Algebraizing Hybrid Logic

Evangelos Tzanis
University of Amsterdam
Institute of Logic, Language and Computation
etzanis@science.uva.nl

May 1, 2005

Contents

1 Introduction **5**
1.1 A guide to this thesis 6

2 Preliminaries **7**
2.1 Introduction 7
2.2 Basic Hybrid Logic 7
2.3 Hybrid Languages with Quantifiers $\mathcal{H}(@, \forall, \exists)$ 9
2.4 Tableau 10
2.5 Translation 13
2.6 Frame Definability 14
2.7 Elements of Universal Algebra 16
2.8 Skolemization 19
2.9 Notation 22

3 Extended Language **23**
3.1 Introduction 23
3.2 $\mathcal{HLON}$ 23
3.3 Bisimulation and Elementary Equivalence 25
3.4 Undefinable Properties 27

4 $\mathcal{HLOV}(@, \forall, \exists)$ **29**
4.1 Introduction 29
4.2 $\mathcal{HLOV}(@, \forall, \exists)$ 29
4.3 $\mathcal{HLON}$ and $\forall - \mathcal{HLOV}$ 31

5 Skolem Functions **33**
5.1 Introduction 33
5.2 Skolemization in Hybrid Logic 33
5.3 Frame Definability and $\mathcal{HLON}$ 35

5.4 Universal equalities 38

6 Tableau **41**
6.1 Motivation 41
6.2 Proofs 43

7 Equational Logic, Term Rewriting and $\mathcal{HLON}$ **47**
7.1 Proof systems 49
7.2 Hyrbid Languages and Equations 51
7.3 Hybrid Logic and Algebras 52

8 Conclusions and Further work **53**

Chapter 1

Introduction

Hybrid logic is the result of extending the basic modal language with a second sort of atomic propositions called nominals, and with satisfaction operators. Precisely, the nominals (denoted by $i, j, ...$) behave similar to ordinary proposition letters, expect that nominals are true uniquely at a world. In other words, a nominal names a state by being true there and nowhere else. An example of a formula involving nominals is $\Diamond\Diamond i \rightarrow \neg\Diamond i$. The language obtained by adding nominals to the basic modal language, is called the *minimal hybrid logic* $\mathcal{H}$. Satisfaction operators allow one to express that a formula holds at the world named by nominal. A formula of the form $@_i\phi$ expresses that p holds at the world named by the nominal i. The extension of the basic modal language with nominals and satisfaction operators is called the *basic hybrid language* $\mathcal{H}(@)$.

In this thesis we introduce and study an extension of hybrid logic in which the set of nominals may be endowed with an algebraic structure. In other words we add modal operators only for nominals. The main motivation of the paper comes from [4]: *you can name states but you can not give them structure.* In this paper we consider an application of hybrid logics to relational structures on algebras, thus a set with a relation and an algebraic structure. Roughly speaking we try to give to nominals a structure, we study the case where this structure is an algebraic structure. As far as we know, the possible algebraic structure of nominals has not been studied in the context of hybrid logic before. We should note that in [11] there is a complete list of papers which study Kripke frames in which the universe of possible worlds has a specific algebraic structure, besides the traditional relational component.

1.1 A guide to this thesis

The thesis consists of six chapters. A brief overview of each of these parts follows.

Chapter 2: This chapter begins with a review of the basics of two hybrid logics $\mathcal{H}(@)$ and $\mathcal{H}(@, \forall, \exists)$, their language and semantics. We define notions that will be discussed further later in the thesis such as: truth preserving translations, the process of skolemizing a first order formula and elementary notions of Universal Algebra.

Chapter 3: We name a version of the basic Hybrid Logic ($\mathcal{H}(@)$) and we call it $\mathcal{HLON}$ (**H**ybrid **L**ogic with **O**perators only for **N**ominals) henceforth. We set up the syntax and we give the notions of Kripke model and validity to it. We give a notion of bisimulation for this language. We provide a standard translation from $\mathcal{HLON}$ to First Order Logic ($\mathcal{FOL}$).

Chapter 4: We name a version of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$ and we call it $\mathcal{HLOV}(@, \forall, \exists)$ (**H**ybrid **L**ogic with **O**perators only for **V**ariables). We set up its language and semantics.

Chapter 5: In this chapter we introduce a natural method for eliminating existential quantifiers in the context of $\mathcal{H}(@, \forall, \exists)$ and $\mathcal{HLON}$. This method is quite similar with the technique of skolemization in first order logic.

Chapter 6: We deal with tableau proof methods applied to $\mathcal{HLON}$. We motivate this result with examples from [17] . Again the technique of skolemization is central here. Note that an important reason for growing popularity of hybrid languages is a general completeness result for logics axiomatized by pure formulas. We will show that this is also the case for $\mathcal{HLON}$.

Chapter 7 We relate our contribution with the following area: Equational Logic. The main motivation for this section comes from [4].

Chapter 8 The thesis ends with a discussion of possible directions in which this thesis can be extended.

Main Result: In this thesis we set up a new hybrid logic $\mathcal{HLON}$ which proposes alternatives to the modal operators by having uninterpreted functions which fulfill the role of existential quantifiers. Based on this, we show that one of the traditional fragments of hybrid logic becomes equivalent to first order logic.

Chapter 2

Preliminaries

2.1 Introduction

Chapter 2: This chapter begins with a review of the basics of two hybrid logics $\mathcal{H}(@)$ and $\mathcal{H}(@, \forall, \exists)$, their language and semantics. We define notions that will be discussed further later in this thesis.

2.2 Basic Hybrid Logic

This section is a short introduction of hybrid logic based on [4]. Hybrid Logic is the result of extending the basic modal language with a second sort of atomic propositions called nominals and with satisfaction operators. The nominals are predicates which are true uniquely at a world, and in that sense are predicates naming worlds. Thus nominals' interpretation in models is restricted to singleton sets. Satisfaction operators allow one to express that a formula holds at the world named by nominal. For instance, $@_i p$ expresses that p holds at the world named by the nominal i.

Formally, let **PROP**$= \{p_1, p_2, \cdots\}$ be a countably infinite set of proposition letters and **NOM**$= \{i, i_1, \cdots\}$ be a countably infinite set of nominals. Let $\Phi =$ **PROP** $\cup$ **NOM**. Note that elements of **NOM** are still propositional variables.

Definition 2.2.1. $\mathcal{H}(@)$ **MODELS**.

A model *for the basic hybrid language is a Kripke model* $\mathcal{M} = (W, R_\Diamond, V)$*, such that the valuation* V *assigns singleton subsets of* W *to nominals.*

- $W \neq 0$*, a set of possible worlds;*

- *$R_\Diamond$ is a binary relation over W;*
- *V: $\Phi \to Pow(W)$ and V is such that for all nominals $i \in$ **NOM**, $V(i)$ is singleton of W.*

Following standard terminology we call the pair $(W, R_\Diamond)$ the *frame* underlying the model.

For any model $\mathcal{M}$ and world w in the domain of $\mathcal{M}$, the model $\mathcal{M}, w$ is called pointed model.

Definition 2.2.2. *The Hybrid Language $\mathcal{H}(@)$ is language that is used for describing models and frames. Its formulas are given by the following recursive definition.*

$$\varphi := i \mid p \mid \neg\varphi \mid \varphi \to \psi \mid \Diamond\varphi \mid @_i\varphi$$

where $p \in PROP$ and $i \in NOM$.

Definition 2.2.3. Semantics for $\mathcal{H}(@)$

Given such a hybrid model $\mathcal{M}$, we interpret our language as follows:

$$\begin{array}{lll}
\mathcal{M}, w \models \alpha & \textit{iff} & w \in V(i), \textit{ where } i \in \Phi \\
\mathcal{M}, w \models \varphi \to \psi & \textit{iff} & \mathcal{M}, w \not\models \varphi \textit{ or } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \neg\varphi & \textit{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \Diamond\varphi & \textit{iff} & \textit{there is a } v \in W \textit{ such that } R_\Diamond(w, v) \textit{and} \\
 & & \mathcal{M}, v \models \varphi. \\
\mathcal{M}, w \models @_i\varphi & \textit{iff} & \mathcal{M}, v \models \varphi \textit{ where } V(i) = \{v\}
\end{array}$$

Note that the truth definition for nominals is the same as for proposition letters. The only difference is in the admissible valuations: only valuation functions are allowed that assign to each nominal a singleton set.

We proceed with basic definitions.

Definition 2.2.4. *If $\mathcal{M}, w \models \phi$ then we say that ϕ is* satisfied *in $\mathcal{M}$ at w.*

Definition 2.2.5. *For any frame $\mathcal{F}$, if ϕ is satisfied in every model $(\mathcal{F}, V)$ at every w in $\mathcal{F}$ no matter which (hybrid) valuation V we choose, then we say that ϕ is* valid on $\mathcal{F}$. *A formula is* valid *if it is valid on every frame. A formula is* valid on a class of frames F *if it is valid on every frame in* F.

Definition 2.2.6. *A pure formula is a formula that contains no propositional letters.*

Example 2.2.1. *Many properties of frames can be defined using pure formulas. For example, transitivity of R could be defined as follows:*

$$@_i(\diamond \diamond j \rightarrow \diamond j)$$

That is, this formula is valid on every transitive frame and falsifiable on every non-transitive frame.

Syntax and semantics of $\mathcal{H}(@)$ can be found in [4] which is available in [14]. Note also that orthodox modal languages do not let us refer to individual states, worlds, times and situations. Hybrid languages solve this problem based on the notion of nominals (more about the use of hybrid logics please look in [4]). Moreover, enriching ordinary propositional modal logic with both nominals and satisfaction operators does not raise the computational complexity:

Theorem 2.2.1. *[23] The satisfiability problem for the basic hybrid language is* **PSPACE-***complete.*

2.3 Hybrid Languages with Quantifiers $\mathcal{H}(@, \forall, \exists)$

Let **NOM** be a countable set of nominals **NOM**= $\{i_1, i_2, ...\}$, let **SVAR** be a countable set o state variables **SVAR**= $\{x_1, x_2...\}$ and let **PROP** be a countable set of propositional letters. We assume that these sets are pairwise disjoint. We call **SSYM**= **NOM** $\cup$ **SVAR** the set of state symbols and Φ =**PROP** $\cup$ **NOM** $\cup$ **SVAR**.

Definition 2.3.1. $\mathcal{H}(@, \forall, \exists)$ **MODELS**.

A model $\mathcal{M}$ for $\mathcal{H}(@, \forall, \exists)$ is $\mathcal{M} = (W, R_\diamond, V)$ such that:

- *$W \neq 0$, a set of possible worlds;*
- *$R_\diamond$ is a binary relation over W;*
- *V: $\Phi \rightarrow$ Pow(W) and V is such that for all nominals $i \in$ **NOM**, V(i) is singleton of W.*

*An assignment g for $\mathcal{M}$ is a mapping g : **SVAR**$\rightarrow \mathcal{M}$. Given an assignment g, the notation $g[x \mapsto d]$ is the function that assigns d to x and differs from g at most with regard to the value x.*

Let $[V, g](\alpha) = g(\alpha)$ if α is a state variable and $\mathrm{V}(\alpha)$ otherwise.

Definition 2.3.2. *Formulas of the Hybrid Language $\mathcal{H}(@, \forall, \exists)$ are given by the following recursive definition:*

$$\varphi := \top \mid s \mid p \mid \neg\varphi \mid \varphi \rightarrow \psi \mid \Diamond\varphi \mid @_s\phi \mid \forall x\phi \mid \exists x\phi$$

*where $p \in$ **PROP**, $x \in$ **SVAR**, $s \in$ **SSYM**.*

Remark 2.3.1. *All types of atomic symbol are formulas. Note that the difference between nominals and state variables is: nominals can not be bound by $\forall, \exists$, whereas state variables can.*

Definition 2.3.3. Semantics for $\mathcal{H}(@, \forall, \exists)$

Let a model $\mathcal{M} = (W, R_\Diamond, V)$ for $\mathcal{H}(@, \forall, \exists)$ and an assignment g such that as (the semantics follow the lines of $\mathcal{H}(@)$, so we just give a few clauses of the definition here, leaving the remainder to the reader):

$$\begin{array}{lll} \mathcal{M}, g, w \models \alpha & \textit{iff} & w \in [V, g](\alpha), \ \textit{where } a \in \Phi \\ \mathcal{M}, g, w \models @_x\varphi & \textit{iff} & \mathcal{M}, g, g(x) \models \varphi \\ \mathcal{M}, g, w \models \forall x\phi & \textit{iff} & (\mathcal{M}, g[x \mapsto d]) \models \phi \ \textit{where for all} \ \ d \in W \end{array}$$

2.4 Tableau

In this section we provide basic definitions and theorems that we shall use later in this thesis. Our main purpose is to define a notion of systematic tableau construction general enough to establish completeness for countable languages. More about hybrid tableau reasoning please read [5]. We will now give the set of tableau rules for the $\mathcal{H}(@)$. First the rules for the booleans and for the satisfaction operators:

$$\frac{@_s\neg\varphi}{@_s\varphi}\ [\neg] \qquad \frac{\neg@_s\neg\varphi}{@_s\varphi}\ [\neg\neg]$$

$$\frac{@_s(\varphi \wedge \psi)}{\begin{array}{c}@_s\varphi \\ @_s\psi\end{array}}\ [\wedge] \qquad \frac{\neg@_s(\varphi \wedge \psi)}{\neg@_s\varphi \ \mid \ \neg@_s\psi}\ [\neg\wedge]$$

$$\frac{@_s@_t\varphi}{@_t\varphi}\ [@] \qquad \frac{\neg@_s@_t\varphi}{\neg@_t\varphi}\ [\neg@]$$

In these rules s, t are nominals. Note that these rules use the resources available in hybrid logic to mimic the Kripke satisfaction definition: They draw conclusions from the input to each rule (the formula(s) above the horizontal line) to the output (the formula(s) below the line). For instance, the $\wedge-$ says that if $\phi \wedge \psi$ is true at s, then both ϕ and ψ are true at s. The rule $\neg\wedge$ is branch rule saying that if $\phi \wedge \psi$ is false at s, then either ϕ or ψ is false at s.

In order to formulate modal theories of state equality and state succession the following rules are needed (rewrite rules):

$$\frac{[s \text{ on branch}]}{@_s s}\ \text{[Ref]} \qquad \frac{@_t s}{@_s t}\ \text{[Sym]} \qquad \frac{@_s t \quad @_t \varphi}{@_s \varphi}\ \text{[Nom]} \qquad \frac{@_s \Diamond t \quad @_t t'}{@_s \Diamond t'}\ \text{[Bridge]}$$

By s `on branch` in the statement of **Ref** we simply mean that some formula on the branch in question contains an occurrence of s.

Rules related to modalities $\Diamond$ and $\Box$ are:

$$\frac{@_s \Diamond \varphi}{\begin{array}{l} @_s \Diamond \tau \\ @_\tau \varphi \end{array}}\ [\Diamond] \qquad \frac{\neg @_s \Diamond \varphi \quad @_s \Diamond t}{\neg @_t \varphi}\ [\neg\Diamond]$$

$$\frac{@_s \Box \varphi \quad @_s \Diamond t}{@_t \varphi}\ [\Box] \qquad \frac{\neg @_s \Box \varphi}{\begin{array}{l} @_s \Diamond \tau \\ \neg @_\tau \varphi \end{array}}\ [\neg\Box]$$

where τ is a new nominal. That is, $@_s \Diamond \varphi$ means that we can make a transition by the relation R from s to some state and then at this $R-$ successor state, φ is true there. We can capture this idea by the $\Diamond-$ rule where we need to introduce a new state by the label τ and to insist that φ is true at τ.

As with any tableau system, we prove formulas by systematically trying to falsify them. Suppose we want to prove ϕ. We choose a nominal i that does not occur in ϕ, prefix ϕ with $\neg @_i \phi$ and start applying rules. In case that tableau closes (that is, if every branch contains some formula and its negation), then ϕ is proved. On the other hand, suppose we reach a stage where we have applied the appropriate connective rule to every complex formula and no application of the rewrite rules yields anything new. If the tableau we have constructed contains

open branches , then ϕ is not valid and hence not provable. Note that the near atomic satisfaction statements on the open branch specify a counter model.

Recall that every formula in a tableau is of the form $@_s\phi$ or $\neg@_s\phi$ and that such formulas are called satisfaction statements. Suppose that Σ is a set of satisfaction statements, and that $\mathcal{R}$ is one of our tableau rules. Then:

- If $\mathcal{R}$ is not a branching rule, and $\mathcal{R}$ takes a single formula as input, and Σ^+ is the set obtained by adding to Σ all the formulas (there are two) yielded by applying $\mathcal{R}$ to $\sigma_1 \in \Sigma$ then we say that Σ^+ is a result of expanding Σ by $\mathcal{R}$.
- If $\mathcal{R}$ is a binary rule, then Σ^+ is the set obtained by adding to Σ the formula yielded by applying $\mathcal{R}$.
- If $\mathcal{R}$ is a branching rule and Σ^+ is a set obtained by adding to Σ the formula yielded by one of the two possible outcomes of applying $\mathcal{R}$ to $\sigma_1 \in \Sigma$, then we say that Σ^+ is a result of expanding Σ by $\mathcal{R}$.
- If a nominal s belongs to some formula in Σ, then $\Sigma \cup \{@_s s\}$ is a result of expanding Σ by **Ref**.

Definition 2.4.1. *(***Consistency and provability***) A branch of a tableau is closed iff it contains some satisfaction statement and its negation, and a branch which is not closed is open. A tableau is closed iff all its branches are closed. A formula ϕ is provable iff there is a closed tableau whose root node is $\neg@_j\phi$ (i, can be any nominal not occurring in ϕ) and ϕ is consistent iff $\neg\phi$ is not provable. A set of formulas Σ is consistent iff for any finite subset Σ^f of Σ, the conjunction of all the formulas in Σ^f is consistent.*

Definition 2.4.2. *(***Satisfied by label***) Suppose that Σ is a set of satisfaction statements and that $\mathcal{M} = (W, R, V)$ is a standard model of $\mathcal{H}(@)$. We say that Σ is satisfied by label in $\mathcal{M}$ iff for all formulas in Σ:*

- *If $@_s\phi \in \Sigma$, then $\mathcal{M}, w \models \phi$ and*
- *If $\neg@_s\phi \in \Sigma$ then $\mathcal{M}, w \not\models \phi$.*

Note that w is the denotation of s under V. We say that Σ is satisfiable by label iff there is a standard model in which it is satisfied by label.

Lemma 2.4.1. *(***Soundness***) Suppose Σ is a set of satisfaction statements that is satisfiable by label. Then, for any $\mathcal{R}$, at least one of the sets obtainable by expanding Σ by $\mathcal{R}$ is satisfiable by label.*

Proof. The only non-trivial case are when $\mathcal{R}$ is an existential rule. But even this is straightforward. □

Remark 2.4.1. *If the formula $@_s\phi$ at the root of the tree is satisfiable, then trivially it is satisfiable by label. But the preceding lemma ensures that all formulas on at least one branch of the tableau will be satisfiable by label too.*

Remark 2.4.2. *The systematic tableau construction is defined in [5]. Roughly speaking, this construction is needed in order to prove strong completeness.*

Theorem 2.4.1. *([5]) Any consistent set of formulas in countable language is satisfiable in a countable standard model.*

The following theorem is the most celebrated result in hybrid languages. Roughly speaking, once a base system has been defined, it is easy to extend it to a complete system for many important classes of frames by adding pure formulas as axioms.

Theorem 2.4.2. *([5]) Let $\mathcal{A}$ be a finite collection of $@-$ prefixed formulas. Let TA be the tableau system given above extended with the following rule: at any stage in the tableau, we are free to choose a formula from $\mathcal{A}$, instantiate it with nominals occurring on some branch B, and then add the result to the end of branch B. Then TA is complete with respect to the class of frames defined by $\bigwedge_{\phi\in A}\phi$*

2.5 Translation

In this section we will answer the question: *what does it mean for a logic to be equivalent or translatable into another one?*

The answer comes from the following definition:

Definition 2.5.1. *We say that a language $\mathcal{L}_1$ is as expressive as a language $\mathcal{L}_2$ with respect to a class K of models for $\mathcal{L}_1$ and $\mathcal{L}_2$ if there is a translation $\cdot^\natural$: $\mathcal{L}_2 \to \mathcal{L}_1$ such that, for every $\mathcal{B} \in \mathsf{K}$, point w in $\mathcal{B}$, and formula $\phi \in \mathcal{L}_2$:*

$$\mathcal{B}, w \models \phi \text{ iff } \mathcal{B}, w \models \phi^\natural$$

we say that $\mathcal{L}_1$ and $\mathcal{L}_2$ are equally expressible, if $\mathcal{L}_1$ is as expressive as $\mathcal{L}_2$ and $\mathcal{L}_2$ is as expressive as $\mathcal{L}_1$.

Note that $\mathcal{L}_2$ can be translated into $\mathcal{L}_1$ in such way that a sentence of $\mathcal{L}_2$ is true in one model **iff** the translation of this sentence is true in the same model viewed

as a $\mathcal{L}_1$ model. As a consequence the model are the same mathematically, but the languages that are interpreted in them are not the same. That is, the satisfiability problem for $\mathcal{L}_1$ in K is decidable iff the satisfiability problem in K is decidable for $\mathcal{L}_2$. The same holds for the finite model property.

2.6 Frame Definability

In this section we would like to summarize some important results of Hybrid logics hoping to be easier for the reader to follow the structure of this thesis.

We start, defining the notion of modally definable frame class. Given a set of modal formulas Σ, the frame class *defined* by Σ is the class of all frames on which each formula in Σ is valid. A frame class is *modally definable* if there is a set of modal formulas that defines it. A frame class is *elementary* if it is defined by a sentence of the first order correspondence language $\mathcal{L}_{fr}$ which is the first order language with equality and binary relation for each modality. We will continue discussing a number of results concerning the relationship between modally definable frame classes and elementary frame classes in the context of hybrid logics.

First of all we should mention that there are some frame properties not definable by pure axioms.

Example 2.6.1. *An example is the class of Church-Rosser frames:*

$$\forall wvu\ \exists t(wRv \wedge wRu \rightarrow uRt \wedge uRt)$$

That is, there is no way to get a handle on the convergence node t, in the above first order definition.

The fact that a frame property is not definable by a pure axiom does not mean that there isn't a formula (containing propositional letters) of the basic hybrid logic defining this particular class of frames.

Example 2.6.2. *Right directed frames satisfy the property:*

$$\forall vu \exists t(vRt \wedge uRt)$$

The formula: $@_i\Box p \rightarrow @_j\Diamond p$ defines the class of right directed frames in the context of Hybrid Logic. But since this formula is not a pure one, we can not apply the completeness result (of [5])related to pure formulas .

From [4], [9] we know that the hybrid languages with $\forall$ and $\exists$ ($\mathcal{H}(@, \forall, \exists)$) can express any first order sentence. That is, using the following translation $\mathcal{HT}(\cdot)$, we map any first order representation to formulas of $\mathcal{H}(@, \forall, \exists)$ ([4]):

$$\begin{aligned}
\mathcal{HT}(\top) &= \top && (2.1)\\
\mathcal{HT}(P(x)) &= p && (2.2)\\
\mathcal{HT}(x = y) &= @_x y && (2.3)\\
\mathcal{HT}(xRy) &= @_x \Diamond y && (2.4)\\
\mathcal{HT}(\neg\varphi) &= \neg\mathcal{HT}(\varphi) && (2.5)\\
\mathcal{HT}(\varphi \wedge \psi) &= \mathcal{HT}(\varphi) \wedge \mathcal{HT}(\psi) && (2.6)\\
\mathcal{HT}(\exists v\phi) &= \exists v\mathcal{HT}(\phi) && (2.7)\\
\mathcal{HT}(\forall v\phi) &= \forall v\mathcal{HT}(\phi) && (2.8)
\end{aligned}$$

Proposition 2.6.1. *Let ϕ be a first order formula. Then for every model $\mathcal{M}$ and any assignment g, $\mathcal{M} \models \phi[g]$ iff $M, g \models \mathcal{HT}(\phi)$.*

Proof. On page 45 of [9], proposition 3.7. □

Example 2.6.3. *Using the translation above, the right directness can be defined by a natural sentence of the strong hybrid language, namely:*

$$\forall xy\exists z(@_x \Diamond z \wedge @_y \Diamond z)$$

Example 2.6.4. *We know that the following pure formula defines transitivity: $@_i(\Diamond \Diamond j \rightarrow \Diamond j)$. The same property can be re-expressed as follows: $\forall x \forall y @_x(\Diamond \Diamond y \rightarrow \Diamond y)$.*

That is, we can substitute nominals with variables taking the universal closure. But also we can have the other way around: the state variables x and y are arbitrary variables of our model, so the universal quantification is implicit. That is, we can substitute state variables with nominals. We summarize this result defining first the PUNF-sentences.

Definition 2.6.1. *Call a sentence of $\mathcal{H}(@, \forall, \exists)$ of the form $\forall x_1 ... x_n \phi$, where ϕ does not contain any quantifiers, propositional letters or nominals a PUNF-sentence (**P**ure, **U**niversal, **N**ominal free sentence).*

Proposition 2.6.2. *1. For any pure formula ϕ of the basic hybrid language there is a PUNF-sentece $\phi^{\forall}$ of the strong hybrid language such that for any frame $\mathcal{F}$, ϕ is valid on $\mathcal{F}$ iff $\phi^{\forall}$ is true on $\mathcal{F}$.*

2. *For any PUNF-sentence ϕ of the strong hybrid language there is a pure formula ϕ^B of the basic hybrid language such that for any frame $\mathcal{F}$, ϕ is true in $\mathcal{F}$ iff ϕ^B is valid on $\mathcal{F}$.*

Proof. 1. Suppose we are given a pure formula $\phi(i_1, ..., i_n)$, where $\phi(i_1, ..., i_n)$, where $i_1, ..., i_n$ are all the nominals in ϕ, then the required $\phi^\forall$ is $\forall x_1 ... x_n \phi$

2. Any PUNF-sentence ϕ of the strong hybrid language must have the form $\forall x_1 ... x_n \phi(x_1, ..., x_n)$ where ϕ contains no quantifiers, proposition letters or nominals. Therefore, the required ϕ^B is $\phi([i_1 \leftarrow x_1, ..., i_n \leftarrow x_n])$.

□

That is, the last proposition tell us that the frame classes expressible by pure axioms are simply the frame classes definable by PUNF-sentences.

To sum up, in this section we mentioned that the strong hybrid languages are quite expressible. In particular they can express any first order representation. We studied also the relation between PUNF-sentences and pure axioms. But PUNF-sentences corresponds to the universal fragment of the strong hybrid languages. Therefore the following question might arise:

Can we find an extension of the basic hybrid language that can define any first order frame property?

This thesis deals with this question.

2.7 Elements of Universal Algebra

In this section we review some basic (universal) algebraic notions. Of primary importance is the concept of an algebra. Also we will discuss the notions of isomorphism and congruence. What follows is a short textbook-style presentation of algebra, following [26].

Roughly speaking an algebra is a set of together with a collection of functions over the set. These functions are usually called operations.

Definition 2.7.1. *(***Similarity Type***)*
An algebraic similarity type is an ordered pair $\mathcal{F} = (F, \rho)$ where F is a non empty set and ρ is a function $F \to \mathbb{N}$. Elements of F are called function symbols; the function ρ assigns to each operator $f \in F$ a finite arity or rank, indicating the number of arguments that f can be applied to. Function symbols of rank zero are called constants. We will usually be sloppy in our notation and terminology and write $f \in \mathcal{F}$ instead of $f \in F$.

Definition 2.7.2. *(***Algebras***)*
Let A be some set and n a natural number; an n-ary operation on A is a function from $A^n \to A$. Let $\mathcal{F}$ be an algebraic similarity type. An algebra of type $\mathcal{F}$ is a pair $\mathcal{U} = (A, I)$ where A is a non empty set called the carrier of the algebra and I is an interpretation, a function assigning, for every n, an $n-$ ary operation $f_{\mathcal{U}}$ on A to each function symbol f of rank n. We often use the notation $\mathcal{U} = (A, f_{\mathcal{U}})_{f \in F}$ for such an algebra.

It is a quite curious fact that the algebras that have been studied in conventional modern algebra do not have fundamental operations of arity greater than two. We proceed with various of examples of algebras. In particular we would like to point out the role of recent directions in logic aimed at providing algebraic models for certain logical systems. The reader will notice that all of the different kinds of algebras listed below are distinguished from each other by their fundamental operations and the fact they satisfy certain identities.

Example 2.7.1. Groups. *A group G is an algebra $< G, \cdot, ^{-1}, 1 >$ with a binary, a unary and nullary operation in which the following identities are true:*

$$\begin{array}{ll} G_1 & x \cdot 1 = 1 \cdot x = x \\ G_2 & x \cdot x^{-1} = x^{-1} \cdot x = 1 \\ G_3 & (x \cdot y) \cdot z = x \cdot (y \cdot z) \end{array}$$

A group G is Abelian (or cummutative) if the following identity is true:

$$G_4 \quad x \cdot y = y \cdot x$$

Groups were one of the earliest concepts studied in algebra.

Remark 2.7.1. *The definition given above is not the one which appears in standard texts on groups; for they use only one binary operation and axioms involving existential quantifiers. We will discuss this in more detail on chapter 3.*

Groups are generalized to semigroups and monoids in one direction, and to quasigroups and loops in another direction.

Example 2.7.2. *(***Semigroups and Monoids***) A semigroup is a groupoid $< G, \cdot >$ in which (G_1) is true. It is commutative or Abelian if (G_4) holds. A monoid is an algebra $< M, \cdot, 1 >$ with a binary and a nullary operation satisfying (G_1) and (G_2).*

Example 2.7.3. *A quasigroup is an algebra* $< Q, /, \cdot, \backslash >$ *with three binary operations satisfying the following identities:*

$$\begin{array}{ll} Q_1 & x\backslash(x \cdot y); \quad (x \cdot y)/x = x \\ Q_2 & x \cdot (x\backslash y) = y; \quad (xy) \cdot y = x \end{array}$$

A loop is a quasigroup with identity, i.e. an algebra $< Q, /, \cdot, \backslash, 1 >$ *which satisfies* (Q_1) Q_2 *and* G_2*.*

Example 2.7.4. *(***Rings***) A ring is an algebra* $< R, +, \cdot, -, 0 >$ *where* $+$ *and* $\cdot$ *are binary,* $-$ *is unary and* 0 *is nularry, satisfying the following conditions:*

$$\begin{array}{lll} R_1 & < R, +, \cdot, -, 0 > & \text{is an Abelian group} \\ R_2 & < R, \cdot > & \text{is a semigroup} \\ R_3 & x \cdot (y + z) = x \cdot y + x \cdot z & \\ & (x + y) \cdot z = x \cdot z + y \cdot z & \end{array}$$

A ring with identity is an algebra $< R, +, \cdot, -, 0, 1 >$ *such that* $R_1 - R_3$ *and* G_2 *hold.*

Definition 2.7.3. *(***Isomorphism***) Let* $\mathcal{A}$ *and* $\mathcal{B}$ *be two algebras of the same type* $\mathcal{F}$*. Then a function* $\alpha : A \to B$ *is an isomorphism from* $\mathcal{A}$ *to* $\mathcal{B}$ *if* α *is one to one and onto, and for every n-ary* $f \in \mathcal{F}$ *for* $(\alpha_1, ..., \alpha_n) \in A$*, we have:*

$$\gamma f^{\mathcal{A}}(\alpha_1, ..., \alpha_n) = f^{\mathcal{B}}(\gamma\alpha_1, ..., \gamma\alpha_n)$$

We say $\mathcal{A}$ *is isomorphic to* $\mathcal{B}$*, written* $\mathcal{A} \cong \mathcal{B}$*, if there is an isomorphism from* $\mathcal{A}$ *to* $\mathcal{B}$*. If* γ *is an isomorphism from* $\mathcal{A}$ *and to* $\mathcal{B}$ *we may simply say* $\gamma : \mathcal{A} \to \mathcal{B}$ *is an isomorphism. Usually we do not distinguish isomorphic algebras. Thus, isomorphic algebras can be regarded as equal or the same.*

There are several important methods of constructing new algebras from given ones. Three of the most fundamental are the formation of subalgebras, homomorphic images and direct products. But we will not cover these issues here. Another important definition is the following one:

Definition 2.7.4. *(***Congruences***) Let* $\mathcal{U}$ *be an algebra for the similarity type* $\mathcal{F}$*. An equivalence relation* $\sim$ *on* $\mathcal{A}$ *(that is, a reflexive, symmetric and transitive relation) is a congruence if it satisfies, for all* $f \in \mathcal{F}$

$$\text{if } a_1 \sim b_1 \text{ and } \cdots \text{ and } a_n \sim b_n \text{ then } f_{\mathcal{U}}(a_1, ..., a_n) \sim f_{\mathcal{U}}(b_1, ..., b_n)$$

where n *is the rank of* f*.*

The importance of congruences is that they are precisely the kind of equivalence relations that allow a natural algebraic structure to be defined on the collection of equivalence classes.

2.8 Skolemization

Skolem, following the work of Lowenheim (1915), introduced a technique to convert a first order sentence ϕ into a sentence ϕ' in prenex form, with only universal quantifiers such that: ϕ is satisfiable **iff** ϕ' is satisfiable.

Roughly speaking, dealing with universal quantified sentences are apparently easier to understand. Note that we will discuss the same topic in the context of Equational Logic in chapter 7. Skolemization allows us to say that the first order language $\mathcal{L}$ with universal, existential quantifiers, relation symbols, function symbols and equality $=$ is equivalent expressible with the first order language $\mathcal{L}^{\forall}$ (the universal fragment of $\mathcal{L}$. In order to do that we use the notion of skolem functions. Please treat this section as a repository of basic definitions for later use.

Definition 2.8.1. *A formula is on Prenex normal formula (PNF) if it has the form*

$$Q_1 x_1 \; Q_2 x_2 \; \cdots \; Q_n x_n \; \phi$$

where $Q_i \in \{\forall, \exists\}$ are quantifiers and ϕ is quantifier free.

Theorem 2.8.1. *Every predicate logic formula of $\mathcal{L}$ is equivalent to a formula in prenex form.*

Briefly, to obtain a prenex form of a formula one can proceed in two steps (more in [25]):

- Rename bound variables, so that no variable occurs both bound and free and no two quantifiers bind the same variable.
- Use the quantifier equivalences to move quantifiers out of the scope of propositional connectives.

Definition 2.8.2. *A first order formula ϕ is* **universal** *if it is in PNF and all quantifiers are universal. That is, ϕ is of the form $\forall x_1, ..., x_n \;\; G(x_1, ..., x_n)$, where G is quantifier-free. G is called the matrix of ϕ.*

We will use the following notation:

- $\overline{x} = x_1 \cdots x_n$.
- Let $\mathcal{A}$ to be a model for the first order logic $\mathcal{L}$.
- Let $\mathcal{A}^{\forall}$ to be a model of the first order logic $\mathcal{L}^{\forall}$ (the universal fragment of $\mathcal{L}$). In our case the model $\mathcal{A}^{\forall}$ is an expansion of the model $\mathcal{A}$ as the lemma 2.8.1 states.

Lemma 2.8.1.
- *Given the sentence $\exists y G(y)$, augment the language with a new constant c and form the sentence $G(c)$. Then*

$$\mathcal{A} \models \exists y G(y) \text{ iff } \mathcal{A}^{\forall} \models G(c)$$

- *Given the sentence $\phi = \forall x_1 \cdots \forall x_n \exists y G(\overline{x}, y)$, we augment the language $\mathcal{L}$ with a new $n-$ ary function symbol f and form the sentence of $\mathcal{L}^{\forall}$: $\phi^S = \forall x_1 \cdots \forall x_n G(\overline{x}, f(\overline{x}))$. Then:*

$$\mathcal{A} \models \phi \text{ iff } \mathcal{A}^{\forall} \models \phi^S$$

Proof. The proof of this lemma is quite well known. We prefer the approach of [25].

In each step we expand the model $\mathcal{A}$ of $\mathcal{L}$ to the model $\mathcal{A}^{\forall}$ of $\mathcal{L}^{\forall}$ by adding the appropriately designated element for the constant, or by adding an appropriate function f.

Also the other way around, we can take the model $\mathcal{A}^{\forall}$ and to remove the interpretation of the constant, respectively function symbol and get the model $\mathcal{A}$. □

Remark 2.8.1. *We should also mention the approach of [29]. In [29] there is a proof of the lemma* 2.7.1 *(pages 274-276), using terms of second order logic.*

Definition 2.8.3. *Consider a* **PNF***-formula $\phi = Q_1 x_1 \cdots Q_n x_n \psi(x_1, ..., x_n)$. For each existential quantifier $Q_i = \exists$, remove $\exists x_i$ and replace x_i by $f_i(x_{j_1}, \cdots, x_{j_m})$ where f_i is a new $m-$ary function symbol and $x_{j_1}, \cdots, x_{j_m}$ are universally quantified variables to the left $Q_i x_i$. The resulting formula ϕ^S is called the* **skolemization** *of φ and the new function symbols are called* **Skolem function symbols**. *This process of converting a sentence to such a universal sentence is called* **skolemizing**.

Example 2.8.1. *(***Right Directness***)*

$$\begin{aligned} \phi &= \forall vu \underline{\exists t}(R(u,t) \wedge R(u,t)) && (2.9)\\ &\mapsto \forall vu(R(u, f(u,v)) \wedge R(u, f(u,v))) \ f \text{ new function symbol} && (2.10)\\ &= \phi^S && (2.11) \end{aligned}$$

Theorem 2.8.2. *Given a first order formula ϕ there is an effective procedure for finding a universal sentence ϕ' (usually in an extended language) such that:*

$$\mathcal{A} \models \phi \text{ iff } \mathcal{A}^{\forall} \models \phi'$$

Furthermore, we can choose ϕ' such that every model of ϕ can be expanded to a model of ϕ' and every model ϕ' can be reducted to model of ϕ.

Proof. First we put ϕ in prenex form. Then we just apply the lemma 2 repeatedly until there are no existential quantifiers. □

The next lemma gives the relation of a formula and its skolemization.

Lemma 2.8.2. *Let ϕ be a closed prenex $\mathcal{L}-$ formula. Let ϕ^S be its skolemization and $f_1, ..., f_m$ the introduced Skolem function symbols. Then for every $L-$ structure $\mathcal{A}$ there is an $\mathcal{L}' = \mathcal{L} \cup \{f_1, ..., f_m\}-$ structure $\mathcal{B}$ expanding $\mathcal{A}$ so that:*

$$\mathcal{B} \models \phi \leftrightarrow \phi^S$$

Proof. The proof goes by successively applying the procedure described above to the out most existential quantifier, and then noting that an equivalence holds in a structure if it already holds in a restriction. We use again the above example and for some f (which we do not know much about):

$$(\mathcal{A}; f) \models \forall vu\ \exists t(R(u,t) \wedge R(u,t)) \leftrightarrow \forall vu(R(u, f(u,v)) \wedge R(u, f(u,v)) \quad (2.12)$$

□

Remark 2.8.2. *Note that the lemma above only states that for a given structure $\mathcal{A}$ and a given formula ϕ there is some expanded structure B in which the formula is equivalent to its Skolemization: $\mathcal{B} \models \phi \leftrightarrow \phi^S$. This equivalence need not hold in every expansion of $\mathcal{A}$.*

Now we extend these ideas to sets of sentences.

Theorem 2.8.3. *Given a set of first order sentences $\mathcal{S}$, there is a set $\mathcal{S}'$ of universal sentences (usually in an extended language) such that:*

$$\mathcal{A} \models \mathcal{S} \text{ iff } \mathcal{A}' \models \mathcal{S}'$$

Furthermore, every model of $\mathcal{S}$ can be expanded to a model of $\mathcal{S}'$ and every model of $\mathcal{S}'$ can be reducted to a model of $\mathcal{S}$.

Proof. We skolemize each sentence in $\mathcal{S}$ as before, making sure that distinct sentences do not have any common skolem constants of functions. □

Example 2.8.2. *An axiom of abelian groups is:*

$$\forall x \exists y \ \ (x + y = 0)$$

by skolemization we have:

$$\forall x (x + f(x) = 0)$$

where f is unary.

2.9 Notation

Before proceeding any further we summarize the notation of logics that we will use later in this thesis.

- $\mathcal{H}$: *Nominal hybrid logic.*
- $\mathcal{H}(@)$: *Basic Hybrid logic.* $\mathcal{H}$ plus satisfaction operators.
- $\mathcal{H}(@, \forall, \exists)$: *Strong Hybrid Logic.* That is, $\mathcal{H}(@)$ plus a second sort of nominals that can be bind from the quantifiers $\forall, \exists$.
- $\mathcal{HLON}$: **H**ybrid **L**ogic with **O**perations on **N**ominals. Extension of $\mathcal{H}(@)$ with operators only for nominals. Please note for further use that $\mathcal{HLON}$ is a language with infinite number of functional symbols of all finite arities.
- $\mathcal{HLOV}(@, \forall, \exists)$: **H**ybrid **L**ogic with **O**perations on **V**ariables. An extension of $\mathcal{H}(@, \forall, \exists)$ with operators only for state variables. Please note for further use that $\mathcal{HLOV}(@, \forall, \exists)$ is a language with infinite number of functional symbols of all finite arities.
- $\mathcal{L}$: the first order language with universal, existential quantifiers, relation symbols, function symbols and equality $=$.
- $\mathcal{L}^{\forall}$: The universal fragment of $\mathcal{L}$.
- $\forall - \mathcal{HLOV}$: The universal fragment of $\mathcal{HLOV}(@, \forall, \exists)$.

Chapter 3

Extended Language

3.1 Introduction

Chapter 3: *We name a version of the basic Hybrid Logic ($\mathcal{H}(@)$) and we call it $\mathcal{HLON}$ (Hybrid Logic with Operators only for Nominals) henceforth. We set up the syntax and we give the notions of Kripke model and validity to it. We give a bisimulation of models for this language providing also that validity is invariant under bisimulation. We provide a standard translation from $\mathcal{HLON}$ to first order logic. Roughly speaking, in $\mathcal{HLON}$ besides the traditional relational component, the universe of possible worlds has a specific algebraic structure.*

3.2 $\mathcal{HLON}$

We are ready now to start our formal work on Hybrid Logics. First some basic definitions:

Definition 3.2.1. *An algebraic similarity type is an ordered pair is $\sigma = (A, \rho)$ where A is a non empty set and ρ is a function $A \to N$. Elements of A are called functions symbols; the function ρ assigns to each operator $f \in A$ a finite arity (rank), indicating the number of arguments that f can be applied to. Function symbols of rank zero are called constants.*

Let **ANOM** be a countable set of atomic nominals: **ANOM**$= \{i, i_1, \cdots\}$. Let σ an algebraic similarity type.

Definition 3.2.2. *An algebraic language of nominals (σ,* **ANOM***) is built up using an algebraic similarity type (A, ρ) and a set of atomic nominals* **ANOM**.

The **NOM**=*FORM(*σ*,* **ANOM***) of nominals formulas over* σ *and* **ANOM** *is given by the rule:*

$$\alpha := i \mid f(\alpha_1, \cdots, \alpha_{\rho(f)})$$

Note that **CNOM** = NOM $\setminus$ ANOM, is the set of non atomic nominals; we call them complex nominals henceforth. Let **PROP**= $\{p_1, p_2, \cdots\}$ a countable set of propositional variables. Let $\Phi =$ **PROP** $\cup$ **NOM**. Note that elements of **NOM** are still propositional variables.

Definition 3.2.3. $\mathcal{HLON}$ **MODELS**.

A model $\mathcal{M}$ *for* $\mathcal{HLON}$ *is* $\mathcal{M} = (W, R_\Diamond, (F^f)_{f\in\sigma}, V)$ *such that:*

- $W \neq 0$*, a set of possible worlds;*
- $R_\Diamond$ *is binary relation over* W*.*
- *for each* $n \geq 0$ *and for all* $f \in \sigma$*, if* f *is an* $n-$*ary function symbol then* F^f *is an* $n-$*ary operation symbol of* W*. Note that* $F^f : W^n \to W$*;*
- *V:* $\Phi \to$ *Pow(W) and V is such that for all nominals* $\alpha \in$ **NOM***, V(*α*) is singleton of* W*. In more detail:*
  - *If* $\alpha \in$ **NOM** *then: V(*α*)=w.*
  - *If* $\alpha \in$ **CNOM** *then:* $V(f(\alpha_1, ..., \alpha_n)) = F^f(V(\alpha_1), ..., V(\alpha_n))$*.*

We call the elements of $\mathcal{M}$ states or worlds, $R_\Diamond$ accessibility relations, F^f operation symbols of W and V the valuation. We call the triple $(W, R_\Diamond, (F^f)_{f\in\sigma})$ the *algebraic frame* underlying the model.

Definition 3.2.4. *The Hybrid Language* $\mathcal{HLON}$ *is a language that is used for describing models and frames. Its formulas are given by the following recursive definition.*

$$\varphi := \alpha \mid p \mid \neg\varphi \mid \varphi \to \psi \mid \Diamond\varphi \mid @_\alpha\varphi$$

where $\alpha \in$ **NOM**.

Note that all types of atomic symbols (propositional variables and nominals) are formulas. Furthermore a formula is **pure** if it is contains no propositional variables and nominal free if it contains no nominals.

Definition 3.2.5. Semantics for $\mathcal{HLON}$

Let a model $\mathcal{M} = (W, R_\Diamond, (F^f)_{f\in\sigma}, V)$ for $\mathcal{HLON}$ and a valuation V which assigns an element of the domain to every nominal.

$\mathcal{M}, w \models i$	*iff*	$w \in V(i)$, *where* $i \in$ ANOM
$\mathcal{M}, w \models f(\alpha_1, \cdots, \alpha_n)$	*iff*	*for some* $v_1, \cdots, v_n \in W$ *with* $F^f(v_1, ..., v_n) = w$ *we have for each* $l, \mathcal{M}, v_l \models \alpha_l$.
$\mathcal{M}, w \models p$	*iff*	$w \in V(p)$, *where* $p \in$ PROP
$\mathcal{M}, w \models @_\alpha\varphi$	*iff*	$\mathcal{M}, v \models \varphi$ *where* $V(\alpha) = \{v\}$
$\mathcal{M}, w \models \varphi \to \psi$	*iff*	$\mathcal{M}, w \not\models \varphi$ *or* $\mathcal{M}, w \models \psi$
$\mathcal{M}, w \models \neg\varphi$	*iff*	$\mathcal{M}, w \not\models \varphi$
$\mathcal{M}, w \models \Diamond\phi$	*iff*	*if there is a* $v \in W$ *such that* $R_\Diamond(w, v)$ *and* $\mathcal{M}, v \models \varphi$.

$(\mathcal{M}, w)$ is called point model. This is a model with designated world, called its point, which is taken to be the actual world.

For simplicity we limit the language to one binary relation $R_\Diamond$ and to one binary function F. In other words our hybrid model is: $\mathcal{M} = (W, R_\Diamond, F^f, V)$ or better $\mathcal{M} = (W, R, F, V)$.

3.3 Bisimulation and Elementary Equivalence

Bisimulation is a useful notion in modal logic. Bisimulation is a relation between two models in which related states have identical atomic information and matching transition possibilities. Roughly speaking the notion of bisimulation defines an equivalence relation between models. In the case of $\mathcal{HLON}$ models the notion of identical atomic information means satisfying the same sentences. A well known result in modal logic is that if two pointed models are bisimilar, then they satisfy the same sentences (see in [6]). In this section we show that such a result holds for $\mathcal{HLON}$ as well.

Definition 3.3.1. Bisimulations for $\mathcal{HLON}$

Let two $\mathcal{HLON}$ models $\mathcal{M} = (W, R, F, V)$ and $\mathcal{M}' = (W', R', F', V')$. be given. A non empty binary relation $\sim$ is a bisimulation when the following holds:

- *(***prop***) if $w \sim w'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all $p \in ANOM \cup PROP$.*

- *(***fun-forth***) if* $w \sim w^{'}$ *and there are* $u_1, u_2 \in W$ *such that* $F(u_1, u_2) = w$, *then there are* $v_1, v_2 \in W^{'}$ *such that for all* l, $u_l \sim v_l$ *and* $F^{'}(v_1, v_2) = w^{'}$.

- *(***fun-back***) A similar condition from* $\mathcal{M}^{'}$ *to* $\mathcal{M}$.

- *(***forth***) if* $w \sim w^{'}$ *and* Rwv, *then there is* $v^{'} \in W^{'}$ *such that* $R^{'}w^{'}v^{'}$ *and* $v \sim v^{'}$.

- *(***back***) A similar condition from* $\mathcal{M}^{'}$ *to* $\mathcal{M}$.

- (@)*: if* $w \in V(\alpha)$ *and* $w^{'} \in V(\alpha)$ *for some* $\alpha \in NOM$, *then* $w \sim w^{'}$.

Since @ is self dual we can collapse the back and forth clauses for this operator into one. **Prop**, **forth** and **back** are the usual conditions for bisimulation. We added **fun-forth** and **fun-back** in order to deal with functional sentences.

If there is a bisimulation between $\mathcal{M}$ and $\mathcal{M}^{'}$ linking w and $w^{'}$ then we write $\mathcal{M}, w \underline{\leftrightarrow} \mathcal{M}^{'}, w^{'}$.

Theorem 3.3.1. *For all models* $(\mathcal{M}, w)$ *and* $(\mathcal{M}^{'}, w^{'})$ *and for all sentences* ϕ, *if* $(\mathcal{M}, w) \underline{\leftrightarrow} (\mathcal{M}^{'}, w^{'})$ *then* $\mathcal{M}, w \models \phi$ *iff* $\mathcal{M}^{'}, w^{'} \models \phi$.

Proof. By induction ϕ. Suppose $\mathcal{M}, w \models \phi \underline{\leftrightarrow} \mathcal{M}^{'}, w^{'} \models \phi$.

- The case where $\varphi \in$ **PROP** $\cup$ **ANOM** follows from clause (**prop**).

- For formulas $\varphi \in$ **CNOM** we want to show that $\mathcal{M}, w \models f(\alpha_1, \alpha_2)$ iff $\mathcal{M}^{'}, w^{'} \models f(\alpha_1, \alpha_2)$. Suppose $\mathcal{M}, w \models f(\alpha_1, \alpha_2)$. This means that are worlds $x, y \in W$ with $\mathcal{M}, x \models \alpha_1$ and $\mathcal{M}, y \models \alpha_2$. Note that $F(x, y) = w$. Using the definition of bisimulation we know that there are worlds $x^{'}, y^{'} \in W^{'}$ such that $\mathcal{M}, x^{'} \models \alpha_1$, $\mathcal{M}, y^{'} \models \alpha_2$, $x \sim x^{'}$ and $y \sim y^{'}$. We also know that $F^{'}(x^{'}, y^{'}) = w^{'}$, thus $\mathcal{M}^{'}, w^{'} \models f(\alpha_1, \alpha_2)$. For the converse direction use the clause (**fun-back**).

- As for formulas $\diamond\varphi$, we have $\mathcal{M}, w \models \diamond\varphi$ iff there exists a u in $\mathcal{M}$ such that Rwu and $\mathcal{M}, u \models \varphi$. As $w, w^{'}$ are bisimilar by clause (**forth**) we know that there is a world $u^{'} \in \mathcal{M}^{'}$ such that $R^{'}w^{'}u^{'}$ and $u \sim u^{'}$. By the induction hypothesis, $\mathcal{M}^{'}, u^{'} \models \varphi$, hence $\mathcal{M}^{'}, w^{'} \models \diamond\varphi$

- For the converse direction use the clause (**back**).

□

3.4 Undefinable Properties

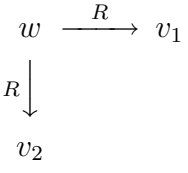
In this section we will show that some class of frames is not closed under bisimulation images.

Example 3.4.1. *A nice example of how to use the bisimulation to show that some first-order property cannot be expressed in our language is the following property: there are two different successors of the current world, defined by the first order formula:*

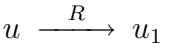
$$\phi = \exists x \exists y (zRx \wedge zRy \wedge \neg x = y)$$

Consider two frames $\mathcal{F}_1$ and $\mathcal{F}_2$. These two frames are bisimilar over relation symbol R and function symbol f, where f is one identity function: $\forall x f(x) = x$. To see this, define the following relation $\mathcal{Z} = \{(w, u), (v_1, u_1), (v_2, u_1)\}$.

$\mathcal{F}_1$:



$\mathcal{F}_2$:



However $\mathcal{F}_1$ satisfies the property ϕ, whereas $\mathcal{F}_2$ does not.

Conclusion: Summarizing, our aim in this section was to develop hybrid logic, incorporating algebraic structure of nominals into it.

Chapter 4

$\mathcal{HLOV}(@, \forall, \exists)$

4.1 Introduction

Chapter 4: *We name a version of the strong hybrid language* $\mathcal{H}(@, \forall, \exists)$ *and we call it* $\mathcal{HLOV}(@, \forall, \exists)$ *(***H***ybrid* **L***ogic with* **O***perators only for* **V***ariables). We set up its language and semantics.*

4.2 $\mathcal{HLOV}(@, \forall, \exists)$

Let **NOM** be a countable set of nominals **NOM**= $\{i_1, i_2, ...\}$, let **SVAR** be a countable set o state variables **SVAR**= $\{x_1, x_2...\}$ and let **PROP** be a countable set of propositional letters. We assume that these sets are pairwise disjoint. Let σ an algebraic similarity type (see definition 2.1.1) and $R_\diamond$ an accessibility relation symbol.

Definition 4.2.1. *An algebraic language of variables (*σ*,* **SVAR***) is built up using an algebraic similarity type* (A, σ)*, and a set of state variables* **SVAR***. The* **VAR**=*FORM(*σ*,* **SVAR***) of variables formulas over* σ *and* **SVAR** *is given by the rule:*

$$s := x \mid f(s_1, \cdots, s_{\rho_{(f)}})$$

Remark 4.2.1. *All types of atomic symbol are formulas. Note that the difference between nominals and state variables is: nominals can not be bound by* $\forall, \exists$*, whereas state variables can.*

$\mathcal{HLOV}(@, \forall, \exists)$ is a natural extension of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$. We call **SSYM**= **NOM** $\cup$ **VAR** the set of state symbols. **ASTATES**= **NOM** $\cup$ **SVAR** the set of atomic states and let Φ =**PROP** $\cup$ **NOM** $\cup$ **VAR**.

Definition 4.2.2. $\mathcal{HLOV}$ **MODELS**.

A model $\mathcal{M}$ for $\mathcal{HLOV}$ is $\mathcal{M} = (W, R_\Diamond, (F^f)_{f\in\sigma}, V)$ such that:

- *$W \neq 0$, a set of possible worlds;*
- *$R_\Diamond$ is binary relation over W;*
- *for each $n \geq 0$ and for all $f \in \sigma$, if f is an $n-$ary function symbol then F^f is an $n-$ary operation symbol of W. Note that $F^f : W^n \to W$;*
- *V: $\Phi \to$ Pow(W) and V is such that for all nominals $i \in$ **NOM**, V(i) is singleton of W.*

*An assignment g for $\mathcal{M}$ is a mapping $g :$ **SVAR**$\to \mathcal{M}$. Given an assignment g, the notation $g[x \mapsto d]$ is the function that assigns d to x and differs from g at most with regard to the value x.*

Let $[V, g](s) = g(s)$ if s is a state variable and V(s) otherwise. In more detail:

- If $s \in$ **ASTATES** then: $[V, g](s) = g(s)$
- If $s \in$ **VAR** $\setminus$ **ASTATES** then: $[V, g](f(s_1, ..., s_n) = F^f(g(s_1), ..., g(s_n))$

Definition 4.2.3. *The Hybrid Language $\mathcal{HLOV}(@, \forall, \exists)$ is a language that is used for describing models and frames. Its formulas are given by the following recursive definition.*

$$\varphi := \top \mid s \mid p \mid \neg\varphi \mid \varphi \to \psi \mid \Diamond\varphi \mid @_s\varphi \mid \forall x\phi \mid \exists x\phi$$

*where $s \in$ **SSYM** and $x \in$ **VAR**.*

Definition 4.2.4. Semantics for $\mathcal{HLOV}(@, \forall, \exists)$

Let a model $\mathcal{M} = (W, R_\Diamond, (F^f)_{f\in\sigma}, V)$ for $\mathcal{HLOV}$ and an assignment g such that as (the semantics follow the lines of $\mathcal{H}(@, \forall, \exists)$, so we just give a few clauses of the definition here, leaving the remainder to the reader):

$$\begin{array}{lll} \mathcal{M}, g, w \models \alpha & \textit{iff} & w \in [V, g](\alpha), \textit{ where } \alpha \in \Phi \\ \mathcal{M}, g, w \models @_s\varphi & \textit{iff} & \mathcal{M}, g, g(s) \models \varphi \\ \mathcal{M}, g, w \models \forall x\phi & \textit{iff} & (\mathcal{M}, g[x \mapsto d]) \models \phi \textit{ where for all } d \in W \end{array}$$

4.3 $\mathcal{HLON}$ and $\forall - \mathcal{HLOV}$

In this section we will show how to move from the universal fragment of $\mathcal{HLOV}(@, \forall, \exists)$ to $\mathcal{HLON}$. For this reason we should modify the definition of PUNF-sentences. By the notation $\forall - \mathcal{HLOV}$ we mean the universal fragment of $\mathcal{HLOV}(@, \forall, \exists)$.

Definition 4.3.1. *Call a sentence of $\mathcal{HLOV}(@, \forall, \exists)$ of the form $\forall x_1 ... x_n \phi(\gamma_1, ..., \gamma_k)$, where ϕ does not contain any quantifiers, propositional letters or nominals a PUNF-sentence (**P**ure, **U**niversal, **N**ominal free sentence). Note that $x_1, ..., x_n$ and $\gamma_1, ..., \gamma_k$ are the state variables and the function symbols over variables occurring in ϕ. (PUNF-sentences are actually $\forall - \mathcal{HLOV}$ sentences)*

Example 4.3.1. *An $\forall - \mathcal{HLOV}$ sentence:*

$$\forall x \quad \Diamond\Diamond f(x) \rightarrow \Diamond f(x)$$

Proposition 4.3.1. *1. For any pure formula ϕ of $\mathcal{HLON}$ there is a PUNF-sentence $\phi^{\forall}$ of $\mathcal{HLOV}(@, \forall, \exists)$ such that for any frame $\mathcal{F}$, ϕ is valid on $\mathcal{F}$ iff $\phi^{\forall}$ is true on $\mathcal{F}$.*

2. For any PUNF-sentence ϕ of $\mathcal{HLOV}(@, \forall, \exists)$ there is a pure formula ϕ^B of $\mathcal{HLON}$ such that for any frame $\mathcal{F}$, ϕ is true in $\mathcal{F}$ iff ϕ^B is valid on $\mathcal{F}$.

Proof. 1. Suppose we are given a pure formula $\phi(i_1, ..., i_n, f_1, ..., f_k)$, where $i_1, ..., i_n$ are all the nominals and the function symbols over nominals in ϕ, then the required $\phi^{\forall}$ is $\forall x_1 ... x_n \phi(\gamma_1, ..., \gamma_k)$

2. Any PUNF-sentence ϕ of $\mathcal{HLOV}(@, \forall, \exists)$ must have the form $\forall x_1 ... x_n \phi(\gamma_1, ..., \gamma_k)$ where ϕ contains no quantifiers, proposition letters or nominals. Therefore, the required ϕ^B is $\phi([i_1 \leftarrow x_1, ..., i_n \leftarrow x_n, f_1, ..., f_k])$.

□

Remark 4.3.1. *This time, we substitute nominals with variables and function symbols over nominals with functions symbols over variables taking the universal closure. And of course the other way around.*

Conclusion: In this section we introduced an extension of the strong hybrid logic which will play an important role in the next section. We should be clear that $\mathcal{HLOV}(@, \forall, \exists)$ is one language with infinite number of functional symbols of all finite arities. Also the following problem remains: *How we can move from the universal fragment of the strong hybrid logic, to the universal existential fragment?*

Chapter 5

Skolem Functions

5.1 Introduction

Chapter 3: *In this chapter we introduce a natural method for eliminating existential quantifiers in the context of* $\mathcal{H}(@, \forall, \exists)$ *and* $\mathcal{HLON}$.

5.2 Skolemization in Hybrid Logic

In this section we will show that is possible to convert first order formulas to $\forall - \mathcal{HLON}$ formulas and this transformation preserves satisfiability. The steps of this transformation should be clear from the beginning:

We write any first order sentence of $\mathcal{L}$ in a prenex form, let ϕ be such a formula. Then we use the truth preserving translation $\mathcal{HT}(\cdot)$ of [4], [9]. That is, we shift from the formula ϕ to the formula $\mathcal{HT}(\phi)$ of $\mathcal{H}(@, \forall, \exists)$. We skolemize the formula $\mathcal{HT}(\phi)$ obtaining the sentence $\phi^{\forall}$ of $\forall - \mathcal{HLON}$. Recall that $\mathcal{HLOV}(@, \forall, \exists)$ is a language with infinitely function symbols. But let us be more precise:

Lemma 5.2.1.

Given the sentence $\exists y\ G(y)$ *(G(y) is a quantifier free formula) of* $\mathcal{H}(@, \forall, \exists)$*, augment the language with a new constant c and form the sentence* $G(c)$ *such that:*

$$\mathcal{M}, w \models \exists y\ G(y) \text{ iff } \mathcal{M}', w \models G(c)$$

Given the sentence $\forall x_1, \cdots, x_n \exists y\ G(\overline{x}, y)$ *of* $\mathcal{H}(@, \forall, \exists)$*, augment the language with a new* $n-$ *ary function symbol* f *and form the sentence* $\forall x_1, \cdots, x_n\ G'(\overline{x}, f(\overline{x}))$ *of* $\forall - \mathcal{HLOV}$*. Then:*

$$\mathcal{M}, w \models \forall x_1, \cdots, x_n \exists y\ G(\overline{x}, y) \text{ iff } \mathcal{M}', w \models \forall x_1, \cdots, x_n\ G'(\overline{x}, f(\overline{x}))$$

where $\mathcal{M}$ and $\mathcal{M}'$ are models of $\mathcal{H}(@, \forall, \exists)$ and $\forall - \mathcal{HLON}$ respectively. Note that w is an arbitrary world of our domain.

Proof. In each step we can expand a model of the sentence on the left hand side to one on the right hand side by adding the appropriately designated element for the constant, or by adding an appropriate function F^f, that is F, is the interpretation of f in $\mathcal{M}$. Conversely, we can take a model of the sentence on the right hand side and remove the constant symbol, respectively function symbol and we get a model for the left hand side. □

Theorem 5.2.1. *Given a formula ϕ of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$ there is an effective procedure for*

finding a universal sentence $\phi^\forall$ of $\forall - \mathcal{HLOV}$ such that:

$$\mathcal{M}, w \models \phi \text{ iff } \mathcal{M}', w \models \phi^\forall$$

Furthermore, we can choose $\phi^\forall$ such that every model of ϕ can be expanded to a model of $\phi^\forall$ and every model $\phi^\forall$ can be reducted to model of ϕ.

Proof. First we put ϕ in prenex form. Then we just apply the lemma 5.2.1 repeatedly until there are no existential quantifiers. □

We conclude with the following theorem:

Theorem 5.2.2. *For every model $\mathcal{M}$ and any assignment g we can associate a first order formula ϕ with one universal formula $\phi^\forall$ of $\mathcal{H}(@, \exists, \forall)$ such that:*

$$\mathcal{M} \models \phi[g] \text{ iff } \mathcal{M}', g \models \phi^\forall$$

Proof. By proposition 2.5.1 we know that:

$$\mathcal{M} \models \phi[g] \text{ iff } \mathcal{M}, g \models \mathcal{HT}(\phi)$$

Skolemizing the sentence $\mathcal{HT}(\phi)$ we obtain the sentence $\phi^\forall$, and by theorem 5.2.1 we have:

$$\mathcal{M}, g \models \mathcal{HT}(\phi) \text{ iff } \mathcal{M}', g \models \phi^\forall$$

□

An important result of this theorem is:

Theorem 5.2.3. *(***Complexity***) The satisfiability problem for* $\forall - \mathcal{HLOV}$ *is undecidable.*

Proof. As we know the satisfiability problem for first-order logic with a single binary relation is undecidable. We can use this fact as follows: Theorem 9 says that the first order logic $\mathcal{L}$ can be converted into $\forall - \mathcal{HLOV}$ in such way that a sentence of $\mathcal{L}$ is true in one model $\mathcal{M}$ **iff** the conversion of this sentence is true in one almost the same model $\mathcal{M}'$ of $\forall - \mathcal{HLOV}$. Therefore the model $\mathcal{M}$ and $\mathcal{M}'$ are almost the same mathematically, but the languages that are interpreted in them are not the same. That is, the satisfiability problem for $\forall - \mathcal{HLOV}$ in $\mathcal{M}'$ is undecidable since the satisfiability problem in $\mathcal{M}$ is undecidable for $\mathcal{L}$. □

The importance of the skolemization in the context of hybrid logic is that we might be able to use the resolution method for developing successful automatic theorem provers for hybrid logics. It is out of the scope of this chapter to provide details of this idea. More about the resolution method and its significance please look [25].

5.3 Frame Definability and $\mathcal{HLON}$

In this section we discuss a number of results concerning the relationship between elementary frame classes and frame classes definable by pure $\mathcal{HLON}-$ formulas. That is, we will show a method of extending the syntax for the basic hybrid language such that *elementary frame classes which are not modally definable in* $\mathcal{H}(@)$, to be almost definable by $\mathcal{HLON}$ pure formulas. We start with one example in order to motivate our formalism.

Example 5.3.1. Right directed. *In [7] the following problem stated:* Many important frame classes cannot be captured using pure axioms....Another (example) is the class of right-directed frames, that is, frames satisfying the property:

$$\forall v, u \exists t(vRt \wedge uRt)$$

We will show how to capture the logic of such frame class in the context of $\mathcal{HLON}$*. First we translate this formula in* $\mathcal{H}(@, \forall, \exists)$*:*

$$\forall v, u \exists t(@_v \diamond t \wedge @_u \diamond t)$$

Introducing a new binary function symbol F *over variables we have:*

$$\forall v, u@_v\diamond F(u,v) \wedge @_u\diamond F(u,v) \tag{5.1}$$

Using the proposition 4.3.1 *we get the following* $\mathcal{HLON}$ *pure formula:*

$$@_i\diamond f(i,j) \wedge @_j\diamond f(i,j) \tag{5.2}$$

We assume that i *and* j *are particular names, but their interpretation in the model is arbitrary, so the universal quantification is implicit. Note that no orthodox modal formula defines the condition of* $right-directed$.

That is,

Theorem 5.3.1. *For any frame* $\mathcal{F}$ *we can associate a first order formula* ϕ *with one* $\mathcal{HLON}$ *pure formula such that:*

$$\phi \textit{ is valid on } \mathcal{F} \textit{ iff } \phi^{nom} \textit{ is valid on } \mathcal{F}'$$

Proof. By the theorem 5.2.1 we know that there is a procedure associating any first order formula ϕ with one universal formula of $\mathcal{HLOV}(@,\forall,\exists)$ such that:

$$\mathcal{M}, w \models \phi \text{ iff } \mathcal{M}', w \models \phi^{\forall}$$

By the definition 2.1.5 we have:

- Since $\mathcal{M}, w \models \phi$ then for any frame $\mathcal{F}$ of $\mathcal{M} = (\mathcal{F}, V)$ we have that ϕ is valid on $\mathcal{F}$.
- Since $\mathcal{M}', w \models \phi^{\forall}$ then for any frame $\mathcal{F}'$ of $\mathcal{M} = (\mathcal{F}', V)$ we have that $\phi^{\forall}$ is valid on $\mathcal{F}'$. By proposition 4.2.3 we get that there is a $\mathcal{HLON}$ pure formula ϕ^{nom} such that: ϕ^{nom} is valid on $\mathcal{F}'$.

  That is,

$$\phi \text{ is valid on } \mathcal{F} \text{ iff } \phi^{nom} \text{ is valid on } \mathcal{F}'$$

□

We can rephrase the previous result as follows: a sentnence φ of $\mathcal{L}$ is valid on a frame $\mathcal{F}$ **iff** the formula ϕ^{nom} of $\mathcal{HLON}$ is valid on a frame $\mathcal{F}'$. That is, the satisfiability problem for φ on $\mathcal{F}$ is decidable or undecidable iff the satisfiability problem is decidable or undecidable for φ^{nom} on $\mathcal{F}'$.

Before we proceed we need the definition of frame satisfiability problem. That is,

Definition 5.3.1. *(***Frame Satisfiability problem***) Given a formula φ, check if there is a frame on which φ is valid.*

Theorem 5.3.2. *The frame satisfiability problem for modal formulas is highly undecidable, in fact not analytical.*

That is, by theorems 5.3.1 and 5.3.2 we get:

Theorem 5.3.3. *The frame satisfiability problem for $\mathcal{HLON}-$ pure formulas is highly undecidable, in fact not analytical.*

To sum up, we showed that the traditional fragment of the hybrid logic becomes in a rather unorthodox interpretation equivalent to first order logic by extending the basic hybrid language with algebraic operators working on nominals denoting worlds in the frame.

We continue with number of examples just in order to motivate our formalism. Note that just for simplicity reasons we skip the step of converting first order sentences in a prenex form. Examples presented here can be found in [23] and in [16].

Example 5.3.2. Any two elements have a greatest lower bound (in the order established by R)*:*

$$\forall x \forall y \exists z (zRx \wedge zRy \wedge \forall u((uRx \wedge uRy) \rightarrow uRz))$$

Using the syntax of the strong hybrid logic $H(@, \forall, \exists)$ we get:

$$\forall x \forall y \exists z (@_z \Diamond x \wedge @_z \Diamond y \wedge \forall u((@_u \Diamond x \wedge @_u \Diamond y) \rightarrow @_u \Diamond z))$$

Then by skolemization we have:

$$\forall x \forall y (@_{F(x,y)} \Diamond x \wedge @_{F(x,y)} \Diamond y \wedge \forall u((@_u \Diamond x \wedge @_u \Diamond y) \rightarrow @_u \Diamond F(x,y)))$$

where $F(x,y) : W^2 \rightarrow W$ is a binary function of our domain.
This can be re-expressed in $\mathcal{HLON}$ as:

$$@_{f(x,y)} \Diamond x \wedge @_{f(x,y)} \Diamond y \wedge (@_u \Diamond x \wedge @_u \Diamond y) \rightarrow @_u \Diamond f(x,y)$$

Example 5.3.3. Two (different) persons are siblings if they share two different parents.

$$x \neq y \wedge \exists z \exists z_1 (z \neq z_1 \wedge zPx \wedge z_1 Px \wedge zPy \wedge z_1 Py)$$

where P is a relation symbol and xPy denotes that x is a parent of y. The interpretation of the binary relation symbol P in $\mathcal{H}(@,\forall,\exists)$ is the modal operator $\Diamond_p$.

In $\mathcal{H}(@,\forall,\exists)$ we have:

$$@_x\neg y \ \wedge \ \exists z\exists z_1\ (@_z\neg z_1 \wedge @_z\Diamond_p x \wedge @_{z_1}\Diamond_p x \wedge @_z\Diamond_p y \wedge @_{z_1}\Diamond_p y)$$

By skolemization we introduce two constant symbols f and f_1:

$$@_x\neg y \wedge \ @_f\Diamond_p x \wedge @_{f_1}\Diamond_p x \wedge @_f\Diamond_p y \wedge @_{f_1}\Diamond_p y$$

Therefore in $\mathcal{HLON}$ we get:

$$@_x\neg y \wedge \ @_i\Diamond_p x \wedge @_{i_1}\Diamond_p x \wedge @_i\Diamond_p y \wedge @_{i_1}\Diamond_p y$$

The interpretation of the constants symbols f and f_1 are the nominals: i and i_1 respectively.

5.4 Universal equalities

In this section we will comment about the remark 4 of chapter 1 (page 9).

Definition 5.4.1. *[?] The set of proper symbols of the theory G of groups is $\{=,\cdot,1\}$ where $\cdot$ is the first $2-$place operation symbol and 1 is the first constant symbol. The proper axioms of G are:*

$$\begin{array}{ll} G_1 & \forall x\ (x\cdot 1 = x) \\ G_2 & \forall x\exists y\ (x\cdot y = 1) \\ G_3 & \forall x\forall y\forall z\ ((x\cdot y)\cdot z = x\cdot(y\cdot z)) \end{array}$$

That is, we skolemize

$$\{\forall x\forall y\forall z\ ((x\cdot y)\cdot z = x\cdot(y\cdot z)), \forall x\ (x\cdot 1 = x), \forall x\exists y\ (x\cdot y = 1)\}$$

and we obtain:

$$\{\forall x\forall y\forall z\ ((x\cdot y)\cdot z = x\cdot(y\cdot z)), \forall x\ (x\cdot 1 = x), \forall x(x\cdot f(x) = 1)\}$$

where $f(x)$ is unary.
That is, using the function symbol f as a reserve operation of $\cdot$ we have:

$$\begin{array}{ll} G_1 & x \cdot 1 = x \\ G_2 & x \cdot x^{-1} = 1 \\ G_3 & (x \cdot y) \cdot z = x \cdot (y \cdot z) \end{array}$$

Note that the variable x is an arbitrary variable of our model, so the universal quantification is implicit. By this example should be clear the relation between universal formulas and algebraic languages. And in the context of $\mathcal{HLON}$ we have:

$$\begin{array}{ll} G_1 & @_{x \cdot 1} x \\ G_2 & @_{x \cdot x^{-1}} 1 \\ G_3 & @_{(x \cdot y) \cdot z} x \cdot (y \cdot z) \end{array}$$

Conclusion: In this section we discussed the complexity (undecidability) of our new logics $\forall - \mathcal{HLOV}$ and $\mathcal{HLON}$. The key point for this conclusion was the skolemization method.

Chapter 6

Tableau

Chapter 4: *In this chapter we present a method of extending the tableau calculus for the basic hybrid language which yields completeness results for many frame classes that can not be defined in terms of $\mathcal{H}(@)$'s pure axioms, for instance right-directed frames. Actually we provide a complete tableau calculus for $\mathcal{HLON}$.*

6.1 Motivation

In this section we will discuss what kind of frames we can capture using the syntax of $\mathcal{HLON}$. Again, the technique of skolemization is central here.

As we know from [17] there is a natural way of presenting logics by combinations of the modal logic K with a relational theory $\mathcal{T}$. In [17], authors work with an orthodox modal language and a labelling algebra through a fixed interface. In our case this result can be rephrased as follows:

the base logic $\mathcal{H}(@)$ stays fixed, and we generate the one we want by combining $\mathcal{H}(@)$ with the appropriate pure formulas.

In order to do that we should consider two things:

- First we should investigate what kind of frame properties we can capture by the syntax of $\mathcal{HLON}$.
- Secondly, to study whether we can extend the completeness theorem of $\mathcal{H}(@)$ in our case.

We discussed the first point in detail in the last chapter. In this section we will accommodate $\mathcal{H}(@)$'s completeness result to our case.

But before we proceed to technicalities we will motivate our study by one example.

A large and important class of modal logics falls under the generalized Geach axiom schema:

$$\diamond^k \Box^l p \rightarrow \Box^m \diamond^n p$$

which corresponds to the semantic notion of (k, l, m, n) convergency:

$$\forall x \forall y \forall z \exists u (xR^k y \wedge xR^m z \rightarrow (yR^l u \wedge zR^n u)) \tag{6.1}$$

where $xR^0 y$ means $x = y$ and $xR^{k+1} y$ means $\exists v (xRv \wedge vR^k y)$.

That is, we start translating formula 23 in the strong hybrid logic $\mathcal{H}(@, \forall, \exists)$ and then we skolemize:

Example 6.1.1. *(***Geach axiom***)*

$$\begin{array}{rcll} \phi & = & \forall x \forall y \forall z \exists u (xR^k y \wedge xR^m z \rightarrow (yR^l u \wedge zR^n u)) & (6.2) \\ & \rightarrow^{\mathcal{H}(\cdot)} & \forall x \forall y \forall z \underline{\exists u} (@_x \diamond^k y \wedge @_x \diamond^m z \rightarrow (@_y \diamond^l u \wedge @_z \diamond^n u)) & (6.3) \\ & \mapsto & \forall x \forall y \forall z (@_x \diamond^k y \wedge @_x \diamond^m z \rightarrow (@_y \diamond^l F(x,y,z) \wedge @_z \diamond^n F(x,y,z))) & (6.4) \end{array}$$

where F is a new function symbol $F : W^3 \rightarrow W$. Then using the syntax of $\mathcal{HLON}$ we get:

$$@_x \diamond^k y \wedge @_x \diamond^m z \rightarrow (@_y \diamond^l f(x,y,z) \wedge @_z \diamond^n f(x,y,z))$$

Property	(k,l,m,n)	Char. Axiom	Pure Formula
Seriality	(0,0,1,1)	$\Box p \rightarrow \diamond p$	$@_x \diamond f(x)$
Convergency	(1,1,1,1)	$\diamond \Box p \rightarrow \Box \diamond p$	$@_x \diamond y \wedge @_x \diamond z \rightarrow (@_y \diamond g(x,y,z) \wedge @_z \diamond g(x,y,z))$
Reflexivity	(0,0,1,0)	$\Box p \rightarrow p$	$@_i \diamond i$
Symmetry	(0,1,0,1)	$\mathrm{p} \rightarrow \Box \diamond p$	$@_i \Box \diamond i$
Transitivity	(0,2,1,0)	$\Box p \rightarrow \Box \Box p$	$\diamond \diamond i \rightarrow \diamond i$

where $f : W \rightarrow W$ and $g : W^3 \rightarrow W$ are skolem functions.

That is, extending the completeness result of $\mathcal{H}(@)$ we could conclude that: *adding the pure formula for convergency to $\mathcal{H}(@)$'s proof system yields a system complete for frames that satisfy this property. Also extending $\mathcal{H}(@)$'s proof system with the pure axioms for reflexivity and transitivity yields a system complete both*

for reflexive and transitive frames. These results are cumulative: adding these three pure axioms, we obtain a complete system for the logic $S4.2$.

A hierarchy of modal logics (a fragment)

$$\begin{array}{ccccc} K & \xrightarrow{reflexivity} & KT(T) & & \\ {\scriptstyle transitivity}\downarrow & & \downarrow{\scriptstyle refl} & & \\ K4 & \xrightarrow{reflexivity} & S4 & \xrightarrow{convergency} & S4.2 \end{array}$$

Remark 6.1.1. *We should note that our method follows the lines of [1] (chapter 5) and [7], in these papers instead of functions symbols a node creating rule was added to the basic hybrid proof system. We hope that our result is more general.*

6.2 Proofs

In this section we will give the completeness proof by constructing Hintikka `sets`. That is, extending the basic tableau methods of $\mathcal{H}(@)$ we suggest a reasoning machinery for systems which combine relational and algebraic formalisms.

Definition 6.2.1. *A substitution is a function from atomic nominals to nominals, which is identity on all but finitely many atomic nominals.*

We typically use Greek letters such as θ, ρ, σ for substitutions. If θ is a substitution and i is an atomic nominal, then we indicate the application of θ to i by $\theta(i)$ as usual. We also extend substitutions homomorphically to nominals. If t is a nominal and θ is a substitution, we indicate by $t\theta$ the application of θ to t. This is defined inductively as follows: If t is an atomic nominal then $t\theta$ is $\theta(t)$. Otherwise, t is of the form $f(t_1, t_2, ..., t_n)$ (a complex nominal) for some n (possibly zero). Then $t\theta$ is $f(t_1\theta, t_2\theta, ..., t_n\theta)$. Thus $t\theta$ is t with all atomic nominals replaced by nominals as specified by θ. We often write a substitution as a set $\{i_1 \mapsto s_1, ..., i_n \mapsto s_n\}$, indicating that the nominal i_k is mapped to s_k by the substitution. We call $t\theta$ an instance of t. Also, if $@_t u$ where t and u are atomic nominals and θ is a substitution, we call $@_{t\theta} u\theta$ an intance of $@_t u$.

Example 6.2.1. *(Substitution) If $@_{x \cdot 1} x$ then $@_{(x \cdot x) \cdot 1} x \cdot x$.*

Since in $\mathcal{HLON}$ we deal with function symbols also the following rules are needed:

$$\frac{@_s t}{@_{f(\ldots,s,\ldots)} f(\ldots,t,\ldots)}\ \text{[Repl]} \qquad \frac{@_t s}{@_{t\theta} u\theta}\ \text{[Sub]}$$

Where θ is a substitution. As we will see in chapter 5 these rules follow the lines of equational logics.

We continue giving completeness proof by constructing Hintikka `sets`. But we should be clear, our tableau calculus are the old rules that we discussed in section 2.4 plus the rules **Repl** and **Sub**. Also the rules **Nom** and **Sym** they apply to nominals, that is both to atomic nominals and to complex nominals.

Definition 6.2.2. Hintikka Set: *A set of satisfaction statements Σ is a Hintikka set if it satisfies the following conditions.*

1. *For all atoms $\pi \in \Phi$, and all nominals s, if $@_s\pi \in \Sigma$ then $\neg@_s\pi \notin \Sigma$.*
2. *For all nominals s and t, if $@_s\Diamond t$ then $\neg@_s\Diamond t \notin \Sigma$.*
3. *If a nominal s occurs in any formula in Σ, then $@_s s \in \Sigma$.*
4. *If Σ contains a formula that one of the branching rules can be applied to, then it contains at least one of the formulas obtainable by making this application.*
5. *If Σ contains a pair of formulas that one of the binary rules can be applied to, then it contains all the formulas obtainable by making this application.*
6. *If Σ contains a formula that one of the existential rules can be applied to, then for some nominal i it also contains the formulas that would be obtained by applying that rule to that formula using i as the new nominal τ.*
7. *For any other rule, if σ contains a formula that one of the rules applies to, then it contains all the formulas obtainable by making this application.*

Definition 6.2.3. *Let Σ be a Hintikka set, we define* $\mathrm{NOM}(\Sigma)$ *to be:*

$$\{i | i \text{ is a nominal that occurs in some formula in} \Sigma\},$$

and define a binary $\sim_\Sigma$ on $\mathrm{TERM}(\Sigma)$ *by $i \sim_\Sigma j$ iff $@_i j \in \Sigma$. Clearly $\sim_\Sigma$ is an equivalence relation. If $k \in \mathrm{NOM}(\Sigma)$, then $|k|$ is the equivalence class of k under $\sim_\Sigma$.*

Remark 6.2.1. *Item* 3 *in the definition of Hintikka sets is an analog for first order reflexivity rule for* $=$*, while closure under* SYM *and* NOM *ensures symmetry and transitivity.*

Lemma 6.2.1. *Let* Σ *be a Hintikka set and assume that* $i, j \in \mathrm{NOM}(\Sigma)$*. Then the following assertions are equivalent:*

1. $i \sim_\Sigma j$
2. *For every formula* ϕ, $@_i\phi \in \Sigma$ *iff* $@_j\phi \in \sigma$

Proof. We start assuming $i \sim_\Sigma j$. Then $@_ij \in \Sigma$ and hence $@_ji \in \Sigma$. Since Hintikka sets are closed under NOM then we have: $@_i\phi \in \Sigma \text{iff} @_k\phi \in \sigma$. For the converse, assume that $@_i\phi \in \Sigma \text{iff} @_j\phi \in \sigma$. j occurs in Σ we have that $@_jj \in \Sigma$. Thus, if ϕ is j we have that $@_ij \in \Sigma$ then $i \sim_\Sigma j$. □

Definition 6.2.4. Model induced by a Hintikka set*: Given any Hintikka set* Σ*, there is a model* $\mathcal{M}_\Sigma = (W, R, V, F)$ — the model induced by Σ — *such that the following hold.*

1. $W = \{|k|_\Sigma \mid k \in \mathrm{NOM}(\Sigma)\}$.
2. $|\alpha_1|_\Sigma R |\alpha_2|_\Sigma$ iff $@_{\alpha_1}\Diamond\alpha_2 \in \Sigma$.
3. $F(|k_1|_\Sigma, |k_2|_\Sigma) = |\alpha|_\Sigma$ iff $@_\alpha f(k_1, k_2) \in \Sigma$
4. $|\alpha|_\Sigma \in V(p)$ iff $@_\alpha p \in \Sigma$
5. $|\alpha|_\Sigma \in V(j)$ iff $@_\alpha j \in \Sigma$. Note j is a nominal.

Lemma 6.2.2. *Let* Σ *be a Hintikka set. Then any model* $\mathcal{M}_\Sigma = (W, R, V)$ *of the kind just above difined is an model induced by* Σ*.*

Proof. In this lemma we should prove that R, V and F are well defined.

To show that F is well defined we need to show that for all $i, j, k, l \in \mathrm{NOM}(\Sigma)$, if $i \sim_\Sigma k$ and $j \sim_\Sigma l$ then $F(|i|, |j|) = |\alpha|$ implies $F(|k|, |l|) = |\alpha|$. That is, we need to show that F is function. Suppose that $i \sim_\Sigma k$, $j \sim_\Sigma l$ and $F(|i|, |j|) = \alpha$. That is, we want to show that $F(|k|, |l|) = |\alpha|$.

By definition this means that $@_\alpha f(i, j) \in \Sigma$, hence as $i \sim_\Sigma k$ and $j \sim_\Sigma l$ then by the replacement rule of $\mathcal{R}$: $@_\alpha f(k, l) \in \Sigma$. But since $@_\alpha f(k, l) \in \Sigma$, we have by item 3 that $F(|k|, |l|) = \alpha$ as required.

By the same way, R is well defined when for all $i, j, k, l \in \mathrm{NOM}(\Sigma)$ if $i \sim_\Sigma k$ and $j \sim_\Sigma l$ then $|i|R|j|$ implies $|k|R|l|$. Let us assume $i \sim_\Sigma k$, $j \sim_\Sigma l$ and $|i|R|j|$.

Since $|i|R|j|$ we have $@_i \Diamond j \in \Sigma$, hence as $i \sim_\Sigma k$ it follows that $@_k \Diamond j$. Also, since $|j|R|l|$ we also have that $@_j \Diamond l \in \Sigma$. Then by the bridge rule we have:

$$\text{Since } @_k \Diamond j \in \Sigma, @_j \Diamond l \in \Sigma \text{ then } @_k \Diamond l$$

Therefore, $|k|R|l|$ as required.

It is straightforward to see that V is well defined. But we need to show that for all nominals , $\mathrm{V}(\alpha)$ is a singleton. As we will see there is no need to study two cases, for atomic nominals and complex ones. Let us assume that α occurs in Σ. Then $\mathrm{V}(\alpha)$ contains $|\alpha|$, and as consequence $@_\alpha \alpha \in \Sigma$. Now, let us assume that $|\beta| \in \mathrm{V}(\alpha)$. Then $@_\beta \alpha \in \Sigma$, that is $\beta \sim_\Sigma \alpha$ which means that $|\beta| = |\alpha|$. So, for all nominals α, $V(\alpha)$ is a singleton subset of W.

□

Lemma 6.2.3. Truth lemma *Let Σ be a Hintikka set and let $\mathcal{M}_\Sigma$ be the model induced by Σ. Then for all terms α and formulas ϕ, the following hold.*

1. *If $@_\alpha \phi \in \Sigma$ then $M_\Sigma, |\alpha|_\Sigma \models \phi$*
2. *If $\neg @_\alpha \phi \in \Sigma$ then $M_\Sigma, |\alpha|_\Sigma \not\models \phi$*

That is, every formula in Σ is satisfied in $\mathcal{M}_\Sigma$.

Proof. The proof follows the lines of [8], so we just give two clauses leaving the remainder to the reader:

- ϕ has the form $f(s_1, s_2)$. Suppose $@_\alpha f(s_1, s_2)$. Using the condition 3 of Lemma 1 we know that $F(|s_1|, |s_2|) = |\alpha|$. By induction hypothesis we get $M, |\alpha| \models f(s_1, s_2)$.

- ϕ is of the form $@_l \psi$, l is a nominal.

  Suppose $@_s \phi \in \Sigma$. Then $@_s @_l \psi \in \Sigma$. Note that s is a nominal. By closure under the tableau rules for @, we can infer that $@_l \psi \in \Sigma$. By induction hypothesis, $M, |l| \models \psi$. Using the fact that the denotation of l in M is $|l|$, concluding that $M, |s| \models @_l \psi$

  Suppose $\neg @_s \phi \in \Sigma$. Then $\neg @_s @_l \psi \in \Sigma$. By closure under the tableau rules for @, we can infer that $\neg @_l \psi \in \Sigma$. By induction hypothesis, $M, |l| \not\models \psi$. Using the fact that the denotation of l in M is $|l|$, we conclude that $M, |s| \not\models @_l \psi$

□

Chapter 7

Equational Logic, Term Rewriting and $\mathcal{HLON}$

An equational system is a set of equations. In other words we can say that one equational system $\mathcal{E}$ is a set of universally closed equations, that is, equations without existential quantifiers. $\mathcal{HLON}$ as we know provides full first order expressetivity: using the technique of skolemization we were able to express any first order concept of $\mathcal{L}$ using only universal quantifiers.

Note that many interesting classes of algebras are defined by equations. We give an example of: semigroups and monoids. Mainly we are interested in knowing whether an equation follows logically from a given set of equations.

Definition 7.0.5. *(Syntax of Algebras) Let σ an algebraic similarity type and a set $\mathcal{X}$ of variables. The language $\mathcal{E}$ of algebras (equational systems) is built up sa follows:*

$$t := x \mid f(t_1, ..., t_{\rho(f)})$$

we call t, $\mathcal{E}-$ terms.

Definition 7.0.6. *(Equation System) An $\mathcal{E}(X, \sigma, =)$ equation system is an expression of the form:*

$$s = t$$

where s, t are $\mathcal{E}-$ terms.

Definition 7.0.7. *(Semantics of Equational systems)*

A `structure` *$\mathcal{M}$ for the equational system $\mathcal{E}$ is $\mathcal{M} = (D, (F^f)_{f\in\sigma}, V)$ such that:*

- *$D \neq 0$, a non empty domain D;*
- *for each $n \geq 0$ and for all $f \in \tau$, if f is an $n-$ary function symbol then F^f is an $n-ary$ operation symbol of D. Note that $F : D^n \to D$;*
- *A variable x is assigned a meaning $V(x)$ which is an element D;*
- *The $V(t)$ where t is a term, is defined as follows:*
  - *If t is a variable x then: $V(x)$ is an element D;*
  - *If t is $f(t_1, \cdots, t_n)$ then: $V(f(t_1, \cdots, t_n)) = F(V(t_1), \cdots, V(t_n))$*

Remark 7.0.2. *In many textbooks instead of $s = t$ the following notation is used:*

$$\forall x_1, ..., x_n \quad s = t$$

where $x_1, ..., x_n$ is the set of variables occurring in the terms s, t. We assume that $x_1, x_2, ..., x_n$ are particular variables, but their interpretation in the model is arbitrary, so the universal quantification is implicit.

Example 7.0.2. *(***Groups***)*

$$\forall x, \quad x \cdot 1 = x \tag{7.1}$$

$$\forall x, \quad x \cdot x^{-1} = 1 \tag{7.2}$$

$$\forall x, \quad (x \cdot y) \cdot z = x \cdot (y \cdot z) \tag{7.3}$$

where $(\cdot)^{-1}$ is a unary operation.

Definition 7.0.8. Truth for Equational Logics

If $\mathcal{M}$ is a structure, we write $\mathcal{M} \models s = t$ to indicate that the equation $s = t$ is true in $\mathcal{M}$, or formally, that $\mathcal{M}$ satisfies the equation $s = t$. That is:

$$\mathcal{M} \models s = t \quad \textit{iff} \quad V(s) = V(t) \ \textit{ where } V(s), V(t) \textit{ are identical elements of } D.$$

We say that a structure $\mathcal{M}$ satisfies a set E of equations, written $\mathcal{M} \models E$, if $\mathcal{M}$ satisfies each element of E. Such a structure is called a **model** of E.

If E_1 and E_2 are two equational systems, we write $\mathcal{E}_1 \models \mathcal{E}_2$ **if all structures** $\mathcal{M}$ satisfy $\mathcal{E}_1$ also satisfy $\mathcal{E}_2$. That is, all models of $\mathcal{E}_1$ are also models of $\mathcal{E}_2$. In this case if $\mathcal{E}_1 \models \mathcal{E}_2$ we call $\mathcal{E}_2$ a logical consequence of $\mathcal{E}_1$.

The main question that this section addresses is to determine when $\mathcal{E}_1 \models \mathcal{E}_2$ for various $\mathcal{E}_1$ and $\mathcal{E}_2$. Mainly $\mathcal{E}_2$ will consist of a single equation. In other words, we are interested in knowing whether this equation is a **logical consequence** of $\mathcal{E}_1$. We will investigate this problem in the context of $\mathcal{HLON}$.

Remark 7.0.3. *Enriching the algebraic language of nominals (Definition 19) with the operator @, we form an equational system. The only difference is how we represent the equality symbol. The relation between equational systems and hybrid logics is mentioned in [4] as follows:* `Thus hybrid logic is genuine hybrid: it brings to modal logic the classical concepts of identity and reference.`

Our contribution makes this relation between Hybrid Logics and Equational systems complete.

Definition 7.0.9. *Suppose $\mathcal{E}$ is a language of algebras.*

- *For $\mathcal{M}$ an $\mathcal{E}-$ structure and $s = t$ an $\mathcal{L}-$ equation we write*

$$\mathcal{M} \models s = t$$

if $V(s) = V(t)$. (read: M satisfies $s = t$ or $s = t$ holds in $\mathcal{M}$.

- *If $\mathcal{M}$ is an $\mathcal{E}-$ structure and $\mathcal{S}$ is a set of $\mathcal{E}-$ equations, then*

$$\mathcal{M} \models \mathcal{S}$$

if $\mathcal{M} \models s = t$ for every equation $s = t$ in $\mathcal{S}$. (read: $\mathcal{M}$ satisfies $\mathcal{S}$ or $\mathcal{S}$ in $\mathcal{M}$)

- *If $\mathcal{S}$ is a set of $\mathcal{E}-$ equations and $s = t$ is an $\mathcal{L}-$ equation, then we write*

$$\mathcal{S} \models s = t$$

if $\mathcal{M} \models s = t$ whenever $\mathcal{M} \models \mathcal{S}$. (read: $s = t$ is a consequence of $\mathcal{S}$ or follows from $\mathcal{S}$)

7.1 Proof systems

In this section we discuss methods for showing that $E \models s = t$ for various E, s and t. We will translate these results in the context of hybrid logic.

Definition 7.1.1. *A* **substitution** *is function from variables to terms, which is identity on all but finitely many variables.*

Example 7.1.1. *Let us assume that $x \cdot y = y \cdot x$. We can deduce that $(x \cdot y) \cdot (z + z) = (z + z) \cdot (x \cdot y)$ substituting every occurrence of x by the term $x \cdot y$ and substituting every occurrence of y by the term $z + z$.*

We will use Greek letters such as θ, ρ, σ for substitution. If θ is substitution and x is a variable, then we indicate the application of θ to x by $\theta(x)$ as usual. We extend these definitions inductively for terms:

- If t is a variable then $t\theta$ is $\theta(t)$.
- Otherwise, t is of the form $f(t_1, ..., t_n)$ for some non -zero n. Then $t\theta$ is $f(t_1\theta, ..., t_n\theta)$. That is, $t\theta$ is t with all variables replaced by terms as specified by θ.

Remark 7.1.1. *Some remarks about the using notation:*

- *It is common to write* $\{x_1 \mapsto s_1, ..., x_n \mapsto s_n\}$*, indicating that the variable* x_i *is mapped to* s_i *by the substitution. We call* $t\theta$ *an instant of* t*.*
- *If* $t = u$ *is an equation and* θ *is a substitution, we call* $t\theta = u\theta$ *an instance of the equation* $t = u$*.*

Rule	Name	Example
$\dfrac{\text{true}}{s = s}$	Reflexive	$\dfrac{}{x + y = x + y}$
$\dfrac{s = t}{t = s}$	Symmetric	$\dfrac{x = x \cdot x}{x \cdot x = x}$
$\dfrac{r = s \quad s = t}{r = t}$	Transitive	$\dfrac{x = x \cdot x \quad x \cdot x = 1}{x = 1}$
$\dfrac{t = u}{t\theta = u\theta}$	Substitution	$\dfrac{x \cdot 1 = x}{(x \cdot x) \cdot 1 = x \cdot x}$
$\dfrac{t = u}{f(..., t, ...) = f(..., u, ...)}$	Replacement	$\dfrac{x \cdot x = x}{(x \cdot x) + x = x + x}$

This result is due to Birkhoff and we will use it in the context of $\mathcal{HLON}$:

Theorem 7.1.1. *If* E *is a set of equations then* $E \models s = t$ **iff** $r = s$ *is derivable from* E *using these rules.*

We can state this result in an equivalent way: $E \models r = s$ iff there is a finite sequence $u_1, ..., u_n$ of terms such that r is u_1 and s is u_n, further for all i, u_{i+1} is obtained from u_i by replacing a subterm t of u_i by a term u, where the equation $t = u$ or the equation $u = t$ is an instance of an equation in E. In more formal way:

Definition 7.1.2. *A derivation of an equation* $s = t$ *of one set of equations* $\mathcal{E}$ *is a sequence:* $s_1 = t_1, ..., s_n = t_n$ *of equations such that each equation is either:*

- *a member of* $\mathcal{E}$ *or*
- *is the result of applying a rule of inference to previous members of the sequence*

and the last equation is $s = t$. *We write* $\mathcal{E} \Vdash s = t$ *if such a derivation exists.*

7.2 Hyrbid Languages and Equations

In this section we rephrase basic notions of Equational Logic in the context of $\mathcal{HLON}$.

Definition 7.2.1. *Sentences of* $\mathcal{HLON}$ *of the form:* $@_s t$ *where* s *and* t *are nominals (atomic or complex nominals) are called* $@-$**sentences**.

Recall that equational sentences are universal first order sentences. Then an immediate consequence of proposition 3 is:

Proposition 7.2.1. *For any equality* $s = t$ *of one equational system* $\mathcal{E}$ *there is an* $@-$**sentence**, *such that for any model* $\mathcal{M}$, $s = t$ *is true in* $\mathcal{M}$ **iff** $@_s t$ *is valid on* $\mathcal{M}$.

Theorem 7.2.1. *(***Birkhoff***)*
If $\mathcal{A}$ *is a set of* $@-$**sentences** *then* $\mathcal{A} \models @_s t$ *if* $@_s t$ *is derivable from* $\mathcal{A}$ *using the following inference rules.*

Rule	Name
$\dfrac{\text{true}}{@_s s}$	Reflexive
$\dfrac{@_s t}{@_t s}$	Symmetric
$\dfrac{@_r s \quad @_s t}{@_r t}$	Transitive
$\dfrac{@_t u}{@_{t\theta} u\theta}$	Substitution
$\dfrac{@_t u}{@_{f(\ldots,t,\ldots)} f(\ldots,u,\ldots)}$	Replacement

7.3 Hybrid Logic and Algebras

Some of the most interesting classes of algebras in mathematics are defined by equations. We will introduce some of them in the context of $\mathcal{HLON}$.

But first let us start with one example.

Example 7.3.1. *Now let us suppose our language has one binary function symbol f. The interpretation of this symbol in our model is $F^f = F$. What we can say about the nominals z, z_1 if $@_{f(x,z)}f(x, z_1)$? The interpretation of $@_{f(x,z)}f(x, z_1)$ is $F(V(x), V(z)) = F(V(x), V(z_1))$. Since F is function, then is one-to-one, therefore $V(z) = V(z_1)$. That is $@_z z_1$.*

Definition 7.3.1. Semigroups $\mathcal{SG}$ *is the following set consisting of one equation (in the language $\mathcal{L}_{\mathcal{SG}} = \{\cdot\}$ of a single binary operation) that defines* `semigroups`*, namely,*

$$\mathcal{SG} = \{(x \cdot y) \cdot z = x \cdot (y \cdot z)\}$$

Using the syntax of $\mathcal{HLON}$ we have :

$$\mathcal{SG} = \{@_{(x\cdot y)\cdot z} x \cdot (y \cdot z)\}$$

Models of $\mathcal{SG}$ are called semigroups and $\mathcal{SG}$ axiomatizes or defines the class of semigroups. Semigroups come up in computer science in the study of languages, or in any subject that is based on the study of strings of symbols.

Definition 7.3.2. Monoids $\mathcal{M}$ *is the following set of three equations (in the Language $\mathcal{L}_{\mathcal{M}} = \{\cdot, 1\}$ where $\cdot$ is binary and 1 is constant symbol) that defines* `monoids`*, namely,*

$$\mathcal{M} = \{(x \cdot y) \cdot z = x \cdot (y \cdot z), x \cdot 1 = x, 1 \cdot x = x\}$$

In $\mathcal{HLON}$ we have the formulas:

$$\mathcal{M} = \{@_{(x\cdot y)\cdot z} x \cdot (y \cdot z), @_{x\cdot 1} x, @_{1\cdot x} x\}$$

Any algebra of nominals that satisfies $\mathcal{M}$ is called monoid, and $\mathcal{M}$ is a set of axioms or defining equations for monoids.

Monoids are very close to semigroups. Every monoid can be viewed as a semigroup (drop the constant symbol).

Chapter 8

Conclusions and Further work

Here we conclude with a discussion of several possible directions in which this work can be extended. Some of them have been addressed to some extend in the preceding:

- **Combination of Logics**: We presented an extension of the basic hybrid logic $\mathcal{H}(@)$: $\mathcal{HLON}$, In this thesis we considered an application of hybrid logics to relational structures on algebras, thus a set with a relation and an algebraic structure. But this is not the only application. The notion of Universal Logic seems to be useful here:

  Universal logic is not a new logic, it is a way of unifying various of logics by developing general tools and concepts that can be applied to all logics

  That is, using techniques of Universal Logic we can design a specific logic required for a given problem.

- **Skolemization**: By the method of skolemization we showed that hybrid modal logic can be used as a worthy alternative to $\mathcal{L}$ logic (with binary relation symbols). Can we extend this result for arbitary relation symbols? Another line of research is to show that the hybrid modal logic with function symbols is **still decidable**, and **expressive enough** to say useful things, that is to derive interesting facts. Obviously, for that purpose one should only consider relatively weak hybrid languages, i.e. just with nominals and @ or universal modality. I want to thank Prof. Valentin Goranko for these observations.

- Herbrand's theorem. It would be interesting to study the theorem of Herbrand in the context of hybrid logics. A guide for further research is the following problem 5.8.12 from [28].

- Hybrid Logics: It would be worth studying the relation of $\mathcal{HLON}$ with other hybrid logics. The following question seems attractive: What kind of function symbols we should add to $\mathcal{H}(@)$ in order to capture the expressive power of $\mathcal{H}(@, \downarrow)$. That is, which function symbols behave like the modal operator $\downarrow$?

- **Nominals**: There is some interest in this extended hybrid logic and its connection to other formalisms that use nominals: for instance term rewriting, monadic second order logic and description logics. Another direction is to investigate the relation of these extended hybrid logics with Fitting's First Order Intensional Logic [21]. Fitting writes: *In addition to objects (variables) there will be what we call intensions or intensional objects or concepts. Typical informal intensions are the morning star, the oldest person in the world*, or simply that. Intensions designate different objects under different objects under different circumstances-they are non rigid designators. As such, they will be modelled by functions from possible worlds to objects. There will be quantification over intensions, as well as quantification over objects.

  The use of these functions seems quite similar with the functions in $\mathcal{HLON}$.

- **Numerical Hybrid Logics:** Designing formal languages for real world applications many times requires the ability to express and to reason numerically. There is a natural extension of $\mathcal{HLON}$ with number terms. That is, we can consider an application of hybrid logics to relational structures on algebras with numerical functions, thus a set with a relation and an algebraic structure and functions assigning numerical values to worlds. The meaning of these values depends on applications. Our main goal is to relate our contribution with two other logics: distance logics [20], [2], [3], [19] and probabilistic logics [22], [18].

  That is, the numerical Hybrid Logic $\mathcal{NHLON}$ views function symbols as terms, these terms do not denote elements of our model, but they denote elements of the set $\mathbb{N}$ or $\mathbb{Q}$ or $\mathbb{R}$. Their choice depends on applications. But of course we can still allow algebraic operators only for nominals: denoting elements of our domain.

  In [20] the following problem addressed: *Some applications might require that we have some algebraic structure on the parameter sets different to the reals, such as finite groups, which we could also incorporate into the formal language.*

  $\mathcal{HLON}$ seems a good candidate for this direction.

Bibliography

[1] Balder ten Cate, Model theory for extended modal languages. PhD thesis, University of Amsterdam, 2005.

[2] Holger Strum, Nobu-Yuki Suzuki, Frank Wolter and Michael Zakharyaschev, Semi-qualitative reasoning about distances: a preliminary report.

[3] Oliver Kutz, Holger Strum, Nobu-Yuki Suzuki, Frank Wolter and Michael Zakharyaschev, Axiomatizing distance logics.

[4] Patric Blackburn, Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto. Proc. of the 1st. Method for Modalities Workshop. Amsterdam. Areces, C., Franconi, E., Gor, R., de Rijke, M., Schlingloff, H., editors. Special Issue of the Logic Journal of the IGPL. Vol 8:3, 339-625, 2000.

[5] Blackburn, P., Internalizing labelled deduction. Journal of Logic and Computation,10(1): 137 - 168, 2000.

[6] Patrick Blackburn, Maarten de Rijke and Yde Venema. Modal Logic, Cambridge University Press, 2001.

[7] Patrick Blackburn and Balder ten Cate, Beyond Pure Axioms: Node Creating Rules in Hybrid Tableaux. In Carlos Areces, Patrick Blackburn, Maarten Marx and Ulrike Sattler, editors, Proceedings of the 4th Workshop on Hybrid Logics (HyLo 2002).

[8] Patrick Blackburn, Internalizing labelled deduction. Journal of Logic and Computation,10(1): 137 - 168, (2000).

[9] Carlos Eduardo Areces. Logic Engineering, The case of Description and Hybrid Logics, PhD thesis, ILLC, University of Amsterdam, 2000.

[10] George Gargov and Valentin Goranko. Modal Logic with names. Journal of Philosophical Logic, 22:607-636, 1993.

[11] Valentin Goranko and Dimiter Vakarelov: Universal Algebra and Modal Logic, in: Advances in Modal Logic, vol. II, CSLI publications, Stanford, 2000, 265-292.

[12] Robin Hirsch, Ian Hodkinson. Relation Algebras by Games, North Holland, volume 147.

[13] Alexander Chargov and Michael Zakharyaschev. Modal Logic, Oxford University Press, 1997.

[14] Hylo page. http://www.hylo.net/, Papers, Bibliography, Course and Slides....

[15] Maarten Marx, Relation Algebra with binders, Journal of Logic and Computation, Vol 11 Nr 5 (2001) 691-700.

[16] A. Tarski, S. Givant, A formalization of Set Theory without variables, volume 41. AMS Coloquim publications, Providence Rhode Island, 1987.

[17] David Basin, Sean Matthews, Luca Vigano, Labelled Propositional Modal Logics: Theory and Practice, Journal of Logic and Computation, 7(6): 685-717,1997, Oxford University Press.

[18] Joseph Y. Halpern, An Analysis of First Order logics of probability, Artificial Intelligence 46, pp. 311-350.

[19] F. Wolter and M. Zakharyaschev, Reasoning about distances, Proceedings of the 18th International Joint Conference on Arificial Intelligence (IJCAI 2003), pages 1275-1280, Acapulco, Mexico.

[20] O.Kutz, E-Connections and Logics of Distance. Ph.D. thesis, University of Liverpool, 2004.

[21] Melvin Fitting, First-order intensional logic, Annals of Pure and Applied Logic, 127: 171–193 (2004).

[22] A.J. Pfeffer, Probabilistic Reasoning for Complex Systems, PhD Thesis, Stanford University, January 2000.

[23] Areces, C., P. Blackburn, and M. Marx, Hybrid logics: Characterization, interpolation and complexity. Journal of Symbolic Logic, 66(3), 977-1010, 2001.

[24] J. Herbrand, Sur la theoerie de la demonstration (On the theory of proof), Comptes Rend. Acad. des Sciences, Paris, 186, pp. 1274-1276, 1928.

[25] Stanley N. Burris, Logic for Mathematics and Computer Science, Prentice Hall, 1998.

[26] Stanley Burris, H P Sankappanavar, A course in universal algebra, Springer-Verlag, New York,1981.

[27] Jean H. Gallier, Logic For Computer Science Foundations of Automatic Theorem Proving, June 2003. (http://www.dit.hcmut.edu.vn/∼ntson/book03.pdf)

[28] Christos Papadimitriou, Computional Complexity, Addison Wesley Longman, 1994.

[29] Herbert B., Enderton Mathematical Introduction to Logic, Academic Pr Inc 2000.