1}$, and $n \in N$ there is $q \in I_j$ extending p and $n \in ran(q)$.

We say that $\mathcal{M}$ and $\mathcal{N}$ are *k-isomorphic* (notation $\mathcal{M} \cong_k \mathcal{N}$) if there is a k-back-and-forth system between them.

The pieces are put together in the following result.

Theorem 1.9. [Ehrenfeucht-Fraïssé Theorem] *Given two first-order models $\mathcal{M}$ and $\mathcal{N}$ on a relational signature, and $k \in \mathbb{N}$, the following are equivalent*

i. Duplicator has a winning strategy in $G_k(\mathcal{M},\mathcal{N})$.

ii. $\mathcal{M} \cong_k \mathcal{N}$.

iii. $\mathcal{M} \equiv_k \mathcal{N}$.

We have now all the machinery we need to embark on the proof of Theorem 1.2 (from [Ebbinghaus and Flum, 1999]).

Proof of Theorem 1.2. We want to show that the relation $TC(R) = R^+$ of transitive closure is not first-order definable. We start by proving a simpler claim. A first-order model over the relational signature $\langle\{R\},\{\},\{\},\{\}\rangle$, where R is binary is called a (simple, possibly infinite) *graph.* We say that a class C of models in this signature is first-order definable if there exists a first-order sentence φ such that $\mathsf{C} = Mod(\varphi)$. We say that a graph $\langle M, R\rangle$ is *connected* if any two different elements in M are related in R^+, the transitive closure of R.

Claim 1.10. The class CONN of connected graphs is not first-order definable.

Proof of Claim. For $l \in \mathbb{N}$, let $\mathcal{C}_l$ be the graph given by a cycle of length $l+1$:

$$\mathcal{C}_l = \langle \{0, \dots, l\}, \{(i, i+1), (i+1, i) \mid 0 \le i < l\} \cup \{(0, l), (l, 0)\} \rangle.$$

We will show that

$$\text{for every } k \in \mathbb{N}, l \ge 2^k,\ \mathcal{C}_l \equiv_k \mathcal{C}_l \uplus \mathcal{C}_l, \tag{1.1}$$

where $\uplus$ denotes disjoint union. The claim follows from (1.1), because suppose φ defines CONN, and let k be the quantifier rank of φ. As $\mathcal{C}_{2^k}$ is connected, then $\mathcal{C}_{2^k} \models \varphi$, as $\mathcal{C}_{2^k} \equiv_k \mathcal{C}_{2^k} \uplus \mathcal{C}_{2^k}$ then also $\mathcal{C}_{2^k} \uplus \mathcal{C}_{2^k} \models \varphi$, and $\mathcal{C}_{2^k} \uplus \mathcal{C}_{2^k} \in \mathit{Mod}(\varphi) = \mathsf{CONN}$, a contradiction.

We use Theorem 1.9 to prove (1.1). Intuitively, the winning strategy of Duplicator works by picking points which are sufficiently far apart. Whenever Spoiler tries to signal a difference between $\mathcal{C}_l$ and $\mathcal{C}_l \uplus \mathcal{C}_l$ by choosing points in the different connected components of $\mathcal{C}_l \uplus \mathcal{C}_l$, during round j, Duplicator chooses points in $\mathcal{C}_l$ which are at least at a distance 2^j. This is enough for the logic to think of them as unreachable. Formally, for a graph $\mathcal{G}$, let $d^{\mathcal{G}} : G \times G \to \mathbb{N}$ be the length of a minimal path between two elements of $\mathcal{G}$ if such path exists, or ∞ otherwise. Define for $j \in \mathbb{N}$, the "truncated" j-distance function $d_j^{\mathcal{G}}$ by

$$d_j^{\mathcal{G}}(e, e') = \begin{cases} d^{\mathcal{G}}(e, e') & \text{if } d^{\mathcal{G}}(e, e') < 2^{j+1}, \\ \infty & \text{otherwise.} \end{cases}$$

It is easy to verify that $(I_j)_{j \le m}$ is an m-back-and-forth system between $\mathcal{C}_l$ and $\mathcal{C}_l \uplus \mathcal{C}_l$ if $l \ge 2^m$, where $I_j = \{p \mid d_j^{\mathcal{C}_l}(e, e') = d_j^{\mathcal{C}_l \uplus \mathcal{C}_l}(p(e), p(e')),\ \text{for } e, e' \in \mathit{dom}(p)\}$. ⊣

From the claim, it follows that $TC(R)$ is not first-order definable. For suppose it were definable; let $\varphi(x, y)$ be a first-order formula defining $TC(R)$, i.e., $\mathcal{M} \models \varphi[m, m']$ iff $(m, m') \in R^+$. Then CONN can be defined as the class of models $\{\mathcal{M} \mid \mathcal{M} \models \forall xy.(\neg(x = y) \to \varphi(x, y))\}$. QED

After all the work we have done in this section we can happily agree that first-order logic comes with many powerful techniques, which is perhaps one of its strongest points. But these same tools let us establish results like Theorems 1.1 and 1.2, which signal important weaknesses when we have certain kind of applications in mind.

In the chapters to come we will discuss how fragments of FO, or fragments of extensions of FO, can be fine-tuned to provide effective reasoning methods, good meta-logical properties, better complexity results, and the exact expressive power needed.

1.2 Fragments of FO, and Extensions

But, there are continuum many different fragments. Obviously, we don't want to consider all of them. How do we define "nice fragments"? A first possibility is to restrict attention to classes presentable in some fixed way. For example, to consider finite classes (i.e., classes with a finite number of models) would be an option, but these classes might be of little applicability if we are interested in inference.

General syntactic and semantic constraints are probably better suited for the job. For example, we could use the semantic consequence relation: for a sentence α, let $Cn(\alpha)$ be the set of all first-order consequences of α. We can now consider the following problem, given a sentence α, decide whether the decision problem for $Cn(\alpha)$ is decidable (i.e., whether for an arbitrary sentence β it is possible to decide if $\beta \in Cn(\alpha)$). A simple reduction argument shows that this problem is as hard as $Sat(\mathsf{FO})$ itself. Another possibility is to choose a certain class of structures (lattices, rings, ...) and attempt to characterize its first-order theory, but this seems to be more the work of (perhaps Universal) Algebra.

In Logic, the syntactic path has been (much) more prevalent. For a long time, logicians have been interested in classes of formulas defined by simple syntactic restrictions. In [1915], Löwenheim gave a decision procedure for the satisfiability of sentences with only unary predicates. He also proved that sentences with only binary predicates form a reduction class for validity, i.e., this set of sentences is such that there is a recursive function f from first-order sentences into sentences with only binary predicates with the property that φ is valid iff $f(\varphi)$ is. Löwenheim's undecidability result was sharpened by Kalmár in [1936]: one binary predicate suffices. This implies, for example, that the first-order theory of graphs is undecidable.

Even if we restrict ourselves to syntactic ways of cutting out fragments, there are different possibilities.

1.2.1 Prenex Normal Form Fragments

Some of the most familiar fragments of first-order logic are defined by means of restrictions on the quantifier prefix of formulas in prenex normal form. As is standard, we will refer to prenex normal form fragments by using finite strings over $\{\exists, \exists^*, \forall, \forall^*\}$. For example, the string $\forall\exists^*$ represents the class of first-order formulas in prenex normal form, where the quantifier prefix starts with a universal quantifier which is followed by zero or more existential quantifiers.

Theorem 1.1 shows that if functional symbols are allowed in the language, then the simple $\forall$ fragment is already undecidable. As is traditional, in what follows we will only discuss fragments over relational signatures. Here are some of the best known results. In 1920, Skolem showed that $\forall^*\exists^*$ sentences form a reduction class for satisfiability. In 1928, Bernays and Schönfinkel gave a decision procedure for the satisfiability of $\exists^*\forall\exists^*$ sentences. Gödel, Kalmár and Schütte, independently in 1931, 1933 and 1934 respectively, discovered decision procedures for the satisfiability of $\exists^*\forall^2\exists^*$ sentences. In 1933, Gödel showed that $\forall^3\exists^*$ sentences form a reduction class for satisfiability. More recently, Kahr in 1962 proved the undecidability of $Sat(\forall\exists\forall)$.

These results are interesting and useful, as is witnessed by the following immediate application. Suppose we built a formal specification (for example, of a certain mathematical structure) by means of first-order sentences which enumerate the needed properties. If all the formulas used lie inside one of the decidable classes mentioned above, then there exist a completely mechanic way of determining if such a structure exists (i.e., by checking the consistency of the specification).

The collection of similar results was eventually organized in what is known today as the *Classification Problem* for first-order logic:

- Which fragments are decidable for satisfiability and which are undecidable?
- Which fragments are decidable for finite satisfiability (satisfiable in a finite model) and which are undecidable?
- Which fragments have the finite model property and which contain axioms of infinity (that is, satisfiable formulas without finite models)?

Finite models are important because for any fragment F of first-order logic having the finite model property, $\mathit{Sat}(\mathsf{F})$ can be decided provided that we can decide membership of a formula in F. The classification problem for prenex normal form fragments admits a complete finite solution. This follows from the Classifiability Theorem of Gurevich [1969]. Also the complexity, for the cases where the fragment is decidable, is fairly well mapped out (see [Börger *et al.*, 1997] for the most up-to-date account).

In other words, the issue of decidability/undecidability for prenex normal form fragments of FO is pretty much settled. But in other aspects these fragments are not satisfactory at all. In almost all cases, the study of their meta-logical properties is difficult as they don't have a neat model theory. The restrictions are "too syntactical" in nature for issues like axiomatization or semantic characterization to be manageable.

1.2.2 Finite Variable Fragments

Let FO^k be the restriction of first-order logic over a relational signature to formulas that contain only the variables $x_1, \ldots, x_k$, from some fixed enumeration of VAR. Finite-variable fragments of first-order logic were introduced for technical reasons in [Henkin, 1967]. Logics with only a finite number of variables are important in many branches of mathematical logic and its applications, including finite model theory, model checking, database query languages and knowledge representation. Notice that these fragments cannot be "turned into prenex normal form." In the transformation to prenex normal form new variables are needed to push out quantifiers. In fact, the clever reuse of variables is crucial.

Example 1.11. [Börger *et al.*, 1997] Given a graph $\mathcal{G}$, we need $n+1$ variables to characterize a path of length n in $\mathcal{G}$, if we require the formula to be in prenex normal form

$$\exists x_0 \ldots x_n.(\bigwedge_{0 \leq i < n} E(x_i, x_{i+1})),$$

but a formula in FO^2 is enough if we allow reuse

$$\exists x_0 x_1.(E(x_0, x_1) \wedge \exists x_0.(E(x_1, x_0) \wedge \ldots)).$$

The satisfiability problem for FO^k is undecidable, even for formulas without equality, for all $k \geq 3$, given that FO^3 already extends the prefix class $\forall\exists\forall$. That FO^2 (without equality) is decidable was first proved in [Scott, 1962], the result was extended to the language with equality in [Mortimer, 1975].

Theorem 1.12. FO^2 *has the finite model property, hence* $\mathit{Sat}(\mathsf{FO}^2)$ *is decidable.*

In [Grädel *et al.*, 1997a] the complexity upper bound set by Mortimer is improved to an essentially optimal single exponential in the model size, proving $\mathit{Sat}(\mathsf{FO}^2) \in \textsc{NTime}(2^{O(n)})$. This result can be extended to FO^2 with constant symbols, but that seems to be the decidability boundary: the *Sat* problem of FO^2 (even without equality) extended with a single unary function is undecidable. Also, the satisfiability problem for the one variable fragment of FO is undecidable if unary functions are allowed.

[Grädel *et al.*, 1997b; Pacholski *et al.*, 1997; Grädel *et al.*, 1999] provide a detailed analysis of (un)decidability phenomena for FO^2 and extensions. A particularly interesting case (see Section 2.2) is C^2, the extension of FO^2 (with equality) by counting quantifiers $\exists^{\geq m}$ and $\exists^{\leq m}$ for $m \in \mathbb{N}$. It is almost immediate to see that C^2 does not have the finite model property. The formula

$$\forall x.\exists^{=1} y.R(x, y) \wedge \forall y.\exists^{\leq 1} x.R(x, y) \wedge \exists y.\forall x.\neg R(x, y)$$

forces R to be the graph of an injective and not surjective function in the domain, which can only be satisfied on an infinite model. However, $\mathit{Sat}(\mathsf{C}^2)$ is shown to be $\textsc{NExpTime}$-complete if unary coding for numbers in quantifiers is used.

For further results and critical discussions of pros and cons of FO^k, see [Andréka *et al.*, 1998]. This paper shows failure of the Łos-Tarski and Craig interpolation theorems inside finite variable fragments. In [Areces and Marx, 1998], failure of (even weak) interpolation in FO^k for $k \geq 2$ is proved as a corollary of a more general result concerning failure of the interpolation property. A short direct proof is also provided, showing that very little is needed to achieve failure of interpolation in finite variable fragments. Let $\vdash_2$ be the derivation system defined as follows

Ax1 Every propositional tautology is an axiom scheme.
Ax2$_i$ $\forall x_i.(\varphi \rightarrow \psi) \rightarrow (\forall x_i.\varphi \rightarrow \forall x_i.\psi)$, for $i \in \{0, 1\}$.
Ax3 $\forall x_1 x_0.\varphi \rightarrow \forall x_0 x_1.\varphi$.
MP From φ and $\varphi \rightarrow \psi$ infer ψ.
UG$_i$ From φ infer $\forall x_i.\varphi$, for $i \in \{0, 1\}$.

Clearly $\vdash_2$ is sound for first-order logic, but hopelessly incomplete. Trivial validities like $\forall x_0.(x_0 = x_0)$ and $\exists x_0 x_0.\varphi \leftrightarrow \exists x_0.\varphi$ are not theorems of $\vdash_2$.

Theorem 1.13. [Areces and Marx, 1998] *For every k, there exist* FO^2 *formulas* φ *and* ψ *such that*

i. $\varphi \vdash_2 \psi$*, and*
ii. for every FO^k *formula* θ *in the common language of* φ *and* ψ*, either* $\varphi \not\models \theta$ *or* $\theta \not\models \psi$.

These formulas can be algorithmically obtained and have size polynomial in k. Either φ *and* ψ *are in disjoint languages but both contain the equality symbol, or they are equality-free but the common language contains one binary predicate.*

Even though it is difficult to put a finger on it, and in spite of the previous negative results, finite variable fragments seem better suited to our purposes than prenex normal form fragments. At least they seem to have better meta-logical properties than the prefix fragments. For instance, we can use the notion of k-back-and-forth-systems to provide a model-theoretical characterization of FO^k. Call a formula *invariant for k-back-and-forth systems* if and only if whenever there is a k-back-and-forth system linking two models $\mathcal{M}$ and $\mathcal{N}$ then $\mathcal{M} \models \varphi \Leftrightarrow \mathcal{N} \models \varphi$.

Theorem 1.14. [van Benthem, 1995] *A first-order formula φ is equivalent to a formula in FO^k iff it is invariant for k-back-and-forth systems.*

And we can use a version of Ehrenfeucht games, called pebble-games (see [Immerman and Kozen, 1989; Ebbinghaus and Flum, 1999]), to obtain game-theoretical characterizations. Furthermore, modified versions of Łos-Tarski and Craig Theorems do go through (see [Andréka *et al.*, 1998]).

The main disadvantage of finite variable fragments, though, seems to be their coarseness: FO^1 is barely interesting, FO^2 is just fine, but most of its extensions aren't. With respect to decidability, FO^2 is not what Vardi [1997] has called *robustly decidable*, i.e., it does not remain decidable when suitable extensions of the language are considered. Also, k-variable fragments have a poor proof theory. No finitely axiomatized Hilbert style system exists [Monk, 1969], and the complexity of the necessary axiom schemes is inevitably high [Andréka, 1991].

1.2.3 Guarded Fragments

Clearly, the expressive power of FO comes from its quantifiers. In recent years, yet another way of defining fragments of first-order logic and its extensions was introduced by Andréka, van Benthem and Németi [1998], by imposing restrictions on the way a quantifier can be introduced in a formula.

Let GF, the *guarded fragment* of first-order logic, be defined as

- Every relational atomic formula $R(\bar{v})$ or $x_i = x_j$ belongs to GF.
- GF is closed under Boolean operations.
- If $\bar{v}$, $\bar{w}$ are tuples of variables, $\alpha(\bar{v}, \bar{w})$ is an atomic formula and φ is a formula in GF such that all free variables of φ occur in α, then also the formulas $\exists \bar{w}.(\alpha(\bar{v}, \bar{w}) \wedge \varphi(\bar{v}, \bar{w}))$ and $\forall \bar{w}.(\alpha(\bar{v}, \bar{w}) \rightarrow \varphi(\bar{v}, \bar{w}))$ belong to GF.

The intuitions behind GF are as follows: the new variables $\bar{w}$, introduced in $\varphi(\bar{v}, \bar{w})$ by the quantifiers are "bound" to a very simple syntactic condition (the atom α). The *guard* α constrains the part of the model that can be reached by the quantifier, thus making decidable the satisfiability problem of the fragment. These intuitions are very closely related to the techniques of algebraic relativization (see [Monk, 1993]). Notice an important property of the guarded fragment: it restricts neither the pattern of alternation of quantifiers as prenex normal form fragments do, nor the number of variables used as is done in FO^k.

In the same spirit, the *loosely guarded fragment* for first-order logic (LGF) is obtained by replacing the last clause in the definition of GF by a less strict one:

- If $\bar{v}$, $\bar{w}$ are tuples of variables, $\alpha(\bar{v}, \bar{w}) = \bigwedge \alpha_i$ is a conjunction of atoms and φ is a formula in LGF, then also the formulas $\exists \bar{w}.(\alpha(\bar{v}, \bar{w}) \wedge \varphi(\bar{v}, \bar{w}))$ and $\forall \bar{w}.(\alpha(\bar{v}, \bar{w}) \rightarrow$

$\varphi(\bar{v}, \bar{w}))$ belong to LGF, provided that $\mathit{Free}(\varphi) \subseteq \mathit{Free}(\alpha)$, and for any two variables $z \in \bar{w}, z' \in \bar{v} \cup \bar{w}$ there is at least one atom α_i that contains both z and z'.

Grädel [2000], proved that the satisfiability problems for GF and LGF is complete for 2Exptime. Even further, the satisfiability problem for *guarded fixed point logic* ($\mu\mathsf{GF}$) is also complete for 2Exptime [Grädel and Walukiewicz, 1999], where $\mu\mathsf{GF}$ is obtained by adding to the definition of GF the clause

– Let R be a k-ary relation variable and let $\bar{v} = v_1, \ldots, v_k$ be a k-tuple of distinct variables. Let $\varphi(R, \bar{v})$ be a guarded formula where R appears only positively and not in guards and that contains no free variables other than $\bar{v}$. Then the formulas $[\mathsf{LFP} R\bar{v}.\varphi(\bar{v})]$ and $[\mathsf{GFP} R\bar{v}.\varphi(\bar{v})]$ are in $\mu\mathsf{GF}$.

$\mu\mathsf{GF}$ is actually a very powerful language, with expressivity beyond FO.

Example 1.15. [Grädel, 1999] Let φ be the conjunction of the formulas

$$\begin{gathered}\exists x_0 x_1 . R(x_0, x_1),\\ \forall x_0 x_1 .(R(x_0, x_1) \rightarrow \exists x_0 . R(x_1 x_0)),\\ \forall x_0 x_1 .(R(x_0, x_1) \rightarrow [\mathsf{LFP} S x_0 . \forall x_1 .(R(x_1, x_0) \rightarrow S x_1)](x_0)).\end{gathered}$$

The first two conjuncts force the model to have an infinite "forward" path through the R relation, while the third says that each point in the path is in the least fixed point of the operator $S \mapsto \{e \mid \text{all } R\text{-predecessors of } e \text{ are in } S\}$, which is the set of points that have only finitely many R-predecessors. In particular, this forces the path to be acyclic and hence the formula has only infinite models.

Many further results show that guarded fragments have nice behavior. Andréka *et al.* [1998] prove a Łos-Tarski theorem for preservation of guarded formulas under submodels, and also a model-theoretical characterization by means of a suitable definition of back-and-forth systems. Hoogland *et al.* [1999] provide the correct notions of interpolation and Beth definability for guarded fragments. Grädel and Walukiewicz' complexity results show that these fragments are, to a large extent, robustly decidable.

It is interesting to investigate how the intuitions built into the guarded fragments came to be. They are deeply rooted in modal languages.

1.3 Why Modal Logic?

Modal Logic [Blackburn *et al.*, 2000], originally conceived as the logic of necessity and possibility, has developed into a powerful mathematical discipline that deals with (restricted) description languages for talking about various kinds of relational structures. For many years, modal logic was viewed as an extension of propositional logic by the addition of the modal operators $\Diamond$ and $\Box$, but nowadays the picture has changed in many directions.

First, $\Diamond$ and $\Box$ have lost their privileged position, as a wide variety of new modalities have been introduced in recent years, witness for instance the work on Since-Until logics [Gabbay *et al.*, 1994], the universal modality [Goranko and Passy, 1992], the difference operator [de Rijke, 1992], propositional dynamic logic [Harel, 1984], counting modalities [de Rijke and van der Hoek, 1995], etc. Moreover, modal logic is no longer seen as just an extension of propositional logic, but also as a restriction of FO.

Again, it is better to do first things first and to start by introducing the basic modal language and its formal semantics.

Definition 1.16. [Basic modal logic] Let $\mathsf{PROP} = \{p_1, p_2, \ldots\}$ be a countable set of *propositional variables.* The *well-formed formulas* of the basic modal language over PROP are

$$\mathsf{FORMS} := \top \mid p_i \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond\varphi,$$

where $p_i \in \mathsf{PROP}$ and $\varphi, \varphi_1, \varphi_2 \in \mathsf{FORMS}$. We take $\Box\varphi$ as a shorthand for $\neg\Diamond\neg\varphi$.

A *modal model* $\mathcal{M}$ for $\mathcal{L}$ is a triple $\mathcal{M} = \langle M, R, V\rangle$ such that M is a non-empty set, R a binary relation on M, and $V : \mathsf{PROP} \to Pow(M)$. Let $\mathcal{M} = \langle M, R, V\rangle$ be a model and $m \in M$, then the *satisfiability relation* $\Vdash$ is defined as

$$\begin{array}{rll}
\mathcal{M}, m \Vdash \top & & \text{always} \\
\mathcal{M}, m \Vdash p_i & \text{iff} & m \in V(p_i),\ p_i \in \mathsf{PROP} \\
\mathcal{M}, m \Vdash \neg\varphi & \text{iff} & \mathcal{M}, m \nVdash \varphi \\
\mathcal{M}, m \Vdash \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M}, m \Vdash \varphi_1 \text{ and } \mathcal{M}, m \Vdash \varphi_2 \\
\mathcal{M}, m \Vdash \Diamond\varphi & \text{iff} & \text{for some } m' \in M \text{ such that } R(m, m') \text{ holds, } \mathcal{M}, m' \Vdash \varphi.
\end{array}$$

From the definition it is clear that the language can be seen as an extension of propositional logic. But we can now make precise our remark that it is also a fragment of first-order logic. Notice that a modal model $\langle M, R, V\rangle$ can be seen as a first-order model by considering V as the part of the interpretation function defining the meaning of unary predicate symbols. Also, the conditions defining $\Vdash$ are purely first-order. Putting these two intuitions together we obtain the following embedding.

Definition 1.17. [Standard translation] Consider the signature $\mathcal{S} = \langle \{R\} \cup \{P_i \mid p_i \in \mathsf{PROP}\}, \{\}, \{\}\rangle$. And define the translation ST from modal formulas to first-order formulas over $\mathcal{S}$ as follows:

$$\begin{array}{rcl}
ST(p_j) & = & P_j(x_0),\ p_j \in \mathsf{PROP} \\
ST(\neg\varphi) & = & \neg ST(\varphi) \\
ST(\varphi \wedge \psi) & = & ST(\varphi) \wedge ST(\psi) \\
ST(\Diamond\varphi) & = & \exists x_j.(R(x_0, x_j) \wedge ST(\varphi)[x_0/x_j]),
\end{array}$$

where j is the smallest index such that x_j does not appear in $ST(\varphi)$.

Theorem 1.18. [Satisfiability preservation] *Let φ be a formula in the modal language, then for any model $\mathcal{M}$ in the appropriate signature, $\mathcal{M}, m \Vdash \varphi$ iff $\mathcal{M} \models ST(\varphi)[m]$.*

Performing the translation with slightly more care and reusing variables, we can actually embed a basic modal formula φ into FO^2.

Definition 1.19. [Standard translation into FO^2] In [Vardi, 1997] the following translation is introduced

$$\begin{array}{rcl}
ST(p_j) & = & P_j(x_0),\, p_j \in \mathsf{PROP} \\
ST(\neg\varphi) & = & \neg ST(\varphi) \\
ST(\varphi \wedge \psi) & = & ST(\varphi) \wedge ST(\psi) \\
ST(\Diamond\varphi) & = & \exists x_1.(R(x_0,x_1) \wedge \forall x_0.(x_0 = x_1 \rightarrow ST(\varphi))),
\end{array}$$

but this uses a more complicated pattern of quantification and also identity. We can do even better by using a pair of mutually recursive functions.

$$\begin{array}{r|r}
ST_{x_0}(p_j) = P_j(x_0),\, p_j \in \mathsf{PROP} & ST_{x_1}(p_j) = P_j(x_1),\, p_j \in \mathsf{PROP} \\
ST_{x_0}(\neg\varphi) = \neg ST_{x_0}(\varphi) & ST_{x_1}(\neg\varphi) = \neg ST_{x_1}(\varphi) \\
ST_{x_0}(\varphi \wedge \psi) = ST_{x_0}(\varphi) \wedge ST_{x_0}(\psi) & ST_{x_1}(\varphi \wedge \psi) = ST_{x_1}(\varphi) \wedge ST_{x_1}(\psi) \\
ST_{x_0}(\Diamond\varphi) = \exists x_1.(R(x_0,x_1) \wedge ST_{x_1}(\varphi)) & ST_{x_1}(\Diamond\varphi) = \exists x_0.(R(x_1,x_0) \wedge ST_{x_0}(\varphi)).
\end{array}$$

Clearly, Theorem 1.18 can also be proved for these new translations. And notice that we have immediately obtained two different proofs of the decidability of the satisfiability problem for the basic modal logic, as the range of ST is both inside FO^2 and GF. Actually, robust decidability obtains, as even the μ-calculus [Kozen, 1983] (the extension of the basic modal language with fixed points) has an exponential-time decision procedure for satisfiability (a double exponential upper bound can easily be obtained via translation into $\mu\mathsf{GF}$).

The embedding of the modal language into the guarded fragments comes as no surprise, as these fragments were actually first thought of as generalizations of the basic modal language. But "generalizations" are at odds with our aim of obtaining specialized, hand-tailored languages. Modal languages instead, can be used *themselves* as ways to identify interesting fragments of first-order logic, and even of extensions as is shown by examples such as the μ-calculus mentioned above, Propositional Dynamic Logic [Harel, 1984], or the Computational Tree Logic of Clarke and Emerson [1982].

Some attractive properties are often shared by modal languages: a propositional flavor, a simple and uniform semantics, elegant completeness results, strong decidability, interpolation and Beth definability, effective reasoning systems, model-theoretical characterizations, adequate tools to compare models. The list is just too long to be given in full. In other words, using modal languages as a means to define well behaved fragments of first-order logic and extensions seems a promising project.

1.4 Some Concrete Examples

Example 1.20. Suppose we want to capture the information conveyed in the following paragraph:

> *'And there is one among them that might have been foaled in the morning of the world. The horses of the Nine cannot vie with him; tireless, swift as the flowing wind. Shadowfax they called him. By day his coat glistens like silver; and by night it is like a shade, and he passes unseen.'*
>
> from "The Lord of the Rings," J. R. R. Tolkien

We can think of the domain of models as "the set of all things" in Tolkien's world and define certain relations among them, like for example the unary relation "tireless-things" which consist of the set of things that never tire, or the binary relation "has-coat-like" which relates a thing a with a thing b if a happens to have a coat that looks like b. "Shadowfax" is, of course, one of the elements in the domain, and we can define a special unary relation symbol `shadowfax`, whose denotation is the singleton subset of the domain containing it. Consider the following formulas,

$$\texttt{shadowfax} \rightarrow \left(\begin{array}{c} \langle \textit{foaled} \rangle \text{ things-existing-in-the-morning-of-the-world} \wedge \\ \text{tireless-things} \ \wedge \ \text{swift-as-the-wind-things} \wedge \\ [\textit{has-coat-like}] \text{ things-glistening-like-silver-by-day} \wedge \\ [\textit{has-coat-like}] \text{ things-like-shades-by-night } \ \wedge [\textit{pass-like}] \text{ unseen-things} \end{array} \right)$$

$$\text{horses-of-the-Nine} \rightarrow \neg\langle \textit{vie} \rangle \ \texttt{shadowfax}.$$

They seem to capture a good deal of the information contained in the text. And we can use some of the extensions to the basic modal language we already mentioned, to define very expressive concepts. For example, if the "purest-breed" are those horses sired by Shadowfax or his descendants, using fixed points we define

$$\text{purest-breed} \leftrightarrow \mathsf{LFP}X.(\texttt{shadowfax} \vee [\textit{sired-by}]X).$$

EXAMPLE 1.21. From a different angle, suppose we want to describe a set of structures, like for example, full binary trees. We can define in the basic modal language, for any $n \in \mathbb{N}$ a formula φ_n such that

- the size of φ_n is quadratic in n,
- φ_n is satisfiable,
- if $\mathcal{M}, w \Vdash \varphi_n$, then $\mathcal{M}$ contains as a substructure an isomorphic copy of the binary tree of depth n with root w.

We fix a set $\{p_0, \ldots, p_{n-1}\}$ of propositional symbols, and start by defining the shorthands $\textit{branch}(p_i) := \Diamond p_i \wedge \Diamond \neg p_i$ and $\textit{store}(p_i) := (p_i \rightarrow \Box p_i) \wedge (\neg p_i \rightarrow \Box \neg p_i)$. Then

$$\varphi_n := \textit{branch}(p_0) \wedge \bigwedge_{1 \leq i < n} \Box^i(\textit{branch}(p_i) \wedge \bigwedge_{0 \leq j < i} \textit{store}(p_j)),$$

in which $\Box^i$ abbreviates an i-long sequence of boxes. The formula works as follows. Suppose $\mathcal{M}, m \Vdash \varphi_n$. Then, because $\Box^i \textit{branch}(p_i)$ is satisfied, every node m' reachable in i R-steps from m has two different successors, one forcing p_i and one forcing $\neg p_i$. But in each R-step we used the *store* formula to ensure that the value of p_j for $j < i$ is "carried over" into the next state. It is easy to check that the interplay of *branch* and *store* forces a binary tree of depth n. As an aside, notice that this implies that the size of the smallest model satisfying φ_n is exponential in $|\varphi_n|$. That the basic modal language is expressive enough to force big models is a well known fact, see [Marx and Venema, 2000] for example, for an up-to-date survey on the topic.

Modal languages are very well suited to describe graphs as we did above and this, together with their effective decision methods, produces a powerful combination.

In [Areces *et al.*, 1999b] for example, we use modal languages to encode the transition diagrams of simple telephony protocols, together with specific *telephony features* — special telephony services like Call Waiting or Call Forwarding. We then formally verify that the behavior of a feature does not interfere with the behavior of others in an unexpected way, by means of automatic tools.

Example 1.22. Believe it or not, modal logics can also help you to properly set a table for a formal dinner. Aiello and van Benthem [1999] discuss modal languages to describe topological structures. In this work, topological spaces are used instead of relational models and the basic modalities $\Diamond$ and $\Box$ are interpreted as the interior and the closure of a region, respectively. When further expressive power is added, the language can express *betweenness* and hence distinguish between the well set table on the left and the messy one on the right in Figure 1.1.

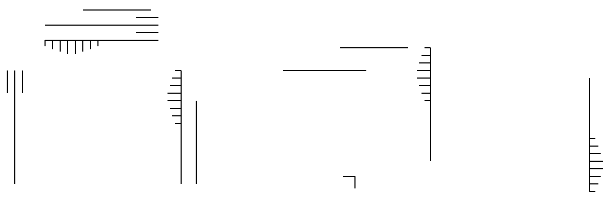

Figure 1.1: Properly setting a table

In [Aiello *et al.*, 1999] we use similar ideas to propose an image description language and an image retrieval engine. Topological operators are used to describe the relations between regions in an image. In addition, further information can be naturally added, similarly as how we did in Example 1.20. A query to the image database can then be recast as a formula, and the retrieval process transformed into inference: given a query φ retrieve all pictures P such that their description δ_P implies the query, $\models \delta_P \rightarrow \varphi$.

1.5 This Thesis

The topics discussed in this chapter should give the reader a taste of what is to come. As we mentioned before, we will investigate modal systems (broadly conceived) as a way to define well behaved fragments. In particular, we will investigate the family of systems known as Description Logics which are probably one of the best examples of "application-driven" formal languages.

Description logics form a collection of languages that come equipped with effective inference methods. These languages were born in the Knowledge Representation community, and this shows in the mixture of theoretical and practical results which is almost a hallmark of the literature in the field. These languages are closely related to the basic modal language, but they usually add further expressivity depending on the specific problem at hand. Systems containing operators like counting, transitive closure, fixed points or nominals, to mention some, have been proposed and widely used. In each case, effective decision methods have been devised and their complexity carefully analyzed. On the other hand, and probably as a result of the fact that many of these languages were introduced to tackle a particular problem, an *unifying* logical background theory to permit for example, the comparison of their respective expressive power, is largely missing. In this thesis, we will put forward the claim that *hybrid logics*, extensions of the basic modal language with the ability to explicitly refer to elements in the domain, provide such a unifying framework.

The themes we discussed in this introduction are at the root of the work we will do throughout the thesis. If you survived up to here, you are ready to go on. Theorem 1.1 focused on undecidability and, more generally on complexity of reasoning problems. For first-order logic we have only discussed satisfiability and validity, but when moving to modal languages we will see different examples of reasoning tasks, in addition to these two. We will also touch on different reasoning methods like tableau and resolution calculi, and on the advantages and disadvantages of each of them, both from a theoretical point of view and in implementations. In Theorem 1.2 we discussed matters related to expressive power. This is a core issue, as little is known about ways of comparing the expressive power of description languages. The tools here are model comparison techniques such as the Ehrenfeucht games, and the central notion of bisimulation (the modal counterpart of partial isomorphisms). Finally, with Theorem 1.13 we touched on particular meta-logical properties like interpolation and Beth definability, which are usually a sign of a "healthy" reasoning system.

Part II of the thesis, starts by introducing the two different families of languages we will study, and concludes by building a bridge between them. Chapter 2 will cover Description Logics, while Chapter 3 will introduce Hybrid Logics. Chapter 4, which is based on early results in [Areces and de Rijke, 1998], and on the more up to date [Areces and de Rijke, 2000], establishes the logical correlation between the two different frameworks. In this chapter we show the strength of the connection between description and hybrid languages, and exemplify in detail how results in one of the fields yield interesting facts in the other.

The three chapters in Part III form the core of the thesis, each of them channeling results into the logical connections we discuss in Chapter 4. In [Areces *et al.*, 1999c], we present a resolution system for modal and description languages. These results are discussed in Chapter 5, and the resolution calculus is extended to deal with hybrid binders. In Chapter 6 we present the characterization results obtained in [Areces *et al.*, 2000b], together with interpolation and definability results for hybrid systems. Chapter 7 is devoted to (un)decidability and complexity issues. The articles [Areces *et al.*, 2000a] and [Areces *et al.*, 1999a] are concerned with sharp undecidability results (what is the borderline between decidability and undecidability for very expressive hybrid systems?), and with the effect that "hybridization" (the addition of the particular ability to refer to states) produces in modal languages from the complexity point of view. These results are related to the already known results for certain description languages. The key results discussed in Part III of the thesis were obtained in collaboration with Patrick Blackburn, Maarten Marx, Hans de Nivelle and Maarten de Rijke. Not only did I learn a great deal from them, but also I enjoyed myself immensely during this joint work, and I can hardly thank them enough for all the help I've received from them.

Finally, Part IV concludes with Chapter 8 where we draw our conclusions and point to directions of further research.

Part II

Two Kingdoms

Every year of her life, she had invented something. The first invention she could remember was the separated-vision experience, conducted in bed with one eye above the blanket and one below. She had been four then. Although she recalled that there had been inventions even before that age, the separated-vision experience had been so seminal that it obliterated its feebler forerunners. Most of the later inventions owed something to the separated-vision experience, which had revealed to her — and still continue to reveal — that if you took up certain positions you could receive dual and conflicting impressions of the universe.

from "Year by Year the Evil Gains," Brian Aldiss

In this part of the thesis we will introduce the basic details of the two families of languages we will study: description and hybrid logics.

Description logics are a collection of specialized languages for the representation and structuring of knowledge, together with efficient methods to perform different "reasoning tasks." Nowadays, the fact that these languages can be regarded as variations of first-order logic, either restrictions or restrictions plus some added operators, is broadly accepted. These variations are mainly motivated by the undecidability of the satisfiability problem for first-order logic, but they are also rooted in a desire to preserve the structure of the knowledge being represented, and to capture a finer grained notion of "reasoning." The main tools used for providing decision methods and studying complexity-theoretical aspects in the area of description logic are based on labeled tableaux.

The history of description logics (or concept or terminological languages as they were initially called) is relatively short, starting around the KL-ONE system of Brachman and Schmolze [1985]. But in virtue of their applicability to problems as varied as deductive databases, image retrieval, system modeling and information classification, they flourished rapidly. Applications and effective inference algorithms are the bread and butter of the field.

Hybrid logics on the other hand, are much older, starting with the work of Arthur Prior on the logic of time and tense in the mid 1960s. The earliest published reference is probably [Prior, 1967, Chapter 5 and Appendix B3]. The original intuitions of Prior were built around the phrase *terms as formulas.* Prior's intentions of providing an implicit account of tense by means of modal languages, collided with the generalized use of explicit references like *now*, *yesterday*, *on May 21st*, etc. in tensed expressions. By treating terms as formulas, he discovered a way of dealing with references in a genuinely modal way. Prior usually worked with what today are known as *strong* hybrid languages, in which references could even be bound by quantifiers. The ideas of Prior were later investigated by his student Robert Bull. [Bull, 1970] contains important technical ideas, and also introduces special "course of history" nominals to name paths through models, bringing into the picture the general theme of sorting.

Hybrid languages developed in a purely logic environment, and bear all the marks of it. The main results concern expressive power, axiomatizations and completeness. The field went into a revival with the work of Gargov, Passy and Tinchev [Passy and Tinchev, 1985a, 1985b; Gargov *et al.*, 1987; Gargov and Passy, 1988], who independently rediscovered the idea of hybrid languages and nominals. The work of the Sofia School, as this group came to be known, focused on hybrid languages built over propositional dynamic logic, and includes some of the first results on (un)decidability. They also examined quantifier-free systems, thus initiating the trend towards weaker hybrid languages.

In this part of the thesis we will first introduce the two families of languages independently in Chapters 3 and 4, defining the basic notions we will need in the chapters to come, and highlighting the strengths that each of them has.

As we will make clear in our presentation, the two families are closely related. It is well-known that the description language $\mathcal{ALC}$ is a notational variant of multi-modal logic, but this relation only holds for very weak inference tasks. Hybrid logic's ability to explicitly refer to elements in the domain together with operators to change the point of reference, will make it possible to account also for reasoning involving terminological definitions and assertional information.

The bridge we will build in Chapter 4, will lead to cross-fertilization between the two fields. Given that the two areas have developed different techniques and evolved in divergent directions, "trading" between them can be specially fruitful. Description logics can export reasoning methods, complexity results and application opportunities; while hybrid logics have their model-theoretical tools, axiomatizations and expressive power analysis to offer.

Chapter 2

Introducing Description Logics

Esas ambigüedades, redundancias y deficiencias recuerdan las que el doctor Franz Kuhn atribuye a cierta enciclopedia china que se titula Emporio celestial de conocimientos benévolos. *En sus remotas páginas está escrito que los animales se dividen en (a) pertenecientes al Emperador, (b) embalsamados, (c) amaestrados, (d) lechones, (e) sirenas, (f) fabulosos, (g) perros sueltos, (h) incluidos en esta clasificación, (i) que se agitan como locos, (j) innumerables, (k) dibujados con un pincel finísimo de pelo de camello, (l) etcétera, (m) que acaban de romper el jarrón, (n) que de lejos parecen moscas.*

from "El Idioma Analítico de John Wilkins," Jorge Luis Borges

2.1 Structure: the Key to Knowledge

Knowledge about anything abounds. For a test, think of any single word and run a query on a search engine on the Internet. It doesn't matter which word, you will probably get (at least) hundreds of hits. The problem of course, is that we are usually not interested in *any kind* of knowledge, and probably only few of the hundreds of hits are actually relevant to us. In other words, information should be structured to be useful, so that we can decide which part of it is important to our problem. But classifying information is a difficult and expensive task.

Structured representation of knowledge aims to address both conceptual and computational complexity. Conceptual economy amounts to building hierarchical structures, where inheritance of attributes through the hierarchy is used to avoid redundancies in the representation. Computational economy refers to the efficiency of reasoning upon such structures. The idea of developing systems based on a structured representation of knowledge has been pursued for a long time in Artificial Intelligence. One of the earliest knowledge representation tools has been the *Semantic Networks*, and the work of Quillian [1967] on the *Semantic Memory Model*. Semantic networks represent knowledge in the form of a labeled directed graph. Specifically, each node is associated with a concept, and the arcs represent the various relations between concepts.

In a similar line, the work of Minsky [1974] on the *Frame Paradigm*, aims to achieve structured knowledge. A frame represents a certain concept (usually a class of individuals), and is characterized by a number of attributes (called *slots*) that members of its class can have. Each slot contains information about the corresponding attribute, such as default values, restrictions on the elements that can fill the attribute (*slot fillers*), attached procedures or methods for computing values when needed, and procedures for propagating side effects when the slot is filled. The values of the attributes are either elements of a concrete domain (e.g., integers, strings) or identifiers of other frames. A frame can also represent a single individual, in this case it is related with the attribute *instance-of* to the frame representing the class of which the individual is an instance.

The main drawback of both proposals was their unclear semantics. As pointed out in [Woods, 1975] with respect to semantic networks, the arcs between the nodes and the nodes themselves can represent different kinds of information, and making this difference explicit can be crucial. Consider the following example.

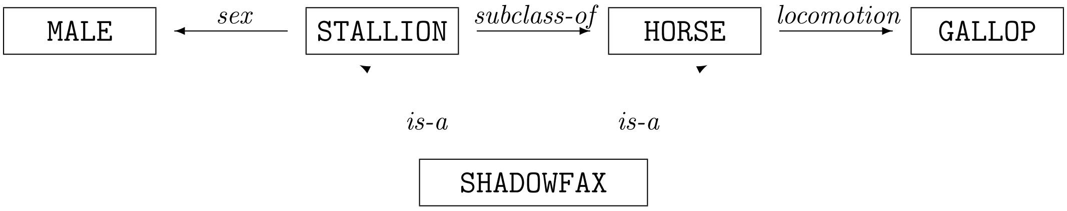

Figure 2.1: Shadowfax the stallion

The network is meant to encode the following information: Shadowfax is a stallion and also a horse, stallions are male, horses gallop, and stallions are horses.

Woods defines two main types of arcs, that he identifies as encoding either *intensional* or *extensional* information. An arc that contributes to the definition of a concept carries intensional knowledge. The arc labeled *locomotion* between `HORSE` and `GALLOP` is of this kind. On the other hand, the arc from `SHADOWFAX` to `STALLION`, labeled *is-a*, asserts the fact that Shadowfax is a stallion, which Woods classifies as extensional knowledge. But things are more subtle, because also the nodes carry different information: `STALLION` is a class, while `SHADOWFAX` is a distinguished individual. And furthermore, the "intensional" relation between `STALLION` and `HORSE` is of a different kind than the relation represented by *locomotion.* The distinction between extensional and intentional links, neither exhausts nor characterizes all the possibilities.

The need for a formal semantics was clear, and [Brachman, 1977, 1979] are some of the early references which aimed to address this problem. The work of Brachman led to the development of KL-ONE [Brachman and Schmolze, 1985], one of the first knowledge maintainance systems for which some kind of formal semantics was specified. The KL-ONE system fathered a number of successors such as KRYPTON [Brachman *et al.*, 1985], LOOM [MacGregor and Bates, 1987], CLASSIC [Borgida *et al.*, 1989], BACK [Quantz and Kindermann, 1990] and KRISS [Baader and Hollunder, 1991]. But, perhaps more importantly, this work also gave rise to theoretical research on what were first named *concept* or *terminological languages*, and which are today called *description logics*.

Description logics (DLs) are a family of formal languages with a clearly specified semantics, usually in terms of first-order models, together with specialized inference mechanisms to account for knowledge classification. It was clearly one of the original and main aims of the research in this field to identify the exact fragments of FO and extensions able to capture the features needed for representing a particular problem, and which can still allow for the design of efficient reasoning algorithms.

DLs have found a variety of applications in diverse areas. To mention some, [Borgida, 1995] is a survey of the application of DLs to the problems of information management, proposing ways to achieve enhanced access to data and knowledge by using descriptions in languages for schema design and integration, queries, updates, rules, and constraints.

Another example is [Devanbu and Jones, 1997]. The increasing size and complexity of software systems demand every day a greater emphasis on capturing and maintaining knowledge at different levels within the software development process. The knowledge-based software engineering (KBSE) paradigm is concerned with systems that use formally represented knowledge, with associated inference procedures, to support the various sub-activities of software development. Devanbu and Jones investigate the application of DLs to KBSE, describing their use in three well-developed KBSE systems: LaSSIE and KITSS in the telephony domain and COMET in the radar tracking domain.

PROSE (PRoduct OfferingS Expertise) is a knowledge based engineering and ordering platform that supports sales and order processing at AT&T Network Systems. The cornerstone of the PROSE architecture is a product knowledge base written in C-CLASSIC. PROSE is used to provide configurations for sales proposals and to generate factory orders for manufacturing. A fairly detailed description of the system can be found in [Wright *et al.*, 1993].

DLs have also been used extensively for general information retrieval as is described in [Meghini *et al.*, 1993] where the system MIRTL (Multimedia Information Retrieval Terminological Logic) is introduced. In [Aiello *et al.*, 1999] we investigated special mixed description languages which coordinate satisfiability and model checking to provide expressive image retrieval which can account for topological relations between objects in pictures. As a final example, in [Areces *et al.*, 1999b] we use description languages to model BCS, the Basic Call System used in telephony domains and formally investigate the issue of feature interaction, analyzing the problems arising when merging new services like Call Waiting, Call Forwarding, etc. DLs have even found their way into general, school level Artificial Intelligence, and they are already discussed in text books like [Russell and Norvig, 1995].

2.2 Basic Issues in Description Logic

Probably rooted in the original distinction of Woods, most description languages split the available knowledge about a given situation into

- *terminological information*: definitions of the basic and derived notions and of the ways they are inter-related. This information is "generic" or "global," been true in every model of the situation and of every individual in the situation. And
- *assertional information*: which records "specific" or "local" information, being true of certain particular individuals in the situation.

All known information is then modeled as a pair $\langle T, A\rangle$, where T is a set of formulas concerning terminological information (the T-Box) and A is a set of formulas concerning assertional information (the A-Box).

Another way to look at this separation of information is from a database point of view: the T-Box is a general schema concerning the classes of individuals to be represented, their general properties and mutual relationships, while the A-Box is a partial instantiation of this schema, containing assertions relating either individuals to classes, or individuals to each other.

Constructor	Syntax	Semantics
concept name	C	$C^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
negation ($\mathcal{C}$)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
disjunction ($\mathcal{U}$)	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
universal quant.	$\forall R.C$	$\{d_1 \mid \forall d_2 \in \Delta^{\mathcal{I}}.(R^{\mathcal{I}}(d_1, d_2) \rightarrow d_2 \in C^{\mathcal{I}})\}$
existential quant. ($\mathcal{E}$)	$\exists R.C$	$\{d_1 \mid \exists d_2 \in \Delta^{\mathcal{I}}.(R^{\mathcal{I}}(d_1, d_2) \wedge d_2 \in C^{\mathcal{I}})\}$
number restr. ($\mathcal{N}$)	$(\geq n\ R)$	$\{d_1 \mid \lvert\{d_2 \mid R^{\mathcal{I}}(d_1, d_2)\}\rvert \geq n\}$
	$(\leq n\ R)$	$\{d_1 \mid \lvert\{d_2 \mid R^{\mathcal{I}}(d_1, d_2)\}\rvert \leq n\}$
one-of ($\mathcal{O}$)	$\{a_1, \ldots, a_n\}$	$\{d \mid d = a_i^{\mathcal{I}}$ for some $a_i\}$
role filler ($\mathcal{B}$)	$\exists R.\{a\}$	$\{d \mid R^{\mathcal{I}}(d, a^{\mathcal{I}})\}$
role name	R	$R^{\mathcal{I}}$
role conjunction ($\mathcal{R}$)	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}}$
inverse roles ($\mathcal{I}$)	R^{-1}	$\{(d_1, d_2) \mid R^{\mathcal{I}}(d_2, d_1)\}$

Table 2.1: Common operators of description logics

2.2.1 Concepts and Roles

Let us make things more precise now.

Definition 2.1. [Description logic semantics] Let $\mathsf{CON} = \{C_1, C_2, \ldots\}$ be a countable set of *atomic concepts*, $\mathsf{ROL} = \{R_1, R_2, \ldots\}$ be a countable set of *atomic roles* and $\mathsf{IND} = \{a_1, a_2, \ldots\}$ be a countable set of *individuals*. For $\mathsf{CON}, \mathsf{ROL}, \mathsf{IND}$ pairwise disjoint, $\mathcal{S} = \langle \mathsf{CON}, \mathsf{ROL}, \mathsf{IND} \rangle$ is a *signature*. Once a signature $\mathcal{S}$ is fixed, an *interpretation* $\mathcal{I}$ for $\mathcal{S}$ is a tuple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where

– $\Delta^{\mathcal{I}}$ is a non empty set.

– $\cdot^{\mathcal{I}}$ is a function assigning an element $a_i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each constant a_i; a subset $C_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each atomic concept C_i; and a relation $R_i^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each atomic role R_i.

In other words, a description logic interpretation is no more than a model for a particular kind of first-order signature (see Definition 1.4), where only unary and binary predicate symbols are allowed and the set of function symbols is empty.

The atomic symbols in a description logic signature can be combined by means of *concept and roles constructors*, to form complex concept and role expressions. Each description logic is characterized by the set of concept and roles constructors they allow. Table 2.1 defines the roles and concepts constructors for the description logics we will discuss, together with their semantics.

Historically, a number of description logics have received a special name. The language $\mathcal{FL}^-$ [Brachman and Levesque, 1984] is defined as the description logic allowing universal quantification, conjunction and unqualified existential quantifications of the form $\exists R.\top$. $\mathcal{FL}^-$ was proposed as a formalization of the core notions of Minsky's frames. Concept conjunction is implicit in the structure of a frame, which requires a *set* of conditions to be satisfied. Role quantifications allow one to characterize slots: the unqualified existentials state the existence of a value for a slot, while the universal quantifier requires that the values of a slot satisfies a certain condition.

The logic $\mathcal{AL}$ [Schmidt-Schauss and Smolka, 1991] extends $\mathcal{FL}^{-}$ with negation of atomic concepts. It is customary to define systems by postfixing the names of these original systems with the names of the added operators from Table 2.1. For example, the logic $\mathcal{ALC}$ is $\mathcal{AL}$ extended with full negation.

We will not discuss in detail all possible languages which can be obtained by combining constructors from Table 2.1. In particular, we will be interested in languages having full Boolean expressivity and usually consider $\mathcal{ALC}$ and its extensions. For a given language $\mathcal{L}$, let $\mathsf{CON}(\mathcal{L})$ be the set of complex concept expressions and $\mathsf{ROL}(\mathcal{L})$ be the set of complex role expressions which can be formed by using the constructors of $\mathcal{L}$. It is interesting to notice that the constructs in Table 2.1 are not necessarily independent of each other. Given a language $\mathcal{L}$ and a constructor $\star$, we say that $\mathcal{L}$ *simulates* $\star$ if for every complex concept of $\mathcal{L}\star$ there exists an equivalent concept in $\mathcal{L}$. Formally, for any $C_1 \in \mathsf{CON}(\mathcal{L}\star)$, there exists $C_2 \in \mathsf{CON}(\mathcal{L})$ such that for all interpretations $\mathcal{I}$, $C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$. Given this definition, proving for example that $\mathcal{ALEU}$ simulates $\mathcal{C}$ or that $\mathcal{ALEO}$ simulates $\mathcal{B}$ is straightforward, by means of simple reductions. We will usually assume that all constructors which can be simulated in $\mathcal{L}$ are already present in the language, e.g., we will say that $\mathcal{U}$ is one of the constructors of $\mathcal{ALC}$.

2.2.2 Knowledge Bases and Inference

In description logics we want to perform inferences given certain background knowledge.

Definition 2.2. [Knowledge bases] Fix a description language $\mathcal{L}$, a *knowledge base* Σ in $\mathcal{L}$ is a pair $\Sigma = \langle T, A\rangle$ such that

- T is the T(erminological)-Box, a finite, possibly empty, set of expressions of the form $C_1 \sqsubseteq C_2$ where C_1, C_2 are in $\mathsf{CON}(\mathcal{L})$. $C_1 \doteq C_2$ is notation for $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$. Formulas in T are called *terminological axioms*.
- A is the A(ssertional)-Box, a finite, possibly empty, set of expressions of the forms $a : C$ or $(a, b) : R$ where C is in $\mathsf{CON}(\mathcal{L})$, R is in $\mathsf{ROL}(\mathcal{L})$ and a, b are individuals. Formulas in A are called *assertions*.

The definitions of terminological axioms and assertions above are among the most general in the description logic literature (and we will generalize them even further in Chapter 4). Terminological axioms were originally thought of as *definitions*, and a number of more restrictive conditions were imposed. The two most important restrictions were the following.

i. *Simple terminological axioms*: in any terminological axiom $C_1 \sqsubseteq C_2$, C_1 is an atomic concept in CON. And any atomic concept in CON appears at most once in the left hand side of a terminological axiom in the T-Box.

ii. *Acyclic definitions*: the graph obtained by assigning a node n_A to each atomic concept A in the T-Box T and drawing an arrow between two nodes n_A and n_B if there is a terminological axiom $C_1 \sqsubseteq C_2$ in T such that A appears in C_1 and B appears in C_2, does not contain cycles.

These restrictions were rooted in the idea of considering terminological axioms as (partial) definitions of concepts. An axiom of the form $A \sqsubseteq C$ was meant to represent the fact that C encoded necessary conditions for A to be the case, $A \doteq C$ indicated that C encoded both necessary and sufficient conditions. From this point of view, conditions *i)* and *ii)* above are very natural. In particular, restriction *ii)* was aimed at avoiding circular definitions, where a concept is defined in terms of itself. It was argued that this kind of definitions called for some sort of special semantics, like for example fixed points or non-well-founded sets [Nebel, 1990a; Baader, 1990; Dionne *et al.*, 1992]. But general terminological axioms as introduced in Definition 2.2 act naturally as constraints on models of the knowledge base, without any need for a special semantics; this way of interpreting terminologies is usually called "descriptive semantics" in the DL literature.

More importantly from an application point of view, restrictions *i)* and *ii)* have a strong impact on the reducibility of certain reasoning tasks into others as we will discuss in Section 2.3. Clearly, this kind of syntactic restrictions undercut the expressive power of the language, but they simplify the definition of decision algorithms. As we discuss in Chapter 4, the complexity results we prove in Chapter 7 show that in many cases we can actually introduce a more general notion of knowledge base than the one given in Definition 2.2 without modifying the worst case complexity of the reasoning tasks for the language.

It is time to define the appropriate notion of inference for description logics.

Definition 2.3. Let $\mathcal{I}$ be an interpretation and φ a terminological axiom or an assertion. Then $\mathcal{I}$ *models* φ (notation, $\mathcal{I} \models \varphi$) if

- $\varphi = C_1 \sqsubseteq C_2$ and $C_1^{\mathcal{I}} \subseteq {C_2}^{\mathcal{I}}$, or
- $\varphi = a{:}C$ and $a^{\mathcal{I}} \in C^{\mathcal{I}}$, or
- $\varphi = (a,b){:}R$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Let $\Sigma = \langle T, A\rangle$ be a knowledge base and $\mathcal{I}$ an interpretation, then $\mathcal{I}$ models Σ (notation, $\mathcal{I} \models \Sigma$) if for all $\varphi \in T \cup A, \mathcal{I} \models \varphi$. We say in this case that $\mathcal{I}$ *is a model of the knowledge base* Σ. Given a knowledge base Σ and a terminological axiom or assertion φ, $\Sigma \models \varphi$ if for all models $\mathcal{I}$ of Σ we have $\mathcal{I} \models \varphi$.

The notion of semantic consequence we have just defined, $\Sigma \models \varphi$, is central in modern logic. Notice that if a deduction theorem fails for the logic, it needs not be reducible to "tautology hood," i.e., to a given formula in the language being satisfied under any interpretation $\mathcal{I}$. We will return to this in Section 4.2.1.

2.3 Reasoning Tasks

In description logics the term *T-Box reasoning* is used to refer to the ability to perform inferences from a knowledge base $\Sigma = \langle T, A\rangle$ where T is non-empty, and similarly, *A-Box reasoning* is inference for A non-empty.

Example 2.4. Let Σ be a knowledge base $\langle T, A\rangle$ where

$$\begin{aligned} T &= \{\texttt{STALLION} \doteq \texttt{HORSE} \sqcap \forall \texttt{Sex.MALE}\} \\ A &= \{\texttt{shadowfax:STALLION}\}. \end{aligned}$$

Informally, the formula in T says that male horses are called stallions, while the formula in A says that a particular horse, named Shadowfax, is a stallion. The formal semantics we gave in Definition 2.3 lets us verify that Σ has at least a model (i.e., it is consistent). And from Σ we can infer further information like, for example, that the concept HORSE is consistent with respect to Σ (there exists some interpretation satisfying Σ which assigns a non-empty extension to HORSE:

$$\Sigma \not\models \texttt{HORSE} \doteq \bot.$$

Notice though, that we cannot express the stronger (true) fact that in any model of $\langle T, A\rangle$, the extension of HORSE is non-empty.

$$\Sigma \models \neg(\texttt{HORSE} \doteq \bot),$$

because of syntactic limitations in the standard definition of assertions. In Chapter 4 we will further discuss this issue, and clarify why $\Sigma \models \neg(\texttt{HORSE} \doteq \bot)$ is not equivalent to $\Sigma \not\models \texttt{HORSE} \doteq \bot$.

We can actually map out a set of *reasoning tasks* or *reasoning services* which can be provided by a knowledge representation system. The following are some of the standard reasoning tasks usually considered in description logics.

Definition 2.5. [Reasoning tasks] Let Σ be a knowledge base, $C_1, C_2 \in \mathsf{CON}(\mathcal{L})$, $R \in \mathsf{ROL}(\mathcal{L})$ and $a, b \in \mathsf{IND}$, we define the following *reasoning tasks*

- *Subsumption*, $\Sigma \models C_1 \sqsubseteq C_2$.
  Check whether for all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$ we have $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.
- *Instance Checking*, $\Sigma \models a{:}C$.
  Check whether for all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$ we have $a^{\mathcal{I}} \in C^{\mathcal{I}}$.
- *Relation Checking*, $\sigma \models (a, b){:}R$.
  Check whether for all interpretations $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$ we have $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.
- *Concept Consistency*, $\Sigma \not\models C \doteq \bot$.
  Check whether for some interpretation $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$ we have $C^{\mathcal{I}} \neq \{\}$.
- *Knowledge Base Consistency*, $\Sigma \not\models \bot$.
  Check whether there exists $\mathcal{I}$ such that $\mathcal{I} \models \Sigma$.

Subsumption is one of the classical reasoning tasks performed in most knowledge representation systems. It is directly related to the quest for information classification: it is useful to organize concepts into "is-a" hierarchies, finding for each class the most specific other classes that subsume it. This classification algorithms relies directly on the subsumption check. The subsumption relation implicitly defines a taxonomy of concepts, which can actually be used to solve some of the other reasoning tasks; for example, a concept is unsatisfiable if it is subsumed by the empty concept $\bot$.

Instance checking is used to verify whether the knowledge base entails that an individual is an instance of a concept, while relation checking determines if two individuals in the knowledge base stand in a given relation. These two operations can be considered the central reasoning tasks for retrieving information about individuals from a knowledge base. Finally, consistency is used for verifying whether the information contained in a knowledge base is coherent.

As we already said, the basic reasoning tasks can be used to define more complex ones. In particular

- *Retrieval*: given a concept, find all the individuals mentioned in the knowledge base that are instances of the concept.
- *Realization*: given an individual mentioned in the knowledge base, find the most specific concepts, with respect to the subsumption relation, of which the individual is an instance.

Let's start by defining the following notation. For any set S of formulas let $\mathsf{IND}(S)$ and $\mathsf{CON}(S)$ be, respectively, the set of individuals and atomic concepts appearing in formulas in S. Then, the retrieval problem can be formulated as follows: given a knowledge base Σ and a concept C, find the set $\{a \in \mathsf{IND}(\Sigma) \mid \Sigma \models a\!:\!C\}$; and it can be performed simply by iterating instance checking for all the individuals in Σ. The realization problem can be solved by finding the set $\{C \in \mathsf{CON}(\Sigma) \mid \Sigma \models a\!:\!C \;\&\; \forall C' \in \mathsf{CON}(\Sigma).(\Sigma \models a\!:\!C' \Rightarrow \Sigma \models C \sqsubseteq C')\}$; this can be done in terms of instance checking and subsumption.

Research on description logics has focused mainly on understanding the relations between the reasoning tasks mentioned above, and on establishing their computational complexity. The study of the computational behavior of DLs has provided a good understanding of the properties of the language constructs and their interaction. This is not only valuable from a theoretical point of view, but also provides insight to the designer of deduction procedures, with indications of which language constructs are difficult to handle and general methods to cope with the computational problems.

Complexity analyses of subsumption originated with the seminal paper of Brachman and Levesque [1984], where they provide a polynomial algorithm to decide subsumption for $\mathcal{FL}^-$. In the early days of description logics, the emphasis was on mapping out the *tractable* reasoning tasks, with an upper bound of polynomial complexity. Clearly, only very weak (non-Boolean) languages were able to survive the test, as already the satisfiability problem of mere propositional logic is NP-complete. The $\mathcal{U}$ and $\mathcal{E}$ constructors were identified as the main culprits of "intractability," the first one introducing nondeterminism while the second can be used to force big models [Baader and Hollunder, 1991; Schmidt-Schauss and Smolka, 1991; Donini *et al.*, 1992, 1997; Buchheit *et al.*, 1993]. Notice that with a bound of polynomial complexity, almost no exploration is possible. In a sense, this is at odds with the intuitive notion of inference, which calls for considering all possible outcomes given the present information.

Interestingly, the technological developments that took place in the intervening years have moved the boundaries of "tractability" further away. During a discussion at the 1998 Description Logic Workshop, Franz Baader put forward a bold claim. Nowadays, he argued, with the advances in both computing power and theorem proving techniques, tractability means something very near to ExpTime. Of course, this completely changes the perspective. The claim is supported by the performance of new theorem provers, as standard test beds like those proposed in [Heuerding and Schwendimann, 1996; Balsiger and Heuerding, 1998] have become too easy and fall in disuse. They are no longer appropriate for measuring the performance of state of the art provers (see [Horrocks *et al.*, 2000a] for an overview).

In sufficiently expressive languages, like $\mathcal{ALC}$ and extensions, all reasoning tasks we introduced can be reduced to instance checking (see [Donini *et al.*, 1994]). Let $\mathcal{L}$ be a language containing the $\mathcal{C}$ and $\mathcal{U}$ constructors, then subsumption and consistency in $\mathcal{L}$ can be reduced to instance checking. If $\mathcal{L}$ contains the $\mathcal{B}$ constructor, then relation checking can be reduced to instance checking.

PROPOSITION 2.6. [Reductions] *Given a knowledge base* Σ, $C_1, C_2 \in \mathsf{CON}(\mathcal{L})$, $R \in \mathsf{ROL}(\mathcal{L})$ *and* $a, b \in \mathsf{IND}$, *the following equivalences hold*

i. $\Sigma \models C_1 \sqsubseteq C_2$ *iff* $\Sigma \models a\!:\!(\neg C_1 \sqcup C_2)$, *for some* $a \in \mathsf{IND} \backslash \mathsf{IND}(\Sigma \cup \{C_1, C_2\})$.
ii. $\Sigma \models C \doteq \bot$ *iff* $\Sigma \models a\!:\!\neg C$, *for some* $a \in \mathsf{IND} \backslash \mathsf{IND}(\Sigma \cup \{C\})$.
iii. $\Sigma \models \bot$ *iff* $\Sigma \models a\!:\!C$, *for some* $C \in \mathsf{CON} \backslash \mathsf{CON}(\Sigma)$.
iv. $\Sigma \models (a, b)\!:\!R$ *iff* $\Sigma \models a\!:\!\exists R.\{b\}$.

A more delicate reduction is related to the transformation of T-Box and A-Box reasoning into *pure reasoning*, i.e., inference from an empty knowledge base. This is similar to what in classical logic is known as a deduction theorem. For example, for first-order logic the following property holds.

THEOREM 2.7. [Deduction theorem for first-order logic] *Let* $\Sigma \cup \{\varphi, \psi\}$ *be a set of sentences of first-order logic, then*

$$\Sigma \cup \{\varphi\} \models \psi \textit{ iff } \Sigma \models \varphi \rightarrow \psi.$$

If Σ is a finite set or if the logic is compact, then $\Sigma \models \varphi$ can be reduced to $\{\} \models \varphi'$ by iterating the application of the deduction theorem. But a property like the deduction theorem does not necessarily hold. Here is where the restriction to simple and acyclic definitions we discussed in Section 2.2.2 comes to help. Clearly, with such restricted definitions we can do the following.

- Let $\Sigma = \langle T \cup \{B \sqsubseteq C\}, A\rangle$ be a knowledge base and let $B' \notin \mathsf{CON}(\Sigma)$, then

  $$\langle T \cup \{B \sqsubseteq C\}, A\rangle \models a\!:\!D \text{ iff } \langle T \cup \{B \doteq C \sqcap B'\}, A\rangle \models a\!:\!D.$$

  This is known as *completion* of definitions.
- Let $\Sigma = \langle T \cup \{B \doteq C\}, A\rangle$ be a knowledge base, then

  $$\langle T \cup \{B \doteq C\}, A\rangle \models a\!:\!D \text{ iff } \langle T[B/C], A[B/C]\rangle \models a\!:\!D[B/C].$$

  This is known as *unfolding* of definitions.

By repeatedly applying completion and unfolding of definitions we can transform $\Sigma \models \varphi$ into an equivalent task $\Sigma' \models \varphi'$ where Σ' has an empty T-Box, provided that definitions in Σ are simple and acyclic [Nebel, 1990b]. Notice, though, that the sizes of Σ' and φ' can be exponential with respect to the original Σ and φ. In certain languages, unfolding can be done "on the fly" avoiding this complexity explosion (see [Baader *et al.*, 1994; Lutz, 1999a, 1999b]). It is also straightforward to see that we have a "deduction theorem" for A-Boxes

$$\langle \{\}, A \cup \{b{:}B\}\rangle \models c{:}C \text{ iff } \langle \{\}, A\rangle \models \neg(b{:}B) \sqcup c{:}C. \tag{2.1}$$

And similarly for assertions $(a, b) : R$. But notice that standard description languages do not allow the Boolean combination of assertions as we did in (2.1). As before, we run into problems with some of the syntactical restrictions imposed by DLs. We will analyze the matter in detail in Chapter 4.

2.4 Constraint Systems

Even though we have formally discussed the different reasoning tasks, up to now we have said nothing about algorithms to decide any of them. Given that we will focus on languages extending $\mathcal{ALC}$, and the properties we listed in Proposition 2.6, we only need an algorithm to decide instance checking.

Calculi based on *constraint systems* were introduced by Schmidt-Schauss and Smolka in [1991] to decide satisfiability for empty knowledge bases in $\mathcal{ALC}$ and its sublanguages. Later, the framework was extended to decide different reasoning tasks for a variety of languages [Donini *et al.*, 1991, 1992, 1997; Baader and Hollunder, 1991]. Constraint systems are generalizations of tableau calculi [Ladner, 1977; Fitting, 1983; Rautenberg, 1983; Goré, 1999]. The constraint system we present below is from [Schaerf, 1994].

We assume without loss of generality that roles and concepts are given in negation normal form (i.e., negation is applied only to atomic concepts or to expressions of the form $\{a_1, \ldots, a_n\}$) and, furthermore, that the $^{-1}$ operator is only applied to atomic roles. Let VAR be a countably infinite set disjoint from IND, a *constraint* is a formula of one of the following forms

$$s{:}C \mid (s, t){:}R \mid s \not\doteq t \mid \forall x.x{:}C,$$

where $s, t \in \mathsf{IND} \cup \mathsf{VAR}$, $C \in \mathsf{CON}(\mathcal{L})$ and $R \in \mathsf{ROL}(\mathcal{L})$.

Let $\mathcal{I}$ be an interpretation, an $\mathcal{I}$*-assignment* is a function α that maps every variable in VAR to an element of $\Delta^{\mathcal{I}}$ and every individual a in IND to $a^{\mathcal{I}}$. We use pairs $\langle\mathcal{I}, \alpha\rangle$ to define *satisfiability of constraints*. Let $s^{\langle\mathcal{I},\alpha\rangle}$ be $s^{\mathcal{I}}$ if $s \in \mathsf{IND}$ and $\alpha(s)$ if $s \in \mathsf{VAR}$,

- $\langle\mathcal{I}, \alpha\rangle \models s{:}C$ iff $s^{\langle\mathcal{I},\alpha\rangle} \in C^{\mathcal{I}}$,
- $\langle\mathcal{I}, \alpha\rangle \models (s, t){:}R$ iff $(s^{\langle\mathcal{I},\alpha\rangle}, t^{\langle\mathcal{I},\alpha\rangle}) \in R^{\mathcal{I}}$,
- $\langle\mathcal{I}, \alpha\rangle \models s \not\doteq t$ iff $s^{\langle\mathcal{I},\alpha\rangle} \neq t^{\langle\mathcal{I},\alpha\rangle}$,
- $\langle\mathcal{I}, \alpha\rangle \models \forall x.x : C$ iff $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$.

A constraint φ is *satisfiable* if there is an interpretation $\mathcal{I}$ and an $\mathcal{I}$-assignment α such that $\langle\mathcal{I}, \alpha\rangle \models \varphi$. A *constraint system* $\mathcal{S}$ is a finite, non-empty set of constraints. A pair $\langle\mathcal{I}, \alpha\rangle$ *satisfies* $\mathcal{S}$ if $\langle\mathcal{I}, \alpha\rangle$ satisfies every constraint in $\mathcal{S}$, in which case we say that $\mathcal{S}$ is *satisfiable*. A knowledge base $\Sigma = \langle T, A\rangle$ in a language with the $\mathcal{C}$ and $\mathcal{U}$ constructors can be easily translated into a constraint system $\mathcal{S}_\Sigma$ by taking

$$\mathcal{S}_\Sigma = A \cup \{\forall x.x{:}\neg C \sqcup D \mid C \sqsubseteq D \in T\}.$$

The following proposition is immediate (notice the relation with Proposition 2.6).

PROPOSITION 2.8. *Given a knowledge base Σ, $C, D \in \mathsf{CON}(\mathcal{L})$, $R \in \mathsf{ROL}(\mathcal{L})$, $a, b \in \mathsf{IND}$, and $x \in \mathsf{VAR}$, the following equivalences hold*

i. $\Sigma \models C \sqsubseteq D$ *iff* $\mathcal{S}_\Sigma \cup \{x : C \sqcap \neg D\}$ *is unsatisfiable.*
ii. $\Sigma \models C \doteq \bot$ *iff* $\mathcal{S}_\Sigma \cup \{x : C\}$ *is unsatisfiable.*
iii. $\Sigma \models \bot$ *iff* $\mathcal{S}_\Sigma$ *is unsatisfiable.*
iv. $\Sigma \models (a, b) : R$ *iff* $\mathcal{S}_\Sigma \cup \{a : \neg \exists R.\{b\}\}$ *is unsatisfiable.*
v. $\Sigma \models a : C$ *iff* $\mathcal{S}_\Sigma \cup \{a : \neg C\}$ *is unsatisfiable.*

In what follows, we will introduce a set of *completion rules* which, when applied to a constraint system $\mathcal{S}$, returns a constraint system $\mathcal{S}'$ such that $\mathcal{S}$ is satisfiable if and only if $\mathcal{S}'$ is. Furthermore, by repeatedly applying completion rules we will eventually either reach a *complete* constraint system where no further rules can be applied (implying that the system is satisfiable), or we will reach a clash (signaling that the system is contradictory). We first need some general definitions.

DEFINITION 2.9. Let $\mathcal{S}$ be a constraint system, $x \in \mathsf{VAR}$, $s, t \in \mathsf{VAR} \cup \mathsf{IND}$, and $R \in \mathsf{ROL}(\mathcal{L})$. Then $\mathcal{S}[x/s]$ is the constraint system obtain by replacing each occurrence of x by s. We say that t is a *direct R-successor* of s in $\mathcal{S}$, if $(s, t) : R \in \mathcal{S}$, and we define *direct R-predecessors* similarly. Furthermore, t is a *direct successor* (*direct predecessor*) of s in $\mathcal{S}$ if it is a direct R-successor (direct R-predecessor) for some R. The *successor* (*predecessor*) relation is defined as the transitive closure of the direct successor (direct predecessor) relation. We say that s and t are *separated* in $\mathcal{S}$ if $s \not\doteq t \in \mathcal{S}$. Finally, we say that t *is a filler of R for s in* $\mathcal{S}$ if either t is an R-successor of s in $\mathcal{S}$, or one of the following conditions holds

- R is an atomic role and $(t, s) : R^{-1} \in \mathcal{S}$,
- R is of the form S^{-1} and $(t, s) : S \in \mathcal{S}$, or
- R is $R_1 \sqcap R_2$ and t is a filler of R_1 and R_2 for s in $\mathcal{S}$.

Figure 2.2 lists the different completion rules. The full set of rules handles the language $\mathcal{ALCNOBRI}$ with non-empty T- and A-boxes, and can, therefore, also handle sublanguages and simpler forms of knowledge bases.

What remains to do is to define the notion of a clash in a constraint system.

DEFINITION 2.10. [Clash] We say that a constraint system $\mathcal{S}$ contains a *clash* if one of the following conditions holds.

i. For some s, $s : \bot \in \mathcal{S}$.
ii. For some s and some atomic concept A, $\{s : A, s : \neg A\} \subseteq \mathcal{S}$.
iii. For some s, $\{s : (\leq n\ R)\} \cup \{(s, t_i) : R \mid 1 \leq i \leq n + 1\} \cup \{t_i \not\doteq t_j \mid 1 \leq i < j \leq n + 1\} \subseteq \mathcal{S}$.
iv. For some s and $n > m$, $\{s : (\geq n\ R), s : (\leq m\ R)\} \subseteq \mathcal{S}$.
v. For some s, $s : \{a_1, \ldots, a_n\} \in \mathcal{S}$, and $s \neq a_j$ for all j.
vi. For some s, $s : \neg\{a_1, \ldots, a_n\} \in \mathcal{S}$, and $s = a_j$ for some j.

In [Schaerf, 1994], the following completeness result is stated.

- $\mathcal{S} \rightarrow_{\sqcap} \{s\!:\!C_1, s\!:\!C_2\} \cup \mathcal{S}$
  if $s\!:\!C_1 \sqcap C_2$ is in $\mathcal{S}$, and either $s\!:\!C_1$ or $s\!:\!C_2$ is not in $\mathcal{S}$;
- $\mathcal{S} \rightarrow_{\sqcup} \{s\!:\!D\} \cup \mathcal{S}$
  if $s\!:\!C_1 \sqcup C_2$ is in $\mathcal{S}$, neither $s\!:\!C_1$ nor $s\!:\!C_2$ is in $\mathcal{S}$,
  and $D = C_1$ or $D = C_2$;
- $\mathcal{S} \rightarrow_{\exists} \{(s, y)\!:\!R, y\!:\!C\} \cup \mathcal{S}$
  if $s\!:\!\exists R.C$ is in $\mathcal{S}$, there is no t such that t is a direct R-successor
  of s in $\mathcal{S}$ and $t\!:\!C$ is in $\mathcal{S}$, and y is a new variable;
- $\mathcal{S} \rightarrow_{\forall} \{t\!:\!C\} \cup \mathcal{S}$
  if $s\!:\!\forall R.C$ is in $\mathcal{S}$, t is a filler of R for s, and $t\!:\!C$ is not in $\mathcal{S}$;
- $\mathcal{S} \rightarrow_{\geq} \{(s, y_1)\!:\!R, \ldots, (s, y_n)\!:\!R\} \cup \{y_i \not\doteq y_j \mid 1 \leq i < j \leq n\} \cup \mathcal{S}$
  if $s\!:\!(\geq n\ R)$ is in $\mathcal{S}$, there do not exist n pairwise separated fillers
  of R for s in $\mathcal{S}$, and $y_1, \ldots, y_n$ are new variables;
- $\mathcal{S} \rightarrow_{\leq} \mathcal{S}[t/z]$
  if $s\!:\!(\leq n\ R)$ is in $\mathcal{S}$, s has more than n fillers of R for s
  and t, z are two fillers of R for s that are not separated;
- $\mathcal{S} \rightarrow_{\mathcal{O}} \mathcal{S}[x/a_i]$
  if $x\!:\!\{a_1, \ldots, a_n\}$ is in $\mathcal{S}$ and $1 \leq i \leq n$;
- $\mathcal{S} \rightarrow_{\mathcal{B}} \{(s, a)\!:\!R\} \cup \mathcal{S}$
  if $s\!:\!\exists R.\{a\}$ is in $\mathcal{S}$, and $(s, a)\!:\!R$ is not in $\mathcal{S}$;
- $\mathcal{S} \rightarrow_{\mathcal{R}} \{(s, t)\!:\!R_1, (s, t)\!:\!R_2\} \cup \mathcal{S}$
  if $(s, t)\!:\!R_1 \sqcap R_2$ is in $\mathcal{S}$, and either $(s, t)\!:\!R_1$ or $(s, t)\!:\!R_2$ is not in $\mathcal{S}$;
- $\mathcal{S} \rightarrow_{-1} \{(t, s)\!:\!R\} \cup \mathcal{S}$
  if $(s, t)\!:\!R^{-1}$ is in $\mathcal{S}$, and $(t, s)\!:\!R$ is not in $\mathcal{S}$;
- $\mathcal{S} \rightarrow_{\forall x} \{s\!:\!C\} \cup \mathcal{S}$
  if $\forall x.x\!:\!C$ is in $\mathcal{S}$, s appears in $\mathcal{S}$, and $s\!:\!C$ is not in $\mathcal{S}$.

Figure 2.2: Completion rules

Theorem 2.11. *Let $\mathcal{S}$ be a constraint system, then it contains a clash only if $\mathcal{S}$ is not satisfiable, and a complete constraint system $\mathcal{S}$ is satisfiable if it contains no clash.*

We can prove that instance checking (and hence all other reasoning tasks we have introduced) for the full language $\mathcal{ALCNOBRI}$ is decidable (actually solvable in NExpTime). We will reduce it to satisfiability in C^2 (recall our discussion in Section 1.2.2) by providing a pair of mutually recursive translation functions ST_x and ST_y as we did in Definition 1.19, and following ideas from [Borgida, 1996].

Proposition 2.12. *The instance checking problem for $\mathcal{ALCNOBRI}$ is decidable, as it can be reduced to satisfiability in C^2.*

Proof. We define ST_x, ST_y being identical but swapping the positions of x and y. We first translate complex roles as follows. Remember that we assumed that the inverse operator $^{-1}$ was applied only to atomic roles. We consider each atomic role R in ROL as a binary predicate symbol in the first-order language.

$$\begin{array}{rcl} ST_x(R) & = & R(x,y),\ \text{for } R \text{ an atomic role} \\ ST_x(R^{-1}) & = & R(y,x),\ \text{for } R \text{ an atomic role} \\ ST_x(R_1 \sqcap R_2) & = & ST_x(R_1) \wedge ST_x(R_2). \end{array}$$

We can now provide the translation of complex concepts. Many constructs can be simulated in $\mathcal{ALCNOBRI}$ and we do not need to translate them. We consider each atomic concept C in CON as a unary predicate symbol.

$$\begin{array}{rcl} ST_x(C) & = & C(x),\ \text{for } C \text{ an atomic concept} \\ ST_x(\neg C) & = & \neg ST_x(C) \\ ST_x(C_1 \sqcap C_2) & = & ST_x(C_1) \wedge ST_x(C_2) \\ ST_x(\exists R.C) & = & \exists y.(ST_x(R) \wedge ST_y(C)) \\ ST_x({\geq} n\, R) & = & \exists^{\geq n} y.ST_x(R) \\ ST_x(\{a_1,\ldots,a_n\}) & = & \bigvee_{1\leq i\leq n}(x = a_i). \end{array}$$

Finally we translate terminological axioms and assertions ($(a,b){:}R$ can be simulated as $a{:}\exists R.\{b\}$).

$$\begin{array}{rcl} ST_x(C \sqsubseteq D) & = & \forall y.(ST_y(C) \rightarrow ST_y(D)) \\ ST_x(a{:}C) & = & ST_x(C)[x/a]. \end{array}$$

It is easy to prove that for a knowledge base $\Sigma = \langle T, A\rangle$ and an assertion $a{:}C$,

$$\Sigma \models a{:}C \text{ iff } \models \bigwedge_{\varphi\in T\cup A}(ST_x(\varphi)) \rightarrow ST_x(a{:}C).$$

As the formula on the right hand side is in C^2 we are done. QED

The reduction above also makes explicit how DLs are fragments of first-order logic. Another interesting fact that comes out from the reduction is that reasoning tasks in description logics are recast as validity of *sentences*. Or in other words, the *local* evaluation which is traditional in modal languages (satisfiability of a *formula with a free variable* at a given point in the model) has not received attention in the DL community. This is another point we will investigate in Chapter 4.

Going back to constraint systems, we should point out that the completion rules in Figure 2.2 do not constitute a decision method, as an infinite sequence of applications can arise (for example, the trivial interaction between the $\rightarrow_{\leq}$ and $\rightarrow_{\geq}$ rules where a new variable is generated and immediately identified with an existing one).

But the rules of the calculus can be carefully specialized, even to the point of obtaining not only decidability but also very sharp complexity results. For example, replacing the $\rightarrow_{\exists}$ and $\rightarrow_{\geq}$ rules by *trace versions* $\rightarrow_{T\exists}$ and $\rightarrow_{T\geq}$:

- $\mathcal{S} \rightarrow_{T\exists} \{(s,y){:}R, y{:}C\} \cup \mathcal{S}$
  if $s{:}\exists R.C$ is in $\mathcal{S}$, there is no t such that t is an R-successor of s in $\mathcal{S}$ and $t{:}C$ is in $\mathcal{S}$, y is a new variable, and for all constraint $(t,x){:}R$ in $\mathcal{S}$, t is a predecessor of s or $s = t$;

– $\mathcal{S} \rightarrow_{T\geq} \{(s,y_1)\!:\!R,\ldots,(s,y_n)\!:\!R\} \cup \{y_i \not\doteq y_j \mid 1 \leq i < j \leq n\} \cup \mathcal{S}$
if $s\!:\!(\geq n\ R)$ is in $\mathcal{S}$, there do not exist n pairwise separated fillers
of R for s in $\mathcal{S}$, $y_1,\ldots,y_n$ are new variables, and for all
constraint $(t,x)\!:\!R$ in $\mathcal{S}$, t is a predecessor of s or $s = t$;

the decidability of concept satisfiability and subsumption for empty knowledge bases in $\mathcal{ALCNR}$ can be proved [Donini *et al.*, 1997]. In a similar way, the complexity of the different reasoning tasks for sublanguages of $\mathcal{ALCO}$ has been carefully mapped out (see [Schaerf, 1994]). In [Tobies, 2000b] the constraint system approach is used to prove that pure concept satisfiability, i.e., with respect to empty knowledge bases, in $\mathcal{ALCQR}$ is PSpace-complete. The constructor $\mathcal{Q}$ of qualified number restrictions is an extension of $\mathcal{N}$ and we will discuss these results further in Section 4.5.5. And in [Horrocks *et al.*, 1999, 2000b, 2000c; Tobies, 2000a] the decidability of both T-Box and A-Box reasoning is proved for very expressive description logics together with a complexity analysis of sublanguages, also by means of constraint systems.

But perhaps the most important characteristic of constraint systems, from a computational logic perspective, is that *they readily provide implementations*. Implementing the completion rules of Figure 2.2 is simple, and many DL theorem provers, like for example FACT [Horrocks, 1999], DLP [Patel-Schneider, 1998] or RACE [Haarslev and Möller, 1999], are based on this technique. These systems have been highly optimized and are able to cope with extensive knowledge bases in very expressive description languages.

Chapter 3

Introducing Hybrid Logics

hybrid: *n. an organism which is the offspring of a union between different races, species, genera or varieties.*

from "The Wordsworth Concise English Dictionary"

3.1 On Naming

There is an asymmetry at the heart of modal logic. Consider the definition of the classical $\Diamond$ modality (see Definition 1.16):

$$\mathcal{M}, m \Vdash \Diamond\varphi \text{ iff for some } m' \in M \text{ such that } R(m, m') \text{ holds, } \mathcal{M}, m' \Vdash \varphi.$$

We see that formulas are evaluated *at a given state* in the model and that their truth values depend on the value of formulas in some other *related states*. And still, nothing in modal syntax gets to grips with *the states themselves*. States have no names, and no rights, even though they carry out the main trust of the work in defining the meaning of a modal language.

In their simplest form, hybrid languages are modal languages which solve this "reference problem" at the root, by introducing special symbols to explicitly name the states in the model. The beauty lies in the fact that these new symbols, which we call *nominals*, enter the stage gracefully: no big change is needed, simply add a new sort of atomic symbols $\mathsf{NOM} = \{i, j, k, \ldots\}$ disjoint from the set PROP of propositional symbols and let them combine freely in formulas. For example

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q) \tag{3.1}$$

is a well formed formula. Now for the important twist: because nominals are supposed to explicitly stand for states in the model, they should be denoted by *singleton sets*. In other words, if i is a nominal, then i should be true at a unique point in any model $\mathcal{M}$. Once we have taken this step, the whole landscape changes. For example, (3.1) becomes a validity: let $\mathcal{M}$ be a model, $m \in M$ and suppose $\mathcal{M}, m \Vdash \Diamond(i \wedge p) \wedge \Diamond(i \wedge q)$, then

$$\mathcal{M}, m' \Vdash i \wedge p \text{ for some } R\text{-successor } m' \text{ of } m, \text{ and}$$
$$\mathcal{M}, m'' \Vdash i \wedge q \text{ for some } R\text{-successor } m'' \text{ of } m.$$

But because i is a nominal, it is true at a unique point in M. Hence $m' = m''$ and we have $\mathcal{M}, m \Vdash \Diamond(p \wedge q)$. Notice that instead, if we change i for a propositional symbol r, the formula $(r \wedge p) \wedge \Diamond(r \wedge q) \rightarrow \Diamond(p \wedge q)$ can be falsified.

Once we have nominals, another interesting idea immediately suggests itself: why not introduce an operator that allows us to jump to the point named by a nominal? This is what the $@_i\varphi$ (read "at i, φ") formula does: it moves the point of evaluation to the state named by i and checks whether φ is true there. $@_i\varphi$ is Prior's $T(i, \varphi)$ construct ("φ is true at time i"), which he used to define his "third grade tense logics" [Prior, 1967]. It is also the $\mathsf{Holds}(i, \varphi)$ operator introduced in [Allen, 1984] for temporal representation in AI. The same operator is used in the "Topological Logic" of Rescher and Urquhart [1971].

The @ operator has many nice logical properties. For a start, it is a normal modal operator as it satisfies distributivity over $\rightarrow$ $(@_i(\varphi \rightarrow \psi) \rightarrow (@_i\varphi \rightarrow @_i\psi))$ and the necessitation rule (from $\vdash \varphi$ infer $\vdash @_i\varphi$). It is also self dual: $@_i\varphi \leftrightarrow \neg @_i \neg\varphi$ is valid. But most importantly, it works as a bridge between semantics and syntax. The intuition here is the following connection

$$\mathcal{M}, w \Vdash \varphi \text{ iff } \mathcal{M} \Vdash @_i\varphi,$$

where i is a nominal naming w. In other words, @ allows the *satisfiability relation itself* to be talked about in the object language. For this reason, @ is sometimes called the *satisfiability operator*.

Once we have realized the potential provided by direct reference to specific points in the model, the way lies open for further enrichments. The most obvious is to regard nominals not as names but as *variables over individual states*, and to add quantifiers. That is, we would be able to write formulas like

$$\varphi := \forall y.\Diamond y.$$

The translation of φ into first-order logic is $\forall y.\exists z.(R(x, z) \wedge z = y)$ or, simply, $\forall y.R(x, y)$, forcing m to be related to all other elements in the domain whenever $\mathcal{M}, m \Vdash \varphi$. But the $\forall$ quantifier is too expressive for our purposes. As discussed in [Blackburn and Seligman, 1998], even the minimal hybrid language extended with the universal quantifier (i.e., no nominals or @ but just state variables and $\forall$) is undecidable. Moreover, $\forall$ and @ give us already the full expressive power of the one-free-variable fragment of first-order logic (see the end of Section 3.3).

However, the $\forall$ quantifier is historically important; it has been introduced on several occasions, for quite different purposes. The earliest treatments are probably those of [Prior, 1967, 1968], and [Bull, 1970]. About fifteen years later (and independently of Prior and Bull) Passy and Tinchev hybridized propositional dynamic logic with the help of $\forall$. They remark that the idea of $\forall$ was suggested to them by Skordev (who in turn was inspired by certain investigations in recursion theory). The idea of binding variables to points underlies much current work on hybrid languages. The recent PhD thesis of Tzakova [1999a] explores very expressive hybrid languages with binding operators in detail, both axiomatically and by means of tableaux systems.

The $\forall$ quantifier is very "classical." If we think modally, and remember again that evaluation of modal formulas takes place *at a given point*, a different kind of binder is born. The $\downarrow$ binder binds variables to points but, unlike $\forall$, it binds to the *current* point. In essence, it enables us to create a name for the here-and-now, and lets us refer to it later in the formula. For example, if we now write

$$\varphi := {\downarrow}y.\Diamond y$$

we have that $\mathcal{M}, m \Vdash \varphi$ will force only the point m to be reflexive on R. The intuitive reading is quite straightforward: the formula says "call the actual point x and ensure that x is reachable." The difference between $\forall$ and $\downarrow$ is subtle, but important. $\forall$ is *global*, as formulas containing $\forall$ are not preserved under generated submodels [Blackburn and Seligman, 1998]. On the other hand, $\downarrow$ is *intrinsically local* and, as we will show in Chapter 6 (Theorem 6.10) it characterizes the operation of taking generated submodels.

Like $\forall$, the $\downarrow$ binder has been independently invented on a number of occasions. For example, [Richards *et al.*, 1989] introduces $\downarrow$ as part of an investigation into temporal semantics and temporal databases, [Sellink, 1994] uses it to aid reasoning about automata, and [Cresswell, 1990] uses it as part of his treatment of indexicality. Nonetheless, none of the systems just mentioned allows the free syntactic interplay of variables with the underlying propositional logic; that is, they make use of $\downarrow$, but in languages that aren't fully hybrid. The earliest paper to introduce it into a fully hybrid language seems to be [Goranko, 1994]. While [Blackburn and Tzakova, 1998a, 1999; Tzakova, 1999a] are the basic references for $\downarrow$, an interesting discussion of this binder as part of a stronger system can be found in [Goranko, 1996].

Note that the @ operator works in perfect coordination with $\downarrow$. Whereas $\downarrow$ "stores" the current point (by binding a variable to it), @ enables us to "retrieve" the information stored by shifting the point of evaluation in the model.

For an introduction to hybrid logics "for the broad audience," read [Blackburn, 2000b]. You will enjoy it.

3.2 Why Hybrid Logics?

The question is ambiguous, as is usually the case in natural language. But two of the different possible interpretations deserve to be commented upon.

First, why are these logics *called* hybrid? One explanation comes from Prior's work. Following McTaggart's [1908] analysis of time in terms of the A-series of past, present and future and the B-series of earlier and later, Prior discusses two logical systems: the U-calculus aims to capture the properties of the A-series and takes variables ranging over instants as primitive, while the T-calculus examines tenses and takes variables ranging over propositions. In Chapter V.6 of [Prior, 1967], he actually proposes a way to develop the U-calculus *inside* the T-calculus, and for this he allows the instant-variables to be used together with propositional symbols. He will call this step "the third grade of tense-logical involvement" in [Prior, 1977, Chapter XI], where instant-variables are treated as representing (special) propositions. From this perspective, the terms hybrid applies to the "confusion" of terms (the variables over instants) with formulas (the proposition symbols). But a hybrid behavior also shows up in a different aspect of hybrid logics: the techniques used when dealing with hybrid languages are a true mixture of modal and first-order ideas, as we will see throughout the thesis.

But probably more interesting is to wonder why hybrid logics are *useful*. We will answer this by example and discuss the connection between hybrid languages and many other fields. In most cases, hybrid languages can be used to provide a flexible and effective background theory.

Hybrid Logics and Proof Theory. One way to see why hybrid languages are proof-theoretically natural, is to observe that nominals and @ can capture the main ideas of labeled deduction. In [Gabbay, 1996] the notation $l : \varphi$ is introduced, where the meta-linguistic symbol : associates the meta-linguistic label l with the object language formula φ. Labeled deduction proceeds by manipulating such labels to guide proof search. The first ideas concerning labeled deduction were probably introduced by Wadge [1975] in his system of natural deduction for the relational calculus. Hybrid languages "internalize" labeled deduction into the object language: nominals are essentially object-level labels, and the formula $@_l\varphi$ asserts in the object language what $l : \varphi$ asserts in the meta-language [Blackburn, 2000a].

Tzakova [1999b] presents a general approach to hybrid proof theory using Fitting-style prefix calculi. When the underlying modal logic is temporal logic, even more flexibility is possible: [Demri, 1999] presents a sequent system for nominal tense logic without @ that has much in common with the @-based internalized labeled deductive systems. In a different line, Seligman's work [1991, 1997] deals with strong ($\forall$-based) systems, but many of the key ideas underlying hybrid deduction (in particular, the deductive significance of @) were first explored in these papers.

Hybrid Logics and Model Theory. Hybrid logics have also a very well developed model theory. Part of this comes by inheritance from modal logics, but as we remarked above, it can amalgamate nicely with first-order ideas. One example is the "Henkin construction of canonical models" used in completeness proofs for hybrid languages [Passy and Tinchev, 1991; Blackburn and Tzakova, 1998a]. These techniques can be used to establish general results, like automatic completeness for pure formulas [Blackburn and Tzakova, 1999], or the general interpolation results we prove in Section 6.2.

Hybrid Logics and Temporal Logic. As indicated in the work of Prior and Bull, hybrid languages allow us to make explicit references to specific times (days, dates, years, etc.), and also to cope with temporal indexicals (such as yesterday, today, tomorrow and now). In addition, they can define many temporally relevant frame properties (such as irreflexivity, asymmetry and trichotomy) that ordinary modal languages cannot express. Furthermore, when nominals and @ are added to interval-based logic, the result is a $\mathsf{Holds}(t, \varphi)$-driven interval logic in the style of those introduced into AI by James Allen [1984], with @ playing the role of Holds (see [Areces *et al.*, 2000a]). Because hybrid logics make temporal reference possible, they remove the most serious obstacle to a modal analysis of temporal representation and reasoning. Nominal tense logics have been studied in detail in [Blackburn, 1990].

Hybrid Logics and PDL. Hybrid languages where rediscovered many years after the work of Prior and Bull by a group of logicians at the Sofia University in Bulgaria. Gargov, Passy and Tinchev were interested on neat axiomatizations of operations in PDL, and they realized that while certain operations like for example the union of programs were easy to capture ($\langle\alpha\cup\beta\rangle p \leftrightarrow \langle\alpha\rangle p \vee \langle\beta\rangle p$), a simple axiomatization of others like intersection or complement called for extra expressive power. In [1985a], Passy and

Tinchev show that the addition of nominals is enough to provide succinct and very natural characterization of these operations. The addition of other kind of "constants" to the language permits the representation of notions like determinism and looping [Gargov and Passy, 1988]. In addition, the work of the Sofia school showed how nominals could also be used to simplify the construction of models during completeness proof [Passy and Tinchev, 1985b]. See [Passy and Tinchev, 1991] for an excellent overview on combinatory dynamic logics.

Hybrid Logics and Natural Language. Hybrid languages are also a powerful resource for studying indexicality in natural language, as an alternative to the more classical use of multi-dimensional modal logic. In the multi-dimensional modal approach, formulas are evaluated at sequences of points, where one point of the sequence is thought of as the point of evaluation, while the others are used as memory locations to store references [Kamp, 1971; Vlach, 1973; Gabbay, 1976; Cresswell, 1990, 1996].

Hybrid languages move multi-dimensional logic's sequence of evaluation points from the meta-language to the object language, with hybrid variables acting as names for indices. See [Blackburn, 1994] for a very clear exposition. Moreover, when equipped with the @ operator, hybrid languages offer the 'de-scoping' behavior typical of such multi-dimensional operators as here and there. There are also links between hybrid logic and mathematical aspects of multi-dimensional modal logic, particularly the multi-dimensional modal perspective on cylindric algebra (cf. [Marx and Venema, 1997]). As we will see in Definition 3.3, $\downarrow$ and @ can be considered as explicit substitution devices.

Hybrid Logics and Feature Logic. The mechanisms underlying PATR-II and other unification-based approaches to grammar are based on the use of attribute value matrices (AVMs) with "tags" to indicate re-entrance of feature structures [Rounds, 1997]. Given the tight connection between AVMs and deterministic multi-modal logic half of the work is done. In addition it is easy to account for re-entrance in the setting of hybrid logic: "tags" are simply nominals. The hybrid approach to feature logic differs from that taken in Kasper-Rounds logic in a number of respects. Kasper-Rounds logic is essentially a fragment of deterministic propositional dynamic logic with intersection, encoding re-entrance in a less direct way than nominals. See [Blackburn and Spaan, 1993; Blackburn, 1993; Reape, 1994].

Hybrid Logics and Information Systems. Nominals have turned up in yet another setting, namely the Polish tradition of modal logics for information systems initiated by Pawlak (see [Orłowska, 1997]). Themes in this tradition include the development of modal logics of similarity (or relative similarity) and there are strong links with the tradition of rough-set theory. Konikowska [1997] has introduced nominals to such logics. Her work is motivated primarily by proof-theoretical considerations: the ability to name states leads to smoother and more intuitive proof systems.

3.3 Hybrid Details

We are now about to start our formal work on hybrid logics. As we did with description logics, it is best if we introduce some standard notation to name the different languages we will be dealing with.

The basic hybrid language is $\mathcal{H}_{\mathsf{N}}$, basic modal logic extended with nominals. We will also consider the extension which adds state variables in addition to nominals and we will call this system $\mathcal{H}_{\mathsf{S}}$. Further extensions will be named by listing the added operators. The most expressive system we will discuss is $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$, the hybrid temporal system extended with the converse (past) and existential modalities, the @ operator and the ↓ binder.

The full $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ language is too expressive for our purposes as already $\mathcal{H}_{\mathsf{S}}(\mathsf{E}, \downarrow)$ can define the $\forall$ quantifier: let A be the dual of E (i.e., $\mathsf{A}\varphi \leftrightarrow \neg\mathsf{E}\neg\varphi$), then $\forall x.\varphi := \downarrow y.\mathsf{A}\downarrow x.\mathsf{A}(y \rightarrow \varphi)$ for y a variable not occurring in φ. By the way, $\mathcal{H}_{\mathsf{S}}(\mathsf{E}, \downarrow)$ can also define the past operator: $\langle R^{-1}\rangle\varphi := \downarrow y.\mathsf{E}(\Diamond(\varphi \wedge y))$ for y a variable not occurring in φ. And $@_i\varphi$ is equivalent to $\mathsf{E}(i \wedge \varphi)$. In other words, $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ is equivalent to $\mathcal{H}_{\mathsf{S}}(\mathsf{E}, \forall)$ and hence too powerful for our enterprise. But we will explore many of its sublanguages in detail.

Definition 3.1. [Syntax] Let $\mathsf{REL} = \{R_1, R_2, \ldots\}$ be a countable set of *relational symbols*, $\mathsf{PROP} = \{p_1, p_2, \ldots\}$ a countable set of *propositional variables*, $\mathsf{NOM} = \{i_1, i_2, \ldots\}$ a countable set of *nominals*, and $\mathsf{SVAR} = \{x_1, x_2, \ldots\}$ a countable set of *state variables*. We assume that these sets are pairwise disjoint. We call $\mathsf{SSYM} = \mathsf{NOM} \cup \mathsf{SVAR}$ the set of *state symbols*, and $\mathsf{ATOM} = \mathsf{PROP} \cup \mathsf{NOM} \cup \mathsf{SVAR}$ the set of *atoms*. The well-formed formulas of the hybrid language $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ in the signature $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR}\rangle$ are

$$\mathsf{FORMS} := \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle R\rangle\varphi \mid \langle R^{-1}\rangle\varphi \mid \mathsf{E}\varphi \mid @_s\varphi \mid \downarrow x.\varphi,$$

where $a \in \mathsf{ATOM}$, $x \in \mathsf{SVAR}$, $s \in \mathsf{SSYM}$, $R \in \mathsf{REL}$ and $\varphi, \varphi_1, \varphi_2 \in \mathsf{FORMS}$. For $T \subseteq \mathsf{FORMS}$, $\mathsf{PROP}(T)$, $\mathsf{NOM}(T)$ and $\mathsf{SVAR}(T)$ denote, respectively, the set of propositional variables, nominals, and state variables which occur in formulas in T.

Note that all types of atomic symbol (i.e., proposition symbols, nominals and state variables) are *formulas*. Further, note that the above syntax is simply that of ordinary (multi-modal) propositional temporal logic extended with clauses for $\mathsf{E}\varphi$, $@_s\varphi$ and $\downarrow x_j.\varphi$. Finally, the difference between nominals and state variables is simply this: nominals cannot be bound by ↓, whereas state variables can.

The notions of *free* and *bound* state variable are defined as in first-order logic, with ↓ as the only binding operator. Similarly, other syntactic notions (such as *substitution*, and of a state symbol t *being substitutable for* x *in* φ) are defined just like the corresponding notions in first-order logic. A *sentence* is a formula containing no free state variables. Furthermore, a formula is *pure* if it contains no propositional variables, and *nominal-free* if it contains no nominals. Now for the semantics, in the rest of the chapter we assume fixed a signature $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR}\rangle$.

Definition 3.2. [Semantics] A (hybrid) *model* $\mathcal{M}$ is a triple $\mathcal{M} = \langle M, \{R_i\}, V\rangle$ such that M is a non-empty set, $\{R_i\}$ is a set of binary relations on M, and $V : \mathsf{PROP} \cup \mathsf{NOM} \rightarrow Pow(M)$ is such that for all nominals $i \in \mathsf{NOM}$, $V(i)$ is a singleton subset of M. We usually call the elements of M *states* or *worlds*, R_i the *accessibility relations*, and V the *valuation*.

An *assignment* g for $\mathcal{M}$ is a mapping $g : \mathsf{SVAR} \to M$. Given an assignment g, we define g^x_m (an *x-variant of g*) by $g^x_m(x) = m$ and $g^x_m(y) = g(y)$ for $x \neq y$.

Let $\mathcal{M} = \langle M, \{R_i\}, V\rangle$ be a model, $m \in M$, and g an assignment. For any atom a, let $[V, g](a) = \{g(a)\}$ if a is a state variable, and $V(a)$ otherwise. Then the *satisfiability relation* is defined as follows

$$\begin{array}{rcl}
\mathcal{M}, g, m \Vdash \top & & \text{always} \\
\mathcal{M}, g, m \Vdash a & \text{iff} & m \in [V, g](a),\ a \in \mathsf{ATOM} \\
\mathcal{M}, g, m \Vdash \neg\varphi & \text{iff} & \mathcal{M}, g, m \nVdash \varphi \\
\mathcal{M}, g, m \Vdash \varphi_1 \wedge \varphi_2 & \text{iff} & \mathcal{M}, g, m \Vdash \varphi_1 \text{ and } \mathcal{M}, g, m \Vdash \varphi_2 \\
\mathcal{M}, g, m \Vdash \langle R\rangle\varphi & \text{iff} & \exists m'.(R(m, m') \ \&\ \mathcal{M}, g, m' \Vdash \varphi) \\
\mathcal{M}, g, m \Vdash \langle R^{-1}\rangle\varphi & \text{iff} & \exists m'.(R(m', m) \ \&\ \mathcal{M}, g, m' \Vdash \varphi) \\
\mathcal{M}, g, m \Vdash \mathsf{E}\varphi & \text{iff} & \exists m'.(\mathcal{M}, g, m' \Vdash \varphi) \\
\mathcal{M}, g, m \Vdash @_s\varphi & \text{iff} & \mathcal{M}, g, m' \Vdash \varphi, \text{ where } [V, g](s) = \{m'\},\ s \in \mathsf{SSYM} \\
\mathcal{M}, g, m \Vdash {\downarrow} x.\varphi & \text{iff} & \mathcal{M}, g^x_m, m \Vdash \varphi.
\end{array}$$

If $\mathcal{M}$ and g are understood from the context, we simply write $m \Vdash \varphi$ for $\mathcal{M}, g, m \Vdash \varphi$. We write $\mathcal{M}, g \Vdash \varphi$ iff for all $m \in M$, $\mathcal{M}, g, m \Vdash \varphi$; and $\mathcal{M} \Vdash \varphi$ iff for all g, $\mathcal{M}, g \Vdash \varphi$. These notions extend to sets of formulas in the standard way.

A formula φ is *satisfiable* if there is a model $\mathcal{M}$, an assignment g on $\mathcal{M}$, and a world $m \in M$ such that $\mathcal{M}, g, m \Vdash \varphi$. A formula φ is *valid* if for all models $\mathcal{M}$, $\mathcal{M} \Vdash \varphi$. A formula φ is a *local consequence* of a set of formulas T if for all models $\mathcal{M}$, valuations g, and points $m \in M$, $\mathcal{M}, g, m \Vdash T$ implies $\mathcal{M}, g, m \Vdash \varphi$. A formula φ is a *global consequence* of a set of formulas T if for all models $\mathcal{M}$, $\mathcal{M} \Vdash T$ implies $\mathcal{M} \Vdash \varphi$. We denote local consequence by $T \models^{loc} \varphi$ and global consequence by $T \models^{glo} \varphi$. As in ordinary propositional modal logic, local consequence is strictly stronger than global consequence. When T is the empty set $\{\} \models^{glo} \varphi$ iff $\{\} \models^{loc} \varphi$, and we will write $\models \varphi$.

The first six clauses in the definition of the satisfiability relation define essentially the standard Kripke satisfiability relation for propositional temporal logic; the only difference is that whereas the standard definition relativizes semantic evaluation to worlds m, we relativize to variable assignments g as well.

Note that the clause for atoms covers all types of atomic symbols (propositional variables, nominals, and state variables) and that given any model $\mathcal{M}$ and assignment g, any state symbol (whether it is a nominal or a state variable) will be forced at a *unique* world. As promised, $\downarrow$ binds state variables to the state where evaluation is being performed (the *current world*), and $@_s$ shifts evaluation to the state named by s.

Just as in first-order logic, if φ is a *sentence* it is irrelevant which assignment g is used to perform evaluation: $\mathcal{M}, g, m \Vdash \varphi$ for *some* assignment g iff $\mathcal{M}, g, m \Vdash \varphi$ for *all* assignments g. Hence for sentences the relativization to assignments of the satisfiability relation can be dropped, and we simply write $\mathcal{M}, m \Vdash \varphi$ instead of $\mathcal{M}, g, m \Vdash \varphi$. In particular, this is always the case in languages with no state variables, like for example $\mathcal{H}_{\mathsf{N}}$. In these cases we completely forget about assignments.

Notice that we have introduced, in one sweep, the syntax and semantics of all sublanguages of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$. Among the different sublanguages, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ will play a special role, as one of the best behaved systems. It is also a prime example of the hybrid ideas at work, as it includes nominals, the satisfiability operator @ and the local binder ↓. We will take this language as our central system, and study extensions and restrictions as variations on a theme.

3.3.1 Translations

Perhaps the best way to get the feeling of hybrid languages is to see what they have to offer in terms of new fragments of first-order logic, reachable by translation. We will take special care in this case, because we will use the basic results presented in this section in our characterizations of Chapter 6.

We focus on two kinds of signature for first-order logic with equality. First we have (multi) *modal* signatures (familiar from correspondence theory [van Benthem, 1983]) which consist of binary predicates R_i, countably many unary predicates, and no function or constant symbols. Thus, a modal signature has the form $\langle \{R_i\} \cup \mathsf{UREL}, \{\}, \{\}, \mathsf{VAR}\rangle$. A *hybrid* signature is an expansion of the modal signature with countably many constant symbols $\langle \{R_i\} \cup \mathsf{UREL}, \{\}, \mathsf{CONS}, \mathsf{VAR}\rangle$.

Any hybrid model $\mathcal{M} = \langle M, \{R_i\}, V\rangle$ can be regarded as a first-order model over the hybrid signature, for the accessibility relations R_i can be used to interpret the binary predicates R_i, unary predicates can be interpreted by the subsets that V assigns to propositional variables, and constants can be interpreted by the worlds that nominals name. We let the context determine whether we are thinking of first-order or hybrid models, and continue to use the notation $\mathcal{M} = \langle M, \{R_i\}, V\rangle$.

We can extend the standard translation ST to $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$, but we have to be careful now with which variables we will be using, and how we treat equality.

Definition 3.3. [Standard translation for $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$] The mutually recursive functions ST_x and ST_y from the hybrid language $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ over $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR}\rangle$ into first-order logic over the signature $\langle \mathsf{REL} \cup \{P_j \mid p_j \in \mathsf{PROP}\}, \{\}, \mathsf{NOM}, \mathsf{SVAR} \cup \{x, y\}\rangle$ are defined as follows

$$\begin{array}{rcl|rcl}
ST_x(i_j) & = & (x = i_j),\ i_j \in \mathsf{NOM} & ST_y(i_j) & = & (y = i_j),\ i_j \in \mathsf{NOM} \\
ST_x(x_j) & = & (x = x_j),\ x_j \in \mathsf{SVAR} & ST_y(x_j) & = & (y = x_j),\ x_j \in \mathsf{SVAR} \\
ST_x(p_j) & = & P_j(x),\ p_j \in \mathsf{PROP} & ST_y(p_j) & = & P_j(y),\ p_j \in \mathsf{PROP} \\
ST_x(\neg\varphi) & = & \neg ST_x(\varphi) & ST_y(\neg\varphi) & = & \neg ST_y(\varphi) \\
ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi) & ST_y(\varphi \wedge \psi) & = & ST_y(\varphi) \wedge ST_y(\psi) \\
ST_x(\langle R\rangle\varphi) & = & \exists y.(R(x,y) \wedge ST_y(\varphi)) & ST_y(\langle R\rangle\varphi) & = & \exists x.(R(y,x) \wedge ST_x(\varphi)) \\
ST_x(\langle R^{-1}\rangle\varphi) & = & \exists y.(R(y,x) \wedge ST_y(\varphi)) & ST_y(\langle R^{-1}\rangle\varphi) & = & \exists x.(R(x,y) \wedge ST_x(\varphi)) \\
ST_x(\mathsf{E}\varphi) & = & \exists y.ST_y(\varphi) & ST_y(\mathsf{E}\varphi) & = & \exists x.ST_x(\varphi) \\
ST_x(@_s\varphi) & = & (ST_x(\varphi))[x/s] & ST_y(@_s\varphi) & = & (ST_y(\varphi))[y/s] \\
ST_x(\downarrow x_j.\varphi) & = & (ST_x(\varphi))[x_j/x] & ST_y(\downarrow x_j.\varphi) & = & (ST_y(\varphi))[x_j/y].
\end{array}$$

The role of nominals and state variables is clear from the translation. They offer us first-order equality, something which is outside the reach of basic modal languages. The "equality effect" of state symbols is strengthened by the effect of @. Notice that $ST_x(@_st)$ gives us $s = t$, i.e., we can not only claim equality with the point of evaluation, but between any two named points in the model.

The translation above also highlights the interaction between @ and $\downarrow$. The original translation in [Blackburn and Tzakova, 1998a] handles $\downarrow$ as follows

$$ST_x(\downarrow x_j.\varphi) \quad = \quad \exists x_j.(x = x_j \wedge ST_x(\varphi)).$$

Blackburn and Tzakova's translation makes the quantificational effect of $\downarrow$ clear, but our translation draws attention to another perspective: in adding $\downarrow$ and @ we have enriched the modal language with an *explicit substitution operator*. Such operators are used in the study of cylindric algebras, and were added to cylindric modal logic in [Venema, 1994]. The link between $\downarrow$ and explicit substitution can be made even more clear if we expand the first-order language with an explicit substitution operator (like s^i_j in the theory of cylindric algebras) and adjust our definition of ST to take advantage of it. We do this as follows. Add the following clause to the grammar generating the first-order language: if φ is a formula, x is a variable and s is a variable or a constant, then $S^x_s\varphi$ is a formula. Interpret S^x_s as follows:

$$\mathcal{M} \models S^x_s\varphi[g] \Leftrightarrow \begin{cases} \mathcal{M} \models \varphi[g] & \text{for } x = s \\ \mathcal{M} \models \varphi[g^x_{g(s)}] & \text{for } s \text{ a variable, } x \neq s \\ \mathcal{M} \models \varphi[g^x_{s^{\mathcal{M}}}] & \text{for } s \text{ a constant.} \end{cases}$$

Clearly $S^x_s\varphi$ and $\varphi[x/s]$ are equivalent. This extension can be axiomatized by adding the formulas $S^x_x\varphi \leftrightarrow \varphi$ and $S^x_s\varphi \leftrightarrow \exists x.(x = s \wedge \varphi)$ for $x \neq s$ as axiom schemas, to a complete axiomatization of first-order logic with equality

And now we can give transparent translations of $\downarrow$ and @:

$$\begin{aligned} ST_x(\downarrow x_j.\varphi) &= S^{x_j}_x ST_x(\varphi) \\ ST_x(@_s\varphi) &= S^x_s ST_x(\varphi). \end{aligned}$$

Theorems like $\downarrow v.@_v\varphi \leftrightarrow \downarrow v.\varphi$ can be proved immediately in this way, for $ST_x(\downarrow v.@_v\varphi)$ $= S^v_x S^x_v ST_x(\varphi)$, which is equivalent to $S^v_x ST_x(\varphi)$, because $S^v_x S^x_v\varphi \equiv S^v_x S^x_x\varphi \equiv S^v_x\varphi$. However we will stick to our original formulation of ST in what follows.

Proposition 3.4. [ST preserves truth] *Let φ be a hybrid formula, then for all hybrid models $\mathcal{M}$, $m \in M$ and assignments g, $\mathcal{M}, g, m \Vdash \varphi$ iff $\mathcal{M} \models ST_x(\varphi)[g^x_m]$.*

Proof. A straightforward extension of the induction familiar from basic modal logic. The only new cases are $ST_x(\downarrow x_j.\varphi)$ and $ST_x(@_s\varphi)$. But, by its semantic definition, $\mathcal{M}, g, m \Vdash \downarrow x_j.\varphi$ iff $\mathcal{M}, g^{x_j}_m, m \Vdash \varphi$, by induction hypothesis, iff $\mathcal{M} \models ST_x(\varphi)[(g^{x_j}_m)^x_m]$, iff $\mathcal{M} \models (ST_x(\varphi))[x_j/x][g^x_m]$. The argument for $ST_x(@_s\varphi)$ is similar. QED

Another way to understand how the new operators work is by example.

Example 3.5. $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ already offers us considerable expressive power over models. For example we can define the Until operator. Remember that $m \Vdash \mathsf{Until}(\varphi, \psi)$ if there is a successor m' of m where φ is true, and all intermediate states between m and m' satisfy ψ. We define

$$\mathsf{Until}(\varphi, \psi) \ := \ \downarrow x.\langle R\rangle \downarrow y.@_x(\langle R\rangle(y \wedge \varphi) \wedge [R](\langle R\rangle y \rightarrow \psi)).$$

That is, we name the current world x, use $\langle R\rangle$ to move to an accessible world which we name y, and then use @ to jump back to x. We then use the modalities to insist that φ holds at the world named y, and ψ holds in all successors of x that precede this y-labeled world.

Another example is counting: $\downarrow$ and @ are expressive enough to encode expressions of the form "there are at least n R-successors satisfying φ." For $n = 2$ we write:

$$\downarrow x.\langle R\rangle\downarrow x_1.(\varphi \wedge @_x\langle R\rangle\downarrow x_2.(\varphi \wedge @_x\langle R\rangle(x_1 \wedge \neg x_2))).$$

But there is an obvious (and modally natural) limit to the expressive power of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$: any nominal-free sentence is preserved under the formation of *point-generated* (or *rooted*) submodels. That is, if a sentence φ is satisfied at a world m in a model $\mathcal{M}$, and we form a submodel $\mathcal{M}_m$ by discarding from $\mathcal{M}$ all the worlds that are *not* reachable by making a finite (possibly empty) sequence of transitions from m, then $\mathcal{M}_m$ also satisfies φ at m. (The key point to observe is that in any subformula of φ of the form $@_t\psi$, t must be a state variable bound by some previous occurrence of $\downarrow$. As $\downarrow$ binds to the current world, t is bound to some world in the submodel generated by m, thus φ is unaffected by the restriction to $\mathcal{M}_m$.) That is, $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ is genuinely local: only worlds reachable from named points are relevant to semantic evaluation. In Chapter 6 we will return to this observation, show that we have not merely preservation but *invariance*, and that it characterizes the expressivity of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ (see Theorem 6.10). The result can easily be generalized to $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle,@,\downarrow)$ as we discuss in Section 6.1.5, and in this language we can actually do without @.

To end this section, we make explicit our previous informal remark that $\forall$ is too expressive for our purposes. We will show that $\mathcal{H}_{\mathsf{S}}(@,\forall)$ gives us already full first-order expressivity. Consider the following translation from a first-order language with identity (over a hybrid signature) into $\mathcal{H}_{\mathsf{S}}(@,\forall)$:

$$\begin{array}{rcl}
HT(R_i(s,s')) & = & @_s\langle R_i\rangle s' \\
HT(P_j(s)) & = & @_s p_j \\
HT(s = s') & = & @_s s' \\
HT(\neg\varphi) & = & \neg HT(\varphi) \\
HT(\varphi \wedge \psi) & = & HT(\varphi) \wedge HT(\psi) \\
HT(\exists x.\varphi) & = & \exists x.HT(\varphi).
\end{array}$$

We have to prove that the translation preserves satisfiability. We first establish the following proposition:

Proposition 3.6. *Let φ be a hybrid formula obtained from formulas whose main operator is @ by use of $\neg$, $\wedge$ and $\exists$. Then for any model $\mathcal{M}$, any assignment g, and $m, m' \in M$*

$$\mathcal{M}, g, m \Vdash \varphi \textit{ iff } \mathcal{M}, g, m' \Vdash \varphi.$$

Proof. The base case is simple, let $\varphi = @_s\psi$, and let $s^{\mathcal{M}}$ be the denotation of s in $\mathcal{M}$. Then, $\mathcal{M}, g, m \Vdash @_s\psi$ iff $\mathcal{M}, g, s^{\mathcal{M}} \Vdash \psi$ iff $\mathcal{M}, g, m' \Vdash @_s\psi$.

The Booleans are trivial, and for $\varphi = \exists x.\psi$ reason as follows. $\mathcal{M}, g, m \Vdash \exists x.\psi$ iff $\mathcal{M}, g', m \Vdash \psi$ for g' an x-variant of g. By induction hypothesis, $\mathcal{M}, g', m \Vdash \psi$ iff $\mathcal{M}, g', m' \Vdash \psi$, iff $\mathcal{M}, g, m' \Vdash \exists x.\psi$. QED

The following result is now straightforward.

Proposition 3.7. *Let φ be a first-order formula in the hybrid signature. Then for every model $\mathcal{M}$ and any assignment g, $\mathcal{M} \models \varphi[g]$ iff $\mathcal{M}, g \Vdash HT(\varphi)$.*

Proof. The proof is by induction on the complexity of φ. We only provide some of the cases. Notice first that given Proposition 3.6, it doesn't matter in which state of $\mathcal{M}$ we evaluate $HT(\varphi)$. Suppose $\varphi = R_i(s,t)$, and let $s^{\mathcal{M}}$, $t^{\mathcal{M}}$ be the denotations of s and t in $\mathcal{M}$. Then $\mathcal{M} \models R_i(s,t)$ iff $(s^{\mathcal{M}}, t^{\mathcal{M}}) \in R_i$. But $\mathcal{M}, g \Vdash @_s\langle R_i\rangle t$ iff $\mathcal{M}, g, s^{\mathcal{M}} \Vdash \langle R_i\rangle t$, iff there is an R_i-successor of $s^{\mathcal{M}}$ satisfying t. Hence, $\mathcal{M}, g \Vdash @_s\langle R_i\rangle t$ iff $t^{\mathcal{M}}$ is an R_i-successor of $s^{\mathcal{M}}$. For $\varphi = \exists x.\varphi$, the result is also straightforward. $\mathcal{M} \models \exists x.\varphi[g]$ iff $\mathcal{M} \models \varphi[g']$ for g' some x-variant of g. By induction hypothesis, iff $\mathcal{M}, g' \Vdash HT(\varphi)$, iff $\mathcal{M}, g \Vdash \exists x.HT(\varphi)$. QED

3.4 Axiomatizations

$\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ is also very expressive with respect to *frames.* We first need to define this notion and what do we mean by a formula defining a property of frames.

Definition 3.8. A *frame* is a model without a valuation, i.e., a tuple $\mathcal{F} = \langle M, \{R_i\}\rangle$. A formula φ is *valid on a frame* $\mathcal{F} = \langle M, \{R_i\}\rangle$ if for every valuation V on $\mathcal{F}$, every assignment g on $\mathcal{F}$, and every $m \in M$, $\langle \mathcal{F}, V\rangle, g, m \Vdash \varphi$. A formula is *valid on a class of frames* F if it is valid on every frame $\mathcal{F}$ in F. A formula φ *defines a class of frames* if it is valid on precisely the frames in F, and it defines a *property* of frames (for example, transitivity of the accessibility relation) if it defines the class of frames with that property.

In what follows we will mainly discuss mono-modal $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ for simplicity.

Example 3.9. Many properties are definable using pure, nominal-free, sentences:

$\downarrow x.\Box\neg x$	$i \to \neg\Diamond i$	Irreflexivity
$\downarrow x.\Box\Box\neg x$	$i \to \neg\Diamond\Diamond i$	Asymmetry
$\downarrow x.\Box(\Diamond x \to x)$	$i \to \Box(\Diamond i \to i)$	Antisymmetry
$\downarrow x.\Box\downarrow y.@_x\Diamond\Diamond y$	$\Diamond i \to \Diamond\Diamond i$	Density
$\downarrow x.\Box\Box\downarrow y.@_x\Diamond y$	$\Diamond\Diamond i \to \Diamond i$	Transitivity

With the exception of transitivity and density, none of these properties are definable in ordinary modal logic. In Section 6.1.4 we will exactly characterize the classes of frames that pure, nominal-free sentences can define.

[Tzakova, 1999a] provides the following complete axiom system for $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$.

Definition 3.10. [Axiomatization] Let φ, ψ be formulas, v a metavariable over state variables, c a metavariable over nominals, and s, t metavariables over state symbols. The hybrid logic $\mathbf{K}[\mathcal{H}_{\mathsf{S}}(@,\downarrow)]$ is the smallest subset of formulas in $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ containing all instances of propositional logic tautologies, all instances of the following axiom schemas, and closed under the following deduction rules

MP	$\vdash\varphi\to\psi, \vdash\varphi\Rightarrow\vdash\psi$
K	$\Box(\varphi\to\psi)\to(\Box\varphi\to\Box\psi)$
N	$\vdash\varphi\Rightarrow\vdash\Box\varphi$
Q1	$\downarrow v.(\varphi\to\psi)\to(\varphi\to\downarrow v.\psi)$, φ without free occurrences of v
Q2	$\downarrow v.\varphi\to(s\to\varphi[v/s])$, s substitutable for v in φ
Q3	$\downarrow v.(v\to\varphi)\to\downarrow v.\varphi$
Self Dual$_\downarrow$	$\downarrow v.\varphi\leftrightarrow\neg\downarrow v.\neg\varphi$
N$_\downarrow$	$\vdash\varphi\Rightarrow\vdash\downarrow v.\varphi$
K$_@$	$@_s(\varphi\to\psi)\to(@_s\varphi\to@_s\psi)$
Self Dual$_@$	$@_s\varphi\leftrightarrow\neg@_s\neg\varphi$
Elim$_@$	$(s\wedge@_s\varphi)\to\varphi$
Label	$@_s s$
Scope	$@_t@_s\varphi\to@_s\varphi$
N$_@$	$\vdash\varphi\Rightarrow\vdash@_s\varphi$
M$_@$	$@_s\varphi\to\Box@_s\varphi$
Naming	$\vdash\varphi\to\neg c\Rightarrow\vdash\varphi\to\bot$, $c\notin\mathsf{NOM}(\varphi)$
Paste	$\vdash\varphi\to@_s\Box(\psi\to\neg c)\Rightarrow\vdash\varphi\to@_s\Box(\psi\to\bot)$, $s\notin\mathsf{NOM}(\{\varphi,\psi,c\})$.

The completeness result proved in [Blackburn and Tzakova, 1999; Tzakova, 1999a] is very general: not only does this axiomatization generate all valid formulas, but it automatically extends to many stronger logics. The relevant theorems read as follows. Let a *pure schema* be a pure formula where we uniformly replace all occurrences of free state variables and nominals by metavariables over state symbols, and bound variables by metavariables over state variables.

Theorem 3.11. [Tzakova, 1999a, Theorems 67 and 68]

i. Let S be a set of pure sentences in $\mathcal{H}_{\mathcal{S}}(@,\downarrow)$, and let S be the extension of the logic $\mathbf{K}[\mathcal{H}_{\mathcal{S}}(@,\downarrow)]$ obtained by adding S as axioms. Then, every S-consistent set of formulas in $\mathcal{H}_{\mathcal{S}}(@,\downarrow)$ is satisfiable in a countable hybrid model, based on a frame that validates every formula in S.

ii. Let S be a set of pure schemas in $\mathcal{H}_{\mathcal{S}}(@,\downarrow)$, and let S be the extension of the logic $\mathbf{K}[\mathcal{H}_{\mathcal{S}}(@,\downarrow)]$ obtained by adding all instances of the schemas in S as axioms. Then, every S-consistent set of formulas in $\mathcal{H}_{\mathcal{S}}(@,\downarrow)$ is satisfiable in a countable hybrid model, based on a frame that validates every formula in S.

Again, the characterization results in Chapter 6 will exactly delineate the boundaries of these general completeness results.

3.5 Bisimulations

The notion of *bisimulation* is a crucial tool in modern modal model theory. Recall that for basic modal logics, bisimulations are non-empty binary relations linking the domains of models, with the restriction that only worlds with identical atomic information and matching accessibility relations are connected (see [van Benthem, 1983, Definition 3.7] where bisimulations are called p-relations). Formally, Definition 3.12. [Bisimulation] Let $\mathcal{M} = \langle M, R^{\mathcal{M}}, V^{\mathcal{M}} \rangle$ and $\mathcal{N} = \langle N, R^{\mathcal{N}}, V^{\mathcal{N}} \rangle$ be two (modal) models. A non-empty binary relation $\sim$ on $M \times N$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{N}$ if the following clauses hold

(prop) If $m \sim n$, then $m \in V^{\mathcal{M}}(p)$ iff $n \in V^{\mathcal{N}}(p)$, for $p \in \mathsf{PROP}$.
(forth) If $m \sim n$ and $R^{\mathcal{M}}(m, m')$, then $\exists n' \in N$ such that $R^{\mathcal{N}}(n, n')$ and $m' \sim n'$.
(back) A similar condition from $\mathcal{N}$ to $\mathcal{M}$.

We will write $\mathcal{M} \sim \mathcal{N}$ if there is a bisimulation between $\mathcal{M}$ and $\mathcal{N}$, and we will call m and n *bisimilar* if $m \sim n$ holds.

Bisimulations are the key to understanding modal expressive power, because bisimilar states satisfy the same basic modal formulas. I.e., bisimulations are to modal languages what partial isomorphisms are to first-order logic (see Definition 1.6). In the case of hybrid languages, the connection between bisimulations and partial isomorphisms will be even stronger (see Proposition 6.8).

We already discussed the fact that hybrid languages seem to blend nicely with first-order notions, and we will investigate the strong connection between hybrid bisimulations and k-back-and-forth systems (Definition 1.8) in Section 6.1. Now, if we want to extend the notion of bisimulation to $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$, we need to take care of assignments to state variables. To this end, hybrid bisimulations will not simply link worlds, rather they will link pairs $(\bar{m}, m)$, where m is a world and $\bar{m}$ is a partial assignment. We start by defining k-seq-bisimulations, which are the correct notion of bisimulation for formulas φ such that $\mathsf{SVAR}(\varphi) \subseteq \{x_1, \ldots, x_k\}$.

Definition 3.13. [k-seq-bisimulation] Let $\mathcal{M}$ and $\mathcal{N}$ be two hybrid models. Let $\overset{k}{\sim}$ be a binary relation between ${}^{k}M \times M$ and ${}^{k}N \times N$. So $\overset{k}{\sim}$ relates tuples $((m_1, \ldots, m_k), m)$ with tuples $((n_1, \ldots, n_k), n)$. We write these tuples as $(\bar{m}, m)$. Note that $\bar{m}$ can be seen as an assignment over $(x_1, \ldots, x_k)$. A non-empty relation $\overset{k}{\sim}$ is called a *k-seq-bisimulation* if it satisfies the following properties:

(prop) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then $m \in V^{\mathcal{M}}(a)$ iff $n \in V^{\mathcal{N}}(a)$, for $a \in \mathsf{PROP} \cup \mathsf{NOM}$.
(var) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then for $j \leq k$, $m_j = m$ iff $n_j = n$.
(forth) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$ and $R^{\mathcal{M}}(m, m')$, then there exists $n' \in N$ such that $R^{\mathcal{N}}(n, n')$ and $(\bar{m}, m') \overset{k}{\sim} (\bar{n}, n')$.
(back) A similar condition from $\mathcal{N}$ to $\mathcal{M}$.
(@) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then for every nominal $i \in \mathsf{NOM}$, if $m' \in V^{\mathcal{M}}(i)$ and $n' \in V^{\mathcal{N}}(i)$ then $(\bar{m}, m') \overset{k}{\sim} (\bar{n}, n')$, and for $j \leq k$, $(\bar{m}, m_j) \overset{k}{\sim} (\bar{n}, n_j)$.
(↓) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then for $j \leq k$, $(\bar{m}^{x_j}_{m}, m) \overset{k}{\sim} (\bar{n}^{x_j}_{n}, n)$.

Since ↓ and @ are self-dual, we can collapse the back and forth clauses for these operators into one. We write $\mathcal{M} \overset{k}{\sim} \mathcal{N}$ if there exists a k-seq-bisimulation between the two models. To extend the notion to the full language we need to add only one further condition.

Definition 3.14. [ω-seq-bisimulation] Let $\mathcal{M}$ and $\mathcal{N}$ be two hybrid models. An ω-seq-bisimulation between $\mathcal{M}$ and $\mathcal{N}$ is a non-empty family of k-seq-bisimulations satisfying the following *storage rule*:

(sto) If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then $(\bar{m} * m, m) \overset{k+1}{\sim} (\bar{n} * n, n)$.

Here and elsewhere, $\bar{m} * m$ denotes the tuple obtained from concatenating $\bar{m}$ and m. Let $\bar{m}$ $(\bar{n})$ be a ${}^k M$-tuple (${}^k N$-tuple). Then $(\mathcal{M}, \bar{m}) \overset{\omega}{\sim} (\mathcal{N}, \bar{n})$ means that there exists an ω-seq-bisimulation between $\mathcal{M}$ and $\mathcal{N}$ such that $(\bar{m}, \bar{m}(0)) \overset{k}{\sim} (\bar{n}, \bar{n}(0))$.

Some remarks. First, k- and ω-seq-bisimulations can be restricted to a given set of propositional variables and nominals $\mathsf{PROP} \cup \mathsf{NOM}$ by restricting **(prop)** and **(@)** accordingly. Second, the *modular character* of the definition of bisimulation will lead to results for reducts and extensions of $\mathcal{H}_\mathsf{S}(@, \downarrow)$ as well. For instance, if we delete $\downarrow$ from the language, we just delete the **($\downarrow$)** clause from the definition of bisimulation and we obtain the appropriate notion for $\mathcal{H}_\mathsf{S}(@)$. Of course, if we also delete the variables from the language and we move to $\mathcal{H}_\mathsf{N}(@)$, we don't need the assignment tuples anymore, and the bisimulation becomes just a relation between worlds, as usual. Then for $\mathcal{H}_\mathsf{N}$, the standard definition of bisimulation applies (the condition **(prop)** takes care of the nominals). If we add @ to this language, we just have to add the following clause

(@′) For all nominals i, if $V^{\mathcal{M}}(i) = \{m\}$ and $V^{\mathcal{N}}(i) = \{n\}$, then $m \sim n$.

Finally, if we add the past operator $\langle R^{-1} \rangle$, we need **(back^{-1})** and **(forth^{-1})** conditions defined over the converse of the accessibility relation, and to account for E we ask for the bisimulation to be a total and surjective relation. In all cases, the extension to many modalities amounts to requiring the **(back)** and **(forth)** conditions (and their $^{-1}$ versions if $\langle R^{-1} \rangle$ is present) of each of the accessibility relations.

An important fact about bisimulations is that they preserve truth:

Proposition 3.15.

i. If $\mathcal{M} \overset{k}{\sim} \mathcal{N}$, with $\overset{k}{\sim}$ over a given set $\mathsf{PROP} \cup \mathsf{NOM}$*, then for all formulas $\varphi \in \mathcal{H}_\mathsf{S}(@, \downarrow)$ over the signature* $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \{x_1, \ldots, x_k\} \rangle$*, $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$ implies $\mathcal{M}, \bar{m}, m \Vdash \varphi \Leftrightarrow \mathcal{N}, \bar{n}, n \Vdash \varphi$.*

ii. If $(\mathcal{M}, m) \overset{\omega}{\sim} (\mathcal{N}, n)$, with $\overset{\omega}{\sim}$ over a given set $\mathsf{PROP} \cup \mathsf{NOM}$*, then for all* sentences *$\varphi \in \mathcal{H}_\mathsf{S}(@, \downarrow)$ over* $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR} \rangle$*, $\mathcal{M}, m \Vdash \varphi \Leftrightarrow \mathcal{N}, n \Vdash \varphi$.*

Proof.

i) By a straightforward inductive argument.

ii) Let $(\mathcal{M}, m) \overset{\omega}{\sim} (\mathcal{N}, n)$ and let φ be a hybrid sentence. Then it contains variables (after renaming) say $\{x_1, \ldots, x_k\}$. We have $(\langle m \rangle, m) \overset{1}{\sim} (\langle n \rangle, n)$, so $k - 1$ applications of the storage rule give us $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, where $\bar{m}$ is a k-tuple consisting of m's and similarly for $\bar{n}$. But then, by i), $\mathcal{M}, \bar{m}, m \Vdash \varphi \Leftrightarrow \mathcal{N}, \bar{n}, n \Vdash \varphi$, whence since φ is a sentence $\mathcal{M}, m \Vdash \varphi \Leftrightarrow \mathcal{N}, n \Vdash \varphi$. QED

Preservation results for all the different sublanguages and extensions can be given by using the adequate notion of bisimulation. We will discuss more in detail some particular cases in Chapters 4 and 6.

Chapter 4

The Connection

Language is a virus from outer space.
William S. Burroughs

from "Home of the Brave," a film by Laurie Anderson

4.1 Similarities and Differences

The language used to define concepts in description logics is very close to the modal language. This similarity was first noticed by Schild [1991], who used it as a bridge to transfer complexity results and axiomatizations from modal logics to description logics. But as Schild carefully noticed, the link between basic modal logics and description logics can only be established at the level of *concept satisfiability*. Basic modal logic is not expressive enough to account for either A-Box reasoning or inference in the presence of definitions (non-empty T-Boxes).

In addition, as we saw in Chapter 2, some very expressive description languages include constructions for building complex roles like intersection, converse, and even transitive closure. By lifting the correspondence to Converse Propositional Dynamic Logic (CPDL) [Fischer and Ladner, 1979], Schild accounts for these constructions and, using the collapsed model property of CPDL and the availability of the Kleene star, also for inference from non-empty T-Boxes. In [1994] De Giacomo and Lenzerini extend these results and in particular they also encode A-Box reasoning into CPDL. As the results in [De Giacomo, 1995] show, the project of embedding description logics into CPDL has proven successful, but it has two important disadvantages.

- With respect to complexity: the local satisfiability problem of CPDL is already ExpTime-complete, and this blurs sharp complexity results.
- With respect to expressive power: the model theory of CPDL is complex, because the Kleene star (and hence a weak notion of induction) needs to be taken into consideration.

In this chapter we will replace CPDL by hybrid languages and in this way improve on the items above.

As we will show in Theorems 4.5 and 4.7 the connection between description and hybrid logics is indeed tight. It doesn't take much to realize some of the similarities between description and hybrid logics. To start with, both can be seen as fragments of a first-order language as we made explicit with the translations given in Proposition 2.12 and Definition 3.3. And the similarities between the two translations are striking: in both cases we used relational similarity types and we can easily spot pairs of operators where the translation literally coincides. In particular notice that

$$\begin{array}{rcl} ST_x(C \sqsubseteq D) & = & \forall y.(ST_y(C) \to ST_y(D)). \\ ST_x(a{:}C) & = & ST_x(C)[x/a]. \end{array}$$

And

$$\begin{array}{rcl} ST_x(\mathsf{A}(\varphi \to \psi)) & = & \forall y.(ST_y(\varphi) \to ST_y(\psi)). \\ ST_x(@_a\varphi) & = & ST_x(\varphi)[x/a]. \end{array}$$

To make things interesting, there are also differences. As we saw in Chapter 3, hybrid languages incorporate variables and the notion of binding, and as we started to investigate in Section 3.3.1, @ and ↓ work together in a nice synchrony. It seems worthwhile to explore what ↓ would have to offer from a knowledge representation perspective. Moving in the other direction, once a tight logical link has been established between the two families of languages, we can export the huge experience on optimization techniques and algorithms developed for description logic, and replace the logically elegant but computationally poor axiomatic systems we introduced in Definition 3.10 for hybrid languages by more effective inference mechanisms, as we will do in Chapter 5. To mention just one more point (and perhaps the one that will be most developed in this thesis), we will be able to take full advantage of modal model-theoretical techniques to explore expressive power (the main theme of Chapter 6) and complexity (as we do in Chapter 7).

But let's start by introducing in detail the work of Schild, and De Giacomo and Lenzerini on the connections between modal and description languages.

4.2 Schild's Terminologies

It is straightforward to map concepts in $\mathcal{ALC}$ into PDL preserving satisfiability, actually basic multi-modal logic is enough. Just define the translation $\cdot^t$ as

$$\begin{array}{rcl} (C_i)^t & = & p_i, \text{ for } C_i \text{ an atomic concept} \\ (\neg C)^t & = & \neg(C^t) \\ (C \sqcap D)^t & = & C^t \wedge D^t \\ (\exists R.C)^t & = & \langle R \rangle C^t. \end{array}$$

It is clear that $\cdot^t$ preserves satisfiability. But we need further expressive power if we want to account for T-Box and A-Box reasoning. The standard notion of bisimulation helps us prove this claim. Consider the signature $\mathcal{S} = \langle \{C_1, C_2\}, \{R\}, \{a\} \rangle$ and the interpretations $\mathcal{I}_1 = \langle \{m_1, m_2\}, \cdot^{\mathcal{I}_1} \rangle$ and $\mathcal{I}_2 = \langle \{m_3, m_4, m_5\}, \cdot^{\mathcal{I}_2} \rangle$ where

$$\begin{array}{rclcrcl} C_1^{\mathcal{I}_1} & = & \{m_1\} & \quad & C_1^{\mathcal{I}_2} & = & \{m_4\} \\ C_2^{\mathcal{I}_1} & = & \{m_1, m_2\} & & C_2^{\mathcal{I}_2} & = & \{m_3\} \\ R^{\mathcal{I}_1} & = & \{\} & & R^{\mathcal{I}_2} & = & \{\} \\ a^{\mathcal{I}_1} & = & m_1 & & a^{\mathcal{I}_2} & = & m_5. \end{array}$$

Clearly, $\mathcal{I}_1$ models both $C_1 \sqsubseteq C_2$ and $a\!:\!C_1$ while $\mathcal{I}_2$ models neither. On the other hand, when we consider $\mathcal{I}_1$ and $\mathcal{I}_2$ as modal models, the relation $\{(m_2, m_3)\}$ is a bisimulation. But we should take care, $C_1 \sqsubseteq C_2$ and $a : C_1$ are *global* notions, they are true of an element of a model if and only if they are true of all elements. On the other hand basic modal formulas are *local*, the point of evaluation is relevant for their truth. Let's go through our argument taking special care of this issue. If a modal formula φ is equivalent to $C_1 \sqsubseteq C_2$ then it would also behave globally, and $C_1 \sqsubseteq C_2$ being true of $\mathcal{I}_1$ would imply φ being true of m_2. By bisimulation φ would also be true of m_3 and by "global behavior" of $\mathcal{I}_2$. But it isn't. We can give a similar argument for $a\!:\!C_1$.

One of the main differences between basic modal languages and description languages is this switch between a local and a global perspective. And this is the reason why we have incorporated the existential modality in hybrid languages. Given that

$$\mathcal{M} \Vdash \varphi \text{ iff } \mathcal{M}, m \Vdash \neg\mathsf{E}\neg\varphi \text{ for some } m \in M,$$

E lets us talk about globality from a local perspective.

Instead of using E, Schild accounts for terminological axioms by using the collapsed model property of CPDL and the availability of the Kleene star. Due to the collapsed model property (which states that any satisfiable CPDL formula is satisfiable in a connected model) we can ignore states which are not reachable by a finite sequence of backwards and forwards transitions through the accessibility relations. Thanks to the Kleene star we can "step over" all these transitions in one step. Formally, extend $\cdot^t$ as follows

$$(C \sqsubseteq D)^t = (C^t \rightarrow D^t).$$

And for a finite set of terminological axioms T, let T^t be $\bigwedge \varphi_i^t$ for $\varphi_i \in T$. Now, let $T \cup \{\varphi\}$ be a finite set of terminological axioms and let $R_1, \ldots, R_n$ be all the roles mentioned in $T \cup \{\varphi\}$, then

$$\langle T, \{\}\rangle \models \varphi \text{ iff } \models [(R_1 \cup R_1^{-1} \cup \cdots \cup R_n \cup R_n^{-1})^*]T^t \rightarrow \varphi^t.$$

As Schild remarks, this translation would not work for an infinite T. On the one hand, T might contain an infinite number of roles, but even in the case of a finite signature, PDL is not compact (see [Harel, 1984, Theorem 2.15]), hence it is not always possible to reduce inference from an infinite set to inference from a finite part of it. In addition, lack of compactness has a striking effect on the complexity of the consequence problem, which becomes highly undecidable, and indication that PDL is not computationally well behaved. The computational problems raised by the Kleene star have been well investigated both in the modal and description logic community [Ladner, 1977; Halpern and Moses, 1992; Sattler, 1996; Horrocks and Gough, 1997]; and authors like Sattler, and Horrocks and Gough have argued that in many cases the ability to define a role as transitive is all what you need in applications, instead of the full power of transitive closure. For example, transitive roles are enough to provide an adequate representation of aggregated objects, as they allow these objects to be described by referring to their parts without specifying a level of decomposition [Horrocks and Sattler, 1999].

Again it pays off to look carefully to the global vs. local issue. To fully appreciate the subtleties here, we will digress into a discussion on global and local notions of consequence.

4.2.1 Global and Local Consequence

In Definition 3.2 we introduced two different notions of consequence for hybrid languages, which we called *local* and *global*:

- $T \models^{glo} \varphi$ iff for all models $\mathcal{M}$, $\mathcal{M} \Vdash T$ implies $\mathcal{M} \Vdash \varphi$.
- $T \models^{loc} \varphi$ iff for all models $\mathcal{M}$, assignments g and $m \in M$, $\mathcal{M}, g, m \Vdash T$ implies $\mathcal{M}, g, m \Vdash \varphi$.

One word of warning to avoid confusion. As we said before, for languages without state variables we can cross out the assignment in the definition of $\models^{loc}$. Still, the two notions of consequence are different because of the relativization to worlds. Perhaps it is simpler to discuss consequence in first-order terms, thinking on the first-order translation of modal, hybrid or description formulas. The availability of the two possibilities above is characteristic of a notion of consequence dealing with formulas instead of sentences. Given a set $\Gamma \cup \{\varphi\}$ of formulas which might contain free variables, the way we define the quantification on models and (first-order) assignments becomes meaningful.

The global consequence relation is the one familiar from first-order logic, but it is always defined for $\Gamma \cup \{\varphi\}$ a set of *sentences* (if they are formulas, the universal closure is usually considered). When $\Gamma \cup \{\varphi\}$ is a set of formulas — and they are indeed treated as formulas — the local definition becomes interesting (see for example the definition just before Proposition 2.3.6 in [Chang and Keisler, 1990]).

Because modal and hybrid formulas may contain free variables when translated into the FO, it is important to understand the connection between these two notions of consequence.

Proposition 4.1. [van Benthem, 1983, Lemma 2.33] *For T a set of basic modal formulas (in a mono-modal language), let $\mathsf{BOXED}(T) = \{\Box^i\psi \mid \psi \in T \ \& \ i \geq 0\}$. Then, for any set $T \cup \{\varphi\}$ of basic modal formulas*

$$T \models^{glo} \varphi \textit{ iff } \mathsf{BOXED}(T) \models^{loc} \varphi.$$

The proof uses the fact that the collapsed model property holds for modal languages. The extension to multi-modal languages is trivial, just redefine BOXED to include all possible boxed prefixes in the multi-modal signature.

The extension to hybrid languages needs more care. As we will see in Chapter 6, if the language does not contain the existential modality E, we can define a natural notion of generated model for hybrid languages and obtain the corresponding collapsed model property. And by defining BOXED properly, Proposition 4.1 also obtains in this case. Basically, if the language contains the @ operator then we should also generate from all named points, and accordingly, start by extending the set T to $T' = T \cup \{@_s\psi \mid \psi \in T\}$, and only then perform boxing by taking $\mathsf{BOXED}(T')$.

If the language does contains E then things are simpler, even though we cannot expect the collapsed model property to hold. Notice that in this case the relation between $\models^{glo}$ and $\models^{loc}$ is straightforward

$$T \models^{glo} \varphi \text{ iff } \{\mathsf{A}\psi \mid \psi \in T\} \models^{loc} \varphi. \tag{4.1}$$

In [1992], Goranko and Passy study the properties of languages containing the existential modality, and prove that the global properties of a logic $\mathcal{L}$ correspond to the local properties of the logic $\mathcal{L}^{\mathsf{E}}$ which arises from $\mathcal{L}$ by adding E. In particular, [Goranko and Passy, 1992] shows that for basic modal logics, global decidability, global finite model property, and global completeness of a logic $\mathcal{L}$ are equivalent to their local versions for $\mathcal{L}^{\mathsf{E}}$ (see [Kracht, 1999, Theorem 3.1.13] for a short proof). This result can be extended to hybrid languages without the $\downarrow$ binder as follows. We first establish a normal form for hybrid formulas not containing $\downarrow$.

PROPOSITION 4.2. [Normal form] *Let φ be a hybrid formula not containing the $\downarrow$ binder. Then φ is equivalent to a formula φ' where subformulas of the form $\mathsf{E}\psi$ and $@_i\psi$ (if any) occur only at modal depth zero. In particular φ' can be taken to be*

$$\bigwedge_{l\in L}(\bigvee_{m\in M} \mathsf{A}\rho_{(l,m)} \vee \mathsf{E}\sigma_l \vee \bigvee_{i\in\mathsf{NOM}(\varphi)} @_i\nu_{(l,i)} \vee \tau_l)$$

for some (possible empty) index sets L, M, where $\rho_{(l,m)}$, σ_l, $\nu_{(l,i)}$ and τ_l contains neither E nor $@$. Furthermore $|\varphi'|$ is polynomial in $|\varphi|$.

PROOF. We start by translating φ into negation normal form. Now we use the following equivalences to "push out" the E and A operators from inside the other modalities

$$\begin{array}{rcl|rcl}
[R_i]\mathsf{A}\psi & \leftrightarrow & [R_i]\bot \vee \mathsf{A}\psi & @_s\mathsf{A}\psi & \leftrightarrow & \mathsf{A}\psi \\
[R_i]\mathsf{E}\psi & \leftrightarrow & [R_i]\bot \vee \mathsf{E}\psi & @_s\mathsf{E}\psi & \leftrightarrow & \mathsf{E}\psi \\
[R_i](\theta \vee \mathsf{A}\psi) & \leftrightarrow & [R_i]\theta \vee \mathsf{A}\psi & @_s(\theta \vee \mathsf{A}\psi) & \leftrightarrow & @_s\theta \vee \mathsf{A}\psi \\
[R_i](\theta \vee \mathsf{E}\psi) & \leftrightarrow & [R_i]\theta \vee \mathsf{E}\psi & @_s(\theta \vee \mathsf{E}\psi) & \leftrightarrow & @_s\theta \vee \mathsf{E}\psi \\
[R_i](\theta \wedge \mathsf{A}\psi) & \leftrightarrow & [R_i]\theta \wedge [R_i]\mathsf{A}\psi & @_s(\theta \wedge \mathsf{A}\psi) & \leftrightarrow & @_s\theta \wedge \mathsf{A}\psi \\
[R_i](\theta \wedge \mathsf{E}\psi) & \leftrightarrow & [R_i]\bot \vee ([R_i]\theta \wedge [R_i]\mathsf{E}\psi) & @_s(\theta \wedge \mathsf{E}\psi) & \leftrightarrow & @_s\theta \wedge \mathsf{E}\psi.
\end{array}$$

Similar equivalences hold for the dual modalities $\langle R_i\rangle$ ($@$ is self dual). For pushing out $@$ we have

$$\begin{array}{rcl|rcl}
[R_i]@_i\psi & \leftrightarrow & [R_i]\bot \vee @_i\psi & @_s@_i\psi & \leftrightarrow & @_i\psi \\
[R_i](\theta \vee @_i\psi) & \leftrightarrow & [R_i]\theta \vee @_i\psi & @_s(\theta \vee @_i\psi) & \leftrightarrow & @_s\theta \vee @_i\psi \\
[R_i](\theta \wedge @_i\psi) & \leftrightarrow & [R_i]\theta \wedge [R_i]@_i\psi & @_s(\theta \wedge @_i\psi) & \leftrightarrow & @_s\theta \wedge @_i\psi.
\end{array}$$

And similarly for the $@$ operators appearing under $\langle R_i\rangle$. Now, it only rests to use propositional equivalences to obtain the normal form for φ. QED

We are now ready to extend Goranko and Passy's result to $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ and its sublanguages.

THEOREM 4.3. *Let the property P be either decidability, finite model property, or completeness, and lel $\mathcal{L}$ be any sublanguage of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$. Then $\mathcal{L}$ has P globally iff $\mathcal{L}^{\mathsf{E}}$ has P locally.*

Proof. The equivalence in (4.1) is enough to prove the left to right implication. For the other, we need a way to relate local validity of a formula in $\mathcal{L}^{\mathsf{E}}$ to consequence in terms of $\models^{glo}_{\mathcal{L}}$. We assume that φ is in the normal form of Proposition 4.2. We can do away with conjunctions in φ, as $\models_{\mathcal{L}^{\mathsf{E}}} \varphi_1 \wedge \varphi_2$ if and only if $\models_{\mathcal{L}^{\mathsf{E}}} \varphi_1$ and $\models_{\mathcal{L}^{\mathsf{E}}} \varphi_2$. Hence, we need only consider $\varphi = (\bigvee \mathsf{A}\rho_i) \vee \mathsf{E}\sigma \vee \theta$. We will prove that $\models_{\mathcal{L}^{\mathsf{E}}} \varphi$ iff $\neg\sigma \models^{glo}_{\mathcal{L}} (\bigvee \rho_i) \vee \theta$.

[$\Leftarrow$]. We reason by contraposition. Assume $\not\models_{\mathcal{L}^{\mathsf{E}}} (\bigvee \mathsf{A}\rho_i) \vee \mathsf{E}\sigma \vee \theta$. Then there exists a model $\mathcal{M}$ and $m \in M$ such that $\mathcal{M}, m \Vdash (\bigwedge \neg\mathsf{A}\rho_i) \wedge \mathsf{A}\neg\sigma \wedge \neg\theta$. Hence $\mathcal{M} \Vdash \neg\sigma$ and $\mathcal{M} \not\Vdash (\bigvee \rho_i) \vee \theta$. So $\neg\sigma \not\models^{glo}_{\mathcal{L}} (\bigvee \rho_i) \vee \theta$.
[$\Rightarrow$]. Again we argue by contraposition. Assume $\neg\sigma \not\models^{glo}_{\mathcal{L}} (\bigvee \rho_i) \vee \theta$. Then there is $\mathcal{M}$ such that $\mathcal{M} \Vdash \neg\sigma$ and $\mathcal{M} \not\Vdash (\bigvee \rho_i) \vee \theta$; i.e., for some $m \in M$, $\mathcal{M}, m \Vdash (\bigwedge \neg\rho_i) \wedge \neg\theta$. But then $\mathcal{M}, m \Vdash (\bigwedge \neg\mathsf{A}\rho_i) \wedge \neg\mathsf{E}\sigma \wedge \neg\theta$ and $\not\models_{\mathcal{L}^{\mathsf{E}}} \varphi$. QED

Going back to description languages, notice that if we use $\models^{glo}$ instead of $\models^{loc}$, then basic modal logic *is* enough to encode terminological axioms, as the following equivalence holds

$$\langle T, \{\}\rangle \models \varphi \text{ iff } T^t \models^{glo} \varphi^t.$$

By using (4.1), in the presence of E we can further move to

$$\langle T, \{\}\rangle \models \varphi \text{ iff } \{\mathsf{A}(T^t)\} \models^{loc} \varphi^t.$$

And given that the local consequence relation satisfies the deduction theorem

$$\langle T, \{\}\rangle \models \varphi \text{ iff } \models \mathsf{A}(T^t) \to \varphi^t.$$

Finally, if the logic is compact we can perform this reduction even for infinite T-Boxes. And by Theorem 4.3, we can investigate logical properties of inference from non-empty knowledge bases by studying the local properties of the language containing E.

4.3 De Giacomo's Individuals

Accounting for assertional information in CPDL is more complicated than encoding terminological axioms. In [De Giacomo and Lenzerini, 1994], a much more involved variation of the translation we discuss below is proposed. De Giacomo and Lenzerini enforce the unique name assumption (i.e., for $a, b \in \mathsf{IND}$, for all interpretation $\mathcal{I}$, $a \neq b$ implies $a^{\mathcal{I}} \neq b^{\mathcal{I}}$), and also deal with complex structure on roles (union, composition, transitive closure, etc.) which makes for the additional complexity. Here we will only discuss the handling of individuals.

Extend $\cdot^t$ to assertions by defining

$$\begin{array}{rcl} (a\!:\!C)^t & = & p_a \to C^t, \\ ((a,b)\!:\!R)^t & = & p_a \to \langle R\rangle p_b, \end{array}$$

where p_a and p_b are propositional symbols.

Let A be a finite set of assertions, define A^t as $\bigwedge \varphi_i^t$ for $\varphi_i \in A$. The problem now is that in translating *individuals* as *propositions* in CPDL we have lost the information that individuals denote a single element in the domain. Hence, we have to explicitly force these symbols to behave as individuals.

Let $\Sigma = \langle T, A\rangle$ be a knowledge base, let $R_1, \dots, R_n$ be the roles appearing in Σ, let $a_1, \dots, a_m$ be the individuals mentioned in Σ, and let $\mathsf{SF}(\varphi)$ be the set of all subformulas of φ. Let $[U]$ stand for $[(R_1 \cup R_1^{-1} \cup \cdots \cup R_n \cup R_n^{-1})^*]$, and let S be a role not appearing in Σ. Let Σ^t be

$$[S][U](A^t \wedge T^t) \wedge \bigwedge_{1\le i\le m}(\langle S\rangle p_{a_i} \wedge (\bigwedge_{\psi\in\mathsf{SF}(T^t\wedge A^t)} [S](\langle U\rangle(p_{a_i}\wedge\psi) \to [U](p_{a_i}\to\psi)))).$$

We will prove that Σ is consistent if and only if Σ^t is satisfiable. This is enough because, as we discussed in Proposition 2.6, in sufficiently expressive languages all reasoning tasks can be reduced to instance checking and, in its turn, $\langle T, A\rangle \models a : C$ is the case if and only if $\langle T, A \cup \{a : \neg C\}\rangle \models \bot$.

PROPOSITION 4.4. *A knowledge base Σ is consistent if and only if Σ^t is satisfiable.*

PROOF.
[⇒]. Let $\mathcal{I} \models \langle T, A\rangle$. For $s \notin \Delta^{\mathcal{I}}$, define a CPDL model $\mathcal{M} = \langle M, \{R_i\} \cup \{S\}, V\rangle$, where $M = \Delta^{\mathcal{I}} \cup \{s\}$, $R_i = R_i^{\mathcal{I}}$, $S = \{(s, m) \mid m \in \Delta^{\mathcal{I}}\}$, $V(C_i) = C_i^{\mathcal{I}}$ and $V(p_{a_i}) = \{a_i^{\mathcal{I}}\}$. We prove that $\mathcal{M}, s \Vdash \Sigma^t$.

For any $m \in \Delta^{\mathcal{I}}$, a simple induction proves that $\mathcal{M}, m \Vdash A^t \wedge T^t$. Hence, $\mathcal{M}, s \Vdash [S][U](A^t \wedge T^t)$. Because s is S-related to all elements in $\Delta^{\mathcal{I}}$, also $\mathcal{M}, s \Vdash \langle S\rangle p_{a_i}$. It rest to prove for any a_i,

$$\mathcal{M}, s \Vdash \bigwedge_{\psi\in\mathsf{SF}(T^t\wedge A^t)} [S](\langle U\rangle(p_{a_i}\wedge\psi) \to [U](p_{a_i}\to\psi)).$$

But this follows from the fact that the denotation of each p_{a_i} is a singleton.

[⇐]. Now suppose $\mathcal{M} = \langle M, \{R_i\} \cup \{S\}, V\rangle$ is a CPDL model, and for $s \in M$ we have $\mathcal{M}, s \Vdash \Sigma^t$. Because of the collapsed model property of CPDL, we can assume that $\mathcal{M}$ is a connected model. Define $\mathcal{M}' = \langle M', \{R_i'\}, V'\rangle$ where $M' = \{m \mid S(s, m)\}$, $R_i' = (R_i)_{\restriction M'}$ and $V' = V_{\restriction M'}$.

Clearly $\mathcal{M}' \Vdash [U](A^t \wedge T^t)$, and hence $\mathcal{M}' \Vdash A^t \wedge T^t$. Also, the following formula is globally true in $\mathcal{M}'$: $\bigwedge_{\psi\in\mathsf{SF}(T^t\wedge A^t)}(\langle U\rangle(p_{a_i}\wedge\psi) \to [U](p_{a_i}\to\psi))$. So for $\psi \in \mathsf{SF}(T^t \wedge A^t)$, if for some $m \in M'$, $\mathcal{M}', m \Vdash p_{a_i} \wedge \psi$ then $\mathcal{M}' \Vdash p_{a_i} \to \psi$. Furthermore, for any p_{a_i}, there is $m \in M'$ such that $\mathcal{M}', m \Vdash p_{a_i}$. Notice that $\mathcal{M}'$ is a modal model. Define $\mathcal{M}^f = \langle M^f, \{R_i^f\}, V^f\rangle$ as a filtration of $\mathcal{M}'$ through $\mathsf{SF}(T^t \wedge A^t)$. We prove that for any a_i, $V^f(p_{a_i})$ is a singleton. Because, let $m_1, m_2 \in V^f(p_{a_i})$ and let $\psi \in m_1$, then $\mathcal{M}', m_1 \Vdash p_{a_i} \wedge \psi$, and hence $\mathcal{M}', m_2 \Vdash \psi$ and $\psi \in m_2$. This proves $m_1 = m_2$.

Now, consider $\mathcal{M}^f$ as a description logic interpretation $\cdot^{\mathcal{I}}$, where $a_i^{\mathcal{I}} = m$, for $m \in V^f(p_{a_i})$. Clearly $\mathcal{I} \models \Sigma$. QED

As remarked in [Horrocks *et al.*, 2000b], De Giacomo's translation is probably too involved and costly to provide effective decision methods. It is also difficult to extract theoretical results from it, except for the general complexity results presented in [De Giacomo and Lenzerini, 1994]. As we already remarked, the model theory of PDL is intricate because of the inductive nature of the Kleene star, and the cryptic translation provides little help on simplifying things out.

The main difficulty of the translation above is on forcing propositional symbols in CPDL to behave as individuals. If we use hybrid logics instead, we can simply use *nominals*. In addition, given our discussion in Section 4.2.1 the E modality gives us access to globality and we don't need to rely on the Kleene star. So, hybrid logic and not CPDL seems to be the language of choice for a modal counterpart of description languages able to deal with full terminological and assertional reasoning.

4.4 Into Hybrid Logics

Consider the following translation $\cdot^h$,

$$\begin{array}{rcl}
(C_i)^h & = & p_i, \text{ for } C_i \text{ an atomic concept}\\
(\neg C)^h & = & \neg(C^h)\\
(C \sqcap D)^h & = & C^h \wedge D^h\\
(\exists R.C)^h & = & \langle R\rangle C^h\\
(\exists R^{-1}.C)^h & = & \langle R^{-1}\rangle C^h\\
(\{a_1,\dots,a_n\})^h & = & a_1 \vee \cdots \vee a_n\\
(\exists R.\{a\})^h & = & \langle R\rangle a\\
(C \sqsubseteq D)^h & = & \mathsf{A}(C^h \to D^h)\\
(a\,{:}\,C)^h & = & @_a C^h\\
((a,b)\,{:}\,R)^h & = & @_a\langle R\rangle b.
\end{array}$$

Theorem 4.5. *Let $\Sigma = \langle T, A\rangle$ be a knowledge base in $\mathcal{ALCOBI}$, and φ a terminological axiom or an assertion, then*

$$\langle T, A\rangle \models \varphi \textit{ iff } \models (\bigwedge_{\psi\in T} \psi^h \wedge \bigwedge_{\psi\in A} \psi^h) \to \varphi^h.$$

The proof in this case is obvious (and the connection between the two languages stronger than with CPDL), as any model of $\langle T, A\rangle$ and φ can be viewed directly as a model of $(\bigwedge_{\psi\in T}\psi^h \wedge \bigwedge_{\psi\in A}\psi^h) \to \varphi^h$ and vice versa. By using additional nominals we can also account for conjunction of roles:

$$(\exists(R_1 \sqcap R_2).C)^h = \langle R_1\rangle i \wedge \langle R_2\rangle i \wedge @_i C^h \text{ for } i \text{ a new nominal, while}$$
$$((a,b)\,{:}\,R_1 \sqcap R_2)^h = @_a\langle R_1\rangle b \wedge @_a\langle R_2\rangle b.$$

Equivalently, we could have put $(\exists(R_1 \sqcap R_2).C)^h = \langle R_1\rangle(i \wedge C^h) \wedge \langle R_2\rangle(i \wedge C^h)$, and do without @. But this is not a linear translation and, as we will soon see, using @ and restricting the use of nominals is more "natural" from a description logic point of view. Notice that in any case, we need to move to an extended language to account for role conjunction (as we need new nominals) in this way. To remain in the spirit (and strength) of the previous translation we would do better by introducing role conjunction into hybrid logics as investigated in [Passy and Tinchev, 1985a]. Similarly, we could add counting modalities to account for the $\mathcal{N}$ constructor.

Blackburn and Tzakova [1998c] also propose using hybrid languages to embed description logics, highlighting the connection between assertional information and nominals, and the use of the existential modality to encode terminological axioms. But Blackburn and Tzakova introduce undecidable hybrid languages (containing the $\forall$ quantifier) for this account, arguing in favor of the gains on expressiveness that these more powerful languages have to offer. Instead, our translation tries to remain as faithful as possible to the original description language, and pay special attention to decidability issues.

It is important to pin down exactly which expressive power we need to encode the different languages and reasoning tasks. For example, the existential modality is required only for translating terminological axioms, while @ is only used for assertions.

Definition 4.6. In the next sections we will discuss properties concerning the following hybrid languages. The first two were introduced already in Section 3.3.

- $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$, in which the full $\cdot^h$ translation can be made.
- $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, in which we can only encode knowledge bases with empty T-Boxes as we have dropped the existential modality.

The next two languages restrict the use of nominals, so that they can only appear as sub-indices of @ and in the construction $@_a\langle R\rangle b$ or $@_a\langle R^{-1}\rangle b$. We have dropped the N in the name to mark this restriction. On the description logic side, these restrictions are equivalent to the absence of the one-of $\mathcal{O}$ operator from the concept language, and the circumscription of nominals to A-box statements.

- $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$, in which we cannot translate the one-of operator $\mathcal{O}$.
- $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$, the "empty T-Boxes" version of the previous language.

$\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$ is a sublanguage of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, and the languages $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ and $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$ are obtained from the other two by the introduction of the existential modality. From a hybrid logic point of view, $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ is probably the most natural language. The other three languages are specially devised to address two issues: restricting the use of nominals to the way they are traditionally introduced in description logics, and obtaining global expressivity to represent terminological axioms. We will also discuss the "pure future" versions of these languages, where we drop the $\langle R^{-1}\rangle$ operator. We will see that this can, in some cases, make an important difference in terms of complexity.

We have defined each of the logics mentioned above to be expressive enough to permit the encoding of certain specific description logics. But it is also important to investigate in which ways we have *extended* the expressive power of the language with the move into hybrid languages. The general answer is: we have incorporated Boolean structure into the knowledge base, and allowed explicit interaction among T-Box definitions, A-Box assertions and concepts.

Take for example the most expressive language $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$. Given Proposition 4.2 we can assume $\varphi \in \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$ to be

$$\bigwedge_{l\in L}(\bigvee_{m\in M} \mathsf{A}\rho_{(l,m)} \vee \mathsf{E}\sigma_l \vee \bigvee_{i\in\mathsf{NOM}} @_i\nu_{(l,i)} \vee \tau_l),$$

where $\rho_{(l,m)}, \sigma_l, \nu_{(l,i)}$ and τ_l contains neither E nor @. By allowing negations in the T-Box we can encode validity of formulas in $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$ as instance checking as follows. Define *Boolean knowledge bases* as pairs $\Sigma = \langle T, A\rangle$ where T is a set of Boolean combinations of terminological axioms, and A a set of Boolean combinations of assertions. For $l \in L$, define $\Sigma^l_\varphi = \langle T^l_\varphi, A^l_\varphi\rangle$ to be

$$T^l_\varphi = \{\neg(\top \sqsubseteq \rho^{h^{-1}}_{(l,m)}) \mid m \in M\} \cup \{\top \sqsubseteq \neg\sigma^{h^{-1}}_l\}$$
$$A^l_\varphi = \{i\!:\!\neg\nu^{h^{-1}}_{(l,i)} \mid i \in \mathsf{NOM}(\varphi)\}$$

where $\cdot^{h^{-1}}$ is the backwards translation from the hybrid language into $\mathcal{ALCOI}$, mapping Boolean and modal operators into the corresponding description logic ones and using singleton one-of sets $\{i\}$ for translating nominals. Then

Theorem 4.7. *For any formula φ in $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$, let $a \notin \mathsf{NOM}(\varphi)$, then φ is valid iff for all $l \in L$, $\Sigma^l_\varphi \models a\!:\!\tau^{h^{-1}}_l$.*

It is interesting to remark that even allowing Boolean knowledge bases, we cannot recast validity of hybrid formulas as inference in terms of a *unique* knowledge base. This is because the separation between terminological axioms, assertions and simple concepts still impose syntactic restrictions which don't exist when we wear our hybrid logic spectacles. Trivially, if the index set L above is a singleton, then a unique knowledge base is sufficient. I.e., we can characterize precisely the fragment of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$ perfectly matching the expressivity of $\mathcal{ALCOI}$ with Boolean knowledge bases.

As we will see in the next section, allowing the extra flexibility that Boolean knowledge bases offer does not modify the complexity class in which the reasoning tasks fall (for the languages we are considering), but it does increase expressivity. Boolean knowledge bases have also been considered by Wolter and Zakharyaschev in a series of papers investigating ways of combining description and modal languages [Wolter and Zakharyaschev, 1998, 1999b, 1999, 1999a, 2000; Wolter, 1999].

4.5 Pay Day

The links between hybrid and description logics are so strong that we can immediately start the harvest by interpreting result in one of the fields in the light of the other. This is what we are going to do now, and from many different perspectives: complexity, expressive power, meta-logical properties, new operators, etc. We will mainly draw results from the work we will carry out in the remaining chapters of the thesis, and also on well known result which can be put to new use. Sometimes, an exhaustive investigation will not be possible, but we will always introduce the main ideas and techniques.

In any case, by the end of this section we should have drilled ourselves into looking at results in hybrid logics with description logic eyes and vice versa.

4.5.1 Complexity

Also for complexity we need to pay attention to the difference between local and global notions. For a modal language, we can distinguish between the local *Sat* problem (determining whether for a given formula φ there exists a model $\mathcal{M}$ and $m \in M$ such that $\mathcal{M}, m \Vdash \varphi$), and the global *Sat* problem (where we require a model $\mathcal{M}$ such that $\mathcal{M} \Vdash \varphi$). Of course, if the logic contains the E modality, both problems collapse to the same. And as we argued in Section 4.2.1, we can study the global *Sat* problem of a language $\mathcal{L}$ by analyzing the local *Sat* problem of $\mathcal{L}^{\mathsf{E}}$. In addition, we can study different classes of frames as is standard in modal logic (transitive, linear, etc.), hence a *Sat* problems should also be qualified with respect to a given class of models. For F a class of models and a language $\mathcal{L}$, F-$Sat(\mathcal{L})$ is the (global or local) satisfiability problem of the language when its class of models is restricted to F. In this section we will be mainly interested in the class K of all models.

The complexity of the satisfiability problem of the four languages in Definition 4.6 can be established by drawing from some of the results we will discuss in detail in Chapter 7. Let us first consider the "pure future" fragments, i.e., we only consider formulas without the $\langle R^{-1}\rangle$ operator.

In Theorem 7.15 we prove that the local K-*Sat* problem for $\mathcal{H}_{\mathsf{N}}(@)$ is PSpace-complete. This results sets also the complexity of $\mathcal{H}(@, @\Diamond)$, because this language contains the basic modal language. We obtain an ExpTime upper bound for the local K-*Sat* problem for $\mathcal{H}_{\mathsf{N}}(@, \mathsf{E})$ as a corollary of Theorem 7.20. Given Spaan's result concerning the ExpTime completeness of modal logic expanded with the existential modality [Spaan, 1993], both $\mathcal{H}(@, @\Diamond, \mathsf{E})$ and $\mathcal{H}_{\mathsf{N}}(@, \mathsf{E})$ are ExpTime-complete.

If we now switch to the description logic perspective, the results above imply that it is the move from empty T-Boxes to full T-Boxes which modifies complexity, independently of whether we consider standard or Boolean knowledge bases, as the same complexity obtains for the knowledge bases introduced in Definition 2.2. Furthermore, the addition of the one-of operator $\mathcal{O}$ and role fillers $\mathcal{B}$ offers more expressivity at no cost (up to a polynomial). Notice how the encoding into hybrid languages instead of CPDL works to our advantage here, as we can identify cases falling into the PSpace complexity class. Let's gather these results neatly.

Theorem 4.8.

i. Instance checking for Boolean knowledge bases with empty T-Boxes is solvable in PSpace *(hence* PSpace-*complete) for the language $\mathcal{ALCROB}$.*

ii. Instance checking for Boolean knowledge bases is solvable in ExpTime *(hence* ExpTime-*complete) for the language $\mathcal{ALCROB}$.*

Notice that we don't need to restrict to empty A-boxes in item *i*), and remember that by Proposition 2.6, the complexity results for instance checking extend to all the reasoning tasks we defined in Section 2.3.

Things are different when the $\langle R^{-1}\rangle$ operator is present. As we prove in Theorem 7.18, adding *just one nominal* to basic temporal logic moves the complexity of the local satisfiability problem over K from PSpace- to ExpTime-hard. As the ExpTime upper bound of Theorem 7.20 actually covers also $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$, we have that the local K-*Sat* problems of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ and $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$ are ExpTime-complete.

A PSpace upper bound for $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$ is easy to establish by using the fact that @ operators need only appear at modal depth zero. We give a sketch of the proof. To avoid confusion we will write $@_i\langle R_r\rangle j$ as $R_r(i,j)$. Let

$$\varphi = \bigwedge_{l\in L}(\bigvee_{i\in\mathsf{NOM}} @_i\nu_{(l,i)} \vee \bigvee T_l \vee \sigma_l),$$

where each T_l is a collection of formulas of the form $R_r(i,j)$ or $\neg R_r(i,j)$, and $\nu_{(l,i)}, \sigma_l$ contain neither @ nor nominals. As PSpace = NPSpace, non-deterministically choose from each conjunct of φ the disjunct satisfied by a model of φ. Call such a set CHOICE. Now, for each i, let $S_i = \{\varphi \mid @_i\varphi \in \mathsf{CHOICE}\}$, create a polynomial model satisfying S_i at the point m_i (notice that all formulas in S_i are basic temporal formulas and hence a PSpace model can be constructed). Similarly, create a polynomial model for all formulas in CHOICE which are not @-formulas. Let $\mathcal{M}$ be de disjoint union of all these models. Finally, if $R_r(i,j) \in \mathsf{CHOICE}$, add the pair (m_i, m_j) to R_r. The model of φ obtained in this way has size polynomial in $|\varphi|$.

Again, evaluating the difference in terms of complexity that the presence or absence of the $\langle R^{-1}\rangle$ makes, wouldn't be possible using the CPDL translation.

Theorem 4.9.

i. Instance checking for Boolean knowledge bases with empty T-Boxes is solvable in PSpace *(hence* PSpace*-complete) for the language $\mathcal{ALCRBI}$.*

ii. Instance checking for knowledge bases with empty T- and A-Boxes is ExpTime*-hard for the language $\mathcal{ALCIO}$.*

iii. Instance checking for Boolean knowledge bases is solvable in ExpTime *(hence* ExpTime*-complete) for the language $\mathcal{ALCROBI}$.*

The ExpTime-hardness result for $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ (basic temporal logic with at least one nominal) contrast sharply with the good complexity behavior of $\mathcal{H}_{\mathsf{N}}(@)$. For example, as we will see in Theorems 7.22 and 7.24, if we move to the class of transitive models, even $\mathcal{H}_{\mathsf{N}}(@, \mathsf{E})$ is PSpace-complete (meaning that there are PSpace algorithms even for inference from non-empty T-Boxes), while $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ remains obstinately in ExpTime.

In Chapter 7 we will investigate further the issue of complexity in different classes of models. One of the main results (Theorem 7.29 and Corollary 7.31) implies that instance checking for Boolean knowledge bases in $\mathcal{ALCROBI}$ can be solved in PSpace if we consider only transitive trees as models.

On the other hand, known complexity results from description logics can be usefully translated into hybrid terms. For example, as we will discuss in Section 4.5.5, little is known with respect to the extension of hybrid languages with counting.

Also, the "folklore" result concerning the PSpace-completeness of instance checking for $\mathcal{ALC}$ when T-Boxes are restricted to simple and acyclic terminological axioms (recall our discussion in Section 2.2.2) implies that when syntactic restrictions are imposed on the use of E we can avoid ExpTime-hardness for the local K-*Sat* problem of $\mathcal{H}(@, @\Diamond, \mathsf{E})$. [Lutz, 1999a, 1999b] provide the first detailed complexity analysis of inference from simple, acyclic T-Boxes. Interestingly, as Lutz proves, the restriction to simple, acyclic T-Boxes not always preserves complexity: instance checking in $\mathcal{ALCF}$ ($\mathcal{ALC}$ extended with features, feature agreement and feature disagreement) is PSpace-complete for empty T-Boxes, but it turns NExpTime-complete even when only simple, acyclic T-Boxes are allowed. Lutz' results are in line with Spaan's [1993, 1996], where it is shown that extensions with the E operator behave rather chaotically, complexity-wise.

4.5.2 Expressive Power

Baader [1996] and Borgida [1996] were the first to address the issue of expressive power for description languages. Each author proposes different means to measure expressive power. Borgida compares complex concepts and roles in description languages with first-order formulas: there should be a translation from complex concept and roles of a DL into first-order formulas in one and two free variables respectively, such that for each interpretation their denotations coincide. Baader, on the other hand, remains "on the description logic side." To compare the expressive power of two DLs $\mathcal{L}_1$ and $\mathcal{L}_2$ he proposes to define translations between the atomic concepts of the T-Boxes in each language, and compare the denotation of concepts in a T-Box $T_1 \in \mathcal{L}_1$ with the denotation of the translated concepts in $T_2 \in \mathcal{L}_2$ on models of T_1 and T_2 respectively. Interestingly, the translation function maps only atomic concept to atomic concepts and is allowed to be different for each T-Box.

More recently, Kurtonina and de Rijke [1999] have taken a modal perspective on the topic and provided a detailed analysis of the expressive power of concepts in DLs by means of (bi-)simulations. The most interesting result, from a logic point of view, discussed by Kurtonina and de Rijke is their "deconstruction" of the notion of bisimulation to address languages which lack full Boolean expressivity. But Kurtonina and de Rijke only address the expressive power of *concepts*.

Instead, in this section we will study the expressive power which full knowledge bases offer, taking advantage of the tools we have introduced in Chapter 3. In particular, we will use hybrid bisimulations. In Definition 3.13 and the discussion that follows, we spelled out almost all the necessary bits for defining the appropriate notions of bisimulation for the languages in Definition 4.6.

Definition 4.10. [Bisimulations] Let $\mathcal{M} = \langle M, \{R_r^{\mathcal{M}}\}, V^{\mathcal{M}}\rangle$ and $\mathcal{N} = \langle N, \{R_r^{\mathcal{N}}\}, V^{\mathcal{N}}\rangle$ be two hybrid models. Let $\sim$ be a non-empty binary relation on $M \times N$, and consider the following properties on $\sim$:

(prop) If $m \sim n$, then $m \in V^{\mathcal{M}}(p)$ iff $n \in V^{\mathcal{N}}(p)$, for $p \in \mathsf{PROP}$.
(nom) If $m \sim n$, then $m \in V^{\mathcal{M}}(i)$ iff $n \in V^{\mathcal{N}}(i)$, for $i \in \mathsf{NOM}$.
(forth) If $m \sim n$ and $R_r^{\mathcal{M}}(m, m')$, then $\exists n' \in N.(R_r^{\mathcal{N}}(n, n')$ & $m' \sim n')$.
(forth^{-1}) If $m \sim n$ and $R_r^{\mathcal{M}}(m', m)$, then $\exists n' \in N.(R_r^{\mathcal{N}}(n', n)$ & $m' \sim n')$.
(back) A condition similar to **(forth)**, but from $\mathcal{N}$ to $\mathcal{M}$.
(back^{-1}) A condition similar to **(forth^{-1})**, but from $\mathcal{N}$ to $\mathcal{M}$.
(@) For all nominals i in NOM, $i^{\mathcal{M}} \sim i^{\mathcal{N}}$.
(@$\diamond$) Let i, j be nominals in NOM, then $R_r(i^{\mathcal{M}}, j^{\mathcal{M}})$ iff $R_r(i^{\mathcal{N}}, j^{\mathcal{N}})$.
(E) $\sim$ is total and surjective.

Now for the final definitions:

i. $\sim$ is an $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$-bisimulation if it satisfy the conditions **(prop)**, **(forth)**, **(forth^{-1})**, **(back)**, **(back^{-1})**, **(@)** and **(@◇)**.

ii. $\sim$ is an $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$-bisimulations if in addition it satisfies **(nom)**. (And in this case **(@◇)** can be derived from the others.)

iii. $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$- and $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$-bisimulations are obtained, respectively, from $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$- and $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$-bisimulations by requiring the additional condition **(E)**.

Definition 4.10 has been devised to obtain the following result.

Proposition 4.11. *Let $\mathcal{H}$ be any of the languages $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$, $\mathcal{H}_{N}(\langle R^{-1}\rangle, @)$, $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ or $\mathcal{H}_{N}(\langle R^{-1}\rangle, @, \mathsf{E})$. Let $\mathcal{M}$ and $\mathcal{N}$ be two models, and $\sim$ an $\mathcal{H}$-bisimulation between $\mathcal{M}$ and $\mathcal{N}$.*

Then for $m \in M, n \in N$, and for any formula φ in $\mathcal{H}$, $m \sim n \Rightarrow (\mathcal{M}, m \Vdash \varphi$ iff $\mathcal{N}, n \Vdash \varphi)$.

Proposition 4.11 alone lets us establish a hierarchy of expressive power.

Definition 4.12. For two logics $\mathcal{H}$ and $\mathcal{H}'$, $\mathcal{H} \preccurlyeq \mathcal{H}'$ denotes that for each formula φ in $\mathcal{H}$ there exists a formula φ' in $\mathcal{H}'$ such that for each model $\mathcal{M}$ and $m \in M$, $\mathcal{M}, m \Vdash \varphi$ iff $\mathcal{M}, m \Vdash \varphi'$. We write $\mathcal{H} \prec \mathcal{H}'$ if $\mathcal{H} \preccurlyeq \mathcal{H}'$ and not $\mathcal{H}' \preccurlyeq \mathcal{H}$.

Our approach to comparing the expressive power of languages is different from the proposals of Borgida and Baader we discussed above. From the local, hybrid logic perspective, we can compare the relative expressive power of two languages by simply requiring the existence of an equivalent formula (a formula which receives the same denotation under all interpretations). This notion is stronger that simple satisfiability preservation. The interesting twist is that we have internalized terminological definitions and assertions into the hybrid language, and hence implemented an approach similar to the one used in [Kurtonina and de Rijke, 1999] but this time accounting for full knowledge bases. Let's see how this works.

It is immediate that $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond) \preccurlyeq \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ and $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E}) \preccurlyeq \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$. More interestingly, each of the relations is strict. Given $\mathcal{H} \preccurlyeq \mathcal{H}'$, to prove $\mathcal{H} \prec \mathcal{H}'$ it is enough to provide models $\mathcal{M}$ and $\mathcal{M}'$, points $m \in M$, $m' \in M'$, an $\mathcal{H}$-bisimulation linking m and m' and a formula in $\mathcal{H}'$ such that $\mathcal{M}, m \Vdash \varphi$ and $\mathcal{M}', m' \nVdash \varphi$. Consider the models in Figure 4.1.a) and the bisimulation relation linking all points in $\mathcal{M}_1$ with all points in $\mathcal{M}_2$. All conditions in Definition 4.10 except (NOM) are easy to check. So, the relation is both an $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$-bisimulation and an $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$-bisimulation. Furthermore, $\mathcal{M}_1, m_1 \Vdash \neg @_i j$ while $\mathcal{M}_2, m_3 \nVdash \neg @_i j$. Hence $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond) \prec \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ and $\mathcal{H}(\langle R^{-1}\rangle, @\Diamond, \mathsf{E}) \prec \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$.

The relation between $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ and $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ is more complex. We can prove both that $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @) \not\preccurlyeq \mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ and $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E}) \not\preccurlyeq \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$. For the first, we need but reuse the models in Figure 4.1.a). While the models in Figure 4.1.b) and the $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$-bisimulation sending m_1 to m_3 proves $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E}) \not\preccurlyeq \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, as $\mathsf{A}p$ holds in m_3 and not in m_1. Nevertheless, we can prove that $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ is at least as expressive as $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$ if we are only interested in satisfiability (and not in the existence of an equivalent formula).

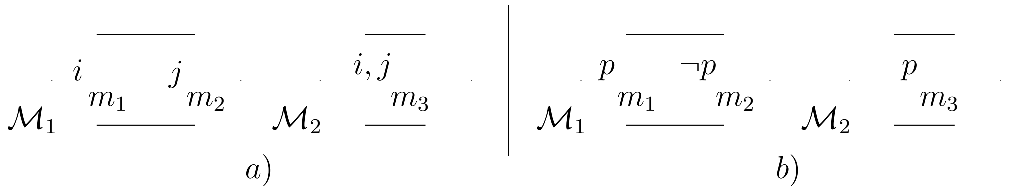

Figure 4.1: Separating languages

PROPOSITION 4.13. *Let φ be a formula in $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, then there exists a formula $\varphi' \in \mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ such that φ is satisfiable iff φ' is satisfiable.*

PROOF. Given a formula $\varphi \in \mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, introduce for each $a_i \in \mathsf{NOM}(\varphi)$ a proposition letter p_{a_k} not in $\mathsf{PROP}(\varphi)$. Define φ' as

$$\varphi' := \varphi[a_1/p_{a_1}, \ldots, a_n/p_{a_n}] \wedge \bigwedge_{a_i \in \mathsf{NOM}(\varphi)} \mathsf{COND}(a_i, \varphi)$$

where

$$\mathsf{COND}(a_i, \varphi) := @_{a_i} p_{a_i} \wedge \mathsf{A}(p_{a_i} \rightarrow \bigwedge_{\psi \in \mathsf{SF}(\varphi[a_1/p_{a_1}, \ldots, a_n/p_{a_n}])} (@_{a_i}\psi \rightarrow \psi)).$$

Notice first that φ' is a formula in $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$. Actually, the range of the translation falls into the weaker language $\mathcal{H}(\langle R^{-1}\rangle, @, \mathsf{E})$. Notice also that the translation is polynomial. The intuition behind $\mathsf{COND}(i, \varphi)$ is as in Proposition 4.4: we should force all states satisfying p_{a_i} to agree in all subformulas of $\varphi[a_1/p_{a_1}, \ldots, a_n/p_{a_n}]$. But now, instead of having to reach for the values through the accessibility relations using $[R_1 \cup R_1^{-1} \cup \cdots \cup R_m^{-1}]$, we can simply jump to a_i by using @. It rests to prove satisfiability preservation.

The left to right direction is simple. Given $\mathcal{M}, w \Vdash \varphi$, define $\mathcal{M}' = \langle M', \{R'_r\}, V'\rangle$ as follows: $M' = M$, $R'_r = R_r$ and $V'(p_{a_i}) = V(a_i)$ for p_{a_i} one of the propositions used in the translation, and $V'(a) = V(a)$ for any other atom. $\mathcal{M}', w \Vdash \varphi'$.

For the other direction, let $\mathcal{M}' = \langle M', \{R'_r\}, V'\rangle$, $w' \in M'$ be such that $w' \Vdash \varphi'$. Let $\mathcal{M}^f$ be a filtration of $\mathcal{M}'$ through $\mathsf{SF}(\varphi[a_1/p_{a_1}, \ldots, a_n/p_{a_n}])$. We can prove that for $i_k \in \mathsf{NOM}(\varphi)$, $V^f(p_{a_k})$ is a singleton with a similar argument than in Proposition 4.4, but now using $@_{a_i}$. Extend V^f by setting $V^f(a_i) = V^f(p_{a_i})$, and $V^f(a) = |w'|$ for any other nominal. We will obtain $\mathcal{M}^f, |w'| \Vdash \varphi$. QED

These expressive separation results easily translate to description languages. For two description languages $\mathcal{L}_1$ and $\mathcal{L}_2$, define $\mathcal{L}_1 \preccurlyeq \mathcal{L}_2$ if for any knowledge bases Σ in $\mathcal{L}_1$ there is a knowledge base Σ' in $\mathcal{L}_2$ such that for all interpretations $\mathcal{I}$, $\mathcal{I} \models \Sigma$ iff $\mathcal{I} \models \Sigma'$. Notice now that the formulas we have used to separate the languages can easily be recast as assertions ($@_i j \Leftrightarrow i\!:\!\{j\}$) or terminological definitions ($\mathsf{A}p \Leftrightarrow \top \sqsubseteq p$), and similarly for the translation used in the proof of Proposition 4.13.

The notions of bisimulation we have defined not only separate the fragments of first-order logic which corresponds to the hybrid logics we have been discussing, they also *characterize* them. For $\mathcal{H}$ any of our hybrid languages, we say that a first-order formula $\alpha(x)$ in the first-order language over $\langle \mathsf{REL} \cup \{P_j \mid p_j \in \mathsf{PROP}\}, \mathsf{NOM}, \{x, y\}\rangle$ is *invariant for $\mathcal{H}$-bisimulations* if for all models $\mathcal{M}$ and $\mathcal{N}$, and all states m in $\mathcal{M}$, n in $\mathcal{N}$, and all $\mathcal{H}$-bisimulations $\sim$ between $\mathcal{M}$ and $\mathcal{N}$ such that $m \sim n$, we have $\mathcal{M} \models \alpha(x)[m]$ iff $\mathcal{N} \models \alpha(x)[n]$.

THEOREM 4.14. *For $\mathcal{H}$ any of $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$, $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$, $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond, \mathsf{E})$ or $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E})$, a first-order formula $\alpha(x)$ over the signature $\langle \mathsf{REL} \cup \{P_j \mid p_j \in \mathsf{PROP}\}, \mathsf{NOM}, \{x, y\}\rangle$ is invariant for $\mathcal{H}$-bisimulations iff it is equivalent to the hybrid translation of a hybrid formula in $\mathcal{H}$.*

The proof is a standard diagram chasing. We will see more details in Chapter 6.

4.5.3 Interpolation and Beth Definability

In Chapter 6 we will investigate the interpolation and Beth definability properties for a variety of hybrid languages. What is the role of these two properties in the setting of description logics?

Let's first introduce some notation. For $\Sigma = \langle T, A\rangle, \Sigma' = \langle T', A'\rangle$ two knowledge bases, let $\Sigma \cup \Sigma'$ be $\langle T \cup T', A \cup A'\rangle$, and $\Sigma[C/D]$ be the knowledge base obtained from Σ by replacing each occurrence of the concept C by D. Now, suppose that for a given knowledge base Σ the following holds,

$$\Sigma[C/D_1] \cup \Sigma[C/D_2] \models D_1 \doteq D_2 \text{ for some } D_1, D_2 \notin \mathsf{CON}(\Sigma). \tag{4.2}$$

Notice that (4.2) needs not be the case for all knowledge bases Σ and concepts C. For example, for the simple knowledge base $\Sigma = \langle\{C \sqsubseteq A\}, \{\}\rangle$ we have

$$\langle\{D_1 \sqsubseteq A, D_2 \sqsubseteq A\}, \{\}\rangle \not\models D_1 \doteq D_2.$$

Actually, (4.2) implies that Σ encodes enough information concerning C to provide a complete — though not necessarily explicit — definition. Now, if the (global) Beth definability property (see Definition 6.18) holds for the language of Σ, then there actually exists an explicit definition of C. I.e., there is a concept D not involving C such that

$$\Sigma \models C \doteq D.$$

Given that description languages take definitions very seriously, the Beth definability property (i.e., the capacity of the language to turn implicit definitions into explicit) seems highly relevant.

There are well know examples of languages for which the Beth definability property fails: the finite variable fragments of first-order logic, the →-fragment of classical propositional logic, or full first-order logic when interpreted on finite models. On the other hand for example, all modal logics extending **K4** have the Beth definability property. The work of Maksimova [1991a, 1991b, 1992a, 1992b, 1992c] is the main reference on interpolation, Beth definability and their interrelations for modal languages.

There doesn't seem to be one uniform direct way of proving or disproving Beth definability. The standard approach to establish the property is via a detour through interpolation (see Definition 6.15). In first-order and modal languages, the (arrow) interpolation property implies the Beth definability property and, as we will discuss in Section 6.2.2, the same relation holds for hybrid languages.

Hence, positive interpolation results for hybrid languages would translate into nice definability properties of the corresponding description language. Sadly, for languages where nominals appear free in formulas, and which do not provide a binding mechanism, failure of arrow interpolation seems to be the norm. In particular, in Section 6.2 we provide counter-examples to the arrow interpolation property for the basic modal language extended with nominals, $\mathcal{H}_{\mathsf{N}}(@)$ and $\mathcal{H}_{\mathsf{S}}(@)$. The extensions of these languages with the $\langle R^{-1}\rangle$ operator fare no better, and adding the E operator doesn't help either. Hence, in all these cases, the most traded path to establish Beth definability is closed for us.

The case is different for $\mathcal{H}(@, @\Diamond)$ and $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$. As we will now show, we can extend the constructive method for establishing arrow interpolation presented in [Kracht, 1999, Section 3.8], to handle @ and $@\Diamond$. Again we will make use of the normal form introduced in Proposition 4.2.

Kracht proves interpolation for a family of modal languages by means of tableaux. Given a complete tableau system for a logic $\mathcal{L}$, consider a closed tableau for $\varphi\wedge\neg\psi$. Now, proceed inductively from the tableau leaves up to the root and for the set of formulas X in each node, provide a splitting $X = X^a \cup X^c$ into antecedent and consequent formulas together with an interpolant for $\bigwedge X^a$ and $\bigwedge X^c$. This is done by analyzing one by one each of the tableau rules. At the end of the process we arrive at a formula θ in the common language such that $\varphi \wedge \neg\theta$ and $\theta \wedge \neg\psi$ have closed tableaux. Hence θ is an interpolant of $\varphi \to \psi$.

Investigating interpolation always involves paying special attention to the exact language in which deduction is carried over. The tableaux used in this kind of proofs should be specially designed along these lines, and be careful on the vocabulary used during a proof. For example, systems introducing new labels, as the constraint systems we discussed in Section 2.4, are usually of no help. The connection between tableaux systems and interpolation for modal languages has been explored in detail in [Rautenberg, 1983].

To prove interpolation for $\mathcal{H}(@, @\Diamond)$, we extend the tableau system $\mathcal{T}$ for $\mathbf{K}$ introduced in [Kracht, 1999] with rules to handle @ and $@\Diamond$, and prove that the inductive construction of the interpolant can be carried over in the extended system. Our work is particularly simple: given the normal form of formulas in $\mathcal{H}(@, @\Diamond)$, the rules for @ and $@\Diamond$ need to be applied *only once* in any closed tableau. The tableau system $\mathcal{T}$ for $\mathbf{K}$ is the following. The rules in $\mathcal{T}$ transform sets of modal formulas into new sets. Below, X, Y are sets of formulas, φ, ψ are formulas, $[R_r]S = \{[R_r]\varphi \mid \varphi \in S\}$, and $X; \varphi = X \cup \{\varphi\}$.

$$\frac{X; \varphi \wedge \psi}{X; \varphi; \psi}\ (\wedge\mathrm{E}) \qquad\qquad \frac{X; \neg\neg\varphi}{X; \varphi}\ (\neg\mathrm{E})$$

$$\frac{X; \varphi \vee \psi}{X; \varphi \mid X; \psi}\ (\vee\mathrm{E}) \qquad\qquad \frac{X; Y}{X}\ (\mathrm{W})$$

$$\frac{[R_r]X; \neg[R_r]\varphi}{X; \neg\varphi}\ ([R_r]\mathrm{E}).$$

Extend $\mathcal{T}$ to $\mathcal{T}'$ with the following rules. Below, $@_iS = \{@_i\varphi \mid \varphi \in S\}$ and we write $@_i\langle R_r\rangle j$ as $R_r(i,j)$ to avoid confusion with our notation $@_iX$.

$$\frac{@_iX}{X}\ (@_i\mathrm{E}) \qquad\qquad \frac{@_jX_1; @_i[R_r]X_2; R_r(i,j)}{X_1; X_2}\ (R_r(i,j)\mathrm{E}).$$

If a hybrid formula is in normal form, then $(@_i\mathrm{E})$ and $(R_r(i,j)\mathrm{E})$ need to be applied only once in a branch of a tableau, because the consequence of each rule is a set of basic modal formulas. Hence any closed tableau in $\mathcal{T}'$ involving some application of the two new rules, can be turned into an equivalent one which starts with a number of applications of the Boolean rules till they cannot be applied further, followed by an application of either ($@_i$E) or ($R_r(i,j)$E) (but not both) and ending as a tableau in $\mathcal{T}$. To complete the argument provided by Kracht for the new two rules, we need only verify that given an interpolant θ for $X = X^a; X^c$, $@_i\theta$ is an interpolant for $@_iX = @_iX^a; @_iX^c$. And similarly, given an interpolant θ for $X_1; X_2 = X_1^a; X_2^a; X_1^c; X_2^c$, $@_j\theta$ is an interpolant for $@_jX_1; @_i[R_r]X_2; R_r(i,j) = @_jX_1^a; @_i[R_j]X_2^a; @_jX_1^c; @_i[R_r]X_2^c; R_r(i,j)^c$.

Notice that the proof above is constructive, i.e., we can explicitly obtain an interpolant for $\varphi \rightarrow \psi$ from a tableau for $\varphi; \neg\psi$. Arrow interpolation for $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$ can be established in a similar way, my means of an extension of the tableau construction for the temporal basic logic $\mathbf{K}_t$. Hence

Theorem 4.15. *$\mathcal{H}(@, @\Diamond)$ and $\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$ have arrow interpolation.*

As we said, arrow interpolation implies global Beth definability: implicit definitions in $\mathcal{H}(@, @\Diamond)$ can be turned into explicit definitions. And we can attempt to transfer this property to the description logic counterpart of $\mathcal{H}(@, @\Diamond)$. We would do as follows, suppose a knowledge base $\Sigma = \langle T, A\rangle$ in $\mathcal{ALC}$ satisfies the conditions in (4.2). Then we can translate Σ into a theory T of $\mathcal{H}(@, @\Diamond)$ (as we are using global consequence this time we don't need E), and $T[p_C/p_{D_1}] \cup T[p_C/p_{D_2}] \models^{glo} p_{D_1} \leftrightarrow p_{D_2}$. Applying Beth definability for $\mathcal{H}(@, @\Diamond)$ we obtain a formula θ such that $T \models^{glo} \theta \leftrightarrow p_C$. Now, θ is an explicit definition of C, but it is in the *full* language $\mathcal{H}(@, @\Diamond)$, i.e., it might contain subformulas of the form $@_i\psi$ and $@_i\Diamond j$. Because of the syntactic restrictions imposed by the division into T- and A-Box information it will not always be possible to translate θ into a concept in $\mathcal{ALC}$. To see an example, suppose θ is of the form $@_i\nu \vee \psi$. Hence we will have that $\Sigma \models (@_i\nu \rightarrow (p_C \leftrightarrow \top)) \wedge (@_i\neg\nu \rightarrow (p_C \leftrightarrow \psi))$. That is, we obtain a definition of C conditioned on assertional information.

More generally, we first write θ in normal form to obtain

$$T \models^{glo} (\bigwedge_{l\in L}(\bigvee_{i\in\mathsf{NOM}} @_i\nu_{(i,l)}) \vee \tau_l) \leftrightarrow p_C.$$

Notice that for a hybrid formula ψ and $@_i\nu \in \mathsf{SF}(\psi)$ such that $@$ does not appear in ν, ψ is equivalent to $(@_i\nu \rightarrow \psi[@_i\nu/\top]) \wedge (@_i\neg\nu \rightarrow \psi[@_i\nu/\bot])$. By iterating this rewriting on $(\bigwedge_{l\in L}(\bigvee_{i\in\mathsf{NOM}} @_i\nu_{(i,l)}) \vee \tau_l) \leftrightarrow p_C$ we finally obtain a series of definitions of C in terms of concepts of $\mathcal{ALC}$, but conditioned on assertional information to be inferred from Σ.

There is an interesting connection between the Beth definability property and our discussion in Section 2.2.2 concerning restricted definitions. As we mentioned there, the restriction to acyclic definitions was aimed at avoiding the introduction of circular concepts, i.e., concepts defined in terms of themselves. This kind of concepts, it was argued, called for some kind of fixed point semantics and this kind of semantics was computationally expensive [Nebel, 1990a; Baader, 1990]. But if the language has the Beth definability property, any concept implicitly defined in a knowledge base also has an explicit definition *without self reference*. Hence, considering only acyclic definitions does not carry any expressivity loss.

4.5.4 Variables and Binders

What about the idea of introducing variables and binders? It turns out that free variables do not fit well in a global perspective. As we prove in Theorem 7.17, the global K-*Sat* problem of $\mathcal{H}_{\mathsf{S}}(@)$ is not decidable. The reason is that with the global notion of consequence, free variables are interpreted as universally quantified and the global hybrid quantifier $\forall$ is surreptitiously creeping into the picture.

Too bad, but we could still consider only sentences if we add the $\downarrow$ binder to the language. Undecidability strikes again: by Theorem 7.1 even the fragment of $\mathcal{H}_{\mathsf{S}}(\downarrow)$ consisting of pure nominal-free sentences has an undecidable local K-*Sat* problem. But if we restrict $\downarrow$ to appear non-nested, the language turns decidable. We will prove this in Theorem 7.10, here instead we will show that this non-nested use of $\downarrow$ actually has a quite natural interpretation in description logic terms.

Extend $\mathcal{ALCOI}$ to $\mathcal{ALCOI}\downarrow$ with the addition of two new operators $\mathsf{THOSE\text{-}X}$ and $\{\mathsf{X}\}$, and allow $\{\mathsf{X}\}$ and $\mathsf{THOSE\text{-}X}.C$ as concepts if C is a concept. Semantics for $\mathcal{ALCOI}\downarrow$ will be defined in terms of extended interpretations which are pairs $\langle \mathcal{I}, i\rangle$ where $\mathcal{I}$ is a standard interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ and $i \in \Delta^{\mathcal{I}}$. Now define,

$$\begin{array}{rcl} \{\mathsf{X}\}^{\langle \mathcal{I},i\rangle} & = & \{i\} \\ \mathsf{THOSE\text{-}X}.C^{\langle \mathcal{I},i\rangle} & = & \{a \in \Delta^{\mathcal{I}} \mid a \in C^{\langle \mathcal{I},a\rangle}\}. \end{array}$$

The best way to understand how $\mathsf{THOSE\text{-}X}$ and $\{\mathsf{X}\}$ work together is by trying our hand with some examples.

Example 4.16. Consider the following definitions in $\mathcal{ALCOI}\downarrow$,

$$\begin{array}{rcl} \mathtt{NOT\text{-}SELF\text{-}EMPLOYED} & \doteq & \mathsf{THOSE\text{-}X}.(\forall \mathtt{EMPLOYED\text{-}BY}.\neg\{\mathsf{X}\}) \sqcap \mathtt{HUMAN} \\ \mathtt{CORRESPONDED\text{-}LOVE} & \doteq & \mathsf{THOSE\text{-}X}.(\exists \mathtt{LOVES}.\exists \mathtt{LOVES}^{-1}.\{\mathsf{X}\}). \end{array}$$

The first concept defines the set of all those elements in the domain which are both human and which are not employed by themselves. While the second, define those happy people loving somebody who loves them back.

It is easy to show with techniques similar to the ones we used in Section 4.5.2, that the addition of these new operators (even with the restriction to non-nested occurrences) indeed provides extended expressive power. If we restrict to sentences and non-nested occurrences of $\mathsf{THOSE\text{-}X}$, we obtain a new decidable description language which seems well suited to define notions involving self reference, as the concepts in Example 4.16 show. In Chapter 5 we will discuss how to provide reasoning methods for handling $\downarrow$.

4.5.5 Accounting for Counting

Graded or counting modalities $\langle n\rangle\varphi$ allow us to restrict the number of possible successors of a given state satisfying φ.

$$\mathcal{M}, m \Vdash \langle n\rangle\varphi \text{ iff } \exists m_1, \ldots, m_n.(\bigwedge_{1\leq i<j\leq n} m_i \neq m_j \ \&\ \bigwedge_{1\leq i\leq n} R(m, m_i) \ \&\ \bigwedge_{1\leq i\leq n} \mathcal{M}, m_i \Vdash \varphi).$$

Even though these modalities have been introduced into modal languages in the 1970s [Goble, 1970; Fine, 1972], their theory is not so well developed. On the other hand, the corresponding operators $(\leq n\ R)C$ and $(\geq n\ R)C$ called qualifying number restrictions are actively used in description languages, as they lend themselves well to represent information like "every human has exactly two parents" or "applicants should provide at least two references from professors":

$$\begin{aligned}\texttt{HUMAN} &\sqsubseteq (\leq 2\ \texttt{Parent})\top \sqcap (\geq 2\ \texttt{Parent})\top,\\ \texttt{APPLICANT} &\sqsubseteq (\geq 2\ \texttt{Reference})\texttt{PROFESSOR}.\end{aligned}$$

Notice that the operator $\mathcal{Q}$ of qualifying number restriction is more expressive than the simple number restrictions $\mathcal{N}$ we introduced in Table 2.1. The definition of APPLICANT above is not possible using just $\mathcal{N}$.

Only recently, and actually stemming from the interaction between the description and modal logic communities, new result concerning counting operators have been presented. The first complexity results appeared in [de Rijke and van der Hoek, 1995], where a PSpace-completeness result is proved for the local satisfiability problem for multi-modal **Gr**(**K**) (multi-modal **K** extended with graded modalities). But the proof only covers the case when numbers in graded modalities are encoded in unary. Tobies [1999], shows that the same result obtains even when the encoding is done in binary.

Concerning model-theoretical results, [de Rijke, 2000] presents the appropriate notion of bisimulation for graded modalities, and provides a simple proof of the finite model property together with a characterization of the fragment of first-order logic corresponding to the basic modal language extended with graded modalities in the line of Theorem 4.14.

As we showed in Example 3.5, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is expressive enough to encode graded modalities. But very little is known concerning counting modalities in less powerful hybrid logics. The only two references we are aware of come from the description logic community. In [Horrocks *et al.*, 2000c] a decision method for determining the consistency of (non Boolean) knowledge bases for the description language $\mathcal{SHIQ}$ is given. $\mathcal{SHIQ}$ is an extension of $\mathcal{ALC}$ which includes transitively closed primitive roles, inverse roles, role hierarchies and qualifying number restrictions. The algorithm is an extension of a previous decision method for consistency of $\mathcal{SHIQ}$ knowledge bases with empty A-Boxes, and relays in techniques similar to the ones used in [Areces *et al.*, 1999c] and the ideas we have been using in previous sections: and A-box can be modeled by a *forest*, a set of trees whose root nodes form an arbitrarily connected graph, where the number of trees is limited by the number of individual names occurring in the A-Box.

In [Tobies, 2000a], complexity results for $\mathcal{ALCQ}$, the description logic counterpart of **Gr**(**K**), are investigated. Tobies considers T-Boxes with *cardinality restrictions*. Cardinality restrictions are expressions of the form

$$(\geq n\ C) \text{ and } (\leq n\ C)$$

for C a concept in $\mathcal{ALCQ}$. An interpretation $\mathcal{I}$ satisfies $(\geq n\ C)$ iff $|C^{\mathcal{I}}| \geq n$. This kind of knowledge bases encodes a form of *global* counting and are more expressive than those containing only terminological axioms. To witness, $(C \sqsubseteq D)$ is equivalent to $(\leq 0\ (C \sqcap \neg D))$. Tobies proves that deciding consistency of knowledge bases containing cardinality restrictions for the language $\mathcal{ALCQ}$ is ExpTime-complete, while it moves to NExpTime-complete for $\mathcal{ALCQI}$. Tobies also reports a behavior similar to what we discuss in Theorem 7.18: even though the satisfiability problem in terms of empty knowledge bases for $\mathcal{ALCQI}$ is PSpace-complete, it jumps to NExpTime-complete by the addition of a single nominal.

One interesting point for further research is the following. It will become clear in Section 6.2 that the counter-examples to arrow interpolation we present are based on a counting argument. Because the language is not expressive enough to bound the number of successors of a given state we can draw bisimulations between points with different number of successors and use this to prove failure of the interpolation property. The language extended with counting operators (even unqualified counting) would, of course, destroy the bisimilarity and hence our counter-examples, opening the way to interpolation.

4.6 Differences and Similarities

As we said in the introduction of Section 4.5, there are many more possible connections between description and hybrid languages which we didn't discuss.

On the complexity line, for example, there are interesting links between the filtration technique and the selection of maximal and minimal witnesses (see the proof of Theorem 7.22), and the blocking technique used to prove termination of completion of constraint systems when transitive roles are allowed (see [Horrocks *et al.*, 2000b]). Also having to do with complexity and decision methods, the tableau systems provided in the hybrid literature [Tzakova, 1999a; Blackburn, 2000a] differ from the constraint systems we introduced for description languages in Section 2.4, and a comparison would surely lead to new discoveries. In addition, as we discussed in Section 3.2 there are important connections between hybrid languages and labeled deduction, and these connections are now made extensive to description languages. And there is of course, the issue of implementations. Many, very powerful provers (dlp, fact, race) are available for a variety of description languages. They can already today deal with many modal languages, and it would be simple to extend them to deal with hybrid languages.

We have only scratched the surface on expressivity issues. For example, definability results for hybrid languages (like those in [de Rijke, 1992; Gargov and Goranko, 1993; de Rijke and Sturm, 2000]) shed light on which are the models which can be captured by the knowledge bases of certain description languages. More generally, [Gargov and Goranko, 1993] discusses transfer results when moving from basic modal languages to languages with nominals, while [Goranko and Passy, 1992] does a similar analysis for the extension with the existential modality. These results are closely related to the move from empty knowledge bases to non-empty A- and T-Boxes, respectively. In his original article, Schild discusses axiomatizations for description languages drawing from modal logics and CPDL. We can explore a similar path by means of the axiomatizations and completeness results for hybrid logic [Passy and Tinchev, 1991; Tzakova, 1999a].

In Sections 4.5.4 and 4.5.5 we picked just two examples of the different directions in which the two families of languages have developed, but the possible options were many. On the hybrid side, for example, the general theme of sorting (inclusion of new sets of symbols to represent a given type of information), instead of just naming, gives rise to hybrid languages which can handle intervals [Areces *et al.*, 2000a], paths [Bull, 1970; Goranko, 2000], or time granularity and reference [Blackburn, 1994]. And the available choices on the description logic side are innumerable: transitive closure, transitive roles, role hierarchies, role composition, disjointness axioms, etc.

It looks like the bridge between description and hybrid logics we have constructed will be well-traveled.

Part III
Going Places

'I know what you're thinking about,'
said Tweedledum, 'but it isn't so, nohow.'
'Contrariwise,' continued Twedledee,
'if it was so, it might be; and if it were so,
it would be; but as it isn't it ain't.
That's logic.'

from "Alice's Adventures in Wonderland," Lewis Carroll

In this part of the thesis we will tread three roads crossing the lands between and around the two kingdoms we presented in Part II. These roads are Reasoning Methods, Expressive Power, and Complexity. Each road has its own main panoramic stops, and even its own rhythm and direction. Each of them can be taken independently, relying only on the notions, results and connections we have introduced up to now.

As we saw in Chapter 4, results on description logics cast their shadows on hybrid logics and vice versa. In the chapters to come we will mainly favor a hybrid logic perspective, but Section 4.5 should have provided enough hints on the way of looking into these matters under two different lights.

In Chapter 5 we draw on lessons from description and hybrid logics to provide a direct resolution method for modal languages. Here again, individuals/nominals play a role in simplifying proof theory: state labels transform previous complex proposals for direct resolution for modal languages into elegant systems. And once the basics have been cleared up, the resolution system can easily be extended to more expressive description/hybrid languages.

In Chapter 6 we discuss expressivity. In the first half of the chapter we provide a precise characterization (both syntactically and semantically) of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ and some of its sublanguages and extensions. These results let us grasp which classes of frames can be defined in these languages. In the second half we turn to interpolation and Beth definability, completing the picture we started drawing on Section 4.5.3.

Finally, in Chapter 7 we discuss complexity results. We have already taken advantage of these results in Section 4.5.1, and we now provide full details. In particular, we will discuss the effect of considering different languages and classes of frames.

Enjoy the ride!

Chapter 5

Improving Reasoning Methods

Luchando por una verdad
pero que sea rentable.

from "El Ente," Los Visitantes

As a warming up to the purely theoretical work we will do in Chapters 6 and 7, we will now show how ideas from description and hybrid logics can be put to work with benefit even when the subject is purely modal. In particular, aided by the notions of nominals or labeling, we will show how to define well behaved direct resolution methods for modal languages. This "case study" is a clear example of how the additional flexibility provided by the ability to name states can be used to greatly simplify reasoning methods. In addition, we can build over the basic resolution system and obtain extensions for description and hybrid languages.

Reasoning methods for modal-like languages, can be broadly divided in two categories: direct and indirect. Indirect methods start by translating modal formulas into some first-order language preserving satisfiability, and then take advantage of reasoning methods for FO [de Rijke *et al.*, 2000]. Direct methods instead, work directly on modal formulas devising specialized algorithms for each modal language [Fitting, 1983].

The most developed reasoning methods for modal and modal-like languages today are direct methods, and they are mainly tableau based. Most indirect methods use first-order resolution. In contrast, *direct modal resolution* methods are poorly developed. By drawing on what we have learned in previous chapters about hybrid and description languages, we will provide a direct resolution-based proof procedure for modal languages which improves many aspects of previous proposals. After explaining in detail the resolution method for basic modal languages we will discuss extensions in the three fields of modal, description and hybrid logics. The main characteristics of the new resolution method can be summarized as follows:

- by using labeled formulas it avoids the complexity of earlier direct resolution-based methods for modal logic.
- it does not involve skolemization beyond the use of constants;
- it does not involve translation into large undecidable languages, working directly on modal, hybrid or description logic formulas instead;
- as far as we know, its extension to DLs is the first to account for knowledge base inference by means of a direct resolution approach;
- it is flexible and conservative in more than one sense: it allows the amalgamation of different ideas. In particular it incorporates the method of prefixes used in tableaux into resolution in such a way that different heuristics and optimizations devised in either field are applicable.

5.1 Resolution

Resolution, introduced originally for FO in [Robinson, 1965], is the most widely spread reasoning method for first-order logic today: most of the available automatic theorem provers for FO are resolution based. The elegance of the resolution method and its appeal for implementation rely on its bare simplicity.

Let us discuss the propositional case. To check whether a propositional formula φ is inconsistent, we first turn it into *clausal form*. To this aim, write φ in conjunctive normal form

$$\varphi = \bigwedge_{l \in L} \bigvee_{m \in M} \psi_{(l,m)},$$

and let the clause set associated with φ be

$$\mathit{ClSet}(\varphi) = \{\{\psi_{(l,m)} \mid m \in M\} \mid l \in L\}.$$

Now define $\mathit{ClSet}^*(\varphi)$ as the smallest set containing $\mathit{ClSet}(\varphi)$ and closed under a unique, very simple to grasp rule,

$$\frac{\mathit{Cl}_1 \cup \{N\} \in \mathit{ClSet}^*(\varphi) \quad \mathit{Cl}_2 \cup \{\neg N\} \in \mathit{ClSet}^*(\varphi)}{\mathit{Cl}_1 \cup \mathit{Cl}_2 \in \mathit{ClSet}^*(\varphi)} \text{ (RES).}$$

If $\{\} \in \mathit{ClSet}^*(\varphi)$, then φ is inconsistent. The intuition behind the (RES) rule is as follows: given that either N or $\neg N$ is always the case in any model they can be "cut away" if the sets of clauses are conjoined. The aim of the whole method is to "cut away everything" and arrive to the empty set.

The resolution method seems to be specially devised for a dumb machine able to crunch symbols quickly. The only computational cost is a search for complementary atoms in the set of clauses. Of course, actual system implementations for first-order logic are not "dumb" at all. On the contrary, the field has developed into an extensive community, with an impressive collection of methods, optimizations, etc. [Bibel and Schmitt, 1998; Robinson and Voronkov, 2000].

In contrast, modern modal theorem provers, as well as the fastest description logic provers we mentioned in Chapter 2, are generally based on tableau methods. Strangely enough, nowadays resolution and modal languages seem to be related only when indirect methods are used. In translation-based resolution calculi for modal logics, one translates modal languages into a large background language (typically first-order logic), and devises strategies that guarantee termination for the fragment corresponding to the original modal language [Fermüller *et al.*, 1993; Hustadt, 1999; de Nivelle *et al.*, 2000; Areces *et al.*, 2000d]. First-order resolution provers like BLIKSEM or SPASS handle modal formulas in this way. This approach has both advantages and disadvantages with respect to the tableau approach. On the one hand we can translate many systems into the same background language and hence explore different, and also combined, systems without the need to modify the prover. But empirical tests show that the price to pay is high [Horrocks *et al.*, 2000a; Areces *et al.*, 2000d]. The undecidability of the full background language usually shows up in degraded performance on the modal fragments, and first-order provers can hardly emulate their tableau based competitors.

It is natural to wonder why *direct* resolution methods for modal languages don't figure in the picture. Designing resolution methods that can directly (that is, without having to perform translations) be applied to modal logics, received some attention in the late 1980s and early 1990s [Enjalbert and Fariñas del Cerro, 1989; Mints, 1989; de Nivelle, 1994]. Also the first (non-clausal) resolution methods for temporal languages go back to that period with the work of Abadi and Manna [1985]. Recently, new results on clausal temporal resolution have been presented (see [Dixon *et al.*, 2000]). But even though we might sometimes think of modal languages as a "simple extension of propositional logic," direct resolution for modal languages has proved a difficult task: in basic modal languages the resolution rule has to operate *inside* boxes and diamonds to achieve completeness. This leads to more complex systems, less elegant results, and poorer performance, ruining the one-dumb-rule spirit of resolution.

5.1.1 Direct Resolution for Modal Languages

To understand exactly how we can use hybrid and description logic ideas to improve direct modal resolution, we introduce the system presented by Enjalbert and Fariñas del Cerro in [1989]. We first provide some definitions and notation from [Enjalbert and Fariñas del Cerro, 1989], as they are not completely standard.

A modal formula is in *disjunctive normal form* if it is a (possibly empty) disjunction of the form

$$\varphi = \bigvee L_i \vee \bigvee \Box D_j \vee \bigvee \Diamond A_k,$$

where each L_i is a literal, each D_j is in disjunctive normal form, and each A_k is in conjunctive normal form. A modal formula is in *conjunctive normal form* if it is a conjunction $\varphi = \bigwedge C_i$, where each C_i is in disjunctive normal form. A formula in disjunctive normal form is called a *clause*. The empty clause is denoted as $\bot$. We identify a conjunction $C_1 \wedge \ldots \wedge C_n$ with the set $(C_1, \ldots, C_n)$. Clearly any modal formula is equivalent to a clause, and from now on we need only consider clauses.

The following examples of applications of the resolution rule "in modal contexts" are discussed in [Enjalbert and Fariñas del Cerro, 1989]

$$\frac{\Box(p \vee q) \qquad \Diamond\neg p}{\Diamond(\neg p, q)} \qquad\qquad \frac{\Box(p \vee q) \qquad \Box\neg p}{\Box q}.$$

Both inferences are sound, and are clearly instances of the (RES) rule. But if we attempt to apply a similar rule to the clauses $\Diamond(p \vee q)$ and $\Diamond\neg p$ to derive $\Diamond(\neg p, q)$ we don't preserve soundness. Also, inferences with only one premise seem to be needed, as for example

$$\frac{\Diamond(\neg p, p \vee q)}{\Diamond(\neg p, p \vee q, q)}.$$

In line with these intuitions, the following resolution system is introduced and proved complete for **K**. Define inductively two relations on clauses $\Sigma(A, B) \rightarrow C$ (C is a direct resolvent of A and B) and $\Gamma(A) \rightarrow C$ (C is a direct resolvent of A), as indicated in Figure 5.1, where α, β, κ, δ_1, δ_2 are clauses, Ψ, Φ are sets (conjunctions) of clauses, and (α, Ψ) denotes the result of appending the clause α to the set Ψ.

Define the simplification relation $A \approx B$ as the least congruence containing

Axioms	
(A1) $\Sigma(p, \neg p) \to \bot$ (A2) $\Sigma(\bot, \alpha) \to \bot$	
Σ-Rules	**Γ-Rules**
$(\vee)\ \dfrac{\Sigma(\alpha,\beta) \to \kappa}{\Sigma(\alpha \vee \delta_1, \beta \vee \delta_2) \to \kappa \vee \delta_1 \vee \delta_2}$	$(\Diamond 1)\ \dfrac{\Sigma(\alpha,\beta) \to \kappa}{\Gamma(\Diamond(\alpha,\beta,\Phi)) \to \Diamond(\alpha,\beta,\kappa,\Phi)}$
$(\Box\Diamond)\ \dfrac{\Sigma(\alpha,\beta) \to \kappa}{\Sigma(\Box\alpha, \Diamond(\beta,\Psi)) \to \Diamond(\beta,\kappa,\Psi)}$	$(\Diamond 2)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\Diamond(\alpha,\Phi)) \to \Diamond(\beta,\alpha,\Phi)}$
$(\Box\Box)\ \dfrac{\Sigma(\alpha,\beta) \to \kappa}{\Sigma(\Box\alpha, \Box\beta) \to \Box\kappa}$	$(\vee)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\alpha \vee \kappa) \to \beta \vee \kappa}$
	$(\Box)\ \dfrac{\Gamma(\alpha) \to \beta}{\Gamma(\Box\alpha) \to \Box\beta}$

Figure 5.1: Resolution rules

$$\begin{array}{rcl} \Diamond\bot & \approx & \bot \\ \bot \vee D & \approx & D \\ (\bot, E) & \approx & \bot \\ A \vee A \vee D & \approx & A \vee D. \end{array}$$

For any formula F there is a unique F' such that $F \approx F'$ and F' cannot be simplified further. We call F' the normal form of F. C is a *resolvent* of A and B (respectively A) iff there is some C' such that $\Sigma(A, B) \to C'$ (respectively, $\Gamma(A) \to C'$) and C is the normal form of C'. We write $\Sigma(A, B) \Rightarrow C$ (respectively, $\Gamma(A) \Rightarrow C$) if C is a resolvent of A and B (respectively, of A).

Given a set of clauses S, let *ClSet*$^*(S)$ be the smallest set containing S and closed under resolvents of elements in *ClSet*$^*(S)$. We say that D is a *resolution consequence* of a set of clauses S (notation $S \vdash D$) iff $D \in$ *ClSet*$^*(S)$.

THEOREM 5.1. *For S a set of clauses and D a clause, $S \vdash D$ iff $\models_{\mathbf{K}} S \to D$.*

So much for the one-rule-spirit of resolution. Let us go through an example to better understand how this resolution method works.

EXAMPLE 5.2. Consider the formula $\Diamond(p \wedge (\neg p \vee \Box r \vee q)) \wedge \Box\neg q \wedge \Box\Diamond\neg r$. In the resolution proof below we underline the literals on which resolution takes place, and simplify many steps for succinctness.

1. $(\Diamond(\underline{p}, \underline{\neg p} \vee \Box r \vee q), \Box\neg q, \Box\Diamond\neg r)$
   by (A1), ($\vee$) and ($\Diamond$1)
2. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee \underline{q}), \Box\underline{\neg q}, \Box\Diamond\neg r)$
   by (A1), ($\vee$) and ($\Box\Diamond$)
3. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee q, \Box\underline{r}), \Box\neg q, \Box\Diamond\underline{\neg r})$
   by (A1) and two applications of ($\Box\Diamond$)
4. $(\Diamond(p, \neg p \vee \Box r \vee q, \Box r \vee q, \Box r, \Diamond(\neg r, \bot)), \Box\neg q, \Box\Diamond\neg r)$
   by the simplification $\Diamond\bot = \bot$, (A2) and ($\Diamond$1)
5. $\bot$

As we see, the direct resolution method for modal logics presented in [Enjalbert and Fariñas del Cerro, 1989] (and similarly those in [Fariñas del Cerro, 1982; Mints, 1989; de Nivelle, 1994] perform resolution "inside" modalities (in a similar way as how new tableaux have to be started in non-prefixed tableaux systems).

In the next sections we develop a direct resolution method for modal, description and hybrid logics that retains as much as possible of the lean one-rule character of traditional resolution methods. The key idea introduced here, from a basic modal logic perspective, is to use labels to decorate formulas with additional information. Labels allow us to make information explicit and resolution can then always be performed at the "top level." From a description or hybrid logic perspective we have just taking advantage of the new expressive power that individuals/nominals provide.

5.1.2 Labeled Resolution

In this section we introduce a direct resolution proof procedure for the basic multi-modal logic $\mathbf{K}_m$. In what follows, we assume fixed a modal similarity type $\mathcal{S} = \langle \mathsf{REL}, \mathsf{PROP} \rangle$, together with a hybrid/description logic similarity type (without state variables) $\mathcal{S}' = \langle \mathsf{REL}, \mathsf{PROP}, \mathsf{LAB} \rangle$ where LAB is a countably infinite set of nominals/individuals.

Definition 5.3. [Weak negation normal form] Define the following rewriting procedure *wnnf* on modal formulas

i. $\neg\neg\varphi \overset{wnnf}{\rightsquigarrow} \varphi$,

ii. $\langle R \rangle \varphi \overset{wnnf}{\rightsquigarrow} \neg([R]\neg\varphi)$,

iii. $(\varphi_1 \vee \varphi_2) \overset{wnnf}{\rightsquigarrow} \neg(\neg\varphi_1 \wedge \neg\varphi_2)$.

For any formula φ, *wnnf* converges to a unique normal form *wnnf*(φ) which is logically equivalent to φ. If we take $\vee$ and $\langle R \rangle\varphi$ as defined operators, then *wnnf* is slightly more than an expansion of definitions.

Definition 5.4. [Clauses] A *clause* is a set Cl such that each element of Cl is a *labeled formula* of the form $t : \varphi$ or $(t_1, t_2) : R$ for $t, t_1, t_2 \in \mathsf{LAB}$, $R \in \mathsf{REL}$ and φ a basic multi-modal formula. Let φ be a basic multi-modal formula. The set S_φ of clauses corresponding to φ is simply $\{\{a : \mathit{wnnf}(\varphi)\}\}$, for a an arbitrary label in LAB.

Notice that formulas in a clause can be seen as assertions in a description language. Let Cl be a clause, and $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ be a description logic interpretation on $\mathcal{S}'$, we write $\mathcal{I} \models Cl$ if $\mathcal{I} \models \bigvee Cl$. A set of clauses S is *satisfiable* if there is interpretation $\mathcal{I}$ such that for all $Cl \in S$, $\mathcal{I} \models Cl$.

The following proposition is straightforward,

Proposition 5.5. *Let φ be a basic multi-modal formula and S_φ its corresponding set of clauses. Then φ is satisfiable iff S_φ is satisfiable.*

Proof. For the left to right implication, given a model $\mathcal{M}$ and $m \in M$ such that $\mathcal{M}, m \Vdash \varphi$, just define $a^{\mathcal{I}} = m$ and give any interpretation to others elements in LAB. For the other direction, just drop the interpretation of elements in LAB. QED

$$(\wedge)\quad \frac{Cl \cup \{t\!:\!\varphi_1 \wedge \varphi_2\}}{\begin{array}{c} Cl \cup \{t\!:\!\varphi_1\} \\ Cl \cup \{t\!:\!\varphi_2\}\end{array}} \qquad (\neg\wedge)\quad \frac{Cl \cup \{t\!:\!\neg(\varphi_1 \wedge \varphi_2)\}}{Cl \cup \{t\!:\!\mathit{wnnf}(\neg\varphi_1), t\!:\!\mathit{wnnf}(\neg\varphi_2)\}}$$

$$(\mathrm{RES})\quad \frac{Cl_1 \cup \{t\!:\!\varphi\} \quad Cl_2 \cup \{t\!:\!\neg\varphi\}}{Cl_1 \cup Cl_2}$$

$$([R])\quad \frac{Cl_1 \cup \{t_1\!:\![R]\varphi\} \quad Cl_2 \cup \{(t_1,t_2)\!:\!R\}}{Cl_1 \cup Cl_2 \cup \{t_2\!:\!\varphi\}}$$

$$(\neg[R])\quad \frac{Cl \cup \{t\!:\!\neg[R]\varphi\}}{\begin{array}{c} Cl \cup \{(t,n)\!:\!R\} \\ Cl \cup \{n\!:\!\mathit{wnnf}(\neg\varphi)\}\end{array}}, \text{ where } n \text{ is new.}$$

Figure 5.2: Labeled resolution rules

Figure 5.2 provides a set of rules transforming sets of clauses into sets of clauses.

If you read the rules with the standard translation ST of Definition 1.19 in the back of your mind, the meaning of $([R])$ and $(\neg[R])$ will be immediately clear. $([R])$ is needed to account for the "hidden" negation in the guard of the quantifier in the translation of the box, and in that sense it is indeed a standard resolution rule which cuts away complementary binary literals. On the other hand, $(\neg[R])$ can be seen as a mild kind of skolemization which only involves the introduction of constants. From this point of view we can consider the $(\wedge)$, $(\neg\wedge)$ and $(\neg[R])$ rules as preprocessing the input formula and feeding it into the resolution rules (RES) and $(\neg[R])$. Equivalently, we can view the system as intermingling the reduction towards a standard clausal form with the resolution steps as in [Fitting, 1990]. One immediate advantage of this method is that resolution can be performed not only on literals, but on complex formulas.

Definition 5.6. [Deduction] A *deduction* of a clause Cl from a set of clauses S is a finite sequence $S_1, \ldots, S_n$ of sets of clauses such that $S = S_1$, $Cl \in S_n$ and each S_i (for $i > 1$) is obtained from S_{i-1} by adding the consequent clauses of the application of one of the resolution rules in Figure 5.2 to clauses in S_{i-1}. Cl is a *consequence* of S if there is a deduction of Cl from S. A deduction of $\{\}$ from S is a *refutation* of S.

The set $\mathit{ClSet}^*(S)$, defined as the smallest set containing S and all its consequences, need not be finite because the rule $(\neg[R])$ can introduce infinitely many clauses which only differ on the label. By restricting $(\neg[R])$ to be "fired only once" in a way similar as how is done for constraint systems in Table 2.2, we can ensure finiteness of $\mathit{ClSet}^*(S)$, and hence termination of the search for consequences.

Before moving on, let's redo Example 5.2 in the new resolution system. Again we underline the part of the formula where a rule applies. Notice that we are now explicitly showing *all* steps.

Example 5.7.

1. $\{i\!:\!\underline{\neg\Box}\neg(p \wedge \neg(p \wedge \neg\Box r \wedge \neg q))\}, \{i\!:\!\Box\neg q\},\ \{i\!:\!\Box\neg\Box r\},$ by $(\neg\Box)$
2. $\{R(i,j)\}, \{j\!:\!(p\underline{\wedge}\neg(p \wedge \neg\Box r \wedge \neg q))\}, \{i\!:\!\Box\neg q\},\ \{i\!:\!\Box\neg\Box r\},$ by $(\wedge)$

$$\begin{array}{llr}
3. & \{R(i,j)\}, \{j:p\}, \{j:\underline{\neg}(p\underline{\wedge}\neg\Box r\underline{\wedge}\neg q)\}, \{i:\Box\neg q\},\ \{i:\Box\neg\Box r\}, & \text{by } (\neg\wedge) \\
4. & \{R(i,j)\}, \{j:\underline{p}\}, \{j:\underline{\neg p}, j:\Box r, j:q\}, \{i:\Box\neg q\},\ \{i:\Box\neg\Box r\}, & \text{by (RES)} \\
5. & \{R(i,j)\}, \{j:\Box r, j:q\}, \{i:\underline{\Box}\neg q\},\ \{i:\Box\neg\Box r\}, & \text{by } (\Box) \\
6. & \{j:\Box r, j:\underline{q}\}, \{j:\underline{\neg q}\},\ \{j:\neg\Box r\}, & \text{by (RES)} \\
7. & \{j:\underline{\Box r}\},\ \{j:\underline{\neg\Box r}\}, & \text{by (RES)} \\
8. & \{\}. &
\end{array}$$

It is straightforward to prove that the resolution rules in Figure 5.2 preserve satisfiability. That is, given a rule, if the premises are satisfiable, then so are the conclusions. In Section 5.2.2, we will extend the system to deal with knowledge bases in the description language $\mathcal{ALCR}$, and prove there in detail, soundness, completeness and termination.

5.2 Extensions and Variations

The system we have just introduced can be extended in different directions. In this section we discuss first how to account for modal systems different from $\mathbf{K}_m$. The next step — one which should by now come very naturally to us — is to internalize into the object language the labels we used to assist resolution. In particular, we will extend the calculus above to deal with (simple, acyclic) knowledge bases in $\mathcal{ALCR}$. Finally, we briefly discuss extensions for hybrid languages.

5.2.1 Modal Logics

From a traditional modal point of view we often want to consider systems above $\mathbf{K}_m$. We choose systems **T**, **D**, and **4** as examples. Each system is axiomatically defined as an extension of the basic system **K** by the addition of an axiom scheme which characterizes certain property of the accessibility relation.

Name	Axiom Scheme	Accessibility Relation	
T	$p \to \Diamond p$	reflexivity:	$\forall x.R(x,x)$
D	$\Box p \to \Diamond p$	seriality:	$\forall x \exists y.R(x,y)$
4	$\Diamond\Diamond p \to \Diamond p$	transitivity:	$\forall xyz.(R(x,y) \wedge R(y,z) \to R(x,z))$

Corresponding to each of the axioms we add a new resolution rule.

$$(\mathbf{T}) \quad \frac{Cl \cup \{t:\Box\varphi\}}{Cl \cup \{t:\varphi\}}$$

$$(\mathbf{D}) \quad \frac{Cl \cup \{t:\Box\varphi\}}{Cl \cup \{t:\neg\Box \mathit{wnnf}(\neg\varphi)\}}$$

$$(\mathbf{4}) \quad \frac{Cl_1 \cup \{t_1:\Box\varphi\} \quad Cl_2 \cup \{(t_1,t_2):R\}}{Cl_1 \cup Cl_2 \cup \{t_2 : \Box\varphi\}}.$$

Soundness for these systems is immediate:

Theorem 5.8. *The resolution methods obtained by adding the rules* (**T**), (**D**) *and* (**4**), *are sound with respect to the class of models where the relation R is reflexive, serial and transitive, respectively.*

For completeness and termination we should modify the constructions in Section 5.2.2 (in particular (**4**) needs a mechanism of cycle detection); this can be done using methods from [Enjalbert and Fariñas del Cerro, 1989].

THEOREM 5.9. *The resolution methods obtained by adding the rules* (**T**), (**D**) *and* (**4**), *are complete and terminate with respect to the class of models where the relation is reflexive, serial and transitive, respectively.*

5.2.2 Description Logics

In this section we will spell out the details of a labeled resolution system to decide consistency of simple, acyclic knowledge bases in the description logic $\mathcal{ALCR}$ (see Section 2.2.2). We assume fixed a description logic signature $\langle \mathsf{CON}, \mathsf{ROL}, \mathsf{IND} \rangle$ together with an additional countable set of labels LAB.

The new definition of weak negation normal form is simply a notational variation, obtained by exchanging $\vee$ by $\sqcup$, $\wedge$ by $\sqcap$, etc. Again, for any concept C, *wnnf* always converges to a unique normal form which we denote as *wnnf*(C). The definition of clauses and set of clauses associated to a knowledge base are only slightly more involved.

DEFINITION 5.10. A *clause* is a set Cl such that each element of Cl is either a concept assertion of the form $t\!:\!C$ where $t \in \mathsf{IND} \cup \mathsf{LAB}$, or a role assertion of the form $(t_1, t_2)\!:\!R$, where t_1, t_2 are in $\mathsf{IND} \cup \mathsf{LAB}$.

We will use the notation $t\!:\!C$ for concept assertions and $(t_1, t_2)\!:\!R$ for role assertions. We use the notation $\bar{t}\!:\!N$ to refer both to concept and role assertions.

A formula in a clause is a *literal* if it is either a role assertion, a concept or negated concept assertion on an atomic concept, or a universal or negated universal concept assertion. The notions of model for a clause and for a set of clauses are as in Definition 5.4.

Let $\Sigma = \langle T, A \rangle$ be a knowledge base with simple, acyclic definitions. As we discussed in Section 2.2.2, any such knowledge base can be transformed into an "unfolded" equivalent knowledge base of the form $\langle \{\}, A \rangle$. Hence, from now on we will only consider knowledge bases with empty T-boxes.

DEFINITION 5.11. [Set of clauses of a knowledge base] The set $S_\Sigma = \langle \{\}, A \rangle$ of clauses corresponding to Σ is the smallest set such that

- if $a\!:\!C_1 \sqcap \cdots \sqcap C_n =$ *wnnf*($a\!:\!C$) for $a\!:\!C \in A$ then $\{a\!:\!C_i\} \in S_\Sigma$,
- if $(a, b)\!:\!R_1 \sqcap \cdots \sqcap R_n \in A$ then $\{(a, b)\!:\!R_i\} \in S_\Sigma$.

We can identify in S_Σ a (possibly empty) subset of clauses RA of the form $\{(a, b)\!:\!R\}$ which we call *role assertions*, and for each label a a (possibly empty) subset CA_a of clauses of the form $\{a\!:\!C\}$ which we call *concept assertions for* a. Because of the format of a knowledge base it is impossible to find in S_Σ *mixed* clauses containing both (in disjunction) concept and role assertions. Furthermore there are no disjunctive concept assertions on different labels, i.e., there is no clause Cl in S_Σ such that $Cl = Cl' \cup \{a\!:\!C_1\} \cup \{b\!:\!C_2\}$ for $a \neq b$. We will take advantage of these properties in the first steps of the completeness proof.

$$(\sqcap)\quad \frac{Cl \cup \{\bar{t}\!:\!N_1 \sqcap N_2\}}{\begin{array}{c}Cl \cup \{\bar{t}\!:\!N_1\}\\ Cl \cup \{\bar{t}\!:\!N_2\}\end{array}} \qquad (\neg\sqcap)\quad \frac{Cl \cup \{t\!:\!\neg(C_1 \sqcap C_2)\}}{Cl \cup \{t\!:\!\mathit{wnnf}(\neg C_1), t\!:\!\mathit{wnnf}(\neg C_2)\}}$$

$$(\mathrm{RES})\quad \frac{Cl_1 \cup \{\bar{t}\!:\!N\} \quad Cl_2 \cup \{\bar{t}\!:\!\neg N\}}{Cl_1 \cup Cl_2}$$

$$(\forall)\quad \frac{Cl_1 \cup \{t_1\!:\!\forall R.C\} \quad Cl_2 \cup \{(t_1,t_2)\!:\!R\}}{Cl_1 \cup Cl_2 \cup \{t_2\!:\!C\}}$$

$$(\neg\forall)\quad \frac{Cl \cup \{t\!:\!\neg\forall R.C\}}{\begin{array}{c}Cl \cup \{(t,n)\!:\!R\}\\ Cl \cup \{n\!:\!\mathit{wnnf}(\neg C)\}\end{array}}, \text{ where } n \text{ is new.}$$

Figure 5.3: Labeled resolution rules for $\mathcal{ALCR}$

Proving that Σ is consistent if and only if S_Σ has a model is straightforward. Figure 5.3 shows the labeled resolution rules, but this time recast for the language $\mathcal{ALCR}$. Before proving soundness, completeness and termination we present a simple example of resolution in our system.

Example 5.12. Consider the following description. Ignoring some fundamental genetic laws, suppose that children of tall people are blond (1). Furthermore, all Tom's daughters are tall (2), but he has a non-blond grandchild (3). Can we infer that Tom has a son (4)?

$$\begin{array}{cc}
(0) & \mathtt{FEMALE} \doteq \neg\mathtt{MALE}\\
(1) & \mathtt{TALL} \sqsubseteq \forall\mathtt{Child}.\mathtt{BLOND}\\
(2) & \mathtt{tom}\!:\!\forall\mathtt{Child}.(\neg\mathtt{FEMALE} \sqcup \mathtt{TALL})\\
(3) & \mathtt{tom}\!:\!\exists\mathtt{Child}.\exists\mathtt{Child}.\neg\mathtt{BLOND}\\
\hline
(4) & \mathtt{tom}\!:\!\exists\mathtt{Child}.\mathtt{MALE}.
\end{array}$$

As is standard, we use a new proposition letter REST-TALL to complete the partial definition in (1) and we resolve with the negation of the formula we want to infer. After unfolding and applying *wnnf* we obtain the following three clauses

1. $\{\mathtt{tom}\!:\!\forall\mathtt{Child}.\neg(\neg\mathtt{MALE} \sqcap \neg((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL}))\}$
2. $\{\mathtt{tom}\!:\!\neg\forall\mathtt{Child}.\forall\mathtt{Child}.\mathtt{BLOND}\}$
3. $\{\mathtt{tom}\!:\!\forall\mathtt{Child}.\neg\mathtt{MALE}\}$.

Now we start resolving,

4. $\{\mathtt{s}\!:\!\neg\forall\mathtt{Child}.\mathtt{BLOND}\}$ by $(\neg\forall)$ in 2
5. $\{(\mathtt{tom},\mathtt{s})\!:\!\mathtt{Child}\}$ by $(\neg\forall)$ in 2
6. $\{\mathtt{s}\!:\!\neg\mathtt{MALE}\}$ by $(\forall)$ in 3
7. $\{\mathtt{s}\!:\!\neg(\neg\mathtt{MALE} \sqcap \neg((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$ by $(\forall)$ in 1
8. $\{\mathtt{s}\!:\!\mathtt{MALE}, \mathtt{s}\!:\!((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$ by $(\neg\sqcap)$ in 7
9. $\{\mathtt{s}\!:\!((\forall\mathtt{Child}.\mathtt{BLOND}) \sqcap \mathtt{REST\text{-}TALL})\}$ by (RES) in 6 and 8
10. $\{\mathtt{s}\!:\!\forall\mathtt{Child}.\mathtt{BLOND}\}$ by $(\sqcap)$ in 9
11. $\{\mathtt{s}\!:\!\mathtt{REST\text{-}TALL}\}$ by $(\sqcap)$ in 9
12. $\{\}$ by (RES) in 4 and 10.

Theorem 5.13. [Soundness] *The rules described in Figure 5.3 are sound. That is, if Σ is a knowledge base, then S_Σ has a refutation only if Σ is unsatisfiable.*

Proof. We prove that labeled resolution rules preserve satisfiability. We only discuss $(\neg\forall)$. Let $\mathcal{I}$ be a model of the antecedent. If $\mathcal{I}$ is a model of Cl we are done. If $\mathcal{I}$ is a model of $t\!:\!\neg\forall R.C$, then there exists d in the domain, such that $(t^{\mathcal{I}}, d) \in R^{\mathcal{I}}$ and $d \in \neg C^{\mathcal{I}}$. Let $\mathcal{I}'$ be identical to $\mathcal{I}$ except perhaps in the interpretation of n where $n^{\mathcal{I}'} = d$. As n is a new label, also $\mathcal{I}' \models t\!:\!\neg\forall R.C$. But now $\mathcal{I}' \models Cl \cup \{(t,n)\!:\!R\}$ and $\mathcal{I}' \models Cl \cup \{n\!:\!\mathit{wnnf}(\neg C)\}$. QED

We now prove completeness. We follow the approach used in [Enjalbert and Fariñas del Cerro, 1989]: given a set of clauses S we aim to define a structure T_S such that

(†) if S is satisfiable, a model can be effectively constructed from T_S; and
(††) if S is unsatisfiable, a refutation can be effectively constructed from T_S.

But in our case we have to deal also with A-Box information, that is, with named objects (concept assertions) and fixed constraints on relations (role assertions). We will proceed in stages. To begin, we will obtain a first structure to account for named states and their fixed relation constraints. After that we can use a simple generalization of results in [Enjalbert and Fariñas del Cerro, 1989]. We base our construction on trees which will help in guiding the construction of the corresponding refutation proof.

Let Σ be a knowledge base and S_Σ its corresponding set of clauses. Let a be a label and CA_a the subset of CA of concept assertions concerning the label a. Construct inductively, for each CA_a, a binary tree T_a. Let the original tree u consist of the single node CA_a and repeat in alternation the following operations.

Operation A1. Repeat the following steps as long as possible:

- choose a leaf w. Replace any clause of the form $\{a\!:\!\neg(C_1 \sqcap C_2)\}$ by $\{a\!:\!\mathit{wnnf}(\neg C_1), a\!:\!\mathit{wnnf}(\neg C_2)\}$; and any clause of the form $\{a\!:\!C_1 \sqcap C_2\}$ by $\{a\!:\!C_1\}$ and $\{a\!:\!C_2\}$.

Operation A2. Repeat the following steps as long as possible:

- choose a leaf w of u and a clause Cl in w of the form $Cl = \{a\!:\!C_1, a\!:\!C_2\} \cup Cl'$;
- add two children w_1 and w_2 to w, where $w_1 = w \backslash \{Cl\} \cup \{\{a : C_1\}\}$ and $w_2 = w \backslash \{Cl\} \cup \{\{a\!:\!C_2\} \cup Cl'\}$.

The leaves of T_a give us the possibilities for "named states" in our model. We can view each leaf as a set S_a^j, representing a possible configuration for state a.

Proposition 5.14. *Operation A (the combination of A1 and A2) terminates, and upon termination*

i. all the leaves $S_a^1, \ldots, S_a^n$ of the tree are sets of unit literal clauses,
ii. if all $S_a^1, \ldots, S_a^n$ are refutable, then CA_a is refutable,
iii. if one S_a^j is satisfiable, then CA_a is satisfiable.

Proof. Termination is trivial. *i*) holds by virtue of the construction, and *ii*) is proved by induction on the depth of the tree. We need only realize that by simple propositional resolution, if the two children of a node w are refutable, then so is w. *iii*) is also easy. Informally, Operation A "splits" disjunctions and "carries along" conjunctions. Hence if some S_a^j has a model we have a model satisfying all conjuncts in CA_a and at least one of each disjunct. QED

We should now consider the set RA of role assertions. Let NAMES be the set of labels which appear in Σ. If a is in NAMES but CA_a is empty in S_Σ, define $S_a^1 = \{\{a : C, a : \neg C\}\}$ for some concept C. We will construct a set of sets of nodes $\mathcal{N} = \{N_i \mid N_i$ contains exactly one leaf of each $T_a\}$. Each N_i is a possible set of constraints for the named worlds in a model of S_Σ.

Proposition 5.15. *If for all i, $\bigcup N_i \cup RA$ is refutable, then so is S_Σ.*

Proof. If for all i, $\bigcup N_i \cup RA$ is refutable, then for some label a we have that for all S_a^j obtained from CA_a, $S_a^j \cup RA$ is refutable. Hence by Proposition 5.14, $CA_a \cup RA$ is refutable, and so is S_Σ. QED

For all i, we will now extend each set in N_i with further constraints. For each $S_a \in N_i$, start with a node w_a labeled by S_a.

Operation B1. Equal to *Operation A1.*

Operation B2. Repeat the following steps as long as possible:

- choose nodes w_a, w_b such that $\{(a,b):R\}$ in RA, $\{a:\forall R_i.C_i\} \in w_a$, $\{b:C_i\} \notin w_b$, where w_b is without children;
- add a child to w_b, $w_b' = w_b \cup \{\{b:C_i\}\}$.

Call N_i^* the set of all leaves obtained from the forest constructed in B.

Proposition 5.16. *Operation B terminates, and upon termination*

i. all nodes created are derivable from $\bigcup N_i \cup RA$, and hence if a leaf is refutable so is $\bigcup N_i \cup RA$,

ii. if some $\bigcup N_i^*$ is satisfiable, then S_Σ is satisfiable.

Proof. To prove termination, notice that in each cycle the quantifier depth of the formulas considered decreases. Furthermore, it is not possible to apply twice the operation to a node named by a and b and a formula $a:\forall R_i.C_i$.

As to i), each node is created by an application of the ($\forall$) rule to members of $N_i \cup RA$ or clauses previously derived by such applications. To prove ii), let $\mathcal{I}$ be a model of N_i^*. Define $\mathcal{I}' = \langle \Delta', \cdot^{\mathcal{I}'} \rangle$ as $\Delta' = \Delta$, $a^{\mathcal{I}'} = a^{\mathcal{I}}$ for all labels a, $C^{\mathcal{I}'} = C^{\mathcal{I}}$ for all atomic concepts C, and $R^{\mathcal{I}'} = R^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid \{(a,b):R\} \in RA\}$.

Observe that $\mathcal{I}'$ differs from $\mathcal{I}$ only in an extended interpretation of role symbols. By definition, $\mathcal{I}' \models RA$. It remains to prove that $\mathcal{I}' \models CA$. By Proposition 5.14, we are done if we prove that $\mathcal{I}' \models \bigcup N_i^*$. Since we only expanded the interpretation of relations, $\mathcal{I}$ and $\mathcal{I}'$ can only disagree on universal concepts of the form $a : \forall R.C$. By induction on the quantifier depth we prove this to be false.

Assume that $\mathcal{I}$ and $\mathcal{I}'$ agree on all formulas of quantifier depth less than n, and let $a:\forall R.C$ be of quantifier depth n, for $\{a:\forall R.C\} \in S_a^*$. Suppose $\mathcal{I}' \not\models \forall R.C$. This holds iff there exists b such that $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) \in R^{\mathcal{I}'}$ and $\mathcal{I}' \not\models b:C$. By the inductive hypothesis, $\mathcal{I} \not\models b:C$. Now, if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ we are done. Otherwise, by definition $\{(a,b):R\} \in RA$. But then $\{b : C\} \in S_b^*$ by construction and as $\mathcal{I} \models S_b^*$, we also have $\mathcal{I} \models b : C$ — a contradiction. QED

As we said above, each N_i^* represents the "named core" of a model of S. The final step is to define the non-named part of the model. The following operations are performed to each set in each of the N_i^* obtaining in such a way a forest F_i.

Fix N_i^*, and a. We construct a tree "hanging" from the corresponding $S_a^* \in N_i^*$. The condition that each node of the tree is named by either an individual or a new label (that is, all the formulas in a node have the same prefix) will be preserved as an invariant during the construction. Set the original tree u to S_a^* and repeat the following operations C1, C2 and C3 in succession until the end-condition holds.

Operation C1. Equal to *Operation A1.*

Operation C2. Equal to *Operation A2.*

Operation C3. For each leaf w of u,

- if for some concept we have $\{C\}, \{\neg C\} \in w$, do nothing;
- otherwise, since w is a set of unit clauses, we can write $w = \{\{t{:}C_1\}, \dots, \{t{:}C_m\}, \{t : \forall R_{k_1}.A_1\}, \dots, \{t : \forall R_{k_n}.A_n\}, \{t : \neg\forall R_{l_1}.P_1\}, \dots, \{t : \neg\forall R_{l_q}.P_q\}\}$. Form the sets $w_i = \{\{\mathit{wnnf}(t' : \neg P_i)\}\} \cup S_i$, where t' is a new label, and $S_i = \{\{t' : A_h\} \mid \{t : \forall R_i.A_h\} \in w\}$, and append each of them to w as children marking the edges as R_i links. The nodes w_i are called the *projections* of w.

End-condition. Operation C3 is inapplicable.

Proposition 5.17. *Operation C cannot be applied indefinitely.*

Definition 5.18. We call nodes to which Operation C1 or C2 has been applied of type 1, and those to which Operation C3 has been applied of type 2. The set of *closed nodes* is recursively defined as follows,

- if for some concept $\{t{:}C\}, \{t{:}\neg C\}$ are in w then w is closed,
- if w is of type 1 and all its children are closed, w is closed,
- if w is of type 2 and some of its children is closed, w is closed.

Let F_i be a forest that is obtained by applying Operations C1, C2, and C3 to N_i^* as often as possible. Then F_i is *closed* if any of its roots is closed.

Lemma 5.19. *If one of the forests F_i is not closed then S_Σ has a model.*

Proof. Let F_i be a non-closed forest. By a simple generalization of the results in [Enjalbert and Fariñas del Cerro, 1989, Lemma 2.7] we can obtain a model $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ of all roots S_a^* in F_i, from the trees "hanging" from them, ie., a model of $\bigcup N_i^*$. By Proposition 5.16, S_Σ has a model. QED

Lemma 5.19 establishes the property (†) we wanted in our structure T_S. To establish (††) we need a further auxiliary result.

Proposition 5.20. *Let w be a node of type 2. If one of its projections w_i is refutable, then so is w.*

Proof. Let w be a set of unit clauses $w = \{\{t\!:\!C_1\}, \ldots, \{t\!:\!C_m\}, \{t\!:\!\forall R_{k_1}.A_1\}, \ldots, \{t\!:\!\forall R_{k_n}.A_n\}, \{t\!:\!\neg\forall R_{l_1}.P_1\}, \ldots, \{t\!:\!\neg\forall R_{l_q}.P_q\}\}$. And let w_i be its refutable projection: $w_i = \{\{wnnf(t'\!:\!\neg P_i)\}\} \cup S_i$, where t' is a new label, and $S_i = \{\{t'\!:\!A_h\} \mid \{t\!:\!\forall R_i.A_h\} \in w\}$. We use resolution on w to arrive at the clauses in w_i from which the refutation can be carried out: Apply $(\neg\forall)$ to $\{t\!:\!\neg\forall R_i.P_i\}$ in w to obtain $\{t'\!:\!wnnf(t'\!:\!\neg P_i)\}$ and $\{(t,t')\!:\!R_i\}$. Now apply $(\forall)$ to all the clauses $\{t\!:\!\forall R_i.A_h\}$ in w to obtain $\{t'\!:\!A_h\}$. QED

Lemma 5.21. *In a forest F_i, every closed node is refutable.*

Proof. For w a node in F_i, let $d(w)$ be the longest distance from w to a leaf.

If $d(w) = 0$, then w is a leaf, thus for some concept C, $\{t\!:\!C\}$ and $\{t\!:\!\neg C\}$ are in w. Using (RES) we immediately derive $\{\}$.

For the induction step, suppose the proposition holds for all w' such that $d(w') < n$ and that $d(w) = n$. If w is of type 1, let $w_1 = w\setminus\{Cl\}\cup\{Cl_1\}$ and $w_2 = w\setminus\{Cl\}\cup\{Cl_2\}$ be its children. By the inductive hypothesis there is a refutation for w_1 and w_2. By propositional resolution there is a refutation of w: repeat the refutation proof for w_2 but starting with w, instead of the empty clause we should obtain a derivation of Cl_2; now use the refutation of w_2. Suppose w is of type 2. Because w is closed, one of its projections is closed. Hence, by the inductive hypothesis it has a refutation. By Proposition 5.20, w itself has a refutation. QED

Theorem 5.22. [Completeness] *The resolution method described above is complete: if Σ is a knowledge base, then S_Σ is refutable whenever Σ is unsatisfiable.*

Proof. We only need to put together the previous pieces. If Σ does not have a model then neither does S_Σ. By Lemma 5.19 all the forests F_i obtained from S_Σ are closed, and by Lemma 5.21, for each N_i^*, one of the sets $S_{a_j}^*$ is refutable. By Proposition 5.16, for all i, $\bigcup N_i \cup RA$ is refutable. By Proposition 5.15, S_Σ is refutable. QED

Because we have shown how to *effectively* obtain a refutation from an inconsistent set of clauses we have also established termination. Notice that during the completeness proof we have used a *specific strategy* in the application of the resolution rules (crucially, the $(\neg\forall)$ rule is never applied twice to the same formula). By means of this strategy, we can guarantee termination of labeled modal resolution when verifying the consistency of any knowledge base in $\mathcal{ALCR}$.

Theorem 5.23. [Termination] *Labeled resolution can effectively decide the consistency of simple, acyclic knowledge bases in $\mathcal{ALCR}$.*

We have spelled out in detail the method for the basic description logic $\mathcal{ALCR}$, the next natural step is to consider extensions. For instance, in [Calvanese *et al.*, 1997] some attention has been given to n-ary roles (in modal logic terms, n-ary modal operators). Our approach generalizes to this case without further problems.

Considering additional structure on roles is another possibility. We have limited ourselves to conjunction, but disjunction, negation, composition, etc. can be considered. And, of course, the addition of counting operators should be hight on our to-do list. A very attractive idea which matches nicely with the resolution approach is to incorporate a limited kind of unification on "universal labels" of the form $x:C$, to account for on the fly unfolding of definitions and more general T-Boxes. The use of such universal labels would make it unnecessary to perform a complete unfolding of the knowledge base as a pre-processing step. The leitmotiv would be "to do expansion by definitions only when needed in deduction." On the fly unfolding has already been implemented in tableaux based systems like KRIS [Baader *et al.*, 1994]. See also our discussion in Section 4.5.1.

5.2.3 Hybrid Logics

It's the turn of hybrid languages now. But of course, we have already been dealing with hybrid languages throughout the previous section: just remember the tight connections between description and hybrid logics that we built in Chapter 4.

But what about binders? Extending the system to account for hybrid sentences using ↓ is fairly straightforward. Consider the rules

$$(\downarrow)\ \ \frac{Cl_1 \cup \{t:\downarrow x.\varphi\}}{Cl_1 \cup \{t:\varphi[x/t]\}} \qquad\qquad (\neg\downarrow)\ \ \frac{Cl_1 \cup \{t:\neg\downarrow x.\varphi\}}{Cl_1 \cup \{t:\downarrow x.\mathit{wnnf}(\neg\varphi)\}}.$$

Notice that the rules transform hybrid sentences into hybrid sentences. If, in addition, we add the following rules to handle nominals

$$(\mathrm{NOM})\ \ \frac{Cl_1 \cup \{t:i\} \qquad Cl_2 \cup \{i:\varphi\}}{Cl_1 \cup Cl_2 \cup \{t:\varphi\}} \qquad\qquad (\mathrm{SYM})\ \ \frac{Cl \cup \{t:i\}}{Cl \cup \{i:t\}}$$

we obtain a complete calculus for sentences in $\mathcal{H}_{\mathsf{S}}(\downarrow)$. Of course, in this case we cannot expect a heuristic ensuring termination as the satisfiability problem for full $\mathcal{H}_{\mathsf{S}}(\downarrow)$ is undecidable. As we discuss in Section 4.5.4, we need strong restrictions in the language to achieve decidability like considering only sentences with non-nested occurrences of ↓ (see Theorem 7.10).

Let's work out a short example. We prove that $\downarrow x.\Diamond(x \wedge p) \to p$ is a tautology. Consider the negation of the formula in clausal form

$$\begin{array}{lll}
1. & \{i:\underline{\downarrow x}.\neg\Box\neg(x \wedge p)\},\ \{i:\neg p\}, & \text{by } (\downarrow) \\
2. & \{i:\underline{\neg\Box}\neg(i \wedge p)\},\ \{i:\neg p\}, & \text{by } (\neg\Box) \\
3. & \{R(i,j)\},\ \{j:(i\underline{\wedge}p)\},\ \{i:\neg p\}, & \text{by } (\wedge) \\
4. & \{j:\underline{i}\},\ \{j:p\},\ \{i:\neg p\}, & \text{by (SYM)} \\
5. & \{i:\underline{j}\},\ \{j:\underline{p}\},\ \{i:\neg p\}, & \text{by (NOM)} \\
6. & \{i:\underline{p}\},\ \{i:\underline{\neg p}\}, & \text{by (RES)} \\
7. & \{\}. &
\end{array}$$

5.3 Reflections

In Section 2.4 we showed how constraint systems for instance checking could decide the different reasoning task we introduced in the previous sections. In Section 3.2 we argued how hybrid languages were able to internalize labeled deduction. The same ideas play a fundamental role in the labeled resolution systems we introduced in this chapter. Once again, individuals/nominals/labels together with the satisfiability operator : or @

are the key to achieve smooth and well behaved reasoning methods. And the systems we introduced in this chapter should have made clear that labeled resolution has many advantages with respect to previous direct resolution proposals, supporting our claim that description/hybrid logic ideas can indeed be used to improve reasoning methods. We complete the chapter with a discussion on a number of independent directions for future research.

Once labels are introduced the resolution method is very close to the tableaux approach, but we are still doing resolution. As we said, the rules ($\sqcap$), ($\neg\sqcap$) and ($\neg\forall$), prepare formulas to be fed into the resolution rules (RES) and ($\forall$). And the aim is still to derive the empty clause instead of finding a model by exhausting a branch. But, is this method any better than tableaux? We don't think this is the correct question to ask. We believe that we learn different things from studying different methods. For example, Horrocks and Patel-Schneider [1999] study a number of interesting optimizations of the tableaux implementation which were tested on the tableaux based theorem prover DLP. Some of their ideas can immediately be (or have already been) incorporated in our resolution method (lexical normalization and early detection of clashes, for instance), and others might perhaps be used in implementations of our method. On the other hand, optimizations for direct resolution such as those discussed in [Auffray *et al.*, 1990] can also be exploited in conjunction with the others. For example, in implementations of the resolution algorithm, strategies for selecting the resolving pairs are critical. This kind of heuristics has been investigated by Auffray *et al.* and some of their results easily extend to our framework. In certain cases, establishing completeness of these heuristics is even simpler because of our explicit use of resolution via labels.

The issue of heuristics is very much connected with complexity. The basic heuristic we used in the proof of Theorem 5.22 keeps the complete clause set "in memory" all the time and hence requires non-polynomial space. A similar situation occurs in clausal propositional resolution where the translation into clausal form can introduce an exponential blow up. We conjecture that a PSPACE heuristic for labeled resolution can be obtained by exploiting further the presence of labels (and given that we don't force a translation into full clausal form). Notice that labels and role assertions let us keep track of the accessibility relation and we can define the notion of "being a member of a branch." Now we can attempt to use the tree property of modal languages to guide resolution. We used similar ideas in [Areces *et al.*, 2000d] to improve the performance of translation based resolution provers.

The ideas behind labeled resolution are simple enough so that adapting available provers should not prove to be a very difficult task. It would be interesting to perform empirical testing on the performance of this resolution prover following the lines drawn in, for example [Horrocks *et al.*, 2000a], both in comparison with translation based resolution provers and those based on tableaux.

Finally, our completeness proof is constructive: when a refutation cannot be found we can actually define a model for the formula or knowledge base. Hence, our method can also be used for model extraction. How does this method perform in comparison with traditional model extraction from tableaux systems?

Chapter 6

Investigating Expressive Power

When once you have taken
the Impossible into your calculations
its possibilities become practically limitless.

from "The Peace of Mowsle Barton," Saki

In this chapter we investigate the expressive power of hybrid languages, both through semantic and syntactic characterizations, and by means of the meta-logical properties of interpolation and Beth definability. The work in the chapter is centered around the language $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$, which is analyzed in detail. Even though the satisfiability problem for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is undecidable, this language presents a particularly good meta-theoretical behavior. We will also comment on how and when the results we prove for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ can be adapted to sublanguages and extensions. From a description logic perspective, these results can be seen as marking the farthest boundaries of expressivity.

We begin by providing both model-theoretical characterizations (via a restricted notion of Ehrenfeucht game and an enriched notion of bisimulation) and a syntactic characterization (in terms of bounded formulas) for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. The key result to emerge is that $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ corresponds precisely to the first-order fragment which is invariant for generated submodels. This is in line with the investigations in [van Benthem, 1983], where characterizations of some natural modal model-theoretical operations were provided. If we add $\langle R^{-1} \rangle$ to the language and modify the notion of generated submodels accordingly, then the @ operator can be dropped. For hybrid languages without binders, we provide characterizations in terms of bisimulations.

We then move to interpolation and Beth definability. After introducing the different ways in which these properties can be defined and explaining their inter-relations, we establish that $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ has (strong) interpolation and, as a corollary, also the Beth definability property (both global and local). The proof of strong interpolation for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ can be generalized to *any pure extension of the logic.* This behavior contrasts sharply with what is known for basic modal languages, where general interpolation results are scarce. Interpolation (both strong and weak) does not go through, though, when we restrict $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ to its finite variable fragments. We also prove that only weak interpolation obtains for $\mathcal{H}_{\mathsf{S}}(@)$.

For languages where nominals appear free in formulas and which do not provide a binding mechanism, failure of arrow interpolation seems to be the norm. In particular, we provide counter-examples to interpolation for the basic modal language extended with nominals, $\mathcal{H}_{\mathsf{N}}(@)$ and $\mathcal{H}_{\mathsf{S}}(@)$. The extensions of these languages with the $\langle R^{-1} \rangle$ or E operators fare no better. But in Section 4.5 we have proved that interpolation reappears if we restrict the use of nominals by moving to the languages $\mathcal{H}(@, @\Diamond)$ and $\mathcal{H}(\langle R^{-1} \rangle, @, @\Diamond)$ (see Theorem 4.15).

6.1 Characterizations

We begin by providing a *syntactic* characterization. In particular, we will specialize the standard translation ST introduced in Definition 3.3 to the $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ language. It will be clear that the range of the translation lies in a certain bounded fragment of FO, and we will define a reverse translation HT which maps the bounded fragment back into the hybrid language. Thus, we are free to think either in terms of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ or the corresponding bounded fragment.

We will then turn to *semantic* characterizations. Clearly, $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ is a genuine hybrid of modal and first-order ideas (after all, it combines Kripke semantics with the idea of binding variables to worlds), thus there are two ways to proceed. The first is essentially first-order: we will look for a weak notion of Ehrenfeucht games (see Definition 1.7). The second is essentially modal, using hybrid bisimulation (Section 3.5). As we will see, both paths yield natural notions of equivalence between models, and by relating them (and drawing on our syntactic characterization) we can provide a detailed picture of what $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ offers.

6.1.1 Translations

In the sections that follow we will mainly discuss uni-modal languages but the results are easy to extend to the multi-modal case.

Recall the standard translation we introduced in Definition 3.3, but now specialized to the $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ language.

Definition 6.1. [Standard translation for $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$] The functions ST_x and ST_y are defined by mutual recursion as follows

$$\begin{array}{rcl|rcl}
ST_x(i_j) & = & (x = i_j),\ i_j \in \mathsf{NOM} & ST_y(i_j) & = & (y = i_j),\ i_j \in \mathsf{NOM} \\
ST_x(x_j) & = & (x = x_j),\ x_j \in \mathsf{SVAR} & ST_y(x_j) & = & (y = x_j),\ x_j \in \mathsf{SVAR} \\
ST_x(p_j) & = & P_j(x),\ p_j \in \mathsf{PROP} & ST_y(p_j) & = & P_j(y),\ p_j \in \mathsf{PROP} \\
ST_x(\neg\varphi) & = & \neg ST_x(\varphi) & ST_y(\neg\varphi) & = & \neg ST_y(\varphi) \\
ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi) & ST_y(\varphi \wedge \psi) & = & ST_y(\varphi) \wedge ST_y(\psi) \\
ST_x(\Diamond\varphi) & = & \exists y.(R(x,y) \wedge ST_y(\varphi)) & ST_y(\Diamond\varphi) & = & \exists x.(R(y,x) \wedge ST_x(\varphi)) \\
ST_x(@_s\varphi) & = & (ST_x(\varphi))[x/s] & ST_y(@_s\varphi) & = & (ST_y(\varphi))[y/s] \\
ST_x(\downarrow x_j.\varphi) & = & (ST_x(\varphi))[x_j/x] & ST_y(\downarrow x_j.\varphi) & = & (ST_y(\varphi))[x_j/y].
\end{array}$$

Now for the interesting question: what is the *range* of ST? In fact, it belongs to a bounded fragment of first-order logic.

Given a first-order signature $\langle\{R\} \cup \mathsf{UREL}, \mathsf{CONS}, \mathsf{VAR}\rangle$ we define the *bounded fragment* BF as the set of formulas generated by the following grammar:

$$\mathsf{FORMS} = R(t,t') \mid P_j(t) \mid t = t' \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x_i.(R(t,x_i) \wedge \varphi) \text{ (for } x_i \neq t),$$

where $x_i \in \mathsf{VAR}$, $t,t' \in \mathsf{VAR} \cup \mathsf{CONS}$, $P_j \in \mathsf{UREL}$, φ, φ_1, $\varphi_2 \in \mathsf{FORMS}$. Notice the important side-condition on the generation of existentially quantified formulas. It prevents sentences like $\exists x.(R(x,x) \wedge x = x)$ from falling into the fragment. As we will see, there is a strong connection between formulas in the bounded fragment and invariance under generated submodels, and formulas like $\exists x.(R(x,x) \wedge x = x)$ are not even preserved under this operation.

The bounded fragment arose in set theory, where the bounding relation R would be interpreted as membership $\in$. Most properties of sets can be formalized in BF. A characterization for BF was provided in [Feferman, 1968] in terms of outer extensions of first-order models using proof-theoretical means (as an instance of a more general result involving infinitary languages). In [1983], van Benthem provides a model-theoretical proof in terms of generated subframes.

Clearly ST generates formulas in the bounded fragment. In addition, we can also translate the bounded fragment into $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$. The translation HT from the bounded fragment over $\langle\{R\} \cup \mathsf{UREL}, \mathsf{CONS}, \mathsf{VAR}\rangle$ into the hybrid language over $\langle\mathsf{UREL}, \mathsf{CONS}, \mathsf{VAR}\rangle$ is defined as follows. For $t, t' \in \mathsf{VAR} \cup \mathsf{CONS}$

$$\begin{array}{rcl} HT(R(t,t')) & = & @_t\Diamond t' \\ HT(P_j(t)) & = & @_t p_j \\ HT(t = t') & = & @_t t' \\ HT(\neg\varphi) & = & \neg HT(\varphi) \\ HT(\varphi \wedge \psi) & = & HT(\varphi) \wedge HT(\psi). \\ HT(\exists x.(R(t,x) \wedge \varphi)) & = & @_t\Diamond\downarrow x.HT(\varphi). \end{array}$$

By construction, $HT(\varphi)$ is a hybrid formula built as a Boolean combination of @-formulas (formulas whose main operator is @). We can now prove the following strong truth preservation result.

Proposition 6.2. [HT preserves truth] *Let $\varphi \in \mathsf{BF}$. Then for every first-order model $\mathcal{M}$ and for every assignment g, $\mathcal{M} \models \varphi[g]$ iff $\mathcal{M}, g \Vdash HT(\varphi)$.*

Proof. We use the following fact: let φ be a Boolean combination of @-formulas, then there exists an m such that $\mathcal{M}, g, m \Vdash \varphi$ iff $\mathcal{M}, g \Vdash \varphi$.

$HT(\exists x.(R(t,x) \wedge \varphi))$ is the interesting case. We have $\mathcal{M} \models \exists x.(R(t,x) \wedge \varphi)[g]$ iff $\mathcal{M} \models (R(t,x) \wedge \varphi)[g^x_m]$ for $m \in M$. Let $\mathbf{t}$ be the denotation of t in $\mathcal{M}$ under g^x_m. Given the restriction on variables in bounded quantification, $t \neq x$, whence $\mathbf{t}$ is also the interpretation of t in $\mathcal{M}$ under g. So $R^{\mathcal{M}}(\mathbf{t}, m)$ and $\mathcal{M} \models \varphi[g^x_m]$. By induction hypothesis, $\mathcal{M}, g^x_m \Vdash HT(\varphi)$ iff $\mathcal{M}, g, m \Vdash \downarrow x.HT(\varphi)$, iff $\mathcal{M}, g, \mathbf{t} \Vdash \Diamond\downarrow x.HT(\varphi)$ iff $\mathcal{M}, g, \mathbf{t} \Vdash @_t\Diamond\downarrow x.HT(\varphi)$ iff $\mathcal{M}, g \Vdash @_t\Diamond\downarrow x.HT(\varphi)$. QED

As simple corollaries we have:

Corollary 6.3. *Let $\varphi(x)$ be a bounded formula with only x free, then for all models $\mathcal{M}$ and for all $m \in M$, $\mathcal{M} \models \varphi[m]$ iff $\mathcal{M}, m \Vdash \downarrow x.HT(\varphi)$.*

Corollary 6.4. *Let φ be a first-order formula in the hybrid signature. Then the following are equivalent*

i. φ is equivalent to the standard translation of a hybrid formula.

ii. φ is equivalent to a formula in the bounded fragment.

Moreover, there are effective translations between $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ and BF.

6.1.2 Generated Back-and-Forth Systems

We now turn to the problem of providing semantic characterizations of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$. In this section we will adopt an essentially first-order approach: we define generated back-and-forth systems, basically a restricted form of Ehrenfeucht games, and link it to the concept of generated submodels.

Definition 6.5. [Generated back-and-forth systems] Let $\mathcal{M}$ and $\mathcal{N}$ be two first-order models in the hybrid signature. A *generated back-and-forth system* between $\mathcal{M}$ and $\mathcal{N}$ is a non-empty family F of finite partial isomorphisms between $\mathcal{M}$ and $\mathcal{N}$ satisfying the following two extension rules:

($\Diamond$-extension)

(forth) if $h \in F$, $m \in dom(F)$ & $R^{\mathcal{M}}(m,m')$, then $h \cup \{(m',n')\} \in F$ for some $n' \in N$,
(back) if $h \in F$, $n \in ran(F)$ & $R^{\mathcal{N}}(n,n')$, then $h \cup \{(m',n')\} \in F$ for some $m' \in M$.

(nominal extension)

(forth) if $h \in F$ and there exists $m \in M$ such that $V^{\mathcal{M}}(i) = \{m\}$ for some nominal i, then there exists $n \in N$ such that $h \cup \{(m,n)\} \in F$,
(back) a similar condition backwards.

Let $\bar{m} \in {}^kM$, $\bar{n} \in {}^kN$, then $(\mathcal{M},\bar{m}) \cong_G (\mathcal{N},\bar{n})$ means that there is a generated back-and-forth system linking $\mathcal{M}$ and $\mathcal{N}$ with a partial isomorphism sending $m(i)$ to $n(i)$.

Note how closely this definition follows Definition 1.8. In fact, if we think of a generated back-and-forth system as describing an Ehrenfeucht game, then the only difference is that in the "generated back-and-forth game" the universal player must choose his moves from R-successors or worlds named by a nominal, whereas any choice is allowed in the full first-order game. Unsurprisingly, restricting the play to *accessible* worlds closely connects generated back-and-forth systems and generated submodels.

Definition 6.6. [Generated submodel] Let $\mathcal{M} = \langle M, R, V\rangle$ be a hybrid model and $S \subseteq M$. Let NAMED denote the subset of M whose elements are the denotation of some nominal. The *submodel of $\mathcal{M}$ generated by S* (or the *S-generated submodel of $\mathcal{M}$*) is the substructure of $\mathcal{M}$ with domain $\{m \in M \mid \text{there is } s \in S \cup \mathsf{NAMED} \text{ such that } R^*(s,m)\}$, for R^* the reflexive and transitive closure of R.

Note that if $\mathsf{NAMED} = \{\}$ we obtain the familiar modal notion of a generated submodel, and that if in addition S is a singleton set, we have the usual modal notion of a point-generated (or rooted) submodel.

We now define two notions of invariance. A first-order formula $\varphi(\bar{x})$ in free variables $\bar{x}$ in a signature with one binary relation R, unary predicates and constants (and equality) is *invariant for generated submodels* if for all pairs $(\mathcal{M},\bar{m})$ and $(\mathcal{M}',\bar{m})$ such that $\mathcal{M}'$ is the $\bar{m}$-generated submodel of $\mathcal{M}$,

$$\mathcal{M} \models \varphi[\bar{m}] \text{ if and only if } \mathcal{M}' \models \varphi[\bar{m}].$$

Similarly, we say that $\varphi(\bar{x})$ is *invariant for generated back-and-forth systems* if for all pairs $(\mathcal{M},\bar{m})$ and $(\mathcal{N},\bar{n})$, $(\mathcal{M},\bar{m}) \cong_G (\mathcal{N},\bar{n})$ implies

$$\mathcal{M} \models \varphi[\bar{m}] \text{ if and only if } \mathcal{N} \models \varphi[\bar{n}].$$

THEOREM 6.7. *Let $\varphi(\bar{x})$ be a first-order formula in the hybrid signature. Then the following are equivalent*

i. *$\varphi(\bar{x})$ is equivalent to a formula in the bounded fragment.*

ii. *$\varphi(\bar{x})$ is invariant for generated submodels.*

iii. *$\varphi(\bar{x})$ is invariant for generated back-and-forth systems.*

PROOF.
i) $\Rightarrow$ *ii)* is obvious.

ii) $\Rightarrow$ *iii)* First note that $\varphi(\bar{x})$ is invariant for generated submodels iff $\neg\varphi(\bar{x})$ is. Now, suppose $\varphi(\bar{x})$ is invariant for generated submodels but *not* preserved under generated back-and-forth systems. Then we have pairs $(\mathcal{M}, \bar{m})$ and $(\mathcal{N}, \bar{n})$ such that $(\mathcal{M}, \bar{m}) \cong_G (\mathcal{N}, \bar{n})$, and $\mathcal{M} \models \varphi[\bar{m}]$ while $\mathcal{N} \models \neg\varphi[\bar{n}]$.

Let $\mathcal{M}'$ ($\mathcal{N}'$) be the $\bar{m}$- ($\bar{n}$-) generated submodel of $\mathcal{M}$ ($\mathcal{N}$). Then still $\mathcal{M}' \models \varphi[\bar{m}]$ and $\mathcal{N}' \models \neg\varphi[\bar{n}]$ by invariance, and clearly $(\mathcal{M}', \bar{m}) \cong_G (\mathcal{N}', \bar{n})$. But then $(\mathcal{M}', \bar{m})$ and $(\mathcal{N}', \bar{n})$ have the same first-order theory by the following argument. Because $(\mathcal{M}', \bar{m}) \cong_G (\mathcal{N}', \bar{n})$, Duplicator has a winning strategy in all games where Spoiler only plays *immediate* R-successors or points named by a nominal. But since the models are generated, in the first-order back-and-forth game Spoiler can only play worlds which are accessible by a finite R-transition from either the root or one of the named worlds. But then Duplicator can compute a winning answer for the classic Ehrenfeucht game from his winning generated back-and-forth strategy. This contradicts the claim that $\mathcal{M}' \models \varphi[\bar{m}]$ and $\mathcal{N}' \models \neg\varphi[\bar{n}]$.

iii) $\Rightarrow$ *i)* This implication follows from a diagram-chasing argument [van Benthem, 1996]. Let $\varphi(\bar{x})$ be as in the hypothesis, and $BC(\varphi(\bar{x}))$ be the bounded consequences of $\varphi(\bar{x})$ (that is, the consequences of $\varphi(\bar{x})$ that belong to BF). We have to show that $BC(\varphi(\bar{x})) \models \varphi(\bar{x})$, from which the result follows by compactness. (Here we interpret the $\bar{x}$ as constants, or equivalently we use the local version of first-order consequence.)

If $BC(\varphi(\bar{x}))$ is inconsistent we are done. Otherwise, let $\mathcal{M}, \bar{m}$ satisfy $BC(\varphi(\bar{x}))$ and $\mathcal{N}, \bar{n}$ satisfy $\varphi(\bar{x})$ together with the bounded theory of $\mathcal{M}, \bar{m}$. (Such a pair of model and assignment can easily be shown to exist.) Take ω-saturated extensions $(\mathcal{M}^+, \bar{m})$ and $(\mathcal{N}^+, \bar{n})$. Create a family F of finite functions between M^+ and N^+ as follows: $f : \bar{x} \mapsto \bar{y}$ is in F iff $(\mathcal{M}^+, \bar{x})$ and $(\mathcal{N}^+, \bar{y})$ make the same bounded formulas true. F is a generated back-and-forth system linking $\bar{m}$ and $\bar{n}$. Now we can start diagram chasing: $\mathcal{N} \models \varphi[\bar{n}]$ then (by elementary extension) $\mathcal{N}^+ \models \varphi[\bar{n}]$, then (by invariance) $\mathcal{M}^+ \models \varphi[\bar{m}]$, then (passing to an elementary submodel) $\mathcal{M} \models \varphi[\bar{m}]$ as desired. QED

6.1.3 Hybrid Bisimulations

We have just seen that by weakening the notion of an Ehrenfeucht game we can link the bounded fragment (and hence $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$) with generated submodels. But in spite of its binding apparatus, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ has a distinctly modal flavor. By using hybrid bisimulation we will now characterize $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ in intrinsically modal terms. The approach has an advantage over the use of generated back-and-forth systems as results can be easily obtained for reducts as well.

Remember that we introduced the truly modal notion of k-seq-bisimulations in Definition 3.13. Well, something very much first-order is hidden behind: partial isomorphisms.

Proposition 6.8. *Let $k \geq 2$, and let $\mathcal{M} \overset{k}{\sim} \mathcal{N}$. If $(\bar{m}, m) \overset{k}{\sim} (\bar{n}, n)$, then the function f defined as $f(m) = n$ and $f(m(i)) = n(i)$ is a partial isomorphism between $\{m(1), \ldots, m(k), m\}$ and $\{n(1), \ldots, n(k), n\}$.*

Proof. The map f is a bijection by **(var)** and **(@)**. By **(prop)** and **(@)**, f preserves nominals and propositional variables. To see that it preserves the accessibility relation suppose $R^{\mathcal{M}}(x, y)$. There are three cases: *i*) Suppose $x = m$ and $y = m_i$. Then by **(forth)** there is n' such that $R^{\mathcal{N}}(n, n')$ and $(\bar{m}, m_i) \overset{k}{\sim} (\bar{n}, n')$. But $\bar{m}(i) = m_i$, so by **(var)**, $n' = \bar{n}(i)$, whence $R^{\mathcal{N}}(n, f(m(i)))$. *ii*) Suppose $x = m_i$ and $y = m$. Let $j \neq i$. Such a j exists because we assumed that $k \geq 2$. By **(↓)**, $(\bar{m}^j_m, m) \overset{k}{\sim} (\bar{n}^j_n, n)$. Then by **(@)**, $(\bar{m}^j_m, m_i) \overset{k}{\sim} (\bar{n}^j_n, n_i)$. Now continue as *i*). *iii*) Finally, suppose $x = m_i$ and $y = m_j$. By **(@)**, $(\bar{m}, m_i) \overset{k}{\sim} (\bar{n}, n_i)$. Continue as in *i*). Thus $R^{\mathcal{M}}(x, y)$ implies $R^{\mathcal{N}}(f(x), f(y))$. For the other direction use **(back)** in the same way. QED

Note that the condition $k \geq 2$ is crucial. We use it together with **(↓)** to store the information about m. Proposition 6.8 shows that there is a clear link with our earlier work on generated back-and-forth systems, and the next theorem shouldn't come as a surprise:

Theorem 6.9. *$(\mathcal{M}, \bar{m}) \overset{\omega}{\sim} (\mathcal{N}, \bar{n})$ if and only if $(\mathcal{M}, \bar{m}) \cong_G (\mathcal{N}, \bar{n})$.*

Proof.
[⇒]. Let $(\mathcal{M}, \bar{m}) \overset{\omega}{\sim} (\mathcal{N}, \bar{n})$. Define a family F of maps as follows: $f \in F$ if there exists $(\bar{x}, x') \overset{k}{\sim} (\bar{y}, y')$ and f is defined as in Proposition 6.8.

Clearly $\bar{m}$ and $\bar{n}$ are connected by a map. By Proposition 6.8 all maps are partial isomorphisms. We show the (forth) side of **(nominal extension)**; all other conditions have a similar proof. Suppose $f \in F$ and let i be a nominal. Then for some $\bar{x}$, x, $\bar{y}$, y, $(\bar{x}, x') \overset{k}{\sim} (\bar{y}, y')$ by definition of F. Then $(\bar{x} * x', x') \overset{k+1}{\sim} (\bar{y} * y', y')$ by **(sto)**. But then by **(@)**, $(\bar{x} * x', i^{\mathcal{M}}) \overset{k+1}{\sim} (\bar{y} * y', i^{\mathcal{N}})$. Thus, the required extension is in F.

[⇐]. Let $(\mathcal{M}, \bar{m}) \cong_G (\mathcal{N}, \bar{n})$. We define the following family of relations: for any $f \in F$, for any k, for any tuple $\bar{m}$ in the k-th power of the domain of f and for any m in the domain of f, we set $(\bar{m}, m) \overset{k}{\sim} (f(\bar{m}), f(m))$. It is easy to check that this is an ω-seq-bisimulation. QED

It is possible to prove a direct characterization result for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ in terms of invariance for k-bisimulations, using a diagram-chasing argument. We are not going to do this here since in the next section we will take a detour via the bounded fragment to reach the same result. It is also possible to develop k-pebble versions of generated back-and-forth systems; this notion takes the exact number of variables used in formulas into account. It is not difficult to see that $k + 1$-pebble generated back-and-forth systems correspond to k-bisimulations.

6.1.4 Harvest

It is time to draw together the threads we have developed. First we note their consequences for the expressive power of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ over models. Then we note the consequences for frames and what this tells us about hybrid completeness.

Expressivity over Models. We have arrived to the following five-fold characterization of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$:

THEOREM 6.10. *Let $\varphi(\bar{x})$ be a first-order formula in the hybrid signature (with equality). Then the following are equivalent*

i. *$\varphi(\bar{x})$ is equivalent to the standard translation of a $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ formula.*
ii. *$\varphi(\bar{x})$ is invariant for generated submodels.*
iii. *$\varphi(\bar{x})$ is invariant for generated back-and-forth systems.*
iv. *$\varphi(\bar{x})$ is invariant for ω-seq-bisimulation.*
v. *$\varphi(\bar{x})$ is equivalent to a formula in the bounded fragment of first-order logic.*

PROOF. By Corollary 6.4, Theorem 6.7, Proposition 3.15 and Theorem 6.9. QED

But these have obvious consequences for the ordinary modal correspondence language. In particular, if we consider *nominal-free* hybrid sentences, then we obtain a five-fold characterization of the fragment of first-order logic in the classical modal signature which is invariant for generated submodels.

COROLLARY 6.11. *Let $\varphi(x)$ be a first-order formula in the modal signature (with equality). Then the following are equivalent*

i. *$\varphi(x)$ is equivalent to the standard translation of a nominal-free $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ sentence.*
ii. *$\varphi(x)$ is invariant for generated submodels (now in the standard modal sense).*
iii. *$\varphi(x)$ is invariant for R-generated back-and-forth systems (an R-generated back-and-forth system is a back-and-forth system satisfying only the* **$\Diamond$-extension** *rule).*
iv. *$\varphi(x)$ is invariant for ω-seq-bisimulation.*
v. *$\varphi(x)$ is equivalent to a formula in the bounded fragment of first-order logic without constants.*

A simple generalization of the reduction of the universal validity of a first-order formula α to bisimulation invariance of a formula α' given in [van Benthem, 1996, Remark 4.19], shows that the problem of verifying if a given formula is equivalent to a formula in the bounded fragment of first-order logic (even with no constants) is undecidable. And hence by the equivalences above, so are all the other problems (i.e., verifying whether a formula is invariant for generated submodels, invariant for generated back-and-forth systems, etc.).

But the problem might admit effective solutions in certain particular cases (like, for example, when the original formula is already in a restricted class), and it is possible that this is easier to establish in terms of one of the five equivalent but different versions.

Frames and Completeness. Since the late 1950s, one of the central topics in modal logic has been linking modal formulas to properties of frames and investigating when they give rise to complete axiomatizations for the frame classes they define. The work of the previous section easily yields a characterization of the frame-defining abilities of pure nominal-free sentences. Moreover, the axiomatic investigations of [Tzakova, 1999a] show that there is a perfect match between definability and completeness for pure nominal-free sentences. By combining these results we obtain matching definability and completeness results for a wide range of first-order definable frame classes.

In modal correspondence theory, the first-order language (with equality) over the signature consisting simply of a binary symbol R is called the (first-order) *frame language*. A *frame condition* is a formula in the frame language containing exactly one free variable. The class of frames defined by a frame condition $\varphi(x)$ is the class in which the universal closure $\forall x.\varphi(x)$ is true; we call this class $\textsc{frames}(\forall x.\varphi(x))$.

Before proceeding, two simple observations are in order. First, note that if we apply the standard translation ST to a pure nominal-free sentence α, then $ST(\alpha)$ is a frame condition with free variable x. Furthermore, note that for any frame $\mathcal{F} = \langle M, R\rangle$ we have that $\mathcal{F} \Vdash \alpha$ iff $\mathcal{F} \models \forall x.ST(\alpha)$; this is an immediate consequence of the definition of frame validity.

Theorem 6.12. *Let $\mathsf{K}[\mathcal{H}_\mathcal{S}(@,\downarrow)]$ be the axiomatization given in Definition 3.10, and for any hybrid sentence α let $\mathsf{K}[\mathcal{H}_\mathcal{S}(@,\downarrow)] + \alpha$ be the system obtained by adding α as an additional axiom. Then, if $\varphi(x)$ is a frame condition and $\varphi(x)$ is invariant under generated submodels (in the usual modal sense) we have that:*

i. If $\varphi(x)$ is in the bounded fragment then the pure nominal free sentence $\downarrow x.HT(\varphi(x))$ defines $\textsc{frames}(\forall x.\varphi(x))$, and $\mathsf{K}[\mathcal{H}_\mathcal{S}(@,\downarrow)] + \downarrow x.HT(\varphi(x))$ is strongly complete with respect to $\textsc{frames}(\forall x.\varphi(x))$.

ii. If $\varphi(x)$ is not in the bounded fragment, there is a pure nominal free sentence α such that α defines $\textsc{frames}(\forall x.\varphi(x))$, and $ST(\alpha)$ is equivalent to $\varphi(x)$. Moreover, $\mathsf{K}[\mathcal{H}_\mathcal{S}(@,\downarrow)] + \alpha$ is strongly complete with respect to $\textsc{frames}(\forall x.\varphi(x))$.

Conversely, if a sentence α is pure and nominal-free, then α defines the class of frames $\textsc{frames}(\forall x.ST(\alpha(x)))$, and $\mathsf{K}[\mathcal{H}_\mathcal{S}(@,\downarrow)] + \alpha$ is a complete axiomatic system.

Proof. The converse condition was proved in [Blackburn and Tzakova, 1998b], so let's examine the other direction.

For item i), we first remark that as $\varphi(x)$ belongs to the frame language, it contains no unary predicate symbols, hence $HT(\varphi(x))$ is a *pure* formula; that $\downarrow x.HT(\varphi(x))$ is a pure nominal-free sentence is thus clear. Now, by Corollary 6.3, for any model $\mathcal{M} = (\mathcal{F}, V)$ and any $m \in M$, $(\mathcal{F}, V) \models \varphi[m]$ iff $(\mathcal{F}, V), m \Vdash \downarrow x.HT(\varphi)$. But this means that $(\mathcal{F}, V) \models \forall x.\varphi$ iff $(\mathcal{F}, V) \Vdash \downarrow x.HT(\varphi)$. As $\varphi(x)$ contains no unary predicate symbols (and $\downarrow x.HT(\varphi)$ no propositional variables) V is irrelevant, and hence $\mathcal{F} \models \forall x.\varphi(x)$ iff $\mathcal{F} \Vdash \downarrow x.HT(\varphi)$. This means that $\downarrow x.HT(\varphi(x))$ defines $\textsc{frames}(\forall x.\varphi(x))$. Completeness follows using the arguments of [Blackburn and Tzakova, 1998b].

For item ii) we know that $\varphi(x)$, being invariant under generated submodels, is equivalent to a formula in the bounded fragment — but is it equivalent to a *frame condition* $\varphi'(x)$? In fact, this can be established by diagram-chasing argument as in the proof of Theorem 6.7. The key point to observe is that instead of showing that $BC(\varphi(x)) \models \varphi(x)$, we can show by the same method that $FC(\varphi(x)) \models \varphi(x)$, where FC are all the frame conditions implied by $\varphi(x)$. Thus there is an equivalent frame condition $\varphi'(x)$, and we can take α to be $\downarrow x.HT(\varphi'(x))$. The remainder of the proof is as for item i). QED

6.1.5 Variations

The $\mathcal{H}_\mathsf{S}(\langle R^{-1}\rangle, @, \downarrow)$ Language. The characterization results given in Section 6.1.4, become particularly natural when we introduce the $\langle R^{-1}\rangle$ operator. To cope with the backward looking operators, we need a slightly more liberal notion of generated submodel: a point t belongs to the submodel *temporally generated* by a subset S if t is reachable from some point $s \in S$ by making a finite sequence of moves through the accessibility relation, where both forward and backward steps are allowed. The characterization results we have proved hold for $\mathcal{H}_\mathsf{S}(\langle R^{-1}\rangle, @, \downarrow)$ under this notion of generated submodel.

But let's press matters a little further. Note that in *nominal-free sentences* of $\mathcal{H}_\mathsf{S}(\langle R^{-1}\rangle, @, \downarrow)$, all occurrences of @ can be eliminated. As a simple example, consider the definition of the Until operator:

$$\mathsf{Until}(\varphi,\psi) \ := \ \downarrow x.\langle R\rangle\downarrow y.@_x(\langle R\rangle(y \wedge \varphi) \wedge [R](\langle R\rangle y \to \psi)).$$

Observe that the following @-free sentence has the same effect:

$$\mathsf{Until}(\varphi,\psi) \ := \ \downarrow x.\langle R\rangle\downarrow y.\langle R^{-1}\rangle(x \wedge (\langle R\rangle(y \wedge \varphi) \wedge [R](\langle R\rangle y \to \psi))).$$

In other words, instead of retrieving the point named by x using the @ operator, we can reach it by means of $\langle R^{-1}\rangle$. This observation (first made in [Blackburn and Tzakova, 1998a]) is completely general. As long as a $\mathcal{H}_\mathsf{S}(\langle R^{-1}\rangle, @, \downarrow)$ formula does not contain nominals or free state variables, it is always possible to simulate @ by zig-zagging back to the binding point using $\langle R\rangle$ and $\langle R^{-1}\rangle$.

More precisely, suppose a nominal free sentence φ has a temporal depth of n (that is, the maximal depth of embedding of tense operators is n) and that φ is satisfied at a state m. When we evaluate a subformula of φ of the form $@_x\psi$ at some point m' — which cannot be more than n forward and backward steps from m — we know that x must be bound to a point m'' which is also not more than n forward and backward steps from m. Hence m' and m'' are separated by at most $2n$ steps. We can define an operator $@^{2n}$ that allows us to zig-zag to a named state lying within $2n$ steps as follows. Let $2n$-ZZ be the set of all non-empty finite sequences of $\langle R\rangle$ and $\langle R^{-1}\rangle$ operators of length at most $2n$. Then for any formula ψ and any variable x we define:

$$@^{2n}_x\psi \ := \ (x \wedge \psi) \vee \bigvee_{z \in 2n\text{-ZZ}} z(x \wedge \psi).$$

Hence, given a nominal free sentence φ of temporal depth n, we eliminate all occurrences of @ as follows. Let $@_x\psi$ be a subformula of φ where ψ contains no occurrences of @. Replace $@_x\psi$ by $@^{2n}_x\psi$ to form φ'. Repeating this procedure (starting with φ') produces an equivalent nominal-free sentence containing no occurrences of @. Thus, in the setting of tense logic, our characterization results for nominal free sentences go through without the help of @. $\langle R \rangle$, $\langle R^{-1} \rangle$, and $\downarrow$ work together beautifully. No auxiliary apparatus (not even @) is required, and the outcome is a language which exactly captures first-order temporal reachability.

The $\mathcal{H}_{\mathsf{S}}(@)$ Language. To close this section we will discuss one further characterization: Which classes of frames are definable using $\mathcal{H}_{\mathsf{S}}(@)$ formulas whose only atoms are state variables?

In a sense, the standard translation ST already gives us an answer to this question. Let F be a class of frames defined by a sentence φ of the first-order frame language. Then F is definable by a formula of $\mathcal{H}_{\mathsf{S}}(@)$ whose only atoms are variables iff there is some formula α in this fragment such that φ is equivalent to the universal closure of $ST(\alpha)$. Unfortunately, this is not very helpful. Ideally we would like a syntactic characterization of the range of ST when restricted to $\mathcal{H}_{\mathsf{S}}(@)$ formulas whose only atoms are state variables, together with a reverse translation (like our earlier HT).

What about a semantic characterization? Here we can do a little better by using the notion of *@-k-seq-bisimulation* (see the discussion following Definitions 3.13 and 3.14). Let $\mathcal{M} \overset{k}{\sim}_{@} \mathcal{N}$ denote the fact that there is an @-k-seq-bisimulation linking $\mathcal{M}$ and $\mathcal{N}$. A first-order formula $\varphi(\bar{x}, y)$, for $|\bar{x}| = k$ is called *invariant for @-k-seq-bisimulation*, if for all models $\mathcal{M}$, $\mathcal{N}$ such that $\mathcal{M} \overset{k}{\sim}_{@} \mathcal{N}$

$$(\bar{m}, m) \overset{k}{\sim}_{@} (\bar{n}, n) \Rightarrow \mathcal{M} \models \varphi[\bar{m}, m] \text{ iff } \mathcal{N} \models \varphi[\bar{n}, n].$$

Theorem 6.13. *A first-order formula $\varphi(\bar{x}, y)$ is invariant for @-k-seq-bisimulation if and only if it is equivalent to the standard translation of an $\mathcal{H}_{\mathsf{S}}(@)$-formula containing the variables $\bar{x}$.*

Proof. Preservation is straightforward. For the characterization part we do a diagram-chasing. Let $\mathcal{M}, \bar{m}, m$ and $\mathcal{N}, \bar{n}, n$ have the same hybrid theory. Define a relation $\sim$ on the ω-saturated extensions $\mathcal{M}^+$ and $\mathcal{N}^+$ of $\mathcal{M}$ and $\mathcal{N}$ as follows:

$$(\bar{m}, x) \sim (\bar{n}, y) \in B \Leftrightarrow \forall \varphi.(\mathcal{M}^+, \bar{m}, x \Vdash \varphi \Leftrightarrow \mathcal{N}^+, \bar{n}, y \Vdash \varphi).$$

The standard proof shows that $\sim$ is a modal bisimulation. We check the extra conditions. For all i, $(\bar{m}, m_i) \sim (\bar{n}, n_i)$ holds by the following argument. $\mathcal{M}^+, \bar{m}, m_i \Vdash \varphi$ iff $\mathcal{M}^+, \bar{m}, m \Vdash @_{x_i}\varphi$ iff $\mathcal{N}^+, \bar{n}, n \Vdash @_{x_i}\varphi$ iff $\mathcal{N}^+, \bar{n}, n_i \Vdash \varphi$. The other two conditions are satisfied because of the following. Let $(\bar{m}, m_i) \sim (\bar{n}, y)$. Since $\mathcal{M}^+, \bar{m}, m_i \Vdash x_i$, also $\mathcal{N}^+, \bar{n}, y \Vdash x_i$. But then $y = n_i$. QED

Using this results it is easy, for example, to show that $\exists y.(R(x, y) \wedge R(y, y))$ is not equivalent to an @-formula with one free variable, and that $R(x, y) \wedge R(x, z) \rightarrow \exists w.(R(y, w) \wedge R(z, w))$ is not equivalent to an @-formula in three free variables. And it does tell us something about frame definability:

Corollary 6.14. *Let F be a class of frames defined by a sentence φ of the first-order frame language. Then F is definable by a formula of $\mathcal{H}_{\mathsf{S}}(@)$ whose only atoms are variables iff φ is equivalent to the universal closure of a formula that is invariant under @-k-seq-bisimulations.*

6.2 Interpolation and Beth Definability

In 1957, Craig proved the interpolation theorem for first-order logic [Craig, 1957]. Since Craig's paper, interpolation has become one of the standard properties that one investigates when designing a logic, though it has not received the status of a completeness or a decidability theorem. One of the main reasons why a logic should have interpolation is because of "modular theory building." As we will see below, interpolation in modal logic is equivalent to the following property (which is the semantic version of Robinson's consistency property).

> If two theories T_1 and T_2 are consistent (have a model), and they are complete and agree on the common language (i.e., for any formula θ built up from atoms occurring both in T_1 and in T_2 either $T_1 \models \theta$ and $T_2 \models \theta$, or $T_1 \models \neg\theta$ and $T_2 \models \neg\theta$), then $T_1 \cup T_2$ has a model.

The property is not only intuitively valid for scientific reasoning, it also has practical consequences. In computer science for example, it shows up in the incremental design, specification and development of software and has received quite some attention in that community, (see, e.g., [Renardel de Lavalette, 1989]). There are also technical reasons why interpolation is desirable. In particular, it can be used to establish the Beth definability property which we already discussed in Section 4.5.3 and we will formally introduce in Section 6.2.2.

6.2.1 Kinds of Interpolation

For first-order logic we find the following notions of interpolation in the literature.

Definition 6.15. [Interpolation property] Let $\mathsf{P}(\varphi)$ be the set of atomic symbols occurring in φ.

- The *Arrow Interpolation Property* (AIP) holds if, whenever $\models \varphi \to \psi$, there exists a formula θ such that $\models \varphi \to \theta$, $\models \theta \to \psi$ and $\mathsf{P}(\theta) \subseteq \mathsf{P}(\varphi) \cap \mathsf{P}(\psi)$.
- The *Turnstile Interpolation Property* (TIP) holds if, whenever $\varphi \models \psi$, there exists a formula θ such that $\varphi \models \theta$, $\theta \models \psi$ and $\mathsf{P}(\theta) \subseteq \mathsf{P}(\varphi) \cap \mathsf{P}(\psi)$.

For first-order logic the two versions are equivalent but in general this is not the case (as we see below this depends on both compactness and the availability of a deduction theorem, see [Czelakowski, 1982].) The meaning of TIP in modal logics depends on the way we define the consequence relation $\varphi \models \psi$. Remember our discussion in Section 4.2.1. In modal and hybrid logic the different interpolation properties are related as follows.

Proposition 6.16.

i. With the local consequence relation $\models^{loc}$, AIP and TIP are equivalent.

ii. If $\models^{loc}$ is compact, then AIP implies TIP.

For this reason, from now on we take TIP to be defined using the *global* consequence relation. AIP and TIP are often referred to as the strong and weak interpolation properties respectively, and we will sometimes use this terminology. In what follows, $\models$ will represent $\models^{loc}$, and we will explicitly write $\models^{glo}$ when we refer to global consequence.

Coming back to merging of theories, by a standard proof one can show that

Proposition 6.17. *If the local consequence relation is compact, the arrow interpolation property and the Robinson consistency property are equivalent.*

6.2.2 Beth Definability

The Beth definability property [Beth, 1953] is usually studied together with interpolation (and in many cases interpolation is considered just a step in the proof of Beth definability). Loosely speaking, a logic has the Beth definability property if any implicit definition has also an explicit definition. More precisely:

Definition 6.18. [Beth definability property]

– A logic has the *local Beth definability property* if for all formulas $\varphi(\bar{a}, a)$ whose atomic symbols occur among $\bar{a}, a$, if $\models \varphi(\bar{a}, a/b_1) \wedge \varphi(\bar{a}, a/b_2) \to (b_1 \leftrightarrow b_2)$ then there is a formula $\theta(\bar{a})$ such that $\models \varphi(\bar{a}, a) \to (\theta(\bar{a}) \leftrightarrow a)$.
– A logic has the *global Beth definability property* if for all formulas $\varphi(\bar{a}, a)$ whose atomic symbols occur among $\bar{a}, a$, if $\varphi(\bar{a}, a/b_1) \wedge \varphi(\bar{a}, a/b_2) \models^{glo} b_1 \leftrightarrow b_2$ then there is a formula $\theta(\bar{a})$ such that $\varphi(\bar{a}, a) \models^{glo} \theta(\bar{a}) \leftrightarrow a$.

In first-order and modal logics the local Beth definability property is equivalent to arrow interpolation (see [Kracht, 1999] for a detailed discussion) and this relation also holds for hybrid languages.

In contrast, the relation between global Beth and turnstile interpolation is not as tight. It can be shown that there exist logics without global interpolation while having the global Beth definability property, and that there are logics with global interpolation without Beth definability. See [Maksimova, 1991b].

In the next section, we will investigate interpolation for certain hybrid languages. Given Proposition 6.19 below, whenever we succeed in establishing arrow interpolation for a hybrid language, we immediately obtain both global and local Beth definability.

Proposition 6.19. *Let $\mathcal{L}$ be a hybrid language having the arrow interpolation property. Then $\mathcal{L}$ has both the global and local Beth definability property.*

Proof. The proof is standard. See [Kracht, 1999]. QED

6.2.3 Interpolation for $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$, Fragments and Extensions

In this section we will prove AIP for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$, disprove AIP and TIP for its finite variable fragments (our earlier work on k-seq-bisimulations will enable us to construct straightforward counterexamples) and some other fragments, while we show that TIP holds for the sublanguage $\mathcal{H}_{\mathsf{S}}(@)$. Interestingly, we will be able to generalize the positive results by drawing on the general completeness result we discussed in Theorem 6.12.

Even though interpolation was originally considered a property of deductive systems and established using proof-theoretical arguments as we did in Section 4.5.3, here we will take it as a property of *consequence relations* and prove it using semantic arguments (as is done, for example, in [Chang and Keisler, 1990]). Jerry Seligman has recently announced a proof-theoretical version of our result which has not yet been made available.

We turn to the technicalities of the interpolation result. As is usual in interpolation proofs, where language related issues require special care, we replace the notion of consistency by the finer-grained notion of separability.

Definition 6.20. [Separability] Let T, U, L be sets of formulas. We say that the pair $\langle T, U\rangle$ *is separable with respect to* L if there exists a formula $\theta \in L$ such that $T \models \theta$ and $U \models \neg\theta$. $\langle T, U\rangle$ *is inseparable with respect to* L if it is not separable with respect to L.

The following theorems can be derived from the axiomatization introduced in Definition 3.10 (see [Blackburn and Tzakova, 1999, Lemma 4.1] and [Blackburn and Tzakova, 1998b, Lemma 7] for details).

Proposition 6.21. [Derived theorems and properties]

$K_{\downarrow}$. $\downarrow v.(\varphi \to \psi) \to (\downarrow v.\varphi \to \downarrow v.\psi)$.

Distr$_{@}$. $@_s(\varphi \wedge \psi) \leftrightarrow (@_s\varphi \wedge @_s\psi)$.

Intr$_{@}$. $(s \wedge \varphi) \to @_s\varphi$.

In addition, if $\vdash \varphi$ *and* i *is a nominal in* φ*, then for some state variable* x *not occurring in* φ*,* $\vdash \downarrow x.\varphi[i/x]$*. If* $\vdash \varphi$ *and* x *is a free variable in* φ*, then for some nominal* i *not occurring in* φ*,* $\vdash \varphi[x/i]$*.*

We are ready to prove the main result of this section.

Theorem 6.22. [Arrow interpolation for $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$] *Let* φ *and* ψ *be formulas in the language* $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$*, such that* $\models \varphi \to \psi$*. Then there exists a formula* θ *such that*

i. $\models \varphi \to \theta$ *and* $\models \theta \to \psi$*.*

ii. $\mathsf{IP}(\theta) \subseteq \mathsf{IP}(\varphi) \cap \mathsf{IP}(\psi)$*.*

Proof. Suppose we are given formulas φ_0 and ψ_0 such that there is no interpolant for $\varphi_0 \to \psi_0$. We will prove that $\not\models \varphi_0 \to \psi_0$ by producing a model $\mathcal{M} = \langle M, R, V\rangle$ and an assignment g such that for some $m \in M$, $\mathcal{M}, g, m \Vdash \varphi_0 \wedge \neg\psi_0$ (the proof follows the method of [Chang and Keisler, 1990] in which two models are simultaneously built using fresh constants).

We can assume that $\{\varphi_0\}$ and $\{\neg\psi_0\}$ are consistent (for if they are not, then either $\bot$ or $\top$ is an interpolant). Furthermore they must be inseparable over the formulas in $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ with atomic symbols in $\mathsf{IP}(\varphi_0) \cap \mathsf{IP}(\psi_0)$.

Let $\mathcal{L}$ be the set of formulas of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ over the expanded signature $\langle\mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR}\rangle$, and $\mathcal{L}'$ the set over the expended signature $\langle\mathsf{PROP}, \mathsf{NOM} \cup \mathrm{N}, \mathsf{SVAR}\rangle$

where $\mathrm{N} = \{n_0, \ldots, n_k, \ldots\}$ is a countably infinite set of new nominals. For a formula φ define the restricted language $\mathcal{L}_\varphi$ as $\{\xi \in \mathcal{L} \mid \mathsf{IP}(\xi) \subseteq \mathsf{IP}(\varphi)\}$ and $\mathcal{L}'_\varphi$ as $\{\xi \in \mathcal{L}' \mid \mathsf{IP}(\xi) \subseteq \mathsf{IP}(\varphi) \cup \mathrm{N}\}$. Let $\mathcal{L}'_{\varphi_0\psi_0} = \mathcal{L}'_{\varphi_0} \cap \mathcal{L}'_{\psi_0}$.

Let $\varphi_1, \ldots, \varphi_k, \ldots$ (respectively, $\psi_1, \ldots, \psi_k, \ldots$) be an enumeration of all formulas in $\mathcal{L}'_{\varphi_0}$ ($\mathcal{L}'_{\psi_0}$). We define the sequences $\{n_0\} \cup \{\varphi_0\} = T_0 \subseteq T_1 \subseteq T_2 \subseteq \ldots$ and $\{n_0\} \cup \{\neg\psi_0\} = U_0 \subseteq U_1 \subseteq U_2 \subseteq \ldots$ as follows:

If $T_j \cup \{\varphi_j\}$ and U_j are separable over $\mathcal{L}'_{\varphi_0\psi_0}$ then $T_{j+1} = T_j$, otherwise

- if $\varphi_j \neq @_s s$ and $\varphi_j \neq @_s\Diamond\varphi'$ for $s \in \mathsf{SSYM}$, then $T_{j+1} = T_j \cup \{\varphi_j\}$,
- if $\varphi_j = @_s s$, then $T_{j+1} = T_j \cup \{\varphi_j\} \cup \{@_s(n_k \wedge s) \mid n_k \in \mathrm{N}\backslash\mathsf{NOM}(T_j \cup U_j)\}$,
- if $\varphi_j = @_s\Diamond\varphi'$, then $T_{j+1} = T_j \cup \{\varphi_j\} \cup \{@_s\Diamond(n_k \wedge \varphi') \mid n_k \in \mathrm{N}\backslash\mathsf{NOM}(T_j \cup U_j \cup \{\varphi_j\})\}$.

If T_{j+1} and $U_j \cup \{\psi_j\}$ are separable over $\mathcal{L}'_{\varphi_0\psi_0}$ then $U_{j+1} = U_j$, otherwise

- if $\psi_j \neq @_s s$ and $\psi_j \neq @_s\Diamond\psi'$ for $s \in \mathsf{SSYM}$, then $U_{j+1} = U_j \cup \{\psi_j\}$,
- if $\psi_j = @_s s$, then $U_{j+1} = U_j \cup \{\psi_j\} \cup \{@_s(n_k \wedge s) \mid n_k \in \mathrm{N}\backslash\mathsf{NOM}(T_{j+1} \cup U_j)\}$,
- if $\psi_j = @_s\Diamond\psi'$, then $T_{j+1} = T_j \cup \{\psi_j\} \cup \{@_s\Diamond(n_k \wedge \psi') \mid n_k \in \mathrm{N}\backslash\mathsf{NOM}(T_{j+1} \cup U_j \cup \{\psi_j\})\}$.

The fresh nominals play the same role as *Henkin witnesses* in first-order proofs: they ensure that we obtain models in which every world has a name. Define

$$T_\omega = \bigcup_{j\in\omega} T_j \quad \text{and} \quad U_\omega = \bigcup_{j\in\omega} U_j.$$

CLAIM 6.23. For all $j \in \omega$, $\langle T_j, U_j\rangle$ is inseparable with respect to $\mathcal{L}'_{\varphi_0\psi_0}$. Whence $\langle T_\omega, U_\omega\rangle$ is an inseparable pair in this language. Furthermore T_ω and U_ω are maximally consistent in $\mathcal{L}'_{\varphi_0}$ and $\mathcal{L}'_{\psi_0}$, respectively. Hence for all $\theta \in \mathcal{L}'_{\varphi_0\psi_0}$: $\theta \in T_\omega \Leftrightarrow \theta \in U_\omega$.

PROOF OF CLAIM. The proof is by induction on j. Separability/inseparability below is with respect to $\mathcal{L}'_{\varphi_0\psi_0}$ except when otherwise mentioned.

BASE CASE $j = 0$. Suppose $\langle T_0, U_0\rangle$ is separable. Then there is a formula $\theta \in \mathcal{L}'_{\varphi_0\psi_0}$ such that $\models n_0 \wedge \varphi_0 \rightarrow \theta$ and $\models n_0 \wedge \neg\psi_0 \rightarrow \neg\theta$. θ might contain some nominals of N, say $\{n_{i_1}, \ldots, n_{i_k}\}, k \geq 0$. Let $x_0, x_{i_1}, \ldots, x_{i_k} \in \mathsf{SVAR}$ which don't occur in $\varphi_0, \psi_0, \theta$. We will write $\theta[x_0 x_{i_1} \ldots x_{i_k}]$ for the formula obtained from θ by replacing n_{i_j} by x_{i_j}, and n_0 by x_0. Then, making use of the complete axiomatization given Definition 3.10, we have

$$\begin{array}{ll}
\models \varphi_0 \rightarrow (n_0 \rightarrow \theta) & \\
\models {\downarrow} x_{i_k}.(\varphi_0 \rightarrow (n_0 \rightarrow \theta[x_{i_k}])) & \text{Proposition 6.21} \\
\models \varphi_0 \rightarrow (n_0 \rightarrow {\downarrow} x_{i_k}.\theta[x_{i_k}]) & \textit{Q1} \text{ twice} \\
\models \varphi_0 \rightarrow (n_0 \rightarrow {\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\theta[x_{i_1}\ldots, x_{i_k}]) & \text{Similarly} \\
\models {\downarrow} x_0.(\varphi_0 \rightarrow (x_0 \rightarrow {\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\theta[x_0 x_{i_1}\ldots, x_{i_k}])) & \text{Proposition 6.21} \\
\models \varphi_0 \rightarrow {\downarrow} x_0.(x_0 \rightarrow {\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\theta[x_0 x_{i_1}\ldots, x_{i_k}])) & \textit{Q1} \\
\models \varphi_0 \rightarrow {\downarrow} x_0.{\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\theta[x_0 x_{i_1}\ldots, x_{i_k}] & \textit{Q3}
\end{array}$$

$$\begin{array}{ll}
\models \neg\psi_0 \rightarrow (n_0 \rightarrow \neg\theta) & \\
\models \neg\psi_0 \rightarrow {\downarrow} x_0.{\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\neg\theta[x_0 x_{i_1}\ldots, x_{i_k}] & \text{As before} \\
\models \neg\psi_0 \rightarrow \neg{\downarrow} x_0.{\downarrow} x_{i_1}.\ldots{\downarrow} x_{i_k}.\theta[x_0 x_{i_1}\ldots, x_{i_k}] & \text{Self Dual}_{\downarrow}
\end{array}$$

$\theta[x_0x_{i_1}\ldots,x_{i_k}]$ is a formula in $\mathcal{L}_{\varphi_0}\cap\mathcal{L}_{\psi_0}$ and thus $\langle\{\varphi_0\},\{\neg\psi_0\}\rangle$ is separable over $\mathcal{L}_{\varphi_0}\cap\mathcal{L}_{\psi_0}$. Contradiction.

By using the inductive hypothesis "$\langle T_j,U_j\rangle$ is an inseparable pair" and going step by step through the construction, the inseparability of $\langle T_{j+1},U_{j+1}\rangle$ is easily established. $\dashv$

We are going to construct now a "named" or "labeled" model. Labeled models have played a crucial role in the development of the model theory of hybrid languages and they were successfully used already by the Sofia school in their axiomatic investigations for combinatory PDL (see for example [Passy and Tinchev, 1985b]). There is also a strong connection with Venema's work on general completeness results for modal logics containing the D difference operator [1991]. Venema discusses logics over a *versatile* signature containing D, but from the proof in Chapter 2 of [1991] we can see that the condition of versatility can be replaced by the inclusion of @ in the signature, and that the D operator is actually used only to force propositions to behave as nominals.

To start the construction we recall the notion of pasted maximal consistent sets (MCS) and labeled models from [Blackburn and Tzakova, 1999]. A maximal consistent set Γ is *pasted* if $@_s\varphi\in\Gamma$ implies $@_s(i\wedge\varphi)\in\Gamma$ for some nominal i, and $@_s\Diamond\varphi\in\Gamma$ implies $@_s\Diamond(i\wedge\varphi)\in\Gamma$ for some nominal i. A pasted MCS Γ is *labeled* by a nominal i precisely when $i\in\Gamma$. Let Γ be a pasted MCS labeled by a nominal, then for all state symbols s appearing in Γ, let $\Delta_s=\{\varphi\mid @_s\varphi\in\Gamma\}$. Then the *labeled model* yielded by Γ is $\mathcal{M}=\langle M,R,V\rangle$, where $M=\{\Delta_s\mid s \text{ is a state symbol in } \Gamma\}$, $R(\Delta,\Delta')$ iff $\{\varphi\mid\Box\varphi\in\Delta\}\subseteq\Delta'$ and $\Delta\in V(p)$ iff $p\in\Delta$, for p a propositional variable or nominal.

We define the natural assignment $g:\mathsf{SVAR}\to M$ by $g(x)=\{m\in M\mid x\in m\}$.

By construction T_ω and U_ω are pasted MCSs labeled by the nominal $n_0\in\mathrm{N}$. Let $\mathcal{M}_{\varphi_0}=\langle M_{\varphi_0},R_{\varphi_0},V_{\varphi_0}\rangle$ be the labeled model obtained from T_ω and $\mathcal{M}_{\psi_0}=\langle M_{\psi_0},R_{\psi_0},V_{\psi_0}\rangle$ the one obtained from U_ω. Finally, let g_{φ_0} and g_{ψ_0} be the natural assignments defined as $g_{\varphi_0}(x)=\{m\in M_{\varphi_0}\mid x\in m\}$ and $g_{\psi_0}(x)=\{m\in M_{\psi_0}\mid x\in m\}$. We use $\Delta_s^{\varphi_0}$ and $\Delta_s^{\psi_0}$ to denote elements of M_{φ_0} and M_{ψ_0}.

Claim 6.24.

i. $\Delta_{n_0}^{\varphi_0}=T_\omega$ and $\Delta_{n_0}^{\psi_0}=U_\omega$.

ii. For $\Delta_s^{\varphi_0}\in M_{\varphi_0}$ there is $n\in\mathrm{N}$ such that $n\in\Delta_s^{\varphi_0}$ (or equivalently $\Delta_s^{\varphi_0}=\Delta_n^{\varphi_0}$). Similarly for $\Delta_s^{\psi_0}\in\mathcal{M}_{\psi_0}$.

Proof of Claim.

i) We show that $\Delta_{n_0}^{\varphi_0}=T_\omega$; the other case is similar. $\Delta_{n_0}^{\varphi_0}$ is an MCS because $@_{n_0}$ is self-dual. So it is sufficient to show that $\Delta_{n_0}^{\varphi_0}\supseteq T_\omega$. Let $\varphi\in T_\omega$. By $\mathrm{Intr}_@\models n_0\wedge\varphi\to @_{n_0}\varphi$. Because $@_{n_0}\varphi\in\mathcal{L}'_{\varphi_0}$, $n_0\in T_\omega$ and T_ω is maximal in $\mathcal{L}'_{\varphi_0}$, $@_{n_0}\varphi\in T_\omega$. By definition $\varphi\in\Delta_{n_0}^{\varphi_0}$.

ii) Lemma 4.3.5 in [Blackburn and Tzakova, 1999] proves that for n a nominal, Δ_n a maximal consistent set labeled by n, and Γ a maximal consistent set, if $n\in\Gamma$ then $\Delta_n=\Gamma$. Using this result we can establish *ii*) as follows.

We prove the case for $\mathcal{M}_{\varphi_0}$. $@_ss$ is a formula in $\mathcal{L}'_{\varphi_0}$, hence $@_ss=\varphi_j$ for some j, and furthermore a theorem. Hence $\{@_ss\}$ will be added to T_{j+1} together with $@_s(s\wedge n_k)$ for a new nominal n_k. Hence $n_k\in\Delta_s^{\varphi_0}$ and $\Delta_{n_k}^{\varphi_0}=\Delta_s^{\varphi_0}$. $\dashv$

From Claim 6.24 and following [Blackburn and Tzakova, 1999, Lemma 4.8], it follows that $\mathcal{M}_{\varphi_0}$, g_{φ_0} and $\mathcal{M}_{\psi_0}$, g_{ψ_0} satisfy a Truth Lemma, and thus

$$\mathcal{M}_{\varphi_0}, g_{\varphi_0}, \Delta_{n0}^{\varphi_0} \Vdash \varphi \text{ and } \mathcal{M}_{\psi_0}, g_{\psi_0}, \Delta_{n0}^{\psi_0} \Vdash \neg\psi. \tag{6.1}$$

Furthermore the two models are very closely related in the following sense.

Claim 6.25. Let a function $h : M_{\varphi_0} \to M_{\psi_0}$ be defined as $h(\Delta_n^{\varphi_0}) = \Delta_n^{\psi_0}$, for $n \in \mathrm{N}$. Then h is a homomorphism in the common language $\mathcal{L}'_{\varphi_0\psi_0}$. Moreover, $g_{\psi_0} = h \circ g_{\varphi_0}$.

Proof of Claim. h is defined at every member of the domain of $\mathcal{M}_{\varphi_0}$ by Claim 6.24.*ii*) and the fact that for any $n \in \mathrm{N}$, both $\Delta_n^{\varphi_0}$ and $\Delta_n^{\psi_0}$ are uniquely defined. Moreover, h is a bijection because $@_n\top \in T_\omega$ iff $@_n\top \in U_\omega$ and in $\mathcal{M}_{\varphi_0}$ and $\mathcal{M}_{\psi_0}$ nominals are interpreted as singleton sets. For any proposition symbol p in $\mathcal{L}'_{\varphi_0\psi_0}$ we have $\Delta_n^{\varphi_0} \in V_{\varphi_0}(p)$ iff $@_n p \in T_\omega$ iff $@_n p \in U_\omega$ iff $h(\Delta_n^{\varphi_0}) \in V_{\psi_0}(p)$. For the relation R, $R_{\varphi_0}(\Delta_n^{\varphi_0}, \Delta_{n'}^{\varphi_0})$ iff $@_n\Diamond n' \in T_\omega$ iff $@_n\Diamond n' \in U_\omega$ iff $R_{\psi_0}(h(\Delta_n^{\varphi_0}), h(\Delta_{n'}^{\varphi_0}))$. A similar argument shows that $g_{\psi_0} = h \circ g_{\varphi_0}$. ⊣

Since the two models share the same frame and agree on the common language, there is a model $\mathcal{M}$ and an assignment g for the union of the two languages which have $\mathcal{M}_{\varphi_0}$, g_{φ_0} and $\mathcal{M}_{\psi_0}$, g_{ψ_0} as reducts. But then by (6.1), $\mathcal{M}, g, \Delta_{n_0} \Vdash \varphi_0 \wedge \neg\psi_0$, and we are finished. QED

Actually, we can prove a stronger result: we can restrict the free variables occurring in the interpolant θ to only those appearing both in φ_0 and ψ_0. Reason as follows. Assume $\models \varphi_0(\bar{x}, \bar{z}) \to \psi_0(\bar{y}, \bar{z})$; here we explicitly show the state variables free in φ_0 and ψ_0 and indicate which are shared. By Proposition 6.21 we can replace sequences of free variables $\bar{x}$, $\bar{y}$, $\bar{z}$ by new nominals $\bar{i_x}$, $\bar{i_y}$, $\bar{i_z}$ such that $\models \varphi_0(\bar{i_x}, \bar{i_z}) \to \psi_0(\bar{i_y}, \bar{i_z})$. Use Theorem 6.22 to find an interpolant $\theta(\bar{s}, \bar{i_z})$. The $\bar{s}$ are free variables that might appear in θ; we can assume $\bar{s}$ to be disjoint from $\bar{x}, \bar{y}, \bar{z}$. By $\mathrm{N}_\downarrow$ and Q_1 we obtain $\models \varphi_0(\bar{i_x}, \bar{i_z}) \to {\downarrow}\bar{s}.\theta(\bar{i_z})$ and $\models {\downarrow}\bar{s}.\theta(\bar{i_z}) \to \psi_0(\bar{i_y}, \bar{i_z})$. We now rename again the formulas to the original $\bar{x}, \bar{y}, \bar{z}$ and the interpolant will only contain free state variables common to φ_0 and ψ_0.

Furthermore, note that nothing in the proof is intrinsically tied to the number of modalities in the language. In other words, arrow interpolation also holds for the multimodal versions of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ if modalities are allowed freely in the interpolant, i.e., if modalities are taken as logical operators and not considered part of the restriction on common languages.

Corollary 6.26. *Multi-modal $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ has AIP if no restriction on occurrences of modalities is imposed for the interpolant.*

However when we restrict the interpolant to contain only the modalities in the common language, then interpolation does *not* follows immediately, especially if the modalities interact (for example, if the theory contains axioms involving more than one modality; see [Marx, 1999] for examples of this type). We conjecture that interpolation goes through even if the interpolant's modalities are restricted to the common language, for logics where the modalities don't interact.

But the most important generalization is that strong interpolation holds not only in the minimal logic of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ *but in any pure axiomatic extension.* As is shown in, for example, [Blackburn and Tzakova, 1998a], labeled models validate pure axioms. Now, we have shown how to use labeled models to prove interpolation in Theorem 6.22. So if we use the same construction for any extension of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ by adding *pure* axioms, the resulting frame will validate the extra axioms. Hence in view of our earlier characterization (Theorem 6.10) of the bounded fragment we have:

THEOREM 6.27. *Let $\varphi(\bar{x})$ be a frame condition in the bounded fragment. The theory in the hybrid language $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ of the class* FRAMES$(\forall\bar{x}.\varphi(\bar{x}))$ *has AIP.*

This result stands in sharp contrast with the scarcity of general interpolation results obtained for the basic modal language; see for example [Maksimova, 1991a].

On the other hand, it is clear from the proof that the number of state variables needed cannot be bounded (they are used to quantify away the nominals in the proof of Claim 6.23). Indeed, if we restrict $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ to only a finite number of variables, then arrow interpolation *fails.* We use the notion of k-seq-bisimulations to provide counterexamples. Let's consider the case of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ restricted to only one state variable.

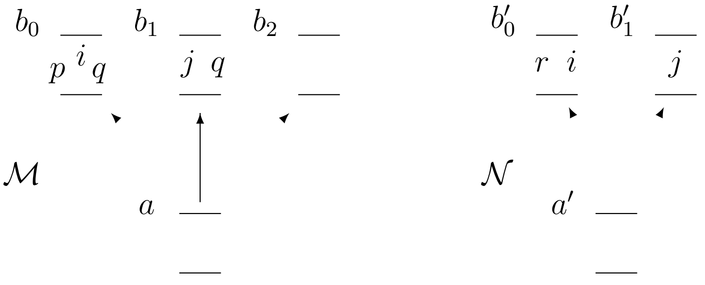

Figure 6.1: Counterexample to interpolation

Take the models in Figure 6.1 and the formulas

$$\begin{aligned} \varphi &= \Diamond(p \wedge q) \wedge \Diamond(\neg p \wedge q) \wedge \Diamond(\neg p \wedge \neg q) \\ \psi &= (\Diamond r \wedge \Box(r \to i)) \to (\Diamond(\neg r \wedge j) \to \Diamond(\neg r \wedge \neg j)). \end{aligned} \tag{6.2}$$

$\varphi \to \psi$ is valid because for any world with at least three different successors, if there is a unique accessible r-world and one of the accessible $\neg r$-worlds is named by the nominal j, then the second accessible $\neg r$-world is named $\neg j$. Furthermore, $\mathcal{M}$ and $\mathcal{N}$ 1-bisimulate in the common (empty) language of φ and ψ via the relation

$$(m,m') \overset{1}{\sim} (n,n') \text{ iff } d(m) = d(n) \ \&\ d(m') = d(n') \ \&\ (m = m' \Leftrightarrow n = n'),$$

where $d(w)$ is the distance from the root to w. Finally, φ is true in $\mathcal{M}, a$, while ψ is false at $\mathcal{N}, a'$ which proves that an interpolant with only one propositional variable does not exist. Actually, the simpler formulas $\varphi = \Diamond p \wedge \Diamond\neg p$ and $\psi = \Diamond i \to \Diamond\neg i$ provide a counterexample to strong interpolation in the one-variable fragment, but they have $\Diamond\top \wedge (\Diamond x \to \Diamond\neg x)$ as weak interpolant. The more complex example proves failure in the weak case also.

Notice that the heart of the counterexample is just a counting argument, which can be reproduced for the other finite variable fragments of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ by taking bigger and bigger models $\mathcal{M}$ and $\mathcal{N}$ exhibiting the same basic pattern. Hence:

THEOREM 6.28. *AIP fails in all finite variable fragments of* $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$.

A more complex counterexample based on the same idea can be set up to prove failure of weak (turnstile) interpolation. Consider again the formulas φ and ψ in (6.2). Clearly $\varphi \models^{glo} \psi$. Take now the model $\mathcal{M}$ and define $\mathcal{M}'$ by linking new copies of b_0, b_1 and b_2 to each terminal world in $\mathcal{M}$. Let $\mathcal{M}_\omega$ be the infinite model obtained by iterating this operation ω times and similarly for $\mathcal{N}_\omega$. Now $\mathcal{M}_\omega$ makes φ globally true. Suppose θ is an interpolant on one variable. Then as $\varphi \models^{glo} \theta$, θ is globally true at $\mathcal{M}_\omega$.

We need something stronger than a mere 1-bisimulation linking $\mathcal{M}_\omega$ and $\mathcal{N}_\omega$, as we want to transfer *global* truth. With ordinary modal languages, requiring $\sim$ to be total and surjective is enough, but we have to take care of assignments as well. We will say that a k-seq-bisimulation between $\mathcal{M}$ and $\mathcal{N}$ is *full* if for every $\langle \bar{m}, m\rangle \in {}^kM \times M$ there is $\langle \bar{n}, n\rangle \in {}^kN \times N$ such that $\langle \bar{m}, m\rangle \overset{k}{\sim} \langle \bar{n}, n\rangle$ and vice versa. If we can define a full 1-bisimulation between $\mathcal{M}_\omega$ and $\mathcal{N}_\omega$ then $\mathcal{N}_\omega \models^{glo} \theta$. But $\overset{1}{\sim}$ defined as in the previous case is indeed full. Hence, as $\theta \models^{glo} \psi$, ψ should be globally true in $\mathcal{N}_\omega$ — but it is not.

THEOREM 6.29. *TIP fails in all finite variable fragments of* $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$.

The models in Figure 6.1 and the formulas in (6.2) can also be used to prove failure of arrow interpolation for a number of fragments of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. It is just a question of verifying that the formulas lie in the appropriate language and that the right bisimulation links a and a'.

THEOREM 6.30. *AIP fails for the basic modal language extended with nominals, for* $\mathcal{H}_{\mathsf{N}}(@)$*, and for their extensions with the* $\langle R^{-1}\rangle$ *and the* E *operators.*

Finally, we see from the proof of Theorem 6.22 that the $\downarrow$ binder is needed in Claim 6.23. So what about interpolation in the sublanguage $\mathcal{H}_{\mathsf{S}}(@)$? We can again use models $\mathcal{M}$ and $\mathcal{N}$ to prove that arrow interpolation fails. We use the restricted version of k-seq-bisimulation which leaves out condition $(\downarrow)$. In this framework we can define for any k, a k-seq-bisimulation between $\mathcal{M}$ and $\mathcal{N}$ such that for any $\bar{m} \in {}^kM$ and any $\bar{n} \in {}^kN$, $(\bar{m}, a) \overset{k}{\sim} (\bar{n}, a')$. This proves that there is no arrow interpolant for $\varphi \to \psi$ in $\mathcal{H}_{\mathsf{S}}(@)$.

THEOREM 6.31. *AIP fails in* $\mathcal{H}_{\mathsf{S}}(@)$.

But *weak* interpolation holds for $\mathcal{H}_{\mathsf{S}}(@)$ because the role of $\downarrow$ is played by the implicit quantification in the definition of $\varphi \models^{glo} \psi$.

THEOREM 6.32. *Let* φ *and* ψ *be sentences of* $\mathcal{H}_{\mathsf{S}}(@)$ *such that* $\varphi \models^{glo} \psi$*. Then there is a formula* θ*, which may contain additional free variables, such that*

i. $\varphi \models^{glo} \theta$ *and* $\theta \models^{glo} \psi$.

ii. $\mathsf{IP}(\theta) \subseteq \mathsf{IP}(\varphi) \cap \mathsf{IP}(\psi)$.

Outline of Proof. We show how to modify the proof of arrow interpolation for $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ (cf. Theorem 6.22) to obtain the result.

First, the construction of the pasted sets T_ω and U_ω needs to be adjusted as we have to ensure that the labeled models obtained from them globally satisfy φ_0 and $\neg\psi_0$. To that end, whenever we run into a formula of the form $@_s s$ or $@_s\Diamond\xi$ we paste not only a new nominal n_k but also the formulas we want to make globally true. For example one clause in the definition of T_{j+1} would read

– $\varphi_j = @_s s$, then $T_{j+1} = T_j \cup \{\varphi_j, @_s(n_k \wedge s \wedge \varphi_0)\}$, for $n_k \in \mathrm{N}\backslash\mathsf{NOM}(T_j \cup U_j)$.

We will need to show that for all $j \in \omega$, $\langle T_j, U_j\rangle$ is (globally) inseparable with respect to $\mathcal{L}'_{\varphi_0\psi_0}$. The base case is simple: if θ (including perhaps some new nominals $\{n_{i_i}, \ldots, n_{i_k}\}$) separates $\langle T_0, U_0\rangle$ on $\mathcal{L}'_{\varphi_0\psi_0}$, then $\theta[x_{i_1} \ldots x_{i_k}]$ separates $\langle\{\varphi_0\}, \{\neg\psi_0\}\rangle$, for new variables $\{x_{i_i}, \ldots, x_{i_k}\}$; this is precisely where the free variables in the interpolant are needed.

What about the inductive step? Consider, for example, the case of $\varphi_j = @_s s$. Assume that $\langle T_j \cup \{\varphi_j\}, U_j\rangle$ is inseparable in $\mathcal{L}'_{\varphi_0\psi_0}$; we want to prove that $\langle T_j \cup \{\varphi_j, @_s(n_k \wedge s \wedge \varphi_0)\}, U_j\rangle$ is inseparable. Suppose θ separates this last pair. Then $U_j \models^{glo} \neg\theta$ while $T_j \cup \{@_s s, @_s(n_k \wedge s \wedge \varphi_0)\} \models^{glo} \theta$. Because $@_s(n_k \wedge s \wedge \varphi_0)$ is an @-formula, this is the case iff $T_j \cup \{@_s s\} \models^{glo} @_s(n_k \wedge s \wedge \varphi_0) \to \theta$. Furthermore, as $\varphi_0 \in T_j$ and n_k is a new nominal by definition, for all $\mathcal{M}$, $\mathcal{M} \Vdash T_j$ implies $\mathcal{M} \Vdash @_s(n_k \wedge s \wedge \varphi_0)$. Hence $T_j \cup \{@_s s\} \models^{glo} \theta$. Contradiction.

From now on the proof follows the same lines as before. We obtain labeled models such that $\mathcal{M}_{\varphi_0} \Vdash \varphi_0$ and $\mathcal{M}_{\psi_0} \Vdash \neg\psi_0$ sharing the same frame, from which we build a model $\mathcal{M}$ where $\varphi_0 \wedge \neg\psi_0$ holds globally. QED

We will prove in Chapter 7 that $\mathcal{H}_{\mathsf{N}}(@)$ is well behaved with respect to complexity: like ordinary uni-modal logic it has a local PSpace-complete satisfiability problem. On the other hand, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is known to be undecidable. $\mathcal{H}_{\mathsf{N}}(@)$ does not have interpolation (not even weak interpolation), while $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ has one of the strongest versions of interpolation for modal languages. Extending $\mathcal{H}_{\mathsf{N}}(@)$ to $\mathcal{H}_{\mathsf{S}}(@)$ gives us weak interpolation, but as we prove in Theorem 7.17, it turns the global *Sat* problem undecidable. It is natural to ask if there is any computationally well behaved hybrid language extending $\mathcal{H}_{\mathsf{N}}(@)$ that enjoys arrow interpolation. Our conjecture is that the addition of graded modalities to $\mathcal{H}_{\mathsf{N}}(@)$ would provide such a system (see the discussion at the end of Section 4.5.5).

The table below summarizes the positive results concerning interpolation we have established in this section and in Section 4.5.3. Let $\varphi(\bar{x})$ be a frame condition in the bounded fragment,

Language	AIP	TIP
$\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ over the class FRAMES$(\forall\bar{x}.\varphi(\bar{x}))$	yes	yes
Multi-modal $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$	yes	yes
$\mathcal{H}_{\mathsf{S}}(@, \downarrow)$	yes	yes
$\mathcal{H}_{\mathsf{S}}(@)$	no	yes
$\mathcal{H}(\langle R^{-1}\rangle, @, @\Diamond)$	yes	yes
$\mathcal{H}(@, @\Diamond)$	yes	yes

On the other hand, all the following logics fail to have both arrow and turnstile interpolation: the finite variable fragments of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$, $\mathcal{H}_{\mathsf{N}}$, $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$, $\mathcal{H}_{\mathsf{N}}(\mathsf{E})$, $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, \mathsf{E})$, $\mathcal{H}_{\mathsf{N}}(@)$, $\mathcal{H}_{\mathsf{N}}(@, \langle R^{-1}\rangle)$, $\mathcal{H}_{\mathsf{N}}(@, \mathsf{E})$ and $\mathcal{H}_{\mathsf{N}}(@, \langle R^{-1}\rangle, \mathsf{E})$.

6.3 Reflections

From a model-theoretical point of view the language $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is surprisingly well behaved. As we discussed in this chapter, it can be characterized in many different and natural ways, it responds with ease to both modal and first-order techniques, and has one of the strongest versions of the interpolation and Beth properties we are aware of for modal languages. This is in addition to the general completeness results we already discussed in Sections 3.4 and 6.1.4.

Sometimes it happens that a particular language is in a state of "perfect equilibrium," offering high expressive power, simplicity, well understood proof and model theory, etc. The first-order language is such an example, and the results above seem to indicate that $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is another. Such systems can be used as a "logical laboratory:" what we learn from them using the plethora of techniques they offer can, in many cases, provide intuitions on restrictions and extensions. We saw this process in action during this chapter, as we were able to transfer certain results from $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ to extensions and sublanguages.

As we said above and as we will prove in detail in the next section, $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is undecidable, just like FO is. And in that respect, it is probably not adequate for applications requiring inference if we can do with a language of lower expressivity. But we conjecture that the intrinsic locality of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ would translate in more efficient decision methods than those available for FO. In particular, it would be interesting to extend the resolution method of Section 5.2.3 to full $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ and compare its performance with respect to, for example, a first-order resolution prover working on formulas from the bounded fragment.

Concerning languages weaker than $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ an interesting phenomena occurs. On the one hand, we have shown that they are extremely well behaved computationally as the presence of nominals and @ lets us define simple and elegant reasoning methods (both tableaux and resolution based), and they also show good behavior complexity wise as we will investigate in Chapter 7. But on the other hand, even though interpolation is usually taken as a sign of a balanced inference system, we have proved that this property usually fails for languages containing nominals but no binders. As we mentioned above, our conjecture is that this "expressivity gap" might be filled with the addition of counting operators.

Chapter 7

Mapping Out Complexity

Jack: [...] That, my dear Algy,
is the whole truth pure and simple.
Algernon: The truth is rarely pure and never simple.
Modern life would be very tedious if it were either,
and modern literature a complete impossibility!

from "The Importance of Being Earnest," Oscar Wilde

In this chapter we analyze the (un)decidability and complexity of a number of hybrid logics. We discuss the effects of extending the basic modal language with nominals and state variables, the $\langle R^{-1}\rangle$ modality, the @ operator, the existential modality E, the difference operator D, and the $\downarrow$ binder. In Chapter 4 we have already transferred some of these results into description logics. A new angle which we will be investigating here is the behavior of hybrid languages (and hence description languages) on classes different from K, the class of all models. In particular, we will discuss the behavior of hybrid languages containing the past operator $\langle R^{-1}\rangle$, on models which satisfy more natural conditions for time structures (i.e., transitivity, strict linearity and branching.) Analyzing complexity results on different classes of models comes naturally in the standard modal logic approach, and can be related to the work of Horrocks and Sattler [1999] on transitive roles in description languages.

We start by proving a sharp undecidability result for $\mathcal{H}_{\mathsf{S}}(\downarrow)$. This result is so general, that it calls for drastic restrictions on $\mathcal{H}_{\mathsf{S}}(\downarrow)$ to come back to decidability. We will prove that the satisfiability problem for formulas of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ where $\downarrow$ cannot be nested is decidable. We will then explore the landscape of logics around $\mathcal{H}_{\mathsf{S}}(@)$. We start by proving that local K-satisfiability for $\mathcal{H}_{\mathsf{S}}(@)$ is PSpace-complete. In other words, (up to a polynomial) there are *no extra computational costs* when expanding uni-modal logic (or even multi-modal logic) with @, nominals and free variables. And, as we discussed in Chapter 3, these extensions do increase the expressive power of the language. The complexity results for $\mathcal{H}_{\mathsf{S}}(@)$ contrast with the case for $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, @)$. Extending uni-modal temporal logic *with just one nominal* brings the complexity of local K-*Sat* to ExpTime-completeness. We will see that this ExpTime-completeness result over arbitrary frames can be tamed either by working with a more appropriate class of frames or by restricting to a fragment of the language. Along the way we make a detour through hybrid PDL: we establish upper bounds for a number of hybrid logics by generalizing results due to Passy and Tinchev [1991], and De Giacomo [1995].

We conclude by drawing attention to the *spypoint technique* from [Blackburn and Seligman, 1995], which we use to prove both upper and lower bounds. We believe this technique may be useful in other settings.

7.1 Into Undecidability and Back

Beautiful as $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is, it has a drawback: its *Sat* problem is undecidable [Blackburn and Seligman, 1995]. The culprit is the $\downarrow$ binder, @ is not needed. And the result of Blackburn and Seligman can be strengthened as follows.

Theorem 7.1. *The fragment of* $\mathcal{H}_{\mathsf{S}}(\downarrow)$ *consisting of pure nominal-free sentences has an undecidable local* K-*Sat problem.*

We begin by quickly sketching an easy undecidability proof for the full language $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. By generalizing the methods used in this simple proof, we will be led to the Spypoint Theorem (Theorem 7.9) and the undecidability result just stated.

Theorem 7.2. *Local* K-*Sat for* $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ *is undecidable.*

Proof. We will use Spaan's result [1993, Theorem 4.2.1] concerning the undecidability of global *Sat* for the class K_{23} of all modal frames in which every state has at most 2 R-successors and at most 3 two-step R-successors. We reduce the global *Sat* problem of this logic to $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. Our reduction uses the spypoint technique. Let *Grid* be the conjunction of the following formulas, for s an arbitrary nominal:

$$
\begin{array}{ll}
G_1 & @_s\neg\Diamond s \\
G_2 & @_s\Diamond\top \\
G_3 & @_s(\Box\Box{\downarrow}x.@_s\Diamond x) \\
G_4 & @_s(\Box{\downarrow}y.\Box{\downarrow}x_1.@_y\Box{\downarrow}x_2.@_y\Box{\downarrow}x_3.(@_{x_1}x_2 \vee @_{x_1}x_3 \vee @_{x_2}x_3)) \\
G_5 & @_s(\Box{\downarrow}y.\Box\Box{\downarrow}x_1.@_y\Box\Box{\downarrow}x_2.@_y\Box\Box{\downarrow}x_3.@_y\Box\Box{\downarrow}x_4.(\bigvee_{1\leq i\neq j\leq 4} @_{x_i}x_j)).
\end{array}
$$

The exact meaning of *Grid* will be discussed below in the proof of Claim 7.3, but intuitively, it is forcing the model to contain an irreflexive point s (the spypoint) which has full access to a grid (i.e., a frame in K_{23}). A warning, during this chapter we will sometimes denote a state in a model simply by a nominal naming it, as we just did.

Claim 7.3. For every formula φ, φ is globally satisfiable on a K_{23}-frame iff $\mathit{Grid} \wedge @_s\Box\varphi$ is satisfiable in some state of a hybrid model.

Proof of Claim. For the left to right direction, let $\mathcal{M} \Vdash \varphi$, where $\mathcal{M} = \langle M, R, V\rangle$ is an ordinary Kripke model. Define $\mathcal{M}'$ as follows: $M' = M \cup \{s\}$, $R' = R \cup \{(s, m) \mid m \in M\}$, $V' = V \cup \{(n, \{s\}) \mid \text{ for all nominals } n\}$. $\mathcal{M}'$ is a hybrid model, for all nominals (including s) are interpreted as the singleton set $\{s\}$, our spypoint. We claim that if $\mathcal{M} \Vdash \varphi$ then $\mathcal{M}, s \Vdash \mathit{Grid} \wedge @_s\Box\varphi$. Clearly, s satisfies G_1, G_2 and G_3. To prove G_4 and G_5, use the fact that $\mathcal{M}$ is based on a K_{23} frame. As φ is globally true at $\mathcal{M}$, s also satisfies $@_s\Box\varphi$.

For the other direction, suppose *Grid* is satisfied in a model $\mathcal{M}$. Then there exists a state named by s (the spypoint). By G_1, s is not related to itself. By G_2, s is related to some state, and by G_3, every state which can be reached from s in two steps can also be reached in one step. This means that in $\mathcal{M}_s$ — the submodel of $\mathcal{M}$ generated by s — every state is reachable from s in one step. Now G_4 and G_5 express precisely the two conditions characterizing the class K_{23} on one step successors of s. So, let $\mathcal{M}, w \Vdash \mathit{Grid} \wedge @_s\Box\varphi$, where $\mathcal{M} = \langle M, R, V\rangle$ is a hybrid model. Define $\mathcal{M}'$ as follows: $M' = \{m \in M \mid R(s,m)\}$, $R' = R_{\restriction M'}$, $V' = V_{\restriction M'}$. Then $\mathcal{M}'$ is based on a K_{23}-frame, and furthermore $\mathcal{M}' \Vdash \varphi$. ⊣

The theorem follows from the claim. QED

We are on our way to prove Theorem 7.1. We start by analyzing the previous proof and generalizing the underlying ideas. The models we used in the proof above had a certain characteristic form. Let's pin this down:

Definition 7.4. A model $\mathcal{M} = \langle M, R, V\rangle$ is called a *spypoint model* if there is an element $s \in M$ (the spypoint) such that $\neg R(s,s)$, and for all $w \in M$, if $w \neq s$, then $R(s,w)$ and $R(w,s)$.

Notice that any spypoint model is generated by its spy point. We will now show that with $\downarrow$ we can easily create spypoint models. On such models we can simulate the $@_x$ operator for every variable x introduced by $\downarrow x$. The following proposition is straightforward:

Proposition 7.5. *Let* $\mathcal{M} = \langle M, R, V\rangle$ *and* $s \in M$ *be such that* $\mathcal{M}, s \Vdash \downarrow s.(\neg\Diamond s \wedge\Box\Box\downarrow x.\Diamond(s \wedge \Diamond x) \wedge \Box\Diamond s)$. *Then*

i. The submodel $\mathcal{M}_s$ *generated by* s *is a spypoint model with* s *the spypoint.*

ii. $@_s\varphi$ *is definable on* $\mathcal{M}_s$ *by* $(s \wedge \varphi) \vee \Diamond(s \wedge \varphi)$.

iii. Let g *be any assignment. Then for all* $m \in M$, $\mathcal{M}_s, g, m \Vdash @_x\varphi$ *iff* $\mathcal{M}_s, g, m \Vdash @_s(\varphi \vee \Diamond(x \wedge \varphi))$.

Spypoint models are very powerful: we can encode lots of information about Kripke models inside a spypoint model. More precisely, for each Kripke model $\mathcal{M}$, we define the notion of *a spypoint model of* $\mathcal{M}$.

Definition 7.6. Let $\mathcal{M} = \langle M, R, V\rangle$ be a Kripke model in which the domain of V is a finite set $\{p_1, \ldots, p_n\}$ of propositional variables. The *spypoint model of* $\mathcal{M}$ (notation $\mathsf{Spy}[\mathcal{M}]$) is the structure $\langle M', R', V'\rangle$ in which

i. $M' = M \cup \{s\} \cup \{w_{p_1}, \ldots, w_{p_n}\}$, for $s, w_{p_1}, \ldots, w_{p_n} \notin M$,

ii. $R' = R \cup \{(s,x),(x,s) \mid x \in M'\backslash\{s\}\} \cup \{(x, w_{p_i}) \mid x \in M \text{ and } x \in V(p_i)\}$,

iii. $V' = \{\}$.

Let $\{s, x_{p_1}, \ldots, x_{p_n}\}$ be a set of state variables. A *spypoint assignment* for this set is an assignment g which sends s to the spypoint s and x_{p_i} to w_{p_i}. We use $\mathbf{m}$ as an abbreviation for $\neg s \wedge \neg x_{p_1} \wedge \ldots \wedge \neg x_{p_n}$. Note that when evaluated under the spypoint assignment, the denotation of $\mathbf{m}$ in $\mathsf{Spy}[\mathcal{M}]$ is precisely M.

$\mathsf{Spy}[\mathcal{M}]$ encodes the valuation on $\mathcal{M}$ and we can take advantage of this fact. Define the following translation from uni-modal formulas to hybrid formulas:

$$\begin{array}{lcl}
IT(p_i) & = & \Diamond(x_{p_i}) \\
IT(\neg\varphi) & = & \neg IT(\varphi) \\
IT(\varphi \wedge \psi) & = & IT(\varphi) \wedge IT(\psi) \\
IT(\Diamond\varphi) & = & \Diamond(\mathbf{m} \wedge IT(\varphi)).
\end{array}$$

Proposition 7.7. *Let $\mathcal{M}$ be a Kripke model and φ a uni-modal formula. Then for any spypoint assignment g, $\mathcal{M} \Vdash \varphi$ iff $\mathsf{Spy}[\mathcal{M}], g, s \Vdash \Box(\mathbf{m} \to IT(\varphi))$.*

Modify the hybrid translation HT given in Section 6.1.1 to its relativized version $HT^{\mathbf{m}}$ which also defines away occurrences of @. Let $HT^{\mathbf{m}}(\exists v.(R(t,v) \wedge \varphi))$ be $@_t\Diamond{\downarrow}v.(\mathbf{m} \wedge HT^{\mathbf{m}}\varphi)$ and replace @ by definition as in Proposition 7.5.*ii*) and *iii*).

The crucial step now is the fact that $\downarrow$ is strong enough to encode many frame-conditions.

Proposition 7.8. *Let $\mathcal{M} = \langle M, R, V\rangle$ be a Kripke model. Let $C(y)$ be a formula in the bounded fragment in the signature $\{R, =\}$. Then for any assignment g, $\langle M, R\rangle \Vdash \forall y.C(y)$ if and only if $\mathsf{Spy}[\mathcal{M}], g, s \Vdash \Box{\downarrow}y.(\mathbf{m} \to HT^{\mathbf{m}}(C(y)))$.*

Proof. Immediate by the properties of HT, Proposition 7.5, and the fact that the spypoint is R-related to all states in the domain of $\mathcal{M}$. QED

Theorem 7.9. [Spypoint theorem] *Let φ be a uni-modal formula in $\{p_1, \ldots, p_n\}$ and $\forall y.C(y)$ a first-order frame condition in $\{R, =\}$ with $C(y)$ in the bounded fragment. The following are equivalent.*

i. There is a Kripke model $\mathcal{M} = \langle M, R, V\rangle$ such that $\langle M, R\rangle \Vdash \forall y.C(y)$ and $\mathcal{M} \Vdash \varphi$.
ii. The following pure hybrid sentence F in the language $\mathcal{H}_{\mathcal{S}}(\downarrow)$ is satisfiable.

$$F := {\downarrow}s.(SPY \wedge \Diamond{\downarrow}x_{p_1}.@_s\Diamond{\downarrow}x_{p_2}@_s \ldots \Diamond{\downarrow}x_{p_n}.@_s(DIS \wedge VAL \wedge FR)),$$

where

$$\begin{aligned} SPY &= \neg\Diamond s \wedge \Box\Box{\downarrow}x.\Diamond(s \wedge \Diamond x) \wedge \Box\Diamond s \\ DIS &= \Box(\textstyle\bigwedge_{1\leq i\leq n}(x_{p_i} \to \bigwedge\{\neg x_{p_j} \mid 1 \leq j \neq i \leq n\})) \\ VAL &= \Box(\mathbf{m} \to IT(\varphi)) \\ FR &= \Box{\downarrow}y.(\mathbf{m} \to HT^{\mathbf{m}}(C(y)). \end{aligned}$$

Proof. The intuitions behind SPY, DIS, VAL and FR are as follows. SPY makes s a spypoint. DIS takes care that x_{p_i} and x_{p_j} do not hold at the same point of the model, for $i \neq j$. VAL makes φ globally true in $\mathbf{m}$. And FR forces the condition $C(y)$ in $\mathbf{m}$. The way we have written it, F contains occurrences of $@_s$ but this does not matter, by Proposition 7.5 all these occurrences can be defined away.

To prove $i) \Rightarrow ii)$, let $\mathcal{M}$ be a Kripke model as in i). We claim that $\mathsf{Spy}[\mathcal{M}], s \Vdash F$. The first conjunct of F is true in $\mathsf{Spy}[\mathcal{M}]$ at s by Proposition 7.5. The diamond part of the second conjunct can be satisfied using any spypoint assignment g. In the spypoint model all w_{p_i} are pairwise distinct, hence $\mathsf{Spy}[\mathcal{M}], g, s \Vdash DIS$. By Propositions 7.7 and 7.8, also $\mathsf{Spy}[\mathcal{M}], g, s \Vdash VAL \wedge FR$.

For the other direction, let $\mathcal{M}, s \Vdash F$. By Proposition 7.5, the submodel $\mathcal{M}_s = \langle M_s, R_s, V_s\rangle$ generated by s is a spypoint model. Let g be the assignment such that $\mathcal{M}, g, s \Vdash DIS \wedge VAL \wedge FR$. By DIS, $g(x_{p_i}) \neq g(x_{p_j})$ for all $i \neq j$, and (since $\neg R(s,s)$) also $g(x_{p_i}) \neq s$, for all i. Define the following Kripke model $\mathcal{M}' = \langle M', R', V'\rangle$, where $M' = M \setminus \{g(s), g(x_{p_1}), \ldots, g(x_{p_n})\}$, $R' = R\restriction_{M'}$ and $V'(p_i) = \{w \mid R(w, g(x_{p_i}))\}$. Note that $\mathsf{Spy}[\mathcal{M}']$ is precisely $\mathcal{M}_s$, and g is a spypoint assignment. But then by Propositions 7.7 and 7.8, and the fact that $\mathcal{M}_s, g, s \Vdash VAL \wedge FR$, we obtain $\mathcal{M}' \Vdash \varphi$ and $\langle M', R'\rangle \Vdash \forall y.C(y)$. QED

The proof of the claimed undecidability result is now straightforward.

Proof of Theorem 7.1. We reduce the undecidable problem of deciding global satisfiability in the uni-modal language over the class K_{23} as we did in Theorem 7.2. The first-order frame conditions defining K_{23} are of the form $\forall y.C(y)$ with $C(y)$ in the bounded fragment. Apply the Spypoint Theorem. The formula F (after all occurrences of $@_s$ have been term-defined) is a pure nominal-free sentence in $\mathcal{H}_{\mathsf{S}}(\downarrow)$. QED

The generality of the Spypoint Theorem can be interpreted as a sign that only very restricted forms of binding will preserve decidability. In the rest of the section, we focus on the fragment of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ containing only sentences where $\downarrow$ cannot be nested. We will prove that this fragment is indeed decidable.

Theorem 7.10. *The set of sentences of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ where $\downarrow$ appears non-nested has a decidable* K*-Sat problem.*

We need some preparation before embarking into the proof of Theorem 7.10. The following are properties of formulas in $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$.

Proposition 7.11.

i. For any $\mathcal{M}$, g, m and x not in φ: $\mathcal{M}, g, m \Vdash \varphi$ iff $\mathcal{M}, g^x_{i^{\mathcal{M}}}, m \Vdash \varphi[i/x]$.

ii. Let φ be a formula with no state variables. Then for any $\mathcal{M}$, m, $\mathcal{M}, m \Vdash i \wedge \varphi$ implies $\mathcal{M}, m \Vdash \downarrow x.\varphi[i/x]$.

iii. Let φ be a formula with only x free. Then for any $\mathcal{M}$,g, m, $\mathcal{M}, g^x_m, m \Vdash i \wedge \varphi$ implies $\mathcal{M}, m \Vdash \varphi[x/i]$.

Proof. *i*) is proved by induction on φ. To prove *ii*), suppose $\mathcal{M}, m \Vdash i \wedge \varphi$. Then by *i*), $\mathcal{M}, [x = i^{\mathcal{M}}], m \Vdash \varphi[i/x]$. But because $\mathcal{M}, m \Vdash i$, $i^{\mathcal{M}} = m$ and hence $\mathcal{M}, m \Vdash \downarrow x.\varphi[i/x]$. For *iii*), suppose $\mathcal{M}, g^x_m, m \Vdash i \wedge \varphi$. Because $m = i^{\mathcal{M}}$, $\mathcal{M}, g^x_{i^{\mathcal{M}}}, m \Vdash \varphi[x/i][i/x]$ and by *i*), $\mathcal{M}, g, m \Vdash \varphi[x/i]$. As x was the only free variable in φ we can drop g. QED

Proof of Theorem 7.10. We prove via filtrations that the fragment consisting of the non-nested sentences of $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ has the finite model property.

Let φ be a sentence in $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E}, @, \downarrow)$ without nested occurrences of $\downarrow$ and $\mathcal{M} = \langle M, R, V\rangle$ be a hybrid model over $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{NOM}, \mathsf{SVAR}\rangle$ such that $\mathcal{M}, m \Vdash \varphi$. We can assume that $\mathcal{M}$ is labeled, i.e., each state in M makes a nominal true.

Define the relation $\sim$ on M as $m \sim m'$ iff for all $\psi \in \mathsf{SF}(\varphi)$, $\mathcal{M}, [x = m], m \Vdash \psi \Leftrightarrow \mathcal{M}, [x = m], m' \Vdash \psi$. Let REP be a subset of M containing exactly one member of each equivalence class in $M/{\sim}$, and let LAB be the set of its labels ($\mathsf{LAB} \subseteq \mathsf{NOM}$). Define $\mathcal{M}^f = \langle M^f, R^f, V^f\rangle$ as follows, $M^f = \mathsf{REP}$, $R^f(m, m')$ iff for some $n \in |m|, n' \in |m'|, R(n, n')$, and $V(a) = \{m \in \mathsf{REP} \mid \mathcal{M}, m \Vdash a\}$ for $a \in \mathsf{PROP} \cup \mathsf{LAB}$. Notice that $\mathcal{M}^f$ is a finite labeled hybrid model over the signature $\langle \mathsf{REL}, \mathsf{PROP}, \mathsf{LAB}, \mathsf{SVAR}\rangle$.

Let $Cl(\varphi)$ be the smallest set containing $\{\varphi\} \cup \mathsf{LAB}$, closed under subformulas, single negation and the condition $\{\downarrow x.\psi, i\} \subseteq Cl(\varphi) \Rightarrow \psi[x/i] \in Cl(\varphi)$. We will prove that $\mathcal{M}^f$ satisfies the following truth lemma

Claim 7.12. If $\psi \in Cl(\varphi)$ is a sentence, then $\mathcal{M}^f, m \Vdash \psi$ iff $\mathcal{M}, m \Vdash \psi$.

Proof of Claim. The proof proceeds by induction on the complexity of ψ. The atomic case is by definition of V^f and the Boolean cases by induction hypothesis. The cases for the modalities, E and @ are straightforward. The only interesting case is $\psi = \downarrow x.\theta$.

[$\Rightarrow$]. Suppose $\mathcal{M}^f, m \Vdash \downarrow x.\theta$, then $\mathcal{M}^f, [x = m], m \Vdash \theta$. Let i be the nominal labeling m, then by Proposition 7.11.*iii*) $\mathcal{M}^f, m \Vdash \theta[x/i]$. Notice that $\theta[x/i]$ is a sentence in $Cl(\varphi)$, hence by inductive hypothesis $\mathcal{M}, m \Vdash \theta[x/i]$. Using Proposition 7.11.*ii*), we obtain $\mathcal{M}, m \Vdash \downarrow x.\theta[x/i][i/x]$ as needed.

[$\Leftarrow$] is proved similarly. $\dashv$

The theorem follows immediately from the claim. QED

7.2 A Note on Nominals, @, E and D

The existential modality E, and to a lesser extent the difference operator D, have played an important role in the development of hybrid languages. Note that in the presence of state variables E can mimic @, for $\mathsf{E}(s \wedge \varphi)$ means exactly the same thing as $@_s\varphi$. This is why practically everyone who has worked with nominals has also experimented with E. In a sense, @ amounts to a *guarded* use of E, where we mean “guarded” in the sense of [Andréka *et al.*, 1998], i.e., the scope of the quantifier introduced by E is restricted to the single point where the nominal holds. As we will see, this kind of guarding can be effective: hybrid logics with @ are often less complex than those which allow unrestricted use of E.

The difference operator D is stronger than E, for we can define $\mathsf{E}\varphi$ as $\varphi \vee \mathsf{D}\varphi$ but E is not strong enough to define D. Actually, D is so strong that it can even simulate nominals: clearly the formula $\mathsf{E}p \wedge \mathsf{A}(p \rightarrow \neg\mathsf{D}p)$ forces p to be true at only one point in the model. This is not a new observation, Gargov and Goranko raise the same point in [1993] and proof systems for D-logics based on Gabbay style rules trade on this [de Rijke, 1992; Venema, 1991, 1993]. The following fact is perhaps more surprising:

Theorem 7.13. *There is a polynomial reduction preserving satisfiability from any hybrid language containing* D *to the fragment containing only* E *and nominals.*

Proof. Let φ be a formula in the full language. In two steps, we construct a formula $\varphi' \wedge \theta'$ without D such that φ is satisfiable iff $\varphi' \wedge \theta'$ is. We take care that this construction can be performed in time polynomial in $|\varphi|$. We use the fact that in any Kripke model $\mathcal{M}$, the denotation $[\mathsf{D}\varphi]_{\mathcal{M}} = \{m \in M \mid \mathcal{M}, m \Vdash \mathsf{D}\varphi\}$ of $\mathsf{D}\varphi$ can only take three values, namely:

$$[\mathsf{D}\varphi]_{\mathcal{M}} = \begin{cases} M & \text{if } |[\varphi]_{\mathcal{M}}| > 1 \\ \{\} & \text{if } [\varphi]_{\mathcal{M}} = \{\} \\ M \setminus \{m\} & \text{if } [\varphi]_{\mathcal{M}} = \{m\}. \end{cases}$$

We now delete all occurrences of D, replacing them with nominals and A. We proceed inductively in the number of D operators in φ. If φ contains no D we are done. Otherwise, consider a subformula of the form $\mathsf{D}\psi$ where ψ contains no occurrences of D. Let φ' be φ with this subformula replaced by a new variable p_k and let $\theta_k = \mathsf{A}(p_k \leftrightarrow \mathsf{D}\psi)$. Clearly

φ is satisfiable iff $\varphi' \wedge \theta_k$ is. An inductive application of this procedure eventually yields a formula φ' without occurrences of D and a conjunction θ of formulas of the form $\mathsf{A}(p_k \leftrightarrow \mathsf{D}\psi)$ with D not in ψ.

Now we "axiomatize" all the $\mathsf{D}\psi$ using nominals, E, and A. For every conjunct $\theta_k = \mathsf{A}(p_k \leftrightarrow \mathsf{D}\psi)$, we create a formula θ'_k which is the conjunction of

$$\begin{array}{l}
\mathsf{A}p_k \vee \mathsf{A}\neg p_k \vee (\mathsf{A}(p_k \leftrightarrow \neg i_k) \wedge \mathsf{E}p_k)\\
\mathsf{A}p_k \rightarrow \mathsf{E}(\psi \wedge i_k) \wedge \mathsf{E}(\psi \wedge \neg i_k)\\
\mathsf{A}\neg p_k \rightarrow \mathsf{A}\neg\psi\\
(\mathsf{A}(p_k \leftrightarrow \neg i_k) \wedge \mathsf{E}p_k) \rightarrow \mathsf{A}(\psi \rightarrow i_k).
\end{array}$$

For every θ_k, we use a new nominal i_k. Let θ' be $\bigwedge \theta'_k$. It is clear that $\varphi' \wedge \theta$ is satisfiable iff $\varphi' \wedge \theta'$ is, and θ' contains no occurrences of D. The translation produces at most a quadratic blow-up in the size of the formula. QED

Corollary 7.14. *Let* F *be any class of Kripke frames, and let* $\mathcal{L}$ *be a modal language. Then*

i. if $\mathcal{L}$ *contains* D *then adding nominals, @, and* E *modifies the complexity of* F-*Sat by at most a polynomial,*

ii. if $\mathcal{L}$ *contains nominals and* E *then adding* D *modifies the complexity of* F-*Sat by at most a polynomial.*

7.3 Restricting the Language

For most of this and the next sections we will study local K-satisfiability problems. Note that as far as local satisfiability problems are concerned, if we replace all state variables in φ by nominals, obtaining φ', then φ is satisfiable if and only if φ' is satisfiable. For this reason we can restrict ourselves to formulas without variables in the proofs, and we won't need to mention variable assignments. However at the end of Section 7.3.1 we examine the complexity of the *global* K-satisfiability problem for $\mathcal{H}_\mathsf{S}(@)$ and an interesting difference between nominals and state variables emerges. Also, if a language without state variables contains the E operator then the local and global satisfiability problems collapse into the same. In other words, complexity results for local *Sat* transfer from $\mathcal{H}_\mathsf{N}$ to $\mathcal{H}_\mathsf{S}$ and if the language contains E then complexity for the local *Sat* transfers to global *Sat* for languages without state variables.

7.3.1 Around $\mathcal{H}_\mathsf{S}(@)$

$\mathcal{H}_\mathsf{S}(@)$ is a very interesting sublanguage of $\mathcal{H}_\mathsf{S}(@, \downarrow)$. As we have seen in Section 6.2, although $\mathcal{H}_\mathsf{S}(@)$ does not enjoy strong interpolation, it does have weak interpolation. Moreover, as is shown in [Blackburn, 2000a], simple tableaux and sequent systems for $\mathcal{H}_\mathsf{S}(@)$ can be defined by exploiting the interplay between nominals (or free variables) and @; the underlying ideas trace back to [Seligman, 1991]. Furthermore, $\mathcal{H}_\mathsf{S}(@)$ provides new expressivity at the level of frames: we can define many properties that are not definable in ordinary propositional modal logic. Moreover, pure formulas automatically yield complete axiomatizations for the frame classes they define.

Thus there are many reasons for being interested in $\mathcal{H}_{\mathsf{S}}(@)$, and a natural question to ask is: how high a computational price do we pay for these benefits? It turns out that (up to a polynomial) there is *no extra computational costs.*

Theorem 7.15. *Local* K*-Sat for* $\mathcal{H}_{\mathsf{S}}(@)$ *is* PSpace*-complete.*

Proof. The lower bound follows from [Ladner, 1977]. We show the upper bound by defining the notion of a ξ-game between two players. We will show that the existential player has a winning strategy for the ξ-game if and only if ξ is satisfiable. Moreover every ξ-game stops after at most as many rounds as the modal depth of ξ and the information on the playing board is polynomial in the length of ξ. Using the close correspondence between alternating Turing machines and two player games [Chlebus, 1986], it is straightforward to implement the problem of whether the existential player has a winning strategy in the ξ-game on a P alternating Turing machine. This in turn can be transformed into a PSpace Turing machine. We present the proof only for uni-modal $\mathcal{H}_{\mathsf{S}}(@)$; it can be straightforwardly extended to the multi-modal case.

Fix a formula ξ. A ξ-Hintikka set is a maximal consistent set of subformulas of ξ. The ξ-game is played as follows. There are two players, ∀belard (male) and ∃loise (female). She starts the game by playing a collection $\{X_0, \ldots, X_k\}$ of Hintikka sets and specifying a relation R on them.

∃loise loses immediately if she cannot meet one of the following conditions:

i. $\xi \in X_0$, and all others X_l contain at least one nominal occurring in ξ,

ii. no nominal occurs in two different Hintikka sets,

iii. for all X_l, for all $@_i\varphi \in \mathsf{SF}(\xi)$, $@_i\varphi \in X_l$ iff $\{i, \varphi\} \subseteq X_k$, for some k,

iv. for all $\Diamond\varphi \in \mathsf{SF}(\xi)$, if $R(X_l, X_k)$ and $\Diamond\varphi \notin X_l$, then $\varphi \notin X_k$.

Now ∀belard may choose an X_l and a "defect-formula" $\Diamond\varphi \in X_l$. ∃loise must respond with a Hintikka set Y such that

i. $\varphi \in Y$ and for all $\Diamond\psi \in \mathsf{SF}(\xi)$, $\Diamond\psi \notin X_l$ implies that $\psi \notin Y$,

ii. for all $@_i\varphi \in \mathsf{SF}(\xi)$, $@_i\varphi \in Y$ iff $\{i, \varphi\} \subseteq X_k$, for some k,

iii. if $i \in Y$ for some nominal i, then Y is one of the Hintikka sets she played at the start. In this case the game stops and ∃loise wins.

If ∃loise cannot find a suitable Y, the game stops and ∀belard wins. If ∃loise does find a suitable Y (one that is not covered by the halting clause in item *iii*) above) then Y is added to the list of played sets, and play continues.

∀belard must now choose a defect $\Diamond\varphi$ from the last played Hintikka set with the following restriction: in round k he can only choose defects $\Diamond\varphi$ such that the modal depth of $\Diamond\varphi$ is less than or equal to the modal depth of ξ minus k. ∃loise must respond as before. She wins if she can survive all his challenges (in other words, he loses if he reaches a situation where he can't choose any more defects).

Clearly the ξ-game stops after at most modal depth of ξ many rounds. At any stage of the game, the size of the information on the board is at most polynomial in the length of ξ. We claim that ∃loise has a winning strategy iff ξ is satisfiable.

The right-to-left direction is clear: ∃loise has a winning strategy if ξ is satisfiable, for she need simply play by reading the required Hintikka sets off the model. The other direction requires more work. Suppose ∃loise has a winning strategy for the ξ-game. We will create a model $\mathcal{M}$ for ξ as follows. The domain M is build in steps by following her winning strategy. M_0 consists of her initial move $\{X_0, \ldots, X_n\}$. Suppose M_j is defined. Then M_{j+1} consists of a copy of those Hintikka sets she plays when using her winning strategy for each of ∀belard's possible moves played in the Hintikka sets from M_j (except when she plays a Hintikka set from her initial move, then of course we do not make a copy). Let M be the disjoint union of all M_j for j smaller than the modal depth of ξ. Set $R(m, m')$ iff for all $\Diamond\varphi \in \mathsf{SF}(\xi)$, $\Diamond\varphi \notin m$ implies $\varphi \notin m'$; and put $V(p) = \{m \in M \mid p \in m\}$. Note that the rules of the game guarantee that nominals are interpreted as singletons. The following truth-lemma holds:

Claim 7.16. For all $m \in M$ which ∃loise plays in round j (i.e., $m \in M_j$), for all φ of modal depth less than or equal to the modal depth of ξ minus j, $\mathcal{M}, m \Vdash \varphi$ iff $\varphi \in m$.

Proof of Claim. By induction, the cases for atoms, Booleans and @ are simple. For $\Diamond$, if $\Diamond\varphi \in m$, then ∀belard challenged this defect, so ∃loise could respond with an m' containing φ. Since for all $\Diamond\varphi \in \mathsf{SF}(\xi)$, $\Diamond\varphi \notin m \Rightarrow \varphi \notin m'$ holds, we have $R(m, m')$ and by induction hypothesis $\mathcal{M}, m \Vdash \Diamond\varphi$. If $\Diamond\varphi \notin m$ but $R(m, m')$ holds, then by our definition of R, $\varphi \notin m'$, so again $\mathcal{M}, m \nVdash \Diamond\varphi$. ⊣

Since she plays a Hintikka set containing ξ in the first round, $\mathcal{M}$ satisfies ξ. QED

Theorem 7.15 shows that for arbitrary frames, the guarding strategy (that is, using @ instead of E) pays off: adding the existential modality to ordinary uni-modal logic results in an ExpTime-complete satisfiability problem (see [Halpern and Moses, 1992] and [Spaan, 1993]).

But what about the *global* K-*Sat* problem? ExpTime-hardness for $\mathcal{H}_{\mathsf{N}}(@)$ obtains as a corollary of Spaan's proof [1993] of the ExpTime-completeness of the local K-*Sat* problem of uni-modal logic plus the existential modality. A matching upper bound follows from Corollary 7.21 below. Interestingly, things are very different if we expand the uni-modal language with *variables* instead of nominals.

Recall from Section 3.3 that $\mathcal{M} \Vdash \varphi$ is defined to hold iff for all g, $\mathcal{M}, g \Vdash \varphi$. That is, g is *not* held constant. Now, if there are no state variables in φ, g is irrelevant and global satisfiability is ExpTime-complete. But if φ is allowed to contain free variables, the implicit universal quantification in the definition certainly does change matters. In effect, we are surreptitiously using the ∀ quantifier, and we wind up with an undecidable global *Sat* problem.

Theorem 7.17. *The global* K-*Sat problem of* $\mathcal{H}_{\mathsf{S}}(@)$ *is undecidable.*

Proof. Let φ be the formula $(\Diamond x_1 \wedge \Diamond x_2 \wedge \Diamond x_3) \rightarrow \Box(x_1 \leftrightarrow x_2 \vee x_1 \leftrightarrow x_3 \vee x_2 \leftrightarrow x_3)$. Then for every model $\mathcal{M}$, $\mathcal{M} \Vdash \varphi$ iff $\mathcal{M} \Vdash \forall y x_1 x_2 x_3.((Ryx_1 \wedge Ryx_2 \wedge Ryx_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3))$. I.e., φ expresses that every world has at most two R-successors. Similarly we can create a formula without propositional variables expressing that every world has at most three two-step R-successors. Hence we can enforce K_{23} models and Spaan's undecidability result applies. QED

Note that this argument makes no use of @, hence undecidability actually obtains for $\mathcal{H}_{\mathsf{S}}$. And the result can be sharpened even further: if the language contains at least three modalities, then we obtain an undecidable global consequence problem for the sublanguage containing only one propositional variable; again, no use is made of @.

7.3.2 Around $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, @)$

The addition of nominals in temporal logics behaves quite differently than for the modal language. In fact, expanding the basic tense logic with even a single nominal or free variable leads to an exponential increase in complexity for the local satisfiability problem (assuming PSPACE ≠ EXPTIME).

THEOREM 7.18. *Local* K-*Sat for* $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ *with at least one nominal is* EXPTIME-*hard.*

PROOF. We will reduce the EXPTIME-complete global K-*Sat* problem for uni-modal languages to the local K-*Sat* problem for $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ containing at least one nominal s which we will use as our spypoint.

Define the following translation function $\cdot^t$: $p^t = p$, $(\neg\varphi)^t = \neg\varphi^t$, $(\varphi \wedge \psi)^t = \varphi^t \wedge \psi^t$, $(\Diamond\varphi)^t = \langle R\rangle(\langle R^{-1}\rangle s \wedge \varphi^t)$. Note that s is a *fixed* nominal in this translation. Clearly $\cdot^t$ is a linear reduction. We claim that for any formula φ, φ is globally K-satisfiable if and only if $s \wedge [R](\langle R^{-1}\rangle s \rightarrow \varphi^t)$ is K-satisfiable.

For the left to right direction, let $\mathcal{M} \Vdash \varphi$, where $\mathcal{M} = \langle M, R, V\rangle$ is an ordinary Kripke model. Define $\mathcal{M}'$ as follows: $M' = M \cup \{s\}$, $R' = R \cup \{(s, m) \mid m \in M\}$, $V' = V \cup \{(n, \{s\}) \mid \text{ for all nominals } n\}$. We claim that for all $m \in M$, for all ψ, we have $\mathcal{M}, m \Vdash \psi$ if and only if $\mathcal{M}', m \Vdash \psi^t$. This follows by a simple induction. The only interesting step is for $\Diamond$:

$$\begin{aligned}
&\mathcal{M}, m \Vdash \Diamond\psi\\
\Leftrightarrow\ &\exists m' \in M.(R(m, m') \;\&\; \mathcal{M}, m' \Vdash \psi)\\
\Leftrightarrow\ &\exists m' \in M'.(R'(m, m') \;\&\; \mathcal{M}', m' \Vdash \psi^t \;\&\; R'(s, m') \text{ (by IH and definition of } R')\\
\Leftrightarrow\ &\mathcal{M}', m \Vdash \langle R\rangle(\langle R^{-1}\rangle s \wedge \psi^t)\\
\Leftrightarrow\ &\mathcal{M}', m \Vdash (\Diamond\psi)^t.
\end{aligned}$$

It follows that $\mathcal{M}', s \Vdash s \wedge [R](\langle R^{-1}\rangle s \rightarrow \varphi^t)$, as desired.

For the other direction, let $\mathcal{M}, w \Vdash s \wedge [R](\langle R^{-1}\rangle s \rightarrow \varphi^t)$, where $\mathcal{M} = \langle M, R, V\rangle$ is a hybrid model. Define $\mathcal{M}'$ as follows: $M' = \{m \in M \mid R(w, m)\}$, $R' = R_{\restriction M'}$, $V' = V_{\restriction M'}$. We claim that for all $m \in M'$, for all ψ, $\mathcal{M}, m \Vdash \psi^t$ if and only if $\mathcal{M}', m \Vdash \psi$. Again we only present the inductive step for $\Diamond$:

$$\begin{aligned}
&\mathcal{M}, m \Vdash \langle R\rangle(\langle R^{-1}\rangle s \wedge \psi^t)\\
\Leftrightarrow\ &\exists m' \in M.(R(m, m') \;\&\; R(w, m') \;\&\; \mathcal{M}, m' \Vdash \psi^t)\\
\Leftrightarrow\ &\exists m' \in M'.(R(m, m') \;\&\; R(w, m') \;\&\; \mathcal{M}, m' \Vdash \psi^t)\\
\Leftrightarrow\ &\exists m' \in M'.(R'(m, m') \;\&\; \mathcal{M}', m' \Vdash \psi) \text{ (by IH and definition of } M')\\
\Leftrightarrow\ &\mathcal{M}', m \Vdash \Diamond\psi.
\end{aligned}$$

For all $m \in M'$, $R(w, m)$ holds and then $\mathcal{M}, m \Vdash \langle R^{-1}\rangle s$. So, since also $\mathcal{M}, w \Vdash [R](\langle R^{-1}\rangle s \rightarrow \varphi^t)$, we have for all $m \in M'$, $\mathcal{M}, m \Vdash \varphi^t$. Hence $\mathcal{M}' \Vdash \varphi$. QED

We can use similar ideas to provide a lower bound for satisfiability of sentences with non-nested $\downarrow$ in $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$. Define $(\Diamond\varphi)^t$ as $\langle R\rangle(\langle Q\rangle s \wedge {\downarrow}x.@_s\langle Q\rangle x \wedge \varphi^t)$. Then again

Proposition 7.19. *A basic modal formula φ is globally satisfiable iff $s \wedge [Q](\langle Q\rangle s \rightarrow \varphi^t)$ is locally satisfiable.*

Hence the local satisfiability problem of the non-nested downarrow fragment of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is ExpTime-hard. Even though we used two modalities in the encoding above, we can do with only one by using the ExpTime-hard global satisfiability problem of the basic modal language over symmetric frames.

We should now provide a matching upper bound for Theorem 7.18, but in fact we will prove a stronger result. Even though the addition of *just one nominal* to the basic tense language yields an ExpTime-hard local K-*Sat*, adding further nominals, multiple forwards and backwards looking modalities, and even the existential modality, does not take us any higher in the complexity hierarchy.

We will establish this by extending known results for nominal PDL. It is known that the local satisfiability problem of nominal PDL enriched with E is solvable in ExpTime [Passy and Tinchev, 1991]. Moreover, De Giacomo's results [1995] on PDL-like description languages containing the $\mathcal{O}$ operator show that the satisfiability problem for nominal PDL with converse is solvable in ExpTime. On *connected* frames — assuming a finite repertoire of atomic programs — the existential modality is definable in converse PDL. But to establish the upper bounds we want, we need to know that we can have access to both *converse programs and* E on arbitrary frames and still stay in ExpTime. And in fact, we can. Once again, we make use of a spypoint argument, but this time to obtain an upper bound.

Theorem 7.20. *Local* K-*Sat for nominal PDL with converse and the existential modality is solvable in* ExpTime.

Proof. Let ξ be a formula in this language. Without loss of generality we may assume that the converse operator is only applied to atomic programs. We will transform ξ into a formula without occurrences of E and then use De Giacomo's result. Let s be a nominal and σ be an atomic program not occurring in ξ. Define ξ^t by recursively replacing every occurrence of $\mathsf{E}\varphi$ in ξ by $\langle\sigma^{-1}\rangle(s \wedge \langle\sigma\rangle\varphi)$, obtaining a formula without occurrences of E. Now transform the programs occurring inside the modalities in ξ by replacing atomic programs p (converse programs p^{-1}) by $p; \langle\sigma^{-1}\rangle s?$ (respectively $p^{-1}; \langle\sigma^{-1}\rangle s?$). We claim that ξ is satisfiable iff $\xi^t \wedge \langle\sigma^{-1}\rangle s$ is satisfiable. Since $\xi^t \wedge \langle\sigma^{-1}\rangle s$ is in the language covered by De Giacomo's ExpTime-algorithm, this claim proves the theorem.

The left to right direction of the claim is obvious: just add a new state to the model, make s true there, and let that state be σ-connected to all other states.

For the other direction, let $\mathcal{M} = \langle M, \{R_i\}, V\rangle$ satisfy $\xi^t \wedge \langle\sigma^{-1}\rangle s$ at w and let $\mathbf{s}$ be the denotation of s. Let $\mathcal{M}'$ be the submodel of $\mathcal{M}$ obtained by restricting the universe M to the set $M' = \{x \in M \mid R_\sigma(\mathbf{s}, x)\}$. We claim that for all subformulas ψ of ξ, for all $x \in M'$, $\mathcal{M}', x \Vdash \psi \Leftrightarrow \mathcal{M}, x \Vdash \psi$.

As $w \in M'$, this provides us with the desired result. The proof of the claim goes by the usual double induction needed for inductive proofs in PDL. The proof is straightforward given the following observations: the translation of E formulas works because we restricted the model to successors of the spypoint **s**; and $\mathcal{M}'$ is a *generated* submodel of $\mathcal{M}$ for the new "atomic" programs $p; \langle\sigma^{-1}\rangle s?$ and $p^{-1}; \langle\sigma^{-1}\rangle s?$ occurring in ξ^t. QED

COROLLARY 7.21. *Local* K-*Sat for* $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E})$ *is* EXPTIME-*complete.*

PROOF. Define a linear translation $\cdot^t$ of formulas of nominal tense logic (without @) into formulas of nominal PDL with converse as follows: $(a)^t{=}a$ for all atoms a, $(\neg\xi)^t = \neg(\xi)^t$, $(\xi \wedge \theta)^t = (\xi)^t \wedge (\theta)^t$, $(\langle R\rangle\xi)^t = \langle r\rangle(\xi)^t$, $(\langle R^{-1}\rangle\xi)^t = \langle r^{-1}\rangle(\xi)^t$, and $(\mathsf{E}\xi)^t = \mathsf{E}(\xi)^t$. Here r is a fixed atomic program. It is easy to see that φ is satisfiable iff $(\varphi)^t$ is satisfiable. QED

7.4 Restricting the Class of Frames

The EXPTIME-complete result for $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ sounds like bad news, but perhaps we can do better. $\langle R^{-1}\rangle$ usually receives a temporal interpretation (we have ourselves been referring to temporal and tense logics in the previous section). So we should investigate the behavior of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ of frames which are "time like" and not just on any arbitrary frame. If we think of the states of a Kripke model as *time points*, view R as the *temporal precedence* (or *earlier-than/later-than*) relation and read $\langle R^{-1}\rangle\varphi$ as "*φ occurs in the past*," then we should require R to be (at least) *transitive*. We will now examine the complexity of hybrid logics over frame classes that are relevant for temporal logic, like strict partial orders (linear time) and transitive trees (branching time).

7.4.1 Transitive Frames

We start with hybrid languages without $\langle R^{-1}\rangle$. We know from [Ladner, 1977] that the local *Sat* problem for ordinary uni-modal logic over transitive frames is PSPACE-complete. What happens when we add nominals and @? Again, nothing. In fact we can even add the existential modality E while staying in PSPACE.

THEOREM 7.22. *Sat over transitive frames for* $\mathcal{H}_{\mathsf{N}}(@, \mathsf{E})$ *is* PSPACE-*complete.*

PROOF. Given Ladner's result, we only need to provide an upper bound. We only consider formulas without occurrences of @ as this operator can be defined away using E: $@_i\varphi$ is equivalent to $\mathsf{E}(i\wedge\varphi)$. The proof will be similar to Theorem 7.15. For a formula ξ we will define a two player ξ-game. The ξ-game is designed so that it halts after at most $|\mathsf{SF}(\xi)|$ rounds. Moreover, at each stage of the game at most $|\mathsf{SF}(\xi)|$ Hintikka sets are on the board.

Fix a formula ξ. The ξ-game is played as follows. ∃loise starts by playing a collection $\{X_0, \ldots, X_k\}$ of Hintikka sets. She must now choose Hintikka sets so that the following three conditions hold:

i. $\xi \in X_0$, and $|\{X_0, \ldots, X_k\}|$ is smaller than $|\mathsf{SF}(\xi)|$,

ii. no nominal occurs in two different Hintikka sets,

iii. for all X_l, for all $\mathsf{E}\varphi \in \mathsf{SF}(\xi)$, $\mathsf{E}\varphi \in X_l$ iff $\varphi \in X_k$, for some k.

If ∃loise cannot find Hintikka sets satisfying these conditions, she loses the game immediately. If she can, the game continues. ∀belard chooses an X_l and a "defect formula" $\Diamond\varphi \in X_l$. ∃loise must respond with a Hintikka set Y such that

i. $\varphi \in Y$ and for all $\Diamond\psi \in \mathsf{SF}(\xi)$, $\Diamond\psi \notin X_l$ implies that $\Diamond\psi \notin Y$ and $\psi \notin Y$,

ii. for all $\mathsf{E}\varphi \in \mathsf{SF}(\xi)$, if $\varphi \in Y$ then $\mathsf{E}\varphi \in Y$, and $\mathsf{E}\varphi \in Y$ iff $\mathsf{E}\varphi \in X_l$,

iii. if $i \in Y$ for some nominal i, then Y is one of the Hintikka sets she played at the start. In this case the game stops and ∃loise wins.

If ∃loise cannot find a suitable Y, the game stops and ∀belard wins. If ∃loise does find a suitable Y (one that is not covered by the halting clause in item *iii*) above) then Y is added to the list of played sets, and play continues. At each stage, ∀belard must choose a defect $\Diamond\varphi$ from the last played Hintikka set. To ensure that the the length of the game is bounded by $|\mathsf{SF}(\xi)|$, we keep a list of the $\Diamond$-formulas ∀belard plays, and we insist that if he plays a formula $\Diamond\varphi$ a second time, ∃loise has to respond with the Hintikka set she played when he challenged with $\Diamond\varphi$ the first time. If this (forced) response does not meet the three criteria just listed, she loses; but if it does meet these criteria, she wins. Either way, the game stops immediately.

Claim 7.23. ∃loise has a winning strategy in the ξ-game iff ξ is satisfiable in the class of transitive hybrid models.

Proof of Claim.
[⇒]. Suppose ∃loise has a winning strategy for the ξ-game. We build a model $\mathcal{M}$ for ξ as follows. The domain M is built as in Theorem 7.15. Set $R(m, m')$ iff for all $\Diamond\varphi \in \mathsf{SF}(\xi)$, $\Diamond\varphi \notin m \Rightarrow [\Diamond\varphi \notin m' \;\&\; \varphi \notin m']$. Let $V(a) = \{m \in M \mid a \in m\}$, for all atoms a in $\mathsf{SF}(\xi)$; and if i is any nominal *not* in $\mathsf{SF}(\xi)$, then $V(i)$ is an arbitrary singleton subset of M. Clearly $\mathcal{M}$ is a transitive hybrid model. The following truth-lemma holds is easy to prove: for all $\psi \in SF(\xi)$, for all $m \in M$, $\mathcal{M}, m \Vdash \psi$ if and only if $\psi \in m$.

[⇐]. Suppose ξ is satisfiable. That is, suppose there is some hybrid model $\mathcal{M}$ and a point m_0 such that $\mathcal{M}, m_0 \Vdash \xi$. We define the model $\mathcal{M}^f = \langle M^f, R^f, V^f\rangle$ as the transitive filtration of $\mathcal{M}$ through $\mathsf{SF}(\xi)$. I.e., or each state $m \in \mathcal{M}$, let $|m|$ be the set of all states in $\mathcal{M}$ that agree on all formulas in $\mathsf{SF}(\xi)$. Then,

i. $M^f = \{|m| \mid \text{ is a state in } \mathcal{M}\}$,

ii. $R^f(|m|, |m'|)$ iff for all ψ, if $\Diamond\psi \in \mathsf{SF}(\xi)$ and $\mathcal{M}, m \not\Vdash \Diamond\psi$, then $\mathcal{M}, m' \not\Vdash \psi$, and $\mathcal{M}, m' \not\Vdash \Diamond\psi$.

iii. $V^f(a) = \{|m| \mid \mathcal{M}, m \Vdash a\}$, for all atomic symbols $a \in \mathsf{SF}(\xi)$, and V^f assigns arbitrary singletons to nominals not in $\mathsf{SF}(\xi)$.

Then $\mathcal{M}^f$ is a *finite* model, $\mathcal{M}^f, |m_0| \Vdash \xi$, and moreover each state in $\mathcal{M}^f$ is a Hintikka set over ξ. Think of ∃loise as consulting $\mathcal{M}^f$ as she plays, and choosing her moves from its states. For her first move, ∃loise chooses $|m_0|$ (as this is a point in $\mathcal{M}^f$ that contains ξ), and each state of $\mathcal{M}^f$ that contains a nominal and for every $\mathsf{E}\psi \in SF(\xi)$, if ψ is satisfied in $\mathcal{M}^f$, just one $|m|$ such that $\mathcal{M}^f, |m| \Vdash \psi$. Clearly her first move satisfies the required conditions. Now for the crucial point: when ∀belard chooses a defect $\Diamond\varphi$ from a Hintikka set X, ∃loise responds with a *maximal* R^f-successor Y of X such that Y contains φ; as $\mathcal{M}^f$ is finite, such a Y exists. It is this choice that enables ∃loise to successfully play the same Hintikka set twice if ∀belard chooses a defect $\Diamond\varphi$ twice. For suppose ∃loise has played Y in response to a defect $\Diamond\varphi$ in X. Then suppose that at some later stage ∀belard points to the defect $\Diamond\varphi$ in Z, for Z a successor of X. ∃loise consults the model, looking for a maximal R^f successor of Z that contains φ. But as $R^f(X, Z)$, any such point will also be a maximal R^f successor of X that contains φ. Thus Y is a suitable choice, and by playing it again she wins immediately. Thus ∃loise has a winning strategy for the ξ-game. This completes the proof of the right to left direction of our claim. ⊣

The theorem follows directly from the claim. QED

So far, so good: nothing strange happens when we add transitivity to the basic hybrid language. But as we will now show, transitivity is not enough to tame hybridization when the backward looking modality $\langle R^{-1}\rangle$ is present.

Theorem 7.24. *Local Sat for transitive frames in $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ is* ExpTime*-hard.*

Proof. We will reduce the ExpTime-complete global K-*Sat* problem for uni-modal languages to the local *Sat* problem over transitive frames of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ with at least one nominal. We assume without loss of generality that the propositional variables in the hybrid language are those of the uni-modal language plus four extra propositional variables 0, 1, 2, and 3. Define the translation $\cdot^t$ as follows:

$$\begin{array}{rcl} p^t & = & p \\ (\neg\varphi)^t & = & \neg\varphi^t \\ (\varphi \wedge \psi)^t & = & \varphi^t \wedge \psi^t \\ (\Diamond\varphi)^t & = & \langle R\rangle(1 \wedge \langle R^{-1}\rangle(2 \wedge \langle R\rangle(3 \wedge \langle R^{-1}\rangle(0 \wedge \langle R^{-1}\rangle i \wedge \varphi^t)))). \end{array}$$

Clearly $\cdot^t$ is a linear reduction. The intuition behind this translation is to mimic one R-step in an ordinary Kripke frame by a zigzag transition in a transitive hybrid frame.

$$\begin{tikzcd} & w_1 \Vdash 1 & & w_3 \Vdash 3 & \\ w \Vdash 0 \arrow[ur] & & w_2 \Vdash 2 \arrow[ul] \arrow[ur] & & w' \Vdash 0 \arrow[ul] \end{tikzcd}$$

The picture above shows how an arrow from w to w' in the original model would be encoded in the transitive model, with intermediate stops at w_1, w_2 and w_3. The propositional symbol 0 will mark the elements of the original model, the others are auxiliary in the encoding.

Claim 7.25. For any uni-modal formula φ, φ is globally K-satisfiable iff $i \wedge \langle R\rangle 0 \wedge [R](0 \rightarrow \varphi^t)$ is satisfiable in a hybrid model based on a transitive frame.

Proof of Claim.
[⇒]. Suppose $\mathcal{M} \Vdash \varphi$, where $\mathcal{M} = \langle M, R, V\rangle$ is a Kripke model. We now define a transitive hybrid model $\mathcal{M}' = \langle M', R', V'\rangle$ as follows:

- $M' = (M \times \{0,1,2,3\}) \cup \{s\}$, for $s \notin M \times \{0,1,2,3\}$. For $x \in \{0,1,2,3\}$, M^x is $\{(m,x) \mid m \in M\}$. In what follows, we will use the notation m_x instead of (m,x), $x \in \{0,1,2,3\}$. Points in M^0 will correspond to the points in the Kripke model $\mathcal{M}$; points in M^1, M^2 and M^3 are a "coding space" in which we can construct the transitive relation we require. The point s is our spypoint.
- R' is defined as follows

$$R'(a,b) \iff \begin{cases} a = x_0, b = x_3 & \text{for some } x \in M, \text{ or} \\ a = x_2, b = x_3 & \text{for some } x \in M, \text{ or} \\ a = x_2, b = x_1 & \text{for some } x \in M, \text{ or} \\ a = y_0, b = x_1 & \text{for some } y, x \in M \;\&\; R(y,x), \text{ or} \\ a = s & \end{cases}$$

  R' codes an R-transition from point y to x in $\mathcal{M}$ as a forward step from y_0 to x_1 followed by a backward, forward, backward sequence from x_1 to x_3. This zig-zag sequence corresponds to the nesting pattern modalities in the translation clause for $\Diamond$. We leave the reader to verify that R' is a *transitive* relation. Note that the spypoint can see every point in the model.
- If p is a propositional variable not in $\{1,2,3,4\}$, then $V'(p) = \{m_0 \mid m \in V(p)\}$; for $p \in \{1,2,3,4\}$, $V'(p) = M^p$, and $V'(i) = \{s\}$.

It follows by induction that for all $m \in M$, for all uni-modal formulas ψ, $\mathcal{M}, m \Vdash \psi$ iff $\mathcal{M}', m_0 \Vdash \psi^t$. The interesting step is $\Diamond\psi$. Proving $\mathcal{M}, m \Vdash \Diamond\psi$ implies $\mathcal{M}', m_0 \Vdash (\Diamond\psi)^t$ is easy. For the other implication, suppose that $\mathcal{M}', m_0 \Vdash (\Diamond\psi)^t$. Then $\mathcal{M}', m_0 \Vdash \langle R\rangle(1 \wedge \langle R^{-1}\rangle(2 \wedge \langle R\rangle(3 \wedge \langle R^{-1}\rangle(0 \wedge \langle R^{-1}\rangle i \wedge \psi^t))))$. This implies that there exist $a_1 \in M^1$, $b_2 \in M^2$, $c_3 \in M^3$ and $n_0 \in M^0$ such that

$$\begin{tikzcd}
 & a_1 & & c_3 & \\
m_0 \arrow[ur] & & b_2 \arrow[ul] \arrow[ur] & & n_0 \Vdash \psi^t \arrow[ul]
\end{tikzcd}$$

But by the definition of R' it follows that $a_1 = n_1$, $b_2 = n_2$, and $c_3 = n_3$. Hence, as $R'(m_0, a_1)$, it follows that $R(m,n)$. Moreover, as $\mathcal{M}', n_0 \Vdash \psi^t$, by the inductive hypothesis $\mathcal{M}, n \Vdash \psi$. Thus $\mathcal{M}, m \Vdash \Diamond\psi$.

From this equivalence it follows that $\mathcal{M}', s \Vdash i \wedge \langle R\rangle 0 \wedge [R](0 \to \varphi^t)$, and we have proved the left to right direction of our claim.

[$\Leftarrow$]. Suppose $\mathcal{M}, w \Vdash i \wedge \langle R\rangle 0 \wedge [R](0 \to \varphi^t)$, where $\mathcal{M} = \langle M, R, V\rangle$ is a transitive hybrid model. Define a Kripke model $\mathcal{M}'$ as follows:

- $M' = \{m \in M \mid R(w,m) \;\&\; \mathcal{M}, m \Vdash 0\}$. Note that $M' \neq \{\ \}$, for $\mathcal{M}, w \Vdash \langle R\rangle 0$.
- $R' = \{(m,n) \in M' \times M' \mid \exists a,b,c \in M.(R(m,a) \;\&\; R(b,a) \;\&\; R(b,c) \;\&\; R(n,c) \;\&\; \mathcal{M}, a \Vdash 1 \;\&\; \mathcal{M}, b \Vdash 2 \;\&\; \mathcal{M}, c \Vdash 3)\}$.
- $V' = V_{\restriction M'}$.

It follows by induction that for all $m \in M'$, for all uni-modal formulas ψ, $\mathcal{M}, m \Vdash \psi^t$ iff $\mathcal{M}', m \Vdash \psi$. Only the step for $\Diamond$ is interesting. That $\mathcal{M}, m \Vdash \psi^t$ implies $\mathcal{M}', m \Vdash \psi$ is straightforward. For the converse, suppose $\mathcal{M}', m \Vdash \Diamond\psi$. Then $\exists n \in M'.(R'(m,n)\ \&\ \mathcal{M}', n \Vdash \psi)$. As $n \in M$, we have that $R(w,n)$, hence by the definition of R', $\exists n, a, b, c \in M.(R(m,a)\ \&\ R(b,a)\ \&\ R(b,c)\ \&\ R(n,c)\ \& R(w,n)\ \&\ \mathcal{M}', n \Vdash \psi)$, Moreover a, b, c satisfy 1, 2, and 3 respectively. By the inductive hypothesis $\mathcal{M}, n \Vdash \psi^t$. Hence, as w denotes i and $R(w,n)$, and as all points in M' satisfy 0, we have $\mathcal{M}, n \Vdash 0 \wedge Pi \wedge \psi^t$. It follows that $\mathcal{M}, m \Vdash (\Diamond\psi)^t$.

From this equivalence and $\mathcal{M}, w \Vdash [R](0 \rightarrow \varphi^t)$ it follows that $\mathcal{M}' \Vdash \varphi$, and we have established the right to left direction of our claim. ⊣

The theorem follows directly from the claim. QED

Clearly, $\langle R^{-1}\rangle$ played a crucial role in this proof, so if we want a PSpace or lower satisfiability result we will need to add further restrictions "to the past" to tame temporal hybridization. We will learn how to do this in the following section. Now for the upper bound, thanks to the Kleene star in PDL we can again use Theorem 7.20.

Corollary 7.26. *Sat for transitive frames in* $\mathcal{H}_{\mathsf{S}}(\langle R^{-1}\rangle, \mathsf{E})$ *is* ExpTime*-complete.*

Proof. Define a linear translation $\cdot^{tt}$ which is identical to $\cdot^{t}$ in Corollary 7.21 for atoms and Booleans, but handles $\langle R\rangle$ and $\langle R^{-1}\rangle$ as follows: $(\langle R\rangle\xi)^{tt} = \langle r; r^*\rangle(\xi)^{tt}$, and $(\langle R^{-1}\rangle\xi)^{tt} = \langle r^{-1}; (r^{-1})^*\rangle(\xi)^{tt}$. It is easy to see that φ is satisfiable over a transitive frames iff $(\varphi)^{tt}$ is satisfiable. QED

Hence, temporal hybridization on transitive frames still moves us outside the complexity class of the original logic. But from now on things get better. Two main kinds of transitive structures are usually considered as standard representations of time: strict total orders for linear time, and transitive trees for branching time. In the next two sections we will prove that in both cases hybridization is tamed.

7.4.2 Strict Linear Orders

For many applications time is modeled as linear. In particular, many temporal logics used in software and hardware verification assume a linear structure which corresponds with the sequence of states that a run of a program goes through [Pnueli, 1977; Manna and Pnueli, 1992].

Of course, choosing linearity leaves many interesting options open, such as whether density or discreteness holds, and whether or not initial and final points in time exist. But complexity-wise such choices are irrelevant: the complexity of the satisfiability problem for $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ *over any subclass* of the class of strict total orders is the same as for $\mathcal{M}(\langle R^{-1}\rangle)$. Given our remarks in Section 7.2 this should not come as a surprise: over strict total orders, $\langle R\rangle$ and $\langle R^{-1}\rangle$ are strong enough to define D, as $\mathsf{D}\varphi \leftrightarrow (\langle R\rangle\varphi \vee \langle R^{-1}\rangle\varphi)$ is valid. Thus we can eliminate all occurrences of nominals (and @ and E) by simulating them using D; in effect, we do the reverse of what we did in Theorem 7.13. Some care has to be taken to avoid a blow up in formula size during the elimination process, but it is easy to define an inductive replacement similar to the one used in the proof of Theorem 7.13. Thus:

THEOREM 7.27. *Let* S *be a subclass of the class of strict total orders. Up to a polynomial, the complexity of local* S*-Sat is the same for* $\mathcal{H}_{\mathsf{N}}(\Diamond, \mathsf{E}, \mathsf{D}, @)$ *as it is for* $\mathcal{M}(\langle R^{-1}\rangle)$.

On many natural linear flows of time (for example, the class of all strict total orders, or the class containing only $\langle \mathbb{Q}, <\rangle$) the complexity of *Sat* for $\mathcal{M}(\langle R^{-1}\rangle)$ is NP-complete [Ono and Nakamura, 1980; Clarke and Sistla, 1985], that is, no worse than propositional calculus, so $\mathcal{H}_{\mathsf{N}}(\Diamond, \mathsf{E}, \mathsf{D}, @)$ will inherit these results.

Theorem 7.27 may seem a bit like cheating. We don't pay any computational cost, but this is because hybridization over strict total orders does not increase the expressive power at our disposal. This is true, but it misses the point. In many applications of temporal logic, reference to times is not an optional extra, it is fundamental. Nominals can be seen as (and have been called) "clock variables" [Bull, 1970], and $@_i p$ read as "at the i-o'clock p". It is clear that they might come handy when specifying the temporal behavior of a system. To handle such problems naturally, we need formalisms which allow us to deal with temporal reference directly, and adding @ and nominals gives us an adequate level of abstraction. It is this level of abstraction (rather than the lower level offered by D) that needs to be isolated and explored.

7.4.3 Transitive Trees

In discussions of temporal logic in philosophy, natural language semantics, and computer science, the following intuition plays an important role: while several possible futures may be allowed, the past has a linear structure. To put it another way, it is often assumed that time has a tree-like structure, with the branching occurring only towards the future. Formal analysis of this conception of time go as far back as the work of William of Ockham in the fourteenth century [Ockham, 1969]. Logics for branching-time have also been actively investigated in computer science [Ben-Ari *et al.*, 1983; Emerson and Halpern, 1986].

Call a directed graph $\langle T, <\rangle$ a *tree* if it is acyclic and connected, and every node has at most one predecessor. A *transitive tree* is the transitive closure of a tree. In this section we are interested in the satisfiability problem of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ over such frames.

Note that the spypoint argument used to prove EXPTIME-hardness in Theorem 7.24 *will not work* for transitive trees: the encoding of transitivity in the model $\mathcal{M}'$ made crucial use of branching towards the past (points in M^1 look back to points in M^0 and M^2). And in fact, if we demand that our frames are transitive *trees*, we tame $\langle R^{-1}\rangle$ and drop back into PSPACE. The key intuition comes from inspecting the structure of the past in a transitive tree: if we generate the submodel by the converse of the accessibility relation from any point in a transitive tree, we obtain a strict linear order. Thanks to this property it will be easy to satisfy past formulas in small structures. Future formulas will require more work, particularly when they interact with nominals. Nonetheless, we will be able to show that all the required computations can be carried out in PSPACE.

NP-Complete Sub-fragments. Before we prove the announced PSPACE-result, consider the sub-fragment of $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle)$ in which we only have $\langle R\rangle$ formulas of the form $\langle R\rangle i \wedge [R](\langle R\rangle i \rightarrow \varphi)$ for one fixed nominal i. Such a formula says that there is a state named i in the future and that φ holds at every state between now and the state named i. In other words, $\langle R\rangle i \wedge [R](\langle R\rangle i \rightarrow \varphi)$ says the same thing as $\mathsf{Until}(i, \varphi)$.

For this fragment every satisfiable formula can be satisfied in a model of size polynomial in the length of the formula, yielding NP-completeness. This is shown by a simple submodel generation argument. Let $\mathcal{M}, m \Vdash \varphi$. If $\mathcal{M}, m \Vdash \langle R \rangle i$, create a submodel by considering the i-named state plus all its predecessors; otherwise create a submodel by taking m plus all its predecessors. In both cases we obtain a linear model which still satisfies φ at m. Since the models are linear, we can now use standard techniques from temporal logic to create a poly-size model for φ. Here again, we obtained a reduction in complexity by *guarding* the modalities with nominals. An inspection of the above argument shows that the restriction to just one nominal as guard is not needed. Whence:

Theorem 7.28. *Let $\mathcal{L}$ be the sublanguage of nominal tense logic in which every occurrence of a formula $\langle R \rangle \psi$ is of the form $\langle R \rangle(\langle R \rangle i \wedge \varphi)$, for some i, φ. Every $\mathcal{L}$ formula which is satisfiable on a transitive tree is satisfiable on a transitive tree of size polynomial in the length of the formula.*

In the proof of the next theorem we use the corollary that the same results hold for formulas of the form $\langle R \rangle(\langle R \rangle(i_1 \vee \ldots \vee i_k) \wedge \varphi)$. You can see this by distributing out the disjunctions. This formula can be used to distinguish future formulas that need their witnessing state before any named state from ones which don't. Suppose in $\mathcal{M}$ there are only finitely many states named by a nominal, say $i_1, \ldots, i_k$. Let $\mathcal{M}, m \Vdash \langle R \rangle \varphi$. Then either $\mathcal{M}, m \Vdash \langle R \rangle(\langle R \rangle(i_1 \vee \ldots \vee i_k) \wedge \varphi)$ or $\mathcal{M}, m \Vdash \langle R \rangle(\neg\langle R \rangle(i_1 \vee \ldots \vee i_k) \wedge \varphi)$. If the latter is false, this means that φ has to be true between m and some named state.

Theorem 7.29. *Local Sat for nominal tense logic over the class of transitive trees is* PSpace-*complete.*

Proof. PSpace-hardness follows from results in [Ladner, 1977]. The real work is to prove the PSpace upper bound. Our argument will be similar to the one used in Theorem 7.22, but now we should construct the appropriate tree structure in the game played by ∀belard and ∃loise. Instead of Hintikka sets, the players will use sequences of Hintikka sets which will play the role of branches in the model. But let us set up the game; first we define the following notions.

For any formula ξ, define the closure set $Cl(\xi)$ as the smallest set containing ξ, closed under subformulas, single negation and under the following rule: if $i \in Cl(\xi)$ then $\langle R^{-1} \rangle i \in Cl(\xi)$. Fix ξ. A *thread* is a finite labeled frame $\langle T, <, l \rangle$ such that:

i. $<$ is a weak total order (i.e., $<$ is transitive and trichotomous),

ii. $l : T \to Pow(Cl(\xi))$,

iii. l labels with maximal consistent Hintikka sets over $Cl(\xi)$,

iv. if $i \in l(x)$ then $\langle R^{-1} \rangle i \notin l(x)$,

v. $|T| \leq |\{\langle R^{-1} \rangle \varphi \in CL(\xi)\} \cup \{\langle R \rangle \varphi \in CL(\xi)\}| + 1$,

vi. (Future coherence) if $\langle R \rangle \varphi \notin l(x)$ and $x < y$, then $\{\langle R \rangle \varphi, \varphi\} \cap l(y) = \{\}$,

vii. (Past saturation) $\langle R^{-1} \rangle \varphi \in l(y)$ iff for some x, $x < y$ and $\varphi \in l(x)$.

The size $|t|$ of a thread $t = \langle T, <, l \rangle$ is $|T|$, and we say that $t_1 = \langle T_1, <_1, l_1 \rangle$ and $t_2 = \langle T_2, <_2, l_2 \rangle$ *fit at* $x \in T_1 \cap T_2$ iff $\langle T_1, <_1, l_1 \rangle_{\restriction\{s \in T_1 \mid s \leq_1 x\}} = \langle T_2, <_2, l_2 \rangle_{\restriction\{s \in T_2 \mid s \leq_2 x\}}$. Two threads *fit* if there exists an x such that they fit at x. You can think of threads as "pieces of branches," we will construct full branches, and in the long ran the full tree model, by "superposing" threads.

We first take care of the trivial case when $|\xi| = 1$. Then ξ is either an atom or a logical constant, and satisfiability is trivial. From now on we assume $|\xi| > 1$. We can now specify the game for ∀belard and ∃loise. She should set up the playing board by specifying a collection M_0 of threads such that:

i. for some thread $t = \langle T, <, l\rangle \in M_0$, for some $x \in T$, $\xi \in l(x)$,

ii. each two threads in M_0 fit,

iii. the number of threads in M_0 is less than $|\mathsf{NOM}(Cl(\xi))| + 1$.

iv. for each nominal i in $\mathsf{SF}(\xi)$ there is a thread $t = \langle T, <, l\rangle \in M_0$, and $x \in T$ such that $i \in l(x)$,

v. let i be a nominal in $\mathsf{SF}(\xi)$, t_1, t_2 two threads in M_0 and $x_1 \in T_1$, $x_2 \in T_2$ be such that $i \in l_1(x_1)$ and $i \in l_2(x_2)$, then $x_1 = x_2$ and t_1 and t_2 fit at x_1,

vi. (Necessary pre-nominal saturation) if there is $t_1 = \langle T_1, <_1, l_1\rangle \in M_0$ such that for $x_1 \in T_1$, $\{\langle R\rangle\varphi, \langle R\rangle i\} \subseteq l_1(x_1)$, then there is $t_2 = \langle T_1, <_2, l_2\rangle \in M_0$ such that t_2 fits t_1 at x_1 and there is an $x_3 \in T_2$ such that $i \in l_2(x_3)$ and either φ or $\langle R\rangle\varphi$ in $l_2(x_3)$, or there is another $x_2 \in T_2$ satisfying $x_1 <_2 x_2 <_2 x_3$ and $\varphi \in l_2(x_2)$.

The "necessary pre-nominal saturation" condition deserves comment. With this condition, we are asking ∃loise to take care of the demands of the form $\Diamond\varphi$ which have to be satisfied at a witnessing state which is a predecessor of some nominal. All in all, the rationale behind M_0 is that — when the threads are fitted together — it can be seen as a transitive tree model in which no nominal occurs in the label of two distinct states and in which all necessarily pre-nominal future formulas have a witnessing state.

If ∃loise cannot set up the board she loses. Otherwise ∀belard chooses one of the threads in M_0, which will be the thread *in use* in the round. To start with, all the elements in the thread in use are *available*. In each round there will be a thread t in use, with a subset of its domain available for ∀belard to pick from. Furthermore we will keep a table S containing for each $\langle R\rangle\varphi$ formula in $Cl(\xi)$ a natural number $S(\langle R\rangle\varphi)$. At the start, for all $\langle R\rangle\varphi \in Cl(\xi)$, $S(\langle R\rangle\varphi) = |\xi|$.

We are now ready to specify the movements of the players. In each round, ∀belard points to an element x among the available elements in the thread t in use, and to a formula $\langle R\rangle\varphi$ in $l(x)$. x should be a maximal element containing $\langle R\rangle\varphi$. Formally, for all $y \in t$, $x < y$ implies $\langle R\rangle\varphi, \varphi \notin l(y)$ and furthermore, there is no thread $t'' = \langle T'', <'', l''\rangle$ on the board, fitting t at x such that either $\langle R\rangle\varphi$ or φ are in $l''(y)$ for some $x <'' y$.

∃loise's answer depends on the value recorded for the formula chosen by ∀belard.

- If $S(\langle R\rangle\varphi) > 1$, then she should play a thread $t' = \langle T', <', l'\rangle$ such that t' fits t at x, there is $y \in T'$ such that $\varphi \in l'(y)$ and $x <' y$, neither i nor $\langle R\rangle i$ is in $l'(z)$ for $x < z$, $i \in \mathsf{NOM}$, and $|\{z \in T' \mid x <' z\}| < S(\langle R\rangle\varphi)$. If she cannot present such a t', she loses. Otherwise, t' becomes the thread in use and $\{z \in T' \mid x <' z\}$ the available Hintikka sets for ∀belard. Furthermore $S(\langle R\rangle\varphi)$ is updated to $|\{z \in T' \mid x <' z\}|$.
- If $S(\langle R\rangle\varphi) = 1$, then $\langle R\rangle\varphi$ has been played by ∀belard before. ∃loise should pick the Hintikka set e' containing φ that fixed the defect the last time ∀belard challenged $\langle R\rangle\varphi$. If $t' = \langle T_{\restriction\{z \in T \mid z \leq x\}} \cup \{n\}, < \cup\{(y, n) \mid y \in T\}, l \cup (n, e')\rangle$ for $n \notin T$ is a thread she wins. Otherwise she loses.

Notice that the values in S decrease as ∀belard repeats choices of $\langle R\rangle\varphi$ (which he cannot avoid as $Cl(\xi)$ is finite). This ensures termination of the game in at most $|\{\langle R\rangle\varphi \in Cl(\xi)\}| * |\xi|$ steps. To prove the theorem we should now establish that ∃loise has a winning strategy in the game described above if and only if ξ is satisfiable; and then observe that in each game only a polynomial number of Hintikka sets are displayed on the board. (The extra information contained in the table S can obviously be coded using just linear space in the length of ξ.)

Claim 7.30. ∃loise has a winning strategy for the ξ-game iff ξ is satisfiable in a model based in a transitive tree frame.

Proof of Claim.
[⇒] Suppose ∃loise has a winning strategy in the ξ-game. We construct a model $\mathcal{N} = \langle N, R, V\rangle$ as follows. Let M be the set of threads ever played by ∃loise in her winning strategy. $\mathcal{N}$ is obtained by "superposing" one by one any two threads in M at their fitting point. More precisely, given two threads t and t' there is a maximal point x such that t and t' fit. Hence for any thread t we can identify a maximal point x such that t fits with some other thread at x. The elements above x are "unique" to t; ensure uniqueness by renaming and let M' be the set of threads obtained in such a way. Notice that after such a renaming, for any element x of any thread in M' we can uniquely identify a Hintikka set corresponding to x, and for any two elements x, y in threads $t = \langle T, <, l\rangle$, $t' = \langle T', <', l'\rangle$ in M, if $x, y \in T \cap T'$ then $x < y$ iff $x <' y$. Let N be the union of all elements in threads in M', R be the union of the $<$ relations, and let V assign to an atomic symbol a the elements x in N such that a is in the Hintikka set corresponding to x. If i is a nominal not in $\mathsf{SF}(\xi)$ then $V(i)$ is any singleton set in N. Since during the setting up of the board nominals in M_0 are assigned to unique elements and ∃loise never plays threads with new Hintikka sets containing nominals in her winning strategy, V is a hybrid valuation.

The following truth lemma holds in $\mathcal{N}$: let $\varphi \in Cl(\xi)$, and s be any element in N, then $\mathcal{N}, s \Vdash \varphi$ iff $\varphi \in H(s)$, where for any x in N, $H(x)$ is the Hintikka set corresponding to x. This would set the left-to-right direction of the claim, except that the threads $t = \langle T, <, l\rangle$ used in the construction of $\mathcal{N}$ might contain *clusters*, i.e., maximal (by inclusion) non empty sets S of T such that for all $x, y \in S$, $x < y$. But we can unravel them by a standard technique, preserving satisfiability. Notice that worlds labeled by nominals will never appear in clusters by conditions *iv*) and *vii*) in the definition of threads, and hence they will not be duplicated by the unraveling. The unraveled model will be a transitive tree.

[⇐] We should now prove that if ξ is satisfiable, then ∃loise has a winning strategy. Let $\mathcal{M}$ be a transitive tree model such that $\mathcal{M}, w \Vdash \xi$. And let $\mathcal{M}^f = \langle M^f, R^f, V^f\rangle$ be a transitive filtration of $\mathcal{M}$ under $Cl(\xi)$. By properties of filtrations we know that $\mathcal{M}^f$ is finite and R^f is transitive. Furthermore R^f has maximal (and minimal) elements, i.e., those $|m| \in M^f$ such that for all $|m'| \in M^f$, $R^f|m||m'| \Rightarrow R^f|m'||m|$. Also, elements of M^f can be uniquely identified with Hintikka sets over $Cl(\xi)$ (and we will treat them as such). But $|M^f|$ can still be exponential in $|\xi|$.

For each element $|m| \in M^f$ we can built a thread t as follows. For each $\langle R^{-1}\rangle\varphi$ in $|m|$ choose a minimal predecessor $|m'|$ of $|m|$ which satisfies φ. Order t by R^f. Since the filtration comes from a transitive tree, a standard argument shows that t is past saturated and hence a thread. Furthermore $|t|$ is linear in $|\xi|$. We call such t a thread built from $|m|$ in M^f. Let φ be a formula satisfiable in $\mathcal{M}^f$. Then a thread for φ is a thread t from some $|m|$, such that $\mathcal{M}^f, |m| \Vdash \varphi$, but we require in addition that $|m|$ is R^f-maximal satisfying φ, and that t is *long*, i.e., for all other threads t' built from $|m'|$ such that $\mathcal{M}^f, |m'| \Vdash \varphi$, $|t'| < |t|$.

Let t be a thread built from $|m|$ and let $|m'|$ be an element in t. Let $|m''|$ be an R^f-successor of m'. It is not difficult to prove that there is a thread t' built from $|m''|$ that fits t at m'. Furthermore t' is linear in $|\xi|$. From this fact,

> $(*)$ if t is a thread built from an element in $\mathcal{M}^f$ and $|m|$ is any element in t such that $\mathcal{M}^f, |m| \Vdash \langle R\rangle\varphi$. Then there is a thread for φ that fits t at $|m|$.

All this machinery is used to specify ∃loise's answers to challenges made by ∀belard. To show that ∃loise can set up the board properly we use Theorem 7.28. The theorem says precisely that all past formulas and all necessarily pre-nominal future formulas can be satisfied using only a polynomial number of states. With this setup it is guaranteed that ∀belard can only point to defects $\langle R\rangle\varphi$ which are *not* necessarily pre-nominal future formulas. This ensures that in the case the antecedent of $(*)$ above obtains, the needed fitting thread can be chosen not to contain any nominal or $\langle R\rangle i$ formula.

With ∃loise's groundwork taken care of, we now specify how ∃loise answers the challenges of ∀belard. Suppose that ∀belard points to a defect formula $\langle R\rangle\varphi$ in an element $|m|$ in t. If $S(\langle R\rangle\varphi) > 1$, she answers with a thread t' provided by $(*)$, again choosing t' to be maximal and long. As argued above she can answer with a thread in which neither i nor $\langle R\rangle i$ appears in t' above the fitting point.

By obtaining threads from $\mathcal{M}^f$, ∃loise can always answer the moves of ∀belard. What remains to be checked is the condition on decreasing lengths when ∀belard plays a repeated formula. So assume that ∀belard has in a previous round chosen $\langle R\rangle\varphi \in |m| \in t_1$, and that ∃loise answered with a long maximal thread t_2. Furthermore he is now choosing $\langle R\rangle\varphi \in |m'| \in t_3$. We know that ∃loise can produce a long maximal thread t_4 to answer the challenge. But it is immediate that if the size of t_4 above $|m'|$ is greater than the size of t_2 above $|m|$, then t_2 was not a long maximal thread.

Suppose now that $S(\langle R\rangle\varphi) = 1$ and let t_1 be the thread previously played by ∃loise as an answer, with $\varphi \in |m_1|$. Again by $(*)$ we could produce a thread t' fitting t at m and containing φ in some element $|m'|$. Furthermore, because of the decreasing condition, we know that this time she can choose t' to be very short. Actually $|m|$ is at "one step" from the R^f-maximal cluster satisfying φ, and ∃loise can choose any element of such a cluster, in particular $|m_1|$. ⊣

As we already mentioned, the game stops in time quadratic in the size of ξ, furthermore all the information ever played in the board can be encoded in polynomial space. Hence the theorem follows. QED

What happens if we add E? Nothing — for we already have it: over transitive trees $\mathsf{E}\varphi$ can be defined to be $\varphi \vee \langle R\rangle\varphi \vee \langle R^{-1}\rangle\varphi \vee \langle R^{-1}\rangle\langle R\rangle\varphi$.

Corollary 7.31. *The transitive tree Sat for $\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, \mathsf{E})$ is* PSpace*-complete.*

7.5 Reflections

The following table summarizes the most important complexity results we presented in the chapter and contrasts the effect of "hybridization" in terms of complexity, with respect to modal languages without nominals ($\mathcal{M}$).

Class of frames	$\mathcal{M}(\langle R^{-1}\rangle)$	$\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @)$	$\mathcal{M}(\langle R^{-1}\rangle, \mathsf{E})$	$\mathcal{H}_{\mathsf{N}}(\langle R^{-1}\rangle, @, \mathsf{E}, \mathsf{D})$
All frames	PSpace	ExpTime	ExpTime	ExpTime
Transitive	PSpace	ExpTime	ExpTime	ExpTime
Strict total orders	NP	NP	NP	NP
Transitive trees	PSpace	PSpace	PSpace	PSpace

Some comments concerning the table. Hybrid temporal logic has an ExpTime-hard satisfiability problem over both arbitrary frames (Corollary 7.21) and transitive frames (Theorem 7.24). These results hold even without @, and only one nominal is needed to establish them. Matching ExpTime upper bounds hold for nominal tense logic, over both arbitrary and transitive frames, even if E is added (Corollaries 7.21 and 7.26). Over any class of strict total orders, the satisfiability problem for nominal tense logic is the same as that of basic tense logic (Theorem 7.27); for example, for the class of all strict total orders, nominal tense logic with E has an NP-complete satisfiability problem. Over trees, nominal tense logic (both with and without @) is PSpace-complete (Theorem 7.29), even when extended with E or D (Corollary 7.31). With respect to pure-future fragments, Theorem 7.15 and Theorem 7.22 show that hybridization does not damage the fundamental complexity results for modal logics. Moreover, while adding E results in an ExpTime-complete satisfiability problem over arbitrary frames [Spaan, 1993], we proved that over transitive frames the logic has a PSpace-complete satisfiability problem (Theorem 7.22). In other words, transitivity is enough to "tame" E, we don't need to restrict ourselves to the guarded form offered by @ in this case.

Concerning hybrid languages with binders, in Theorem 7.1 we proved that even the fragment of $\mathcal{H}_{\mathsf{S}}(\downarrow)$ consisting of pure nominal-free sentences has an undecidable local K-*Sat* problem. Only when we imposed severe restrictions, like considering only non-nested occurrences of $\downarrow$, we were able to regain decidability (Theorem 7.10).

But perhaps the lesson to learn from this chapter lies rather on the methods we used to arrive to these results. We have used (indirectly through Spaan's result for K_{23}) the technique of tiling to prove undecidability, or more directly the encoding of satisfiability problems of known complexity to establish lower bounds. We also used encodings for upper bounds, but in some of the most interesting cases (Theorems 7.15 and 7.29) we resorted to model constructions games. We were able to enhance these methods by using the spypoint technique which deserves to be more widely known. We believe that it will prove useful in more general settings, and the description logic community, for one, has already started to take advantage of it (see [Tobies, 2000a]).

Part IV
The Things We've Learned

Vagando por el Quai des Célestins piso unas hojas secas y cuando levanto una y la miro bien la veo llena de polvo de oro viejo, con por debajo unas tierras profundas como el perfume musgoso que se me pega en la mano. Por todo eso traigo las hojas secas a mi pieza y las sujeto en la pantalla de una lámpara. Viene Ossip, se queda dos horas y ni siquiera mira la lámpara. Al otro día aparece Etienne, y todavía con la boina en la mano, Dis donc, c'est épatant, ça!*, y levanta la lámpara, estudia las hojas, se entusiasma, Durero, las nervaduras, etcétera.*
Una misma situación y dos versiones . . . Me quedo pensando en todas las hojas que no veré yo, el juntador de hojas secas, en tanta cosa que habrá en el aire y que no ven estos ojos, pobres murciélagos de novelas y cines y flores disecadas.

from "Rayuela," Julio Cortázar

We have completed our homework (and it was about time, being as we are on page 131). There remain, of course, many directions in which we only took the first steps, but that is good and well. We will explore those trails further during other trips through the Kingdoms of Description and Hybrid Logics.

But did we advance on our general theme, the one introduced right in the opening of Part I? What did we learn about Logic Engineering?

Well, there are probably as many answers to that question as people reading this thesis. We have tried at least to make clear and provide support to our claim that it is possible to have a wide range of options, when the time comes to choose a logic for a specific reasoning or modeling task. We have also shown that there are many dimensions to this range of options: expressive power, complexity, availability of effective reasoning methods, presence or absence of important meta-logical properties, to mention some. We can take any of the results or techniques we have discussed in the previous chapters as an example of how to gauge the pros and cons of these possibilities.

We have centered our discussion on the case of description and hybrid logics, as they provided an excellent playground to explore. Description languages offered us variety, application examples, realistic and highly optimized implementations; hybrid languages offered the proper model-theoretical tools to investigate and better understand them. But we should take the work we have done in this thesis as an example of a more general philosophy: choose the logic you will use as you choose any tool for a particular job. You don't use a pair of scissors to water your plants, or do you?

In Chapter 8 we will take a short look back to recall what we have done and where we have been. We will browse through the snapshots of our trip and choose the nicest to send home for all the family to see.

Chapter 8

Conclusions

Begin at the beginning
and go on till you come to the end;
then stop.

from "Alice's Adventures in Wonderland," Lewis Carroll

We started our work in Part I by discussing the general topic of Logic Engineering, or the "subtle art of choosing the proper language."

In Chapter 1, we argue that first-order logic, which has been *the* logic for years, need not always be the proper choice. We back up this claim by first proving that the satisfiability problem for FO is undecidable. This can be reason enough to disqualify FO if the problem at hand requires one to deal with logical consequence in an effective way. Notice that if instead we would be interested in a problem requiring model checking, then FO would again enter the list of candidates. One way to interpret the undecidability result for FO is as a sign that the language is too expressive — it lets us encode problems which are too complex. If we were looking for effective reasoning techniques then, conceivably, our problem should be simpler, i.e. decidable. But even with all its expressive power, FO might just not provide the *right kind* of expressivity we need. We show that this might be the case by proving that FO cannot characterize the transitive closure of a binary relation. Once more, the point is this: we don't need the scissors of first-order logic when we are watering the roses, but this doesn't imply that they won't be handy during the pruning season.

But Chapter 1 also aimed to introduce a methodology to obtain alternatives to FO. We discussed different ways of identifying interesting fragments (and fragments of extensions) of first-order logic. We argued that traditional methods, like prenex normal form or finite variable fragments, are not completely satisfactory, and we proposed, instead, to capture relevant fragments *via translations.* The semantics of many formal languages (including modal, description and hybrid languages) is given in terms of classical logics, and as such they can be considered fragments of classical languages. But now, these fragments come together with an extremely simple representation (modal languages for example are usually introduced as "simple extensions of *propositional* logic") and with novel and powerful proof- and model-theoretical tools (simple tableaux systems, elegant axiomatizations, fine-grained notions of equivalence between models, new model-theoretical constructions, game-theoretical characterizations, etc.) which let us investigate their properties in detail.

We chose our case study, description and hybrid logics, starting on the description logic side. DLs are the best example we know about of a collection of formal languages which have been hand-tailored for specific tasks. The choice of hybrid languages came later, when we searched for the proper "modal counterpart" of description logics. After introducing these two kingdoms in Chapters 2 and 3 (one of them built on the realms of Computer Science, the other on those of Mathematical Logic), we brought them together in Chapter 4 by carefully mapping out the roads that connect them.

The results in Chapter 4 show that cross-fertilization between hybrid and description logics is possible and indeed rewarding. We have devised hybrid languages which work nicely as the counterpart of certain description languages, allowing us to account neatly for full knowledge bases in a modal way. We can now exploit to their maximum the model-theoretical techniques in our modal tool-box in the widely diverse landscape of description languages. Going in the other direction, hybrid languages will benefit from a computational boost. Description logic provers can easily be adapted to handle certain hybrid languages in a very efficient manner. And the two fields will, from now on, share their "application space": any given concrete problem will benefit from both a description-like and a hybrid-like perspective.

In Part III we set forth to walk some of the paths we discovered in Chapter 4. In particular, in Chapter 5 we discuss direct resolution methods for modal-like languages. As with tableaux, the addition of labels produces a simplification with respect to previous proposals. We also gain in terms of flexibility as we have shown how easily extensions of the basic labeled resolution method can be obtained. Excellent proof-theoretical behavior seems to be a general characteristic of hybrid and description logics, and we argue that the presence of nominals/individuals and the satisfiability operator @ (assertions in the description logic case) goes a long way towards explaining it.

The work in this chapter shows that simple, direct resolution methods for modal languages are indeed possible, and that the complexities of previous proposals hinged on a certain lack of expressive power. The introduction of labels lets us perform resolution at the "top level" only (outside modalities) and greatly simplifies the task of the prover.

In Chapters 6 and 7 we take a hybrid logic perspective as we dive into model-theoretical issues. But we have already demonstrated in Chapter 4 how hybrid logic results shed their light on description languages. Actually, in Section 4.5 we already took advantage of the most important results in these two chapters and analyzed what was their interpretation in description logic terms.

Chapter 6 covers issues related with expressive power. We took the language $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ as the main tune, and explored restrictions and extensions as variations on a theme. By using a mixture of modal and first-order techniques (a hallmark of hybrid languages), we have shown that $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ captures an intrinsically modal first-order fragment: this language corresponds to the formulas of FO which are invariant under generated submodels, mirroring the key modal notion of locality. It is very pleasing that this notion can be pinned down so simply. Furthermore, the very general result on interpolation for all pure extensions of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ tends to confirm that we are dealing with a natural collection of ideas. These new results, complement the general completeness results for the language provided in [Tzakova, 1999a]. The model theory of $\mathcal{H}_{\mathsf{S}}(@, \downarrow)$ is extremely elegant. The language seems to be in a state of "perfect equilibrium," much in the same way as the first-order language is. Investigating in detail this kind of logical systems is always a worthwhile activity: much is to be learned by means of the wide collection of tools they offer, and these lessons sometimes transfer to restrictions and extensions.

We see this process in action in Chapter 6, where we are able to transfer results from $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ to extensions and restrictions.

The expressive power analysis we have carried out in this chapter tells us where the boundaries fall. And for the case of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ we have obtained extremely clear boundaries. But the notions of bisimulations we have investigated also show us how to delimit the expressive power of weaker languages. The work concerning the interpolation and Beth definability properties also highlighted an interesting phenomena. The interpolation property is usually taken as a "thermometer" for balanced language design. Following that guide, $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ is indeed well balanced but, as we show in Chapter 7, even small fragments of the language are already undecidable. We conjecture, though, that the intrinsic locality of $\mathcal{H}_{\mathsf{S}}(@,\downarrow)$ would translate in more efficient decision methods than those available for FO. On the other hand, weaker hybrid languages and most description languages are computationally tractable but they usually fail to have interpolation. We believe that the results regarding failure of interpolation we prove in the chapter points to an expressivity gap. We know from the description logic community that the addition of counting operators does not disturb the good computational behavior and we conjecture that they will provide the needed expressivity to regain interpolation.

Chapter 7 is devoted to complexity. After proving a sharp undecidability result for $\mathcal{H}_{\mathsf{S}}(\downarrow)$ and showing that decidability can be regained by imposing stringent restriction on the binder, we turned to weaker languages in the proximities of $\mathcal{H}_{\mathsf{S}}(@)$. These languages are very close to standard description languages and hence transfer of results is easy in this case. We show that the addition of nominals and @ to the basic modal logic **K** does not modify its complexity. Hence, for this language, hybridization brings extra expressiveness at no cost (except perhaps by a polynomial). When we explore the basic temporal language $\mathbf{K}_t$ instead, things are very different: the addition of a single nominal shifts the complexity to ExpTime. The rest of the chapter is devoted to "taming" this complexity jump, and we show that in the most interesting classes of temporal models (linear and branching time structures) complexity drops again and coincides with the complexity of $\mathbf{K}_t$ over these structures. As we discuss in their respective sections, the results for linear and branching time are different in nature: the first amounts to the realization that certain expressivity was already present in the language and the identification of the proper level of abstraction; while the second covers a strict increment of expressive power and requires a much more elaborated proof.

The lessons to learn in this chapter are related to the methods we employed in our proofs. We obtained the complexity results we discussed above in a very homogeneous way. We basically used encodings of satisfiability problems of known complexity for lower bounds, and model construction games and encodings for upper bounds. In many cases we "power-uped" translations by means of the spy-point technique (i.e., the use of a single irreflexive point which have full access to a part of the model). The Spypoint Theorem (Theorem 7.9) is a clear example of the strengths of this method. These tools and methodologies are so powerful and versatile that it is usually possible to adapt them to many diverse situations.

To complete Chapter 8 we will take up description and hybrid logics as separate fields again, and discuss some of the main lines we touched on and clarified in our work.

8.1 On Description Languages

It takes time for a modal logician to get used to how things are done in the description logic community (we know by experience). And one of the reasons for this is the shift from a local to a global perspective. The basic notion of validity in modal languages is truth of a modal formula *at a point* in a model. Description logicians instead, are interested in global notions (definitions and assertions) which are true throughout the model, and hence take global consequence as basic.

In Chapter 4 we carefully analyzed this issue and explained how the local and global notions of consequence interrelate. We also designed the hybrid logic counterparts of description languages so that we can investigate this issue, and results like Theorems 4.8 and 4.9 or the discussion in Section 4.5.2 exploit this fact. By means of the existential modality E we made available modal model-theoretical tools to the investigation of properties of inference in terms of non-empty T-Boxes. The work also turned out to be fruitful in pure hybrid logic terms as we were able to identify a useful normal form for hybrid languages without binders (Proposition 4.2). The issue of globality vs. locality has deep roots and can impact heavily on, for example, complexity issues. It is well known that instance checking in $\mathcal{ALC}$ is in PSpace for empty T-Boxes and ExpTime without this restriction; or similarly in modal terms, the addition of E to the basic modal language produces an exponential blow-up in the local satisfiability problem. Notice though that this behavior is not because of “globality.” The modal logic **S5** is globality itself and its local satisfaction problem falls inside NP! We need both locality “to tell things apart” and globality “to spread these differences throughout the model.” We can witness the same behavior on the issue of transitivity vs. transitive closure we discussed in Section 4.2. Languages with the ability to refer to the transitive closure of a relation are usually computationally more expensive than those which can simply define a relation to be transitive. In the former case we have two kinds of expressivity (local and global), while we only have globality in the latter.

The work of this thesis has also brought more light to the relation between the A-Box and T-Box in a description logic knowledge base. As we explained in Chapter 2, there are methodological reasons why it is worth to attempt such a separation on the available information we aim to model. And there are also important reasons which have to do with implementations: enforcing this separation can lead to simple and efficient reasoning algorithms. Some provers, like for example race, are actually able to classify the T-Box component of a knowledge base independently of its A-Box, thus allowing for inference in terms of different instantiations at a lower computational cost. But enforcing this separation also has its price. As we showed in Section 4.5.3, only by allowing the interplay of T- and A-Box information we can prove that a certain notion of definability holds for the language. Furthermore, results like Theorem 7.15 show that T- and A-Box information can indeed “live together” without further complexity costs.

Description logics seem to be finding their way in more and more diverse environments each day, and this is a trend that will not stop in the immediate future, on the contrary. They offer a wide range of inference services to chose from, and they deliver their goods in the form of extremely fast and optimized provers, together with a wealth of expertise concerning how to better structure and exploit complex information.

8.2 On Hybrid Languages

In their long (if sparse) history, hybrid languages have attracted a number of enthusiastic advocates. Some have claimed that hybridization is a natural way to increase the expressive power of modal languages, others have been impressed by the proof-theoretical options they open up, or the ease with which general results can be proved. Underneath most of this work lies a simple idea: that by exploiting the notion of formulas as terms to the full, we will be able to define systems combining the best of modal and classical techniques.

We believe that the results we have presented confirm the interest of hybridization. In writing this thesis it has become very clear that working with hybrid languages involves a genuine interplay of modal and classical methods. For example, both Ehrenfeucht games (or back-and-forth systems) and bisimulations were involved in the expressivity results of Chapter 6, and the general interpolation result in Theorem 6.27 was proved by combining the modal notion of canonical models with the classical idea of Henkin models. The natural way these methods blend bodes well for further developments.

Nominals seem to add a new dimension to modal logic (like filling a hole that only now we notice was there). And as we discussed in Chapter 3 they "cure" an asymmetry at the heart of modal logic. Modal logic is locality itself, and once a local point of view is adopted, once we evaluate formulas at a particular point in a model, the concept of "terms as formulas" comes very naturally. And the existential modality E can simply be added if we need to shift to a global perspective. But nominals are only an instance of a more general theme which deserves much further analysis: sorting. Hybrid languages are obtained from modal languages by extending the language with a new collection of symbols together with a restriction on the interpretation this symbols will receive on models. But the denotation of nominals (i.e., singletons) is probably just the most simple extension. What about exploring more complex sorts like paths, connected components, etc? And how do these sorts interact with one another? A neat example of sorted modal logics are the computational tree logics CTL and CTL*. In a very general perspective, hybrid logics as we know them today are just our first steps towards investigating the more general class of sorted modal logics.

And of course there is the issue of binders. The classical quantifiers $\forall$ and $\exists$ are clearly interesting and a straightforward alternative, but the modal perspective gives rise to new options like $\downarrow$, a truly modal binder. And there are other possibilities (like the $\downarrow^1$ of [Blackburn and Tzakova, 1998b] or, more generally, the $\downarrow^n$ hierarchy), and of course the many combinations of different binders for different sorts. These operators let us capture powerful and natural new fragments of first-order logic without the complexities of actually moving into first-order modal languages. But actually, we can also hybridize first-order modal languages, and some recent preliminary results in our ongoing work seem to indicate that again hybridization would lead to general results concerning for example completeness and interpolation.

Hybrid logics will probably continue to play a role in the future. They offer high expressive power, an elegant proof theory and plenty of connections with other fields like temporal reasoning and knowledge representation. For the moment they taught us a bit more about the structure of the landscape of fragments we are exploring.

8.3 What the Future Brings

It is always difficult, and perhaps unwise, to cast bold predictions about what the future will bring. But it doesn't seem risky at all to say that in a number of years we will be able to chose from a very broad menu of language options when working on a given computational logic enterprise. And we will know in advance what the properties of these languages are: which are the boundaries of the expressive range they offer, which are their complexity prices, which are the available reasoning tools they offer, etc.

As the Queen says to Alice, on her trip on the other side of the looking-glass, *'It's a poor sort of memory that only works backwards.'* It would be more satisfying to remember today a bit of what is to come. To remember (even if vaguely) of when we will be able to understand how to identify the *correct language* for a given, specific need.

Some beautiful results have already been given to us, like the guarded fragments of Andréka, van Benthem and Németi [1995] ("if a logic can be mapped *here* then it is decidable") or the work on complexity of modal logics of Spaan [1993] ("if a logic is able to express *this* then it has at least *this* complexity"). The conditions for failure of interpolation provided in [Areces and Marx, 1998] are in a similar line. The discussions on robust decidability of Vardi [1997] and Grädel [1999] are another example.

What are the well behaved fragments and, more interestingly, what are the reasons of their good properties? These are indeed important questions which the new field of Logic Engineering is only just starting to unravel.

Bibliography

The numbers in parenthesis after each bibliographic entry refer to the pages in which the entry has been mentioned.

from "Logic Engineering," Carlos Areces

[Abadi and Manna, 1985] (75) M. Abadi and Z. Manna. Nonclausal temporal deduction. *Lecture Notes in Computer Science*, 193:1–15, 1985.

[Ackermann, 1954] (6) W. Ackermann. *Solvable Cases of the Decision Problem.* North-Holland Publishing Co., Amsterdam, 1954. Studies in Logic and the Foundations of Mathematics.

[Aiello and van Benthem, 1999] (17) M. Aiello and J. van Benthem. Logical patterns in space. Technical Report PP-99-24, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, 1999. Submitted to CSLI for publication. Available at `http://www.wins.uva.nl/~aiellom/publications/lops.ps.gz`.

[Aiello *et al.*, 1999] (17, 23) M. Aiello, C. Areces, and M. de Rijke. Spatial reasoning for image retrieval. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 23–27, 1999.

[Allen, 1984] (36, 38) J. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, July 1984.

[Andréka *et al.*, 1995] (138) H. Andréka, J. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *Journal of the IGPL*, 3(5):685–720, 1995.

[Andréka *et al.*, 1998] (11, 12, 13, 114) H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.

[Andréka, 1991] (12) H. Andréka. Complexity of the equations valid in algebras of relations, 1991. Thesis for D.Sc. (a post–habilitation degree) at the Institute of Mathematics, Hungarian Academy of Sciences, Budapest, Hungary.

[Areces and de Rijke, 1998] (18) C. Areces and M. de Rijke. Expressiveness revisited. In Franconi et al. [1998], pages 35–43. Available at `http://sunsite.informatik.rwth-aachen.de/publications/ceur-ws`.

[Areces and de Rijke, 2000] (18) C. Areces and M. de Rijke. Description $^{\text{and}}/_{\text{or}}$ hybrid logics. In *Proceedings of AiML 2000. Advances in Modal Logic*, Leipzig. Germany, October 2000. University of Leipzig.

[Areces and Marx, 1998] (11, 138) C. Areces and M. Marx. Failure of interpolation in combined modal logics. *Notre Dame Journal of Formal Logic*, 39(2):253–273, 1998.

[Areces *et al.*, 1999a] (18) C. Areces, P. Blackburn, and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo, editors, *Computer Science Logic*, number 1683 in LNCS, pages 307–321. Springer, 1999. Proceedings of the 8th Annual Conference of the EACSL, Madrid, September 1999.

[Areces *et al.*, 1999b] (17, 23) C. Areces, W. Bouma, and M. de Rijke. Feature interaction as a satisfiability problem. In *Proceedings of MASCOTS'99*, pages 339–346, Los Alamitos, California, October 1999. IEEE Computer Society.

[Areces *et al.*, 1999c] (18, 68) C. Areces, H. de Nivelle, and M. de Rijke. Prefixed resolution: a resolution method for modal and description logics. In *Automated Deduction—CADE-16 (Trento, 1999)*, pages 187–201. Springer, Berlin, 1999.

[Areces *et al.*, 2000a] (18, 38, 70) C. Areces, P. Blackburn, and M. Marx. The computational complexity of hybrid temporal logics. *Logic Journal of the IGPL*, 8(5):653–679, 2000.

[Areces *et al.*, 2000b] (18) C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 2000. To appear.

[Areces *et al.*, 2000c] C. Areces, E. Franconi, R. Goré, M. de Rijke, and H. Schlingloff, editors. *Methods for Modalities 1*, volume 8(3). Logic Journal of the IGPL, 2000.

[Areces *et al.*, 2000d] (74, 87) C. Areces, J. Heguiabehere R. Gennari, and M. de Rijke. Tree-based heuristics in modal theorem proving. In *Proceedings of ECAI'2000*, Berlin, Germany, 2000.

[Auffray *et al.*, 1990] (87) Y. Auffray, P. Enjalbert, and J. Hebrard. Strategies for modal resolution: results and problems. *Journal of Automated Reasoning*, 6(1):1–38, 1990.

[Baader and Hollunder, 1991] (22, 28, 30) F. Baader and B. Hollunder. A terminological knowledge representation system with complete inference algorithm. In Springer-Verlag, editor, *Proceedings of the Workshop on Processing Declarative Knowledge (PDK-91)*, number 567 in Lecture Notes in Artificial Intelligence, pages 67–86, 1991.

[Baader *et al.*, 1994] (30, 86) F. Baader, B. Hollunder, B. Nebel, H. Profitlich, and E. Franconi. An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on. *Journal of Applied Intelligence*, 4:109–132, 1994.

[Baader, 1990] (26, 66) F. Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-90)*, pages 621–626, 1990.

[Baader, 1996] (61) F. Baader. A formal definition for the expressive power of terminological knowledge representation languages. *Journal of Logic and Computation*, 6:33–54, 1996.

[Balsiger and Heuerding, 1998] (29) P. Balsiger and A. Heuerding. Comparison of theorem provers for modal logics — introduction and summary. In H. de Swart, editor, *Proceedings of the Conference on Tableaux Calculi and Related Methods (TABLEAUX), Berlin, Germany*, volume 1397 of *LNAI*, pages 25–26. Springer Verlag, 1998.

[Ben-Ari *et al.*, 1983] (125) M. Ben-Ari, A. Pnueli, and Z. Manna. The temporal logic of branching time. *Acta Informatica*, 20(3):207–226, 1983.

[Berger, 1966] (5) R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:72, 1966.

[Beth, 1953] (100) E. Beth. On Padoa's method in the theory of definition. *Proceedings of the Nederland Akaddemie van Wetenschappen, Series A., 56(4)*, 15:330–339, 1953.

[Bibel and Schmitt, 1998] (74) W. Bibel and P. Schmitt, editors. *Automated Deduction: A Basis for Applications.* Kluwer, 1998.

[Blackburn and Seligman, 1995] (109, 110) P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995. Special issue on decompositions of first-order logic.

[Blackburn and Seligman, 1998] (36, 37) P. Blackburn and J. Seligman. What are hybrid languages? In M. Kracht, M. de Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 1, pages 41–62. CSLI Publications, Stanford University, 1998.

[Blackburn and Spaan, 1993] (39) P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language and Information*, pages 129–169, 1993.

[Blackburn and Tzakova, 1998a] (37, 38, 43, 97, 105) P. Blackburn and M. Tzakova. Hybrid completeness. *Logic Journal of the IGPL*, 6:625–650, 1998.

[Blackburn and Tzakova, 1998b] (96, 137) P. Blackburn and M. Tzakova. Hybrid languages and temporal logics (full version). Technical Report CLAUS-Report 96, Computerlinguistik, Universität des Saarlandes, 1998. `http://www.coli.uni-sb.de/cl/claus`.

[Blackburn and Tzakova, 1998c] (57) P. Blackburn and M. Tzakova. Hybridizing concept languages. *Annals of Mathematics and Artificial Intelligence*, 24:23–49, 1998.

[Blackburn and Tzakova, 1999] (37, 38, 46, 103, 104) P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):27–54, 1999.

[Blackburn *et al.*, 2000] (13) P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2000. To appear.

[Blackburn, 1990] (38) P. Blackburn. *Nominal Tense Logic and Other Sorted Intensional Frameworks.* PhD thesis, Centre for Cognitive Science, University of Edinburgh, Edinburgh, United Kingdom, 1990.

[Blackburn, 1993] (39) P. Blackburn. Modal logic and attribute value structures. In M. de Rijke, editor, *Diamonds and Defaults*, Synthese Language Library, pages 19–65. Kluwer Academic Publishers, Dordrecht, 1993.

[Blackburn, 1994] (39, 70) P. Blackburn. Tense, temporal reference and tense logic. *Journal of Semantics*, 11:83–101, 1994.

[Blackburn, 2000a] (38, 69, 115) P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10:136–168, 2000.

[Blackburn, 2000b] (37) P. Blackburn. Representation, reasoning, and relational structures: a Hybrid Logic manifesto. In Areces et al. [2000c], pages 339–365.

[Börger *et al.*, 1997] (6, 10) E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem.* Springer-Verlag, Berlin, 1997. With an appendix by Cyril Allauzen and Bruno Durand.

[Borgida *et al.*, 1989] (22) A. Borgida, R. Brachman, D. McGuinness, and L. Alperin Resnick. CLASSIC: a structural data model for objects. In *Proceedings of the ACM SIGMOD International Conference of Management of Data*, pages 59–67, 1989.

[Borgida, 1995] (22) A. Borgida. Description logics in data management. *IEEE Transactions on Knowlede and Data Engineering*, 7:671–682, 1995.

[Borgida, 1996] (32, 61) A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82:353–367, 1996.

[Brachman and Levesque, 1984] (24, 28) R. Brachman and H. Levesque. The tractability of subsumption in frame-based description languages. In *AAAI-84*, pages 34–37, 1984.

[Brachman and Schmolze, 1985] (19, 22) R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Brachman *et al.*, 1985] (22) R. Brachman, V. Pigman Gilbert, and H. Levesque. An essential hybrid reasoning system: knowledge and symbol level accounts in KRIPTON. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 532–539, Los Angeles, 1985.

[Brachman, 1977] (22) R. Brachman. What's in a concept: structural foundations for semantic networks. *International Journal of Man-Machine Studies*, 9:127–152, 1977.

[Brachman, 1979] (22) R. Brachman. On the epistemological status of semantic networks. In N. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, 1979.

[Buchheit *et al.*, 1993] (28) M. Buchheit, F. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.

[Bull, 1970] (20, 36, 70) R. Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[Calvanese *et al.*, 1997] (85) D. Calvanese, G. De Giacomo, and M. Lenzerini. Conjunctive query containment in description logics with n-ary relations. In *Proceedings of DL-97*, pages 5–9, June 1997.

[Chang and Keisler, 1990] (52, 101) C. Chang and H. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics.* North-Holland Publishing Co., Amsterdam, third edition, 1990.

[Chlebus, 1986] (5, 116) B. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32(3):374–392, 1986.

[Church, 1936] (5) A. Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1936.

[Clarke and Emerson, 1982] (15) E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logics of Programs (Yorktown Heights, N.Y., 1981)*, pages 52–71. Springer, Berlin, 1982.

[Clarke and Sistla, 1985] (125) E. Clarke and A. Sistla. Complexity of propositional linear temporal logics. *Journal of the Association for Computing Machinery*, 32(3):733–749, 1985.

[Craig, 1957] (99) W. Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22:269–285, 1957.

[Cresswell, 1990] (37, 39) M. Cresswell. *Entities and Indices.* Reidel, Dordrecht, 1990.

[Cresswell, 1996] (39) M. Cresswell. *Semantic Indexicality.* Kluwer, Dordrecht, 1996.

[Czelakowski, 1982] (99) J. Czelakowski. Logical matrices and the amalgamation property. *Studia Logica*, XLI(4):329–341, 1982.

[Davidson *et al.*, 1994] (1) G. Davidson, M. Seaton, and J. Simpson, editors. *The Wordsworth Concise English Dictionary.* Wordsworth Reference. Wordsworth Editions Ltd., Hertfordshire, 1994.

[De Giacomo and Lenzerini, 1994] (49, 54, 55) G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94)*, pages 205–212, 1994.

[De Giacomo, 1995] (49, 109, 119) G. De Giacomo. *Decidability of Class-based Knowledge Representation Formalisms.* PhD thesis, Università di Roma "La Sapienza", 1995.

[de Nivelle *et al.*, 2000] (74) H. de Nivelle, R. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. In Areces et al. [2000c], pages 239–264.

[de Nivelle, 1994] (75, 77) H. de Nivelle. *Ordering Refinements of Resolution.* PhD thesis, Technische Universiteit Delft, Delft. The Netherlands, October 1994.

[de Rijke and Sturm, 2000] (69) M. de Rijke and H. Sturm. Global definability in basic modal logic. In H. Wansing, editor, *Essays on Non-classical Logic.* King's College University Press, 2000.

[de Rijke and van der Hoek, 1995] (14, 68) M. de Rijke and W. van der Hoek. Counting objects. *Journal of Logic and Computation*, 5(3):325–345, 1995.

[de Rijke *et al.*, 2000] (73) M. de Rijke, D. Gabbay, A. Nonnengart, and H. Ohlbach. Encoding two-valued non-classical logics in classical logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning.* Elsevier Science Publishers, 2000. To appear.

[de Rijke, 1992] (14, 69, 114) M. de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57(2):566–584, 1992.

[de Rijke, 2000] (68) M. de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.

[Demri, 1999] (38) S. Demri. Sequent calculi for nominal tense logics: a step towards mechanization? In N. Murray, editor, *Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 140–154. Springer Verlag, 1999.

[Devanbu and Jones, 1997] (23) P. Devanbu and M. Jones. The use of description logics in KBSE systems. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 6(2):141–172, April 1997.

[Dionne *et al.*, 1992] (26) R. Dionne, E. Mays, and F. Oles. A non-well-founded approach to terminological cyles. In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, pages 761–766. AAAI Press / The MIT Press, 1992.

[Dixon *et al.*, 2000] (75) C. Dixon, M. Fisher, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic*, 2000. To appear.

[Donini *et al.*, 1991] (30) F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Tractable concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 458–463, Sidney, 1991.

[Donini *et al.*, 1992] (28, 30) F. Donini, M. Lenzerini, D. Nardi, B. Hollunder, W. Nutt, and A. Marchetti-Spaccamela. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 53(2-3):309–327, 1992.

[Donini *et al.*, 1994] (29) F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in concept languages: from subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[Donini *et al.*, 1997] (28, 30, 34) F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134(1):1–58, 1997.

[Ebbinghaus and Flum, 1999] (7, 12) H-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer-Verlag, Berlin, second edition, 1999.

[Ehrenfeucht, 1961] (6) A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[Emerson and Halpern, 1986] (125) E. Emerson and J. Halpern. "Sometimes" and "not never" revisited: on branching versus linear time temporal logic. *Journal of the Association for Computing Machinery*, 33(1):151–178, 1986.

[Enjalbert and Fariñas del Cerro, 1989] (75, 77, 80, 82) P. Enjalbert and L. Fariñas del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65(1):1–33, 1989.

[F. and Zakharyaschev, 1999] (58) Wolter F. and M. Zakharyaschev. Temporalizing description logics. In M. de Rijke and D. Gabbay, editors, *Frontiers of Combining Systems, (FroCoS'98).* Kluwer Academic Publishers, 1999.

[Fariñas del Cerro, 1982] (77) L. Fariñas del Cerro. A simple deduction method for modal logic. *Information Processing Letters*, 14:49–51, April 1982.

[Feferman, 1968] (91) S. Feferman. Persistent and invariant formulas for outer extensions. *Compositio Mathematica*, 20:29–52, 1968.

[Fermüller *et al.*, 1993] (74) C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Methods for the Decision Problem.* Springer-Verlag, Berlin, 1993. Lecture Notes in Artificial Intelligence.

[Fine, 1972] (68) K. Fine. In so many possible worlds. *Notre Dame Journal Formal Logic*, 13:516–520, 1972.

[Fischer and Ladner, 1979] (49) M. Fischer and R. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[Fitting, 1983] (30, 73) M. Fitting. *Proof Methods for Modal and Intuitionistic Logics.* D. Reidel Publishing Co., Dordrecht, 1983.

[Fitting, 1990] (78) M. Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.

[Fraïssé, 1954] (7) R. Fraïssé. Sur quelques classifications des systèmes de relations. *Université d'Alger, Publications Scientifiques, Série A*, 1:35–182, 1954.

[Franconi *et al.*, 1998] E. Franconi, G. De Giacomo, R. MacGregor, W. Nutt, and C. Welty, editors. *Proceedings of the 1998 International Workshop on Description Logics (DL'98)*, 1998. Available at `http://sunsite.informatik.rwth-aachen.de/publications/ceur-ws`.

[Gabbay *et al.*, 1994] (14) D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic*, volume Volume 1: Mathematical Foundations and Computational Aspects. Oxford University Press, 1994.

[Gabbay, 1976] (39) D. Gabbay. *Investigations in Modal and Tense Logics with Applications to Problems in Philosophy and Linguistics.* Reidel, Dordrecht, 1976.

[Gabbay, 1996] (38) D. Gabbay. *Labelled Deductive Systems. Vol. 1.* The Clarendon Press Oxford University Press, New York, 1996. Oxford Science Publications.

[Gargov and Goranko, 1993] (69, 114) G. Gargov and V. Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22(6):607–636, 1993.

[Gargov and Passy, 1988] (20, 39) G. Gargov and S. Passy. Determinism and looping in combinatory pdl. *Theoretical Computer Science*, 61:259–277, 1988.

[Gargov *et al.*, 1987] (20) G. Gargov, S. Passy, and T. Tinchev. Modal environment for boolean speculations. In D. Scordev, editor, *Mathematical Logic and its Applications*, pages 253–263. Plenum Press, New York, 1987.

[Goble, 1970] (68) L. Goble. Grades of modality. *Logique et Analyse (N.S.)*, 13:323–334, 1970.

[Goranko and Passy, 1992] (14, 53, 69) V. Goranko and S. Passy. Using the universal modality: gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[Goranko, 1994] (37) V. Goranko. Temporal logic with reference pointers. In *Temporal Logic (Bonn, 1994)*, pages 133–148. Springer, Berlin, 1994.

[Goranko, 1996] (37) V. Goranko. Hierarchies of modal and temporal logics with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.

[Goranko, 2000] (70) V. Goranko. Temporal logics with reference pointers and computation tree logics. *Journal of Applied Non-classical Logics*, 2000. To appear.

[Goré, 1999] (30) R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Haenle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[Grädel and Walukiewicz, 1999] (13) E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, 1999.

[Grädel *et al.*, 1997a] (11) E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *The Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[Grädel *et al.*, 1997b] (11) E. Grädel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*, pages 306–317, Warsaw, Poland, 1997. IEEE Computer Society Press.

[Grädel *et al.*, 1999] (11) E. Grädel, M. Otto, and E. Rosen. Undecidability results on two-variable logics. *Archive for Mathematical Logic*, 38(4-5):313–354, 1999. Logic Colloquium '95 (Haifa).

[Grädel, 1999] (13, 138) E. Grädel. Why are modal logics so robustly decidable? *Bulletin of the European Association for Theoretical Computer Science. EATCS*, 68:90–103, 1999.

[Grädel, 2000] (13) E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 2000. To appear.

[Gurevich, 1969] (10) Y. Gurevich. The decision probelm for logic of predicates and operations. *Algebra and Logic*, 8(160–3174), 1969.

[Gurevich, 1976] (5) Y. Gurevich. The decision problem for standard classes. *Journal of Symbolic Logic*, 41(2):460–464, 1976.

[Haarslev and Möller, 1999] (34) V. Haarslev and R. Möller. RACE system description. In Lambrix et al. [1999], pages 130–132.

[Halpern and Moses, 1992] (51, 117) J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[Harel, 1984] (14, 15, 51) D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic. Vol. II*, volume 165 of *Synthese Library*, pages 497–604. D. Reidel Publishing Co., Dordrecht, 1984. Extensions of classical logic.

[Hemaspaandra, 1996] (61) E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):174–203, 1996. Combining logics.

[Henkin, 1967] (10) L. Henkin. *Logical Systems Containing Only a Finite Number of Symbols*, volume 21 of *Séminaire de Mathématiques Supérieures.* Les Presses de l'Université de Montréal, 1967.

[Heuerding and Schwendimann, 1996] (29) A. Heuerding and S. Schwendimann. A benchmark method for the propositional modal logics K, KT , S4. Technical Report IAM-96-015, University of Bern, Switzerland, 1996.

[Hoogland *et al.*, 1999] (13) E. Hoogland, M. Marx, and M. Otto. Beth definability for the guarded fragment. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, 6th International Conference LPAR99, Tbilisi, Georgia*, number 1705 in LNAI, pages 273–285. Springer, 1999.

[Horrocks and Gough, 1997] (51) I. Horrocks and G. Gough. Description logics with transitive roles. In M. Rousset, R. Brachmann, F. Donini, E. Franconi, I. Horrocks, and A. Levy, editors, *Proceedings of the International Workshop on Description Logics*, pages 25–28, Gif sur Yvette, France, 1997.

[Horrocks and Patel-Schneider, 1999] (87) I. Horrocks and P. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3), 1999.

[Horrocks and Sattler, 1999] (51, 109) I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[Horrocks *et al.*, 1999] (34) I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In Harald Ganzinger, David McAllester, and Andrei Voronkov, editors, *Proceedings of the 6th International Conference on Logic for Programming and*

Automated Reasoning (LPAR'99), number 1705 in Lecture Notes in Artificial Intelligence, pages 161–180. Springer-Verlag, September 1999.

[Horrocks *et al.*, 2000a] (29, 74, 87) I. Horrocks, P. Patel-Schneider, and R. Sebastiani. An analysis of empirical testing for modal decision procedures. In Areces et al. [2000c], pages 293–324.

[Horrocks *et al.*, 2000b] (34, 55, 69) I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. In Areces et al. [2000c], pages 239–264.

[Horrocks *et al.*, 2000c] (34, 68) I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In David MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, Lecture Notes in Computer Science, Germany, 2000. Springer Verlag.

[Horrocks, 1999] (34) I. Horrocks. FaCT and iFaCT. In Lambrix et al. [1999], pages 133–135. FACT is available under the GNU public license at `http://www.cs.man.ac.uk/~horrocks`.

[Hustadt, 1999] (74) U. Hustadt. *Resolution-Based Decision Procedures for Subclasses of First-Order Logic.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.

[Immerman and Kozen, 1989] (12) N. Immerman and D. Kozen. Definability with bounded number of bound variables. *Information and Computation*, 83(2):121–139, 1989.

[Kalmár, 1936] (9) L. Kalmár. Zurückführung des Entscheidungsproblems auf den Fall von Formeln mit einer einzigen, binären Funktionsvariablen. *Composition Mathematica*, 4:137–144, 1936.

[Kamp, 1971] (39) H. Kamp. Formal properties of "now". *Theoria*, 37:227–273, 1971.

[Konikowska, 1997] (39) B. Konikowska. A logic for reasoning about relative similarity. *Studia Logica*, 58:185–226, 1997.

[Kozen, 1983] (15) D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.

[Kracht, 1999] (65, 100) M. Kracht. *Tools and Techniques in Modal Logic.* North-Holland Publishing Co., Amsterdam, 1999.

[Kurtonina and de Rijke, 1999] (61, 62) N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2):303–333, 1999.

[Ladner, 1977] (30, 51, 116, 120, 126) R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.

[Lambrix *et al.*, 1999] P. Lambrix, A. Borgida, R. Möller M. Lenzerini, and P. Patel-Schneider, editors. *Proceedings of the 1999 International Workshop on Description Logics (DL'99)*, 1999.

[Lewis, 1979] (6) H. Lewis. *Unsolvable Classes of Quantificational Formulas.* Addison-Wesley Publishing Co., Reading, Mass., 1979.

[Löwenheim, 1915] (9) L. Löwenheim. Über Möglichkeiten im Relativkalkül. *Math. Annalen*, 76:447–470, 1915. English translation in [van Heijenoort, 1967].

[Lutz, 1999a] (30, 60) C. Lutz. Complexity of terminological reasoning revisited. In *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning LPAR'99*, Lecture Notes in Artificial Intelligence, pages 181–200. Springer-Verlag, September 6 – 10, 1999.

[Lutz, 1999b] (30, 60) C. Lutz. On the complexity of terminological reasoning. LTCS-Report 99-04, LuFg Theoretical Computer Science, RWTH Aachen, Germany, 1999.

[MacGregor and Bates, 1987] (22) R. MacGregor and R. Bates. The Loom knowledge representation language. Technical Report ISI/RS-87-188, Information Science Institute, University of Southern California, Marina del Rey, California, 1987.

[Maksimova, 1991a] (64, 105) L. Maksimova. Amalgamation and interpolation in normal modal logics. *Studia Logica*, L(3/4):457–471, 1991.

[Maksimova, 1991b] (64, 100) L. Maksimova. Beth properties, interpolation and amalgamation in varieties of modal algebras. *Doklady Akademii Nauk SSSR*, 319(6):1309–1312, 1991. Translation in *Soviet Mathematics. Doklady.* 44:1 (1992), 327–331.

[Maksimova, 1992a] (64) L. Maksimova. An analogue of Beth's theorem in normal extensions of the modal logic k4. *Rossiĭskaya Akademiya Nauk. Sibirskoe Otdelenie. Sibirskiĭ Matematicheskiĭ Zhurnal*, 33(6):118–130, 231, 1992. translation in *Siberian Mathematical Journal*, 33:6, 1052–1065 (1993).

[Maksimova, 1992b] (64) L. Maksimova. Definability and interpolation in classical modal logics. In *Proceedings of the International Conference on Algebra, Part 3 (Novosibirsk, 1989)*, pages 583–599, Providence, RI, 1992. Amer. Math. Soc.

[Maksimova, 1992c] (64) L. Maksimova. Modal logics and varieties of modal algebras: the Beth, interpolation, and amalgamation properties. *Akademiya Nauk SSSR. Sibirskoe Otdelenie. Institut Matematiki. Algebra i Logika*, 31(2):145–166, 220–221, 1992. Translation in *Algebra and Logic*, 31:2, 90–105 (1993).

[Manna and Pnueli, 1992] (124) Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, New York, 1992. Specification.

[Marx and Venema, 1997] (39) M. Marx and Y. Venema. *Multidimensional Modal Logic*, volume 4 of *Applied Logic Series.* Kluwer Academic Publishers, Dordrecht, 1997.

[Marx and Venema, 2000] M. Marx and Y. Venema. Local variations on a loose theme: modal logic and decidability. In M. Vardi and S. Weinstein, editors, *Finite Model Theory and its Applications*. Springer-Verlag, 2000. To appear.

[Marx, 1999] (104) M. Marx. Interpolation in (fibered) modal logic. In A. Haeberer, editor, *Proceedings of AMAST 1998*, 1999.

[McTaggart, 1908] (37) J. McTaggart. The unreality of time. *Mind*, pages 457–474, 1908.

[Meghini *et al.*, 1993] (23) C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In R. Korfhage et al., editor, *Proceedings of SIGIR-93*, pages 298–307. ACM Press, Baltimore, 1993.

[Minsky, 1974] (21) M. Minsky. A framework for representing knowledge. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, pages 246–262. Morgan-Kaufmann, Los Altos, 1974.

[Mints, 1989] (75, 77) G. Mints. Resolution calculi for modal logics. *American Mathematical Society Translations*, 143:1–14, 1989.

[Monk, 1969] (12) J. D. Monk. Nonfinitizability of classes of representable cylindric algebras. *Journal of Symbolic Logic*, 34:331–343, 1969.

[Monk, 1993] (12) J. Monk. Lectures on cylindric set algebras. In C. Rauszer, editor, *Algebraic Methods in Logic and in Computer Science*, volume 28, pages 253–290. Banach Center Publications, Polish Academy of Sciences, Warsaw, 1993.

[Mortimer, 1975] (10) M. Mortimer. On languages with two variables. *Z. Math. Logik Grundlagen Math.*, 21:135–140, 1975.

[Nebel, 1990a] (26, 66) B. Nebel. Terminological cycles: semantics and computational properties. In J. Sowa, editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, Los Altos, 1990.

[Nebel, 1990b] (29) B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235–249, 1990.

[Ockham, 1969] (125) W. of Ockham. *Predestination, God's Foreknowledge, and Future Contingents.* Appleton-Century-Crofts, New York, 1969.

[Ono and Nakamura, 1980] (125) H. Ono and A. Nakamura. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 34:325–333, 1980.

[Orłowska, 1997] (39) E. Orłowska, editor. *Incomplete Information: Rough Set Analysis.* Studies in Fuzziness and Soft Computing. Physica-Verlag, 1997.

[Pacholski *et al.*, 1997] (11) L. Pacholski, W. Szwast, and L. Tendera. Complexity of two-variable logic with counting. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*, pages 318–327, Warsaw, Poland, 1997. IEEE Computer Society Press.

[Passy and Tinchev, 1985a] (20, 38, 57) S. Passy and T. Tinchev. PDL with data constants. *Information Processing Letters*, 20(1):35–41, 1985.

[Passy and Tinchev, 1985b] (20, 39) S. Passy and T. Tinchev. Quantifiers in combinatory PDL: completeness, definability, incompleteness. In *Fundamentals of Computation Theory FCT 85*, volume 199 of *LNCS*, pages 512–519. Springer, 1985.

[Passy and Tinchev, 1991] (38, 39, 69, 109, 119) S. Passy and T. Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.

[Patel-Schneider, 1998] (34) P. Patel-Schneider. DLP system description. In Franconi et al. [1998], pages 87–89. DLP is available at `http://www.bell-labs.com/user/pfps`.

[Pnueli, 1977] (124) A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, pages 46–57. IEEE Comput. Sci., Long Beach, Calif., 1977.

[Prior, 1967] (20, 36, 37) A. Prior. *Past, Present and Future.* Oxford University Press, 1967.

[Prior, 1968] (36) A. Prior. Now. *Nous*, 2:101–119, 1968.

[Prior, 1977] (37) A. Prior. *Papers on Time and Tense.* University of Oxford Press, 1977.

[Quantz and Kindermann, 1990] (22) J. Quantz and C. Kindermann. Implementation of the BACK system, version 4. Technical Report KIT-Report 78, FB Informatik, Technische Universität Berlin, Berlin, Germany, 1990.

[Quillian, 1967] (21) M. Ross Quillian. Word concepts: a theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967.

[Rautenberg, 1983] (30, 65) W. Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12:403–423, 1983.

[Reape, 1994] (39) M. Reape. A feature value logic. In C. Rupp, M. Rosner, and R. Johnson, editors, *Constraints, Language and Computation*, Synthese Language Library, pages 77–110. Academic Press, 1994.

[Renardel de Lavalette, 1989] (99) G. Renardel de Lavalette. Modularisation, parametrisation and interpolation. *Journal of Information Processing and Cybernetics*, 25(5/6):283–292, 1989.

[Rescher and Urquhart, 1971] (36) N. Rescher and A. Urquhart. *Temporal Logic.* Springer, 1971.

[Richards *et al.*, 1989] (37) B. Richards, I. Bethke, J. van der Does, and J. Oberlander. *Temporal Representation and Inference.* Academic Press, London, 1989.

[Robinson and Voronkov, 2000] (74) A. Robinson and A. Voronkov, editors. *Handbook of Automated Reasoning.* Elsevier Science Publishers B. V., 2000. To appear.

[Robinson, 1965] (74) J. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, 12(1):23–41, January 1965.

[Rounds, 1997] (39) W. Rounds. Feature logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 475–533. North-Holland Publishing Co., Amsterdam, 1997.

[Russell and Norvig, 1995] (23) S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

[Sattler, 1996] (51) U. Sattler. A concept language extended with different kinds of transitive roles. In G. Görz and S. Hölldobler, editors, *20. Deutsche Jahrestagung für Künstliche Intelligenz*, number 1137 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1996.

[Schaerf, 1994] (30, 31, 34) A. Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues.* PhD thesis, Università di Roma "La Sapienza", 1994.

[Schild, 1991] (49) K. Schild. A correspondence theory for terminological logics. In *Proceedings of the 12th IJCAI*, pages 466–471, 1991.

[Schmidt-Schauss and Smolka, 1991] (25, 28, 30) M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[Scott, 1962] (10) D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:377, 1962.

[Seligman, 1991] (38, 115) J. Seligman. A cut-free sequent calculus for elementary situated reasoning. Technical Report HCRC-RP 22, HCRC, Edinburgh, 1991.

[Seligman, 1997] (38) J. Seligman. The logic of correct description. In M. de Rijke, editor, *Advances in Intensional Logic*, pages 107–135. Kluwer, 1997.

[Sellink, 1994] (37) M. Sellink. Verifying modal formulae over I/O-automata by means of type theory. Logic group preprint series, Utrecht University, 1994.

[Spaan, 1993] (59, 61, 110, 117, 130, 138) E. Spaan. *Complexity of Modal Logics.* PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 1993.

[Tobies, 1999] (68) S. Tobies. A PSpace algorithm for graded modal logic. In *Automated Deduction—CADE-16 (Trento, 1999)*, pages 52–66. Springer, Berlin, 1999.

[Tobies, 2000a] (34, 68, 130) S. Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research*, 2000. To appear.

[Tobies, 2000b] (34) S. Tobies. PSPACE reasoning for graded modal logics. *Journal of Logic and Computation*, 2000. To appear.

[Turing, 1937] (5) A. Turing. On computable numbers, with an application to the 'Entscheidungsproblem'. *Proceedings of the London Mathematical Society 2nd. series*, 42:230–265, 1937.

[Tzakova, 1999a] (36, 37, 45, 46, 69, 96, 134) M. Tzakova. *Hybrid Languages.* PhD thesis, Technischen Fakultät der Universität des Saarlandes, 1999.

[Tzakova, 1999b] (38) M. Tzakova. Tableaux calculi for hybrid logics. In N. Murray, editor, *Proceedings of the Conference on Tableaux Calculi and Related Methods (TABLEAUX), Saratoga Springs, USA*, volume 1617 of *LNAI*, pages 278–292. Springer Verlag, 1999.

[van Benthem, 1983] (42, 52, 91) J. van Benthem. *Modal Logic and Classical Logic.* Bibliopolis, Naples, 1983.

[van Benthem, 1995] (12) J. van Benthem. Temporal logic. In D. Gabbay, C. Hoggar, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 241–350. Oxford University Press,, 1995.

[van Benthem, 1996] (93, 95) J. van Benthem. *Exploring Logical Dynamics.* Studies in Logic, Language and Information. CSLI Publications, Stanford, 1996.

[van Heijenoort, 1967] J. van Heijenoort. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931.* Harvard University Press, Cambridge, Massachusetts, 1967.

[Vardi, 1997] (12, 15, 138) M. Vardi. Why is modal logic so robustly decidable? In *Descriptive Complexity and Finite Models (Princeton, NJ, 1996)*, pages 149–183. Amererican Mathematical Society, Providence, RI, 1997.

[Venema, 1991] (103, 114) Y. Venema. *Many-dimensional modal logic.* PhD thesis, Institute for Logic, Language and Computation. University of Amsterdam, 1991.

[Venema, 1993] (114) Y. Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58:1003–1034, 1993.

[Venema, 1994] (43) Y. Venema. A modal logic for quantification and substitution. *Bulletin of the IGPL*, 2(1):31–45, 1994.

[Vlach, 1973] (39) F. Vlach. *Now and Then: a formal study in the logic of tense anaphora.* PhD thesis, UCLA, 1973.

[Wadge, 1975] (38) W. Wadge. A complete natural deduction system for the relational calculus. Research Report CS-RR-005, Department of Computer Science, University of Warwick, Coventry, UK, March 1975.

[Wolter and Zakharyaschev, 1998] (58) F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In A. Cohn, L. Schubert, and S. Shapiro, editors, *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 512–552. Morgan Kaufman, 1998.

[Wolter and Zakharyaschev, 1999a] (58) F. Wolter and M. Zakharyaschev. Dynamic description logics. In M. de Rijke, K. Segerberg, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic, vol 2*, Stanford, 1999. CSLI Publications.

[Wolter and Zakharyaschev, 1999b] (58) F. Wolter and M. Zakharyaschev. Multi-dimensional description logics. In *Proceedings of IJCAI'99*, pages 104–109, Stockholm, 1999. Morgan Kaufman.

[Wolter and Zakharyaschev, 2000] (58) F. Wolter and M. Zakharyaschev. Modal description logics: modalizing roles. *Fundamenta Informaticae*, 2000. To appear.

[Wolter, 1999] (58) F. Wolter. *The Decision Problem for Combined (Modal) Logics.* Institut für Informatik, Universität Leipzig, 1999. Habilitationsschrift.

[Woods, 1975] (22) W. Woods. What's in a link: foundations for semantic networks. In D. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–82. Academic Press, 1975.

[Wright *et al.*, 1993] (23) J. Wright, E. Weixelbaum, K. Brown, G. Vesonder, S. Palmer, J. Berman, and H. Moore. A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T Network Systems. In *Proceedings of the 5th. Innovative Applications of Artificial Intelligence Conference (IAAI-93)*, pages 183–193, Washington, 1993.

Index