Resolution for Higher-Order
Constrained Horn Clauses

submitted by
Dominik Wagner
Magdalen College
on 3rd September 2018

Supervisor
Prof. Dr. Luke Ong

A thesis submitted in partial fulfilment of the requirements for the degree of
MSc in Mathematics and Foundations of Computer Science
Trinity Term 2018
Abstract

In this work we continue the investigation into the algorithmic and model theoretic properties of higher-order constrained Horn clauses, a fragment of higher-order logic with background theories proposed in [Cathcart Burn et al., 2018] for the verification of higher-order functional programs. As an important stepping stone, we consider theories with a single (standard) model, such as linear arithmetic, although we demonstrate that our approach is more versatile and can also deal with some theories with an infinite number of models.

Notwithstanding the fact that full higher-order logic with respect to standard semantics is not recursively enumerable, we present a simple resolution proof system for higher-order constrained Horn clauses and prove its soundness and (refutational) completeness with respect to both standard and Henkin semantics. As a corollary, we obtain an alternative proof of the equivalence of standard, continuous and monotone semantics for higher-order constrained Horn clauses.

For the completeness proof we establish novel model theoretic properties which are refinements of known negative results: (i) The structure obtained by iterating the immediate consequence operator is a least model for a carefully chosen relation and (ii) the immediate consequence operator is “quasi-continuous”.

Finally, we prove that the well-known translation from higher-order to first-order logic is not only sound and complete for Henkin semantics but also for standard semantics. This gives rise to another semi-decision procedure provided that the background theory is decidable.
Contents

1 Introduction 1
  1.1 Background .......................................................... 1
  1.2 Contributions .................................................... 2
  1.3 Outline ............................................................... 3
  1.4 Preliminaries ...................................................... 4
    1.4.1 Relational Higher-Order Logic ............................... 4
    1.4.2 Reduction .................................................... 10
    1.4.3 Theorem Proving for Hierarchic First-Order Theories .... 10

2 (Higher-Order) Constrained Horn Problems and Clauses 13
  2.1 Syntax ............................................................... 13
  2.2 Semantics ........................................................... 15
  2.3 Normal Form and Translations .................................. 15
  2.4 Undecidability ....................................................... 16
  2.5 Discussion .......................................................... 19

3 Resolution and Model Theoretic Properties 21
  3.1 Resolution Proof System ........................................ 21
    3.1.1 Outline of the Completeness Proof ......................... 24
  3.2 Canonical Structure ................................................ 25
    3.2.1 Construction and Basic Properties ......................... 25
    3.2.2 Quasi-Monotonicity ......................................... 27
    3.2.3 Quasi-Continuity ........................................... 30
  3.3 Refutational Completeness of the Proof System .............. 34
    3.3.1 Syntactic Unfolding ....................................... 34
    3.3.2 Leftmost Reduction ....................................... 35
    3.3.3 Refutational Completeness ................................. 39

4 Ramifications 43
  4.1 Equivalence of Semantics ....................................... 43
  4.2 Elimination of Nested Terms ................................. 45
Chapter 1

Introduction

1.1 Background

Verification of Higher-Order Programs  Since the early days of computer science, researchers have been concerned with the correctness of computer programs and the question of what being “correct” actually should mean. A common thread was that pioneers like Sir Tony Hoare, Robert W. Floyd, John C. Reynolds, Dana Scott and Christopher Strachey used logical methods in their seminal work to analyse computer programs (see e.g. [Hoare, 1969, Floyd, 1967, Reynolds, 1972, Scott and Strachey, 1971]).

In the following decades, significant breakthroughs have been made both in terms of theory and the implementation of verification systems. Current state-of-the-art tools leverage the power of SAT and SMT-solvers (see e.g. [Barrett et al., 2009]), which have seen enormous progress in the past two decades, and can be employed to verify code arising in industry with several thousand lines of code. Furthermore, the authors of [Bjørner et al., 2015, Bjørner et al., 2012, Grebenshchikov et al., 2012] advocate using constrained Horn clauses, a fragment of first-order logic with respect to background theories, as a basis for program verification.

The verification community has mostly focused on (first-order) imperative programs, notwithstanding the fact that ideas of the functional programming paradigm have become widespread in practical use. Yet, the verification of higher-order functional programs has received comparatively little attention and has resulted in the development of higher-order model checking (for an overview see [Ong, 2015]) and refinement type inference [Unno et al., 2013].

More recently, [Cathcart Burn et al., 2018] extended (first-order) Horn clauses to a higher-order counterpart, which is suitable for the verification of functional programs.

Consider the (ML-like) functional program in Fig. 1.1, which evaluates to 5. We can axiomatise the overapproximations of the graphs of the functions by the following (higher-order) constrained Horn clauses [Cathcart Burn et al., 2018]:

\[-(z = x + y) \lor \text{Add } x \ y \ z\]
\[-(n \leq 0) \lor -(s = x) \lor \text{Iter } f \ s \ n \ x\]
\[-(n > 0) \lor -(\text{Iter } f \ s \ (n - 1) \ y) \lor -(f \ n \ y \ x) \lor \text{Iter } f \ s \ n \ x.\]
let add x y = x + y

in let rec iter f s n = if n <= 0 then s
else f n (iter f s (n-1))

in iter add 2 2

Figure 1.1: A simple functional program

Besides, the verification of the (safety) conjecture “for every \( n \geq 1 \), it holds that \( \text{iter add } n n > n + n \)” corresponds to the additional clause

\[ \neg(n \geq 1) \lor \neg(\text{Iter Add } n n x) \lor \neg(x \leq n + n). \]

**Higher-Order Logic**  Whilst first-order logic enjoys many attractive meta-logical properties, such as being compact and having refutationally complete proof systems [Gödel, 1929], higher-order logic suffers from an intrinsic incompleteness: It is a consequence of Gödel’s seminal result [Gödel, 1931] that it is impossible to devise a method which allows us to prove all valid sentences of higher-order logic. For that reason, it has been largely disregarded by mathematicians and computer scientists alike.

Two decades later, Leon Henkin showed that it is in fact possible to obtain refutationally complete proof systems if the notion of models is relaxed by not insisting that quantifiers range over all functions and relations [Henkin, 1950]. Conceptually, higher-order logic with Henkin semantics is nothing but a (many-sorted) first-order theory.

However, there are very few known fragments of higher-order logic possessing desirable properties with respect to standard semantics (as opposed to Henkin semantics). Notable exceptions include some decidable monadic theories such as the monadic second-order theory of the infinite binary tree [Rabin, 1969].

### 1.2 Contributions

The main contribution of this dissertation is the design of a simple resolution proof system for higher-order constrained Horn clauses with a single background theory, and its refutational completeness proof with respect to arbitrary Henkin frames (in particular standard frames). As far as we are aware, this is the first proof system for a significant non-monadic fragment of higher-order logic modulo a background theory which is refutationally complete for standard semantics.

We achieve the simplicity of the proof system by resorting to a normal form which allows us to keep reasoning about logical connectives at a minimum level and to focus on the essentials. In the course of this work, we prove that this is indeed a normal form, i.e. also seemingly more general formulas can be transformed in a meaning-preserving way to this normal form.

As a commendable corollary of the soundness and completeness results, we prove that satisfiability is independent of the choice of a particular Henkin frame. In particular, this constitutes an alternative proof of the equivalence of standard, monotone and continuous semantics for higher-order constrained Horn clauses without establishing translations
explicitly. Furthermore, this demonstrates that, in contrast to full higher-order logic, satisfiability with respect to standard and Henkin semantics coincide.

Moreover, we show that a variant of the standard (applicative) translation of higher-order logic into first-order logic is sound and complete not only for Henkin but also for standard semantics. To underpin the practicality of this approach, we argue that the target logic of this encoding is semi-decidable if the background theory is decidable by building on work on automated theorem proving for first-order theories. In particular it is not necessary to include extensionality axioms and only a weak form of comprehension is required.

The completeness proof of our resolution proof system hinges on a number of novel model-theoretic insights. It is well-known that higher-order constrained Horn clauses do not necessarily possess least models with respect to the pointwise ordering. We prove that the structure obtained by iterating the immediate consequence operator does constitute a least model of definite clauses for a carefully chosen relation which respects formulas without negation and universal quantification. Therefore, this canonical structure is a model of any satisfiable set of constrained Horn clauses.

By a similar technique, we prove that the immediate consequence operator is in fact “quasi-continuous”, although it is not continuous in the standard sense (i.e. with respect to the Scott topology). Consequently, if the canonical structure falsifies a goal clause, this already holds after a finite number of iterations of the construction.

Having established these essential properties, the completeness proof follows a similar route to existing proofs for extensional higher-order logic programming. However, our presentation makes a connection to the standardisation theorem of the $\lambda$-calculus, which makes the argumentation very transparent and mostly reduces the completeness proof to straightforward inductions on inductively defined relations.

Moreover, we demonstrate that our approach is more versatile than appears at first glance: it is not only appropriate for background theories with a single standard model, but it can also deal with the (satisfiability) problem considered in extensional higher-order logic programming.

To round off the presentation, we give an account of classic proofs showing the undecidability of the satisfiability problem for (higher-order) constrained Horn clauses.

1.3 Outline

This dissertation is structured as follows: Before defining higher-order constrained Horn clauses in Chapter 2, we review the preliminaries in Section 1.4. Chapter 2 also defines (higher-order) constrained Horn problems in program form, their normal form and it gives an account of undecidability results.

Chapter 3 is devoted to the introduction of our resolution proof system, its soundness and the proof of its refutational completeness with respect to arbitrary Henkin frames.

In Chapter 4, we draw a number of conclusions from the technical results of the previous chapter: In Section 4.1, we show that the satisfiability problem is invariant under the choice of Henkin frames. In Section 4.2 we justify using normal forms and in Section 4.3 we argue that our approach can also be used in some unconstrained settings. Section 4.4 is dedicated to a reduction to first-order logic with background theories.
Finally, in Chapter 5 we relate our efforts to existing work, draw conclusions from our results and discuss future directions.

1.4 Preliminaries

In this section, we review basics about higher-order logic, the $\lambda$-calculus and theorem proving for hierarchic first-order theories.

1.4.1 Relational Higher-Order Logic

This subsection introduces the syntax and semantics of a restricted form of higher-order logic as well as the notions of substitutions and contexts.

1.4.1.1 Syntax

The set of argument types, relational types, first-order types and types are mutually recursively defined by

$$
\tau ::= \iota | \rho \\
\rho ::= o | \tau \rightarrow \rho \\
\sigma_{\text{FO}} ::= o | \iota | \iota \rightarrow \sigma_{\text{FO}} \\
\sigma ::= \rho | \sigma_{\text{FO}},
$$

respectively. We sometimes use the abbreviations $\iota^n \rightarrow \iota$ and $\iota^n \rightarrow o$ for the (first-order) types $\iota \rightarrow \cdots \rightarrow \iota \rightarrow \iota$ and $\iota \rightarrow \cdots \rightarrow \iota \rightarrow o$, respectively. Besides, for types $\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow \sigma$ we also write $\tau \rightarrow \sigma$. Intuitively, the type $\iota$ corresponds to the individuals and the type $o$ to the truth values (or Booleans). Besides, $\sigma_{\text{FO}}$ contains all (first-order) types of the form $\iota^n \rightarrow \iota$ or $\iota^n \rightarrow o$, i.e. all arguments are of type $\iota$. Moreover, each relational type has the form $\tau \rightarrow o$.

A type environment is a set of distinct typed variables $x : \tau$. We assume that for each type there are at least countably infinite many variables of that type and write $\Delta(x) = \tau$ if $\tau$ is the unique argument type such that $x : \tau \in \Delta$. Besides, $\text{dom}(\Delta)$ is the set of variables such that $x : \tau \in \Delta$.

A signature $\Sigma$ is a set of distinct typed symbols $c : \sigma$. It is first-order if for each $c, c : \sigma_{\text{FO}}$. A symbol is logical if it is one of $\neg : o \rightarrow o$, $\wedge, \vee : o \rightarrow o \rightarrow o$, $\exists_\tau : (\tau \rightarrow o) \rightarrow o$ where $\tau$ is an argument type. The set of $\Sigma$-pre-terms is given by

$$
M ::= x | c | \neg | \wedge | \vee | \exists_\tau | MM | \lambda x. M,
$$

where $c \in \Sigma$. Following the usual conventions we assume that application associates to the left and the scope of abstractions extend as far to the right as possible. We also write $M N$ and $\lambda x. M'$ for $M N_1 \cdots N_n$ and $\lambda x_1. \cdots \lambda x_n. M'$, respectively, assuming implicitly that $M$ is not an application. Besides, we abbreviate $\exists_\tau, \lambda x. M$ as $\exists_\tau x. M$ and sometimes we even drop the subscript. Moreover, we identify terms up to $\alpha$-equivalence and we use the following variable convention taken from [Barendregt, 2012]:

...
Convention 1.1 (Variables). “If $M_1, \ldots, M_n$ occur in a certain mathematical context (e.g., definition, proof), then in these terms all bound variables are chosen to be different from the free variables.”

The notion of typing judgement $\Delta \vdash M : \sigma$ is defined by

\[
\begin{align*}
\Delta \vdash x : \Delta(x) \\
\Delta \vdash c : \sigma & \quad c \in \Sigma \\
\Delta \vdash M_1 : \sigma_1 \rightarrow \sigma_2 & \quad \Delta \vdash M_2 : \sigma_1 \\
\Delta \vdash M_1M_2 : \sigma_2 \\
\Delta \vdash \lambda x. M : \Delta(x) \rightarrow \rho \\
\Delta \vdash \circ : o \rightarrow o \rightarrow o & \quad \circ \in \{\land, \lor\} \\
\Delta \vdash \exists \tau : (\tau \rightarrow o) \rightarrow o \\
\Delta \vdash \neg M : o
\end{align*}
\]

We say that $M$ is well-typed if $\Delta \vdash M : \sigma$ for some type $\sigma$. A $\Sigma$-pre-term which is well-typed is called a $\Sigma$-term and a $\Sigma$-term of type $o$ is a $\Sigma$-formula. We use the notion of subterms in the standard sense. A $\Sigma$-term ($\Sigma$-formula) $M$ is a first-order $\Sigma$-term (first-order $\Sigma$-formula) if for all subterms $N$ of $M$, $\Delta \vdash N : \sigma_{FO}$. We call a pre-term of the form $\lambda x. M$ a $\lambda$-abstraction. Besides, for a $\Sigma$-term $M$, the set of free variables $\text{free}(M)$ is defined in the standard way and $M$ is a ground $\Sigma$-term if $\text{free}(M) = \emptyset$.

Remark 1.2. In contrast to similar definitions in the literature, each term $\Delta \vdash M : \iota$ in particular can only contain variables of type $\iota$ and constants of non-relational first-order type, and contains neither $\lambda$-abstractions nor logical symbols. A similar approach is adopted in [Charalambidis et al., 2013].

Furthermore, the negation symbol ($\neg$) can only occur in a term if another term is applied to it (and not in pre-terms of the form $\neg R$).

In this work, the following kind of terms will be of particular significance:

Definition 1.3. (i) A $\Sigma$-term $M$ is positive existential if the logical constant “$\neg$” is not a subterm of $M$.

(ii) A $\Sigma$-formula $F$ is positive existential if the logical constant $\neg$ is not a subterm of $F$.

Since we chose not to include a logical constant for universal quantification this means that it is impossible to (implicitly) quantify variables universally in positive existential formulas.

1.4.1.2 Semantics

A pre-Henkin frame $\mathcal{H}$ assigns to each type $\sigma$ a non-empty set $\mathcal{H}[\sigma]$ satisfying

(i) $\mathcal{H}[o] = \mathbb{B} = \{0, 1\}$,

(ii) for each type $\sigma_1 \rightarrow \sigma_2$, $\mathcal{H}[\sigma_1 \rightarrow \sigma_2] \subseteq [\mathcal{H}[\sigma_1] \rightarrow \mathcal{H}[\sigma_2]]$,

(iii) and, or $\in \mathcal{H}[o \rightarrow o \rightarrow o]$,

(iv) for every argument type $\tau$, exists $\mathcal{H}(\tau \rightarrow o) \rightarrow o$,

(v) for every relational type $\rho$, $\top^\mathcal{H}_\rho, \bot^\mathcal{H}_\rho \in \mathcal{H}[\rho]$ and

(vi) for every relational type $\rho$ and $\mathcal{R} \subseteq \mathcal{H}[\rho]$, $\bigcup_{\rho} \mathcal{R} \in \mathcal{H}[\rho]$,.
where \( \mathcal{H}[[\sigma_1]] \to \mathcal{H}[[\sigma_2]] \) is the set of functions \( \mathcal{H}[[\sigma_1]] \to \mathcal{H}[[\sigma_2]] \),

\[
\begin{align*}
    \text{and} & (b_1)(b_2) = \min\{b_1, b_2\} \\
    \text{or} & (b_1)(b_2) = \max\{b_1, b_2\} \\
    \text{exists} & \mathcal{H}(r) = \max\{r(s) | s \in \mathcal{H}[[\tau]]\} \\
    \mathcal{H}_{\tau\to o}(\mathcal{F}) & = 0 \\
    \mathcal{H}_{\tau\to o}(\mathcal{I}) & = 1
\end{align*}
\]

and for relational types \( \rho \) and \( \mathfrak{R} \subseteq \mathcal{H}[[\rho]] \), \( \bigsqcup_{\rho} \mathfrak{R} \) is defined inductively by

\[
\begin{align*}
    \bigsqcup_{\rho} \mathfrak{R} & = \max \mathfrak{R} & \text{for} \mathfrak{R} \subseteq \mathcal{H}[[o]] \\
    \left( \bigsqcup_{r\to\rho} \mathfrak{R} \right)(s) & = \bigsqcup_{\rho} \{ r(s) | r \in \mathfrak{R} \} & \text{for} \mathfrak{R} \subseteq \mathcal{H}[[\tau \to \rho]] \text{ and } s \in \mathcal{H}[[\tau]].
\end{align*}
\]

Furthermore, for a singleton set \( \{ f \} \subseteq \mathcal{H}[[\iota^n \to i]] \) we define \( \bigsqcup_{\iota^n \to i} \{ f \} = f \). In the following we omit the subscript from \( \bigsqcup_{\sigma} \) since it can be inferred. Note that in particular for \( \rho = \tau \to o \) and \( \tau \in \mathcal{H}[[\tau]] \), \( \bigsqcup_{\rho} \mathfrak{R}(\mathcal{S}) = 1 \) if and only if there exists \( r \in \mathfrak{R} \) such that \( r(\mathcal{S}) = 1 \). Moreover, for types \( \sigma \) we inductively define a relation \( \sqsubseteq^\mathcal{H}_\sigma \) by

\[
\begin{align*}
    f & \sqsubseteq^\mathcal{H}_\iota^n \iota^n \iff f = f' & \text{for} f, f' \in \mathcal{H}[[\iota^n \to i]] \\
    b & \sqsubseteq^\mathcal{H}_\iota^n b' \iff b \leq b' & \text{for} b, b' \in \mathcal{H}[[o]] \\
    r & \sqsubseteq^\mathcal{H}_\tau \rho r' \iff \forall s \in \mathcal{H}[[\tau]], r(s) \sqsubseteq^\mathcal{H}_\rho r'(s) & \text{for} r, r' \in \mathcal{H}[[\tau \to \rho]]
\end{align*}
\]

Clearly, \( \sqsubseteq^\mathcal{H}_\sigma \) is an ordering for each \( \sigma \). Intuitively, this is the discrete ordering on \( \mathcal{H}[[\iota^n \to i]] \) and a pointwise ordering on relational types. In the following, we omit the subscript since it can be inferred, we omit the superscript whenever the pre-Henkin frame is clear from the context and just write \( r \sqsubseteq r' \).

In fact, for relational \( \rho \), the poset \( (\mathcal{H}[[\rho]], \sqsubseteq_\rho) \) is a semilattice in which all suprema exist. The proof of the following can be found in [Abramsky and Jung, 1994, Proposition 2.1.4]

**Lemma 1.4.** Let \( I \) be a set and \( \mathfrak{R}_i \subseteq \mathcal{H}[[\rho]] \) for each \( i \in I \). Then

\[
\bigsqcup \left\{ \bigsqcup \mathfrak{R}_i \mid i \in I \right\} = \bigsqcup \{ r_i \mid i \in I \land r_i \in \mathfrak{R}_i \}.
\]

Next, we consider some examples of pre-Henkin frames [Cathcart Burn et al., 2018, Abramsky and Jung, 1994]:

**Example 1.5.** Let \( D \) be an arbitrary set. We define \( S_D, \mathcal{M}_D \) and \( \mathcal{C}_D \) inductively by

\[
\begin{align*}
    S_D[\iota^n \to \iota^n] & = [D^n \to D] & S_D[o] & = \mathbb{B} & S_D[\tau \to \rho] & = [S_D[\tau] \to S_D[\rho]] \\
    \mathcal{M}_D[\iota^n \to \iota^n] & = [D^n \to D] & \mathcal{M}_D[o] & = \mathbb{B} & \mathcal{M}_D[\tau \to \rho] & = [\mathcal{M}_D[\tau] \overset{\tau}{\to} \mathcal{M}_D[\rho]] \\
    \mathcal{C}_D[\iota^n \to \iota^n] & = [D^n \to D] & \mathcal{C}_D[o] & = \mathbb{B} & \mathcal{C}_D[\tau \to \rho] & = [\mathcal{C}_D[\tau] \cong \mathcal{C}_D[\rho]].
\end{align*}
\]

where \( [\mathcal{M}_D[\tau] \overset{\tau}{\to} \mathcal{M}_D[\rho]] \) (\( [\mathcal{C}_D[\tau] \cong \mathcal{C}_D[\rho]] \)) is the set of monotone (continuous) functions \( \mathcal{M}_D[\tau] \to \mathcal{M}_D[\rho] \) (\( \mathcal{C}_D[\tau] \to \mathcal{C}_D[\rho] \)) with respect to the respective pointwise
For the purpose of the remainder of this subsection, we fix a witness $h_\sigma$ of the non-
emptiness of $H[\sigma]$ for each $\sigma$.

Let $\Sigma$ be a signature. A $(\Sigma, H)$-structure $A$ assigns to each $c: \sigma \in \Sigma$ an element $c^A \in H[\sigma]$ and a $(\Delta, H)$-valuation $\alpha$ is a function such that for every $x: \tau \in \Delta$, $\alpha(x) \in H[\tau]$. For a $(\Delta, H)$-valuation $\alpha, x_1, \ldots, x_n$ and $r_1 \in H[\Delta(x_1)], \ldots, r_n \in H[\Delta(x_n)]$, $\alpha[x_1 \mapsto r_1, \ldots, x_n \mapsto r_n]$ is the $(\Delta, H)$-valuation defined by

$$
\alpha[x_1 \mapsto r_1, \ldots, x_n \mapsto r_n](x) = \begin{cases} 
  r_i & \text{if } x = x_i \text{ for some } i \\
  \alpha(x) & \text{otherwise}
\end{cases}
$$

Furthermore, for a type environment $\Delta$ let $\top^H_\Delta$ be the valuation defined by

$$
\top^H_\Delta(x) = \begin{cases} 
  \top^H \rho & \text{if } \Delta(x) = \rho \\
  h_\iota & \text{otherwise (i.e. } \Delta(x) = \iota)
\end{cases}
$$

We lift $\sqsubseteq$ in the obvious pointwise manner to $(\Sigma, H)$-structures $A$ and $A'$ and $(\Delta, H)$ valuations $\alpha$ and $\alpha'$ by defining

$$
\alpha \sqsubseteq \alpha' \iff \forall x \in \text{dom}(\Delta). \alpha(x) \sqsubseteq \alpha'(x)
$$

$$
B \sqsubseteq B' \iff \forall c \in \Sigma. c^B \sqsubseteq c^{B'}.
$$

The denotation $A^H[\sigma](\alpha)$ of a term $M$ with respect to $A$, $\alpha$ and $H$ is defined inductively by

$$
A^H[\sigma](\alpha) = \alpha(x)
$$

$$
A^H[c](\alpha) = c^A
$$

$$
A^H[\neg M](\alpha) = 1 - A^H[M](\alpha)
$$

$$
A^H[\land](\alpha) = \text{and}
$$

$$
A^H[\lor](\alpha) = \text{or}
$$

$$
A^H[\exists x](\alpha) = \text{exists}_{H}^x
$$

$$
A^H[M_1 \cdot M_2](\alpha) = A^H[M_1](\alpha)(A^H[M_2](\alpha))
$$

$$
A^H[\lambda x. M](\alpha) = \begin{cases} 
  A^H(\lambda x. M)(\alpha) & \text{if } A^H(\lambda x. M)(\alpha) \in H[\Delta(x) \rightarrow \rho] \\
  h_\rho & \text{otherwise}
\end{cases}
$$

if $\Delta \vdash M : \rho$

where

$$
A^H(\lambda x. M)(\alpha) = \lambda r \in H[\Delta(x)]. A^H[M](\alpha[x \mapsto r]).
$$

Note that for each term $\Delta \vdash M : \sigma$, $A^H[\sigma](\alpha) \in H[\sigma]$. For a $\Sigma$-formula $M$, we write $A, \alpha \models_H M$ if $A^H[\sigma](\alpha) = 1$. 

A pre-Henkin frame $\mathcal{H}$ is a Henkin frame if for each signature $\Sigma$, type environment $\Delta$, $(\Sigma, \mathcal{H})$-structure $A$, $(\Delta, \mathcal{H})$-valuation $\alpha$ and positive existential $\Sigma$-term $\lambda x. M$, $A^\mathcal{H}(\lambda x. M)(\alpha) = A^\mathcal{H}[\lambda x. M](\alpha)$.

Note that in contrast to most treatments of the literature we restrict this (implicit) form of comprehension axioms to positive existential formulas.

**Example 1.6.** $S_D$ is trivially a Henkin frame. In Section 4.1 we will show that also $M_D$ and $C_D$ are Henkin frames. We call $S_D$, $M_D$ and $C_D$ standard, monotone and continuous (Henkin) frames, respectively.

For a Henkin frame $\mathcal{H}$, a $(\Sigma, \mathcal{H})$-structure $A$ and $\Sigma$-formula $F$ we write $A \models_\mathcal{H} F$ if for all $(\Delta, \mathcal{H})$-valuations $\alpha$, $A, \alpha \models F$. If $S$ is a set of $\Sigma$-formulas we write $A \models_\mathcal{H} S$ if for each $F \in S$, $A \models_\mathcal{H} F$. For sets $S$ and $S'$ of $\Sigma$-formulas we use the notation $S \models_\mathcal{H} S'$ if for any $\Sigma$-structure $A$, $A \models_\mathcal{H} S$ implies $A \models_\mathcal{H} S'$, and $S \models S'$ if $S \models_\mathcal{H} S'$ for any Henkin frame $\mathcal{H}$. Furthermore, we occasionally identify formulas with the corresponding singleton set.

Since terms of type $\iota^n \to \iota$ contain neither relational variables, $\lambda$-abstractions nor logical symbols (Remark 1.2), the following is obvious:

**Lemma 1.7.** Let $\mathcal{H}$ be a Henkin frame, $\Delta \vdash M : \iota^n \to \iota$ be a $\Sigma$-term, $B$ be a $(\Sigma, \mathcal{H})$-structure and $\alpha, \alpha'$ be $(\Delta, \mathcal{H})$-valuations such that $\alpha'(x) = \alpha(x)$ for all $x : \iota \in \text{free}(M)$.

Then $B^\mathcal{H}[M](\alpha) = B^\mathcal{H}[M](\alpha')$.

**First-Order Structures** Let $\Sigma$ be a first-order signature. A first-order $\Sigma$-structure with domain $D$ is a $(\Sigma, S_D)$-structure $A$ and we write $\text{dom}(A)$ for $D$. Note that by taking standard frames this coincides with the usual definition of structures in a purely first-order setting (cf. e.g. [Chang and Keisler, 2013]).

**Example 1.8.** (i) In the examples we will be primarily concerned with the signature of linear integer arithmetic\(^1\) $\Sigma_{\text{LIA}} = \{0, 1, +, <, \leq, =, \geq, >\}$ and its standard model $A_{\Sigma_{\text{LIA}}}$.

(ii) Let $\Sigma$ be a signature which only contains $\approx : \iota \to \iota \to \omega$, some $c : \iota$ and symbols of type $\iota \to \cdots \to \iota$. The $\Sigma$-Herbrand structure $A_{\Sigma, \text{Her}}$ is the first-order $\Sigma$-structure with domain\(^2\) $D = \{M \text{ ground } | \Delta \vdash M : \iota\}$ which satisfies $\approx A_{\Sigma, \text{Her}} = \{(M, M) \mid M \in D\}$ (identifying the set with its characteristic function) and

\[
\begin{align*}
f_{A_{\Sigma, \text{Her}}} : D^n &\to D \\
(M_1, \ldots, M_n) &\mapsto f M_1 \cdots M_n
\end{align*}
\]

for $f : \iota^n \to \iota \in \Sigma \setminus \{\approx\}$. Clearly this is well-defined. If the signature is fixed we usually omit it from $A_{\Sigma, \text{Her}}$ and just write $A_{\text{Her}}$.

\(^1\)with the usual types $0, 1 : \iota$, $+ : \iota \to \iota \to \iota$ and $\vDash : \iota \to \iota \to \omega$ for $\vDash \in \{<, \leq, =, \geq, >\}$

\(^2\)Due to $c : \iota \in \Sigma$ this is non-empty.
1.4. PRELIMINARIES

1.4.1.3 Substitution and Contexts

Let $M$ and $N_1, \ldots, N_n$ be $\Sigma$-terms and let $x_1, \ldots, x_n$ be variables satisfying $\Delta \vdash N_i : \Delta(x_i)$ for $1 \leq i \leq n$. Then the well-typed (simultaneous) substitution $M[N_1/x_1, \ldots, N_n/x_n]$ of $N_1, \ldots, N_n$ for $x_1, \ldots, x_n$ in $M$ is defined by

$$
x_i[N_1/x_1, \ldots, N_n/x_n] = N_i
$$

$$
x[N_1/x_1, \ldots, N_n/x_n] = x \quad \text{for } x \notin \{x_1, \ldots, x_n\}
$$

$$
c[N_1/x_1, \ldots, N_n/x_n] = c \quad \text{for } c \in \Sigma \cup \{\neg, \land, \lor, \exists\}
$$

$$(M_1 M_2)[N_1/x_1, \ldots, N_n/x_n] = M_1[N/x] M_2[N/x]
$$

$$(\lambda y. M)[N_1/x_1, \ldots, N_n/x_n] = \lambda y. M[N_1/x_1, \ldots, N_n/x_n]
$$

and we use the abbreviation $M[N/x]$ for $M[N_1/x_1, \ldots, N_n/x_n]$. Note that by the variable convention (Convention 1.1) it holds that $y \notin \{x_1, \ldots, x_n\}$ and $y \notin \text{free}(N_1) \cup \cdots \cup \text{free}(N_n)$.

A unifier of two terms $M$ and $N$ is a substitution $\theta$ such that $M\theta = N\theta$.

The following lemma is completely standard and can be proven by an easy structural induction (exploiting the variable convention).

**Lemma 1.9** (Substitution). Let $\mathcal{H}$ be a pre-Henkin frame, $A$ be a $(\Sigma, \mathcal{H})$-structure and $\alpha$ be a $(\Delta, \mathcal{H})$-valuation. Furthermore, let $x \in \text{dom}(\Delta)$ and let $M$ and $N$ be terms such that $\Delta \vdash N : \Delta(x)$. Then $A^\mathcal{H}[M[N/x]](\alpha) = A^\mathcal{H}[M]\alpha[x \mapsto A^\mathcal{H}[N](\alpha)]$.

Next, the set of $\Sigma$-pre-contexts is given by

$$
\widetilde{\Sigma} : [-] : \sigma
$$

where $c \in \Sigma$. We define a typing judgement $\vdash_{\sigma}$ in a similar way as for terms and additionally we set

$$
\Delta \vdash_{\sigma} [-] : \sigma
$$

A $\Sigma$-context with $\sigma$-hole is a $\Sigma$-pre-context $M[-]$ such that $\Delta \vdash_{\sigma} M : \sigma$ for some $\sigma$. For all specialisations of the notion of terms (such as formula), we introduce an analogous notion for contexts.

If $M[-]$ is a $\Sigma$-context with a $\sigma$ hole and $N$ is a $\Sigma$-term such that $\Delta \vdash M[-] : \sigma$ for some $\sigma$ and $\Delta \vdash N : \sigma$ then $M[N]$ is the $\Sigma$-pre-term obtained by replacing $[-]$ in $M$ by $N$ in such a way that free variables of $N$ may become bound\(^3\).

The following illustrates the usefulness of contexts:

**Lemma 1.10.** Let $M$ and $N$ be a $\Sigma$-terms and suppose that $\Delta \vdash N : \sigma$. Then there exists a $\Sigma$-context $\widetilde{M}[-]$ with a $\sigma$-hole such that $M[N] = \widetilde{M}$ and $N$ does not occur in $\widetilde{M}[-]$.

**Proof.** Straightforward induction on (the typing judgement of) $M$. (If $M = N$ then set $\widetilde{M} = [-].$) \qed

In a similar spirit to the Substitution Lemma 1.9, we get the following for contexts (and different signatures).

\(^3\)i.e. this is not done up to $\alpha$-conversion
Lemma 1.11 (Context). Let $\Sigma \subseteq \Sigma'$ be signatures, $M[-]$ be a $\Sigma$-context with a hole of type $\tau$, let $B$ be a $(\Sigma, H)$-structure and $B'$ be a $(\Sigma', H)$-structure, let $N$ be a $\Sigma$-term and $N'$ be a $\Sigma'$ term such that

(i) $\Delta \vdash N : \tau$ and $\Delta \vdash N' : \tau$,
(ii) for each $c \in \Sigma$, $c^B = c^{B'}$ and
(iii) for each $(\Delta, H)$-valuation $\alpha$, $B^H[N](\alpha) = B'^H[N'](\alpha)$.

Then for each $(\Delta, H)$-valuation $\alpha$, $B^H[M[N]](\alpha) = B'^H[M[N']](\alpha)$.

1.4.2 Reduction

Let $R$ be a (binary) relation on terms. The compatible closure $\rightarrow_R$ of $R$ is defined inductively by

$M \rightarrow_R N$ if $M, N \in R$

$M_1 \rightarrow_R N_1$ and $M_2 \rightarrow_R N_2$ then $M_1M_2 \rightarrow_R M_1N_2$

$M \rightarrow_R N$

and let $\rightarrow_R$ be its reflexive, transitive closure.

Let $\beta = \{(\lambda x.M, M[N/x])\}$ and $\eta = \{(\lambda x. M x, M) \mid \Delta \vdash M : \Delta(x) \rightarrow \rho\}$. A term $N$ is a $\beta$-normal form of a term $M$ if $M \rightarrow_\beta N$ and there is no term $L$ such that $N \rightarrow_\beta L$. Since terms are well-typed, they have a unique $\beta$-normal form (see e.g. [Girard, 1989, Barendregt, 2012]).

The following lemma states that in Henkin frames, the denotation is stable under $\beta$- and $\eta$-conversion.

Lemma 1.12. Let $H$ be a Henkin frame, let $M$ and $M'$ be $\Sigma$-terms, $A$ be a $(\Sigma, H)$-structure and let $\alpha$ be a $(\Delta, H)$-valuation. Then

(i) if $M \rightarrow_\beta M'$ then $A^H[M](\alpha) = A^H[M'](\alpha)$;
(ii) if $M \rightarrow_\eta M'$ then $A^H[M](\alpha) = A^H[M'](\alpha)$.

Proof. We prove the lemma by induction on the compatible closure. The only interesting cases are the base cases $((\lambda x. N)N', N[N'/x]) \in \beta$ and $(\lambda y. Ly, L) \in \eta$, respectively. Then

$A^H[(\lambda x. N)N'](\alpha) = A^H[N](\alpha[x \mapsto A^H[N'](\alpha)]) \quad H$ is a Henkin frame

$= A^H[N[N'/x]](\alpha). \quad \text{Lemma 1.9}$

and

$A^H[\lambda y. Ly](\alpha) = \lambda r \in H[\Delta(y)]. A^H[L](\alpha)(r) \quad H$ is a Henkin frame

$= A^H[L](\alpha). \quad \Box$

For non-Henkin frames this is clearly not true, in general.
1.4.3 Theorem Proving for Hierarchic First-Order Theories

Since the 90s, there have been attempts to extend reasoning for full first-order logic (with equality) with background theories, which is known as theorem proving for hierarchic first-order theories. Nowadays, the focus has shifted to (quantifier-free) fragments of first-order logic with background theories and to relax refutational completeness standards, which is referred to as SMT solving. Next, we briefly review some basics of theorem proving for hierarchic first-order theories.

1.4.3.1 Many-Sorted First-Order Logic

A many-sorted (first-order) signature $\Sigma$ consists of sets $\Sigma_s$, $\Sigma_c$, $\Sigma_f$ and $\Sigma_p$, where (i) $\Sigma_s$ is a non-empty set of sorts, (ii) $\Sigma_c$ is a set of sorted constant symbols $c : \iota$, where $\iota \in \Sigma_s$, (iii) $\Sigma_f$ is a set of sorted function symbols $f : \iota_1 \rightarrow \cdots \rightarrow \iota_n \rightarrow \iota$, where $n \in \mathbb{N}_{\geq 1}$ and $\iota_1, \ldots, \iota_n \in \Sigma_s$ and (iv) $\Sigma_p$ is a non-empty set of sorted predicate symbols $P : \iota_1 \rightarrow \cdots \rightarrow \iota_n \rightarrow \alpha$, where $n \in \mathbb{N}_{\geq 1}$ and $\iota_1, \ldots, \iota_n \in \Sigma_s$. A sort environment $\Delta$ is a set of distinct sorted variables $x : \iota$, where $\iota \in \Sigma_s$. $\Sigma$-terms are defined in a similar way as for higher-order logic, but we disallow $\lambda$-abstractions and require that $\Sigma$-terms are well-sorted instead of well-typed. For details refer e.g. to [Kruglov, 2013].

A $\Sigma$-structure $\mathcal{A}$ assigns (i) to each sort $\iota \in \Sigma_s$ a set $\mathcal{A}[\iota]$, (ii) to each constant symbol $c : \iota$ an element $c^A \in \mathcal{A}[\iota]$, (iii) to each function symbol $f : \iota_1 \rightarrow \cdots \rightarrow \iota_n \rightarrow \iota$ an element $f^A \in [\mathcal{A}[\iota_1]] \rightarrow \cdots \rightarrow [\mathcal{A}[\iota_n]] \rightarrow \mathcal{A}[\iota]$ and (iv) to each predicate symbol $P : \iota_1 \rightarrow \cdots \rightarrow \iota_n \rightarrow \alpha$ an element $P^A \in [\mathcal{A}[\iota_1]] \rightarrow \cdots \rightarrow [\mathcal{A}[\iota_n]] \rightarrow \mathcal{B}$. A $\Delta$-valuation assigns to each $x : \iota \in \Delta$ an element of $\mathcal{A}[\iota]$. The denotation $\mathcal{A}[M] (\alpha)$ of a term $M$ with respect to $\mathcal{A}$ and $\alpha$ is defined in a very similar way as to higher-order logic [Kruglov, 2013]. A $\Sigma$-structure $\mathcal{A}$ is term-generated if for every $a \in \text{dom}(\mathcal{A})$ there exists a term $M$ without free variables such that $\mathcal{A}[M] = a$.

1.4.3.2 Hierarchic Specifications

A hierarchic specification consists of a many-sorted signature $\Sigma$, a set $\mathcal{B}$ of $\Sigma$-structures and an extension $\Sigma' \supseteq \Sigma$. A set of $\Sigma'$-clauses $S$ is satisfiable with respect to the hierarchic specification if there exists $\mathcal{A} \in \mathcal{B}$ and a $\Sigma'$-structure $\mathcal{A}'$ such that $\mathcal{A}' \models S$ and $\mathcal{A}$ and $\mathcal{A}'$ agree on the symbols and sorts of $\Sigma$. A specification is term-generated if each $\mathcal{A} \in \mathcal{B}$ is term-generated and it is compact if for each set $S$ of $\Sigma$-formulas there exists a $\mathcal{A} \in \mathcal{B}$ such that $\mathcal{A} \models S$ if and only if for each finite $S' \subseteq S$ there exists $\mathcal{A} \in \mathcal{B}$ such that $\mathcal{A} \models S'$.

1.4.3.3 Hierarchic Superposition

Superposition was originally developed by Leo Bachmair and Harald Ganzinger as a sound and refutationsally complete proof system for first-order logic with equality [Bachmair and Ganzinger, 1990]. It cuts down the number of necessary inferences significantly by imposing restrictions based on term orderings and selection functions. Therefore, it is the basis of most modern automated theorem provers (e.g. SPASS [Weidenbach et al., 2009] or Vampire [Kovács and Voronkov, 2013]), which are successfully used in practice.

\footnote{i.e. $\Sigma_s \supseteq \Sigma_s'$, $\Sigma_c \supseteq \Sigma_c'$, $\Sigma_f \supseteq \Sigma_f'$ and $\Sigma_p \supseteq \Sigma_p'$}
Some years later, superposition has been extended to hierarchic specifications [Bachmair et al., 1994]. It turns out that hierarchic superposition is not automatically complete. Rather, it is necessary to require that the hierarchic specification is term-generated, compact and that it satisfies the notion of sufficient completeness with respect to simple instances. However, there is a simple criterion for the latter [Bachmair et al., 1994]: If there is no function \( f : \iota_1 \to \cdots \to \iota_n \to \iota \in \Sigma' \setminus \Sigma_f \), where \( \iota \in \Sigma_s \) then the hierarchic specification is sufficiently complete with respect to simple instances for all sets of clauses. Furthermore, if \( \mathcal{B} \) is a singleton then the hierarchic specification is obviously compact and can be made term-generated by introducing a constant for each element of the domains. Consequently, we get

**Theorem 1.13** ([Bachmair et al., 1994]). Let \((\Sigma, \{A\}, \Sigma')\) be a hierarchic specification such that there is no function \( f : \iota_1 \to \cdots \to \iota_n \to \iota \in \Sigma' \setminus \Sigma_f \), where \( \iota \in \Sigma_s \), and let \( S \) be a set of \( \Sigma' \)-clauses.

Then, \( S \) is satisfiable with respect to \((\Sigma, \{A\}, \Sigma')\) if and only if hierarchic superposition can refute \( S \).

There are few implementations of hierarchic superposition. Notably, the theorem prover SPASS has been extended for hierarchic specifications with respect to linear rational arithmetic [Althaus et al., 2009, Kruglov, 2013]. Moreover, there are other systems like Vampire that are, however, incomplete in general [Kovács and Voronkov, 2013].
Chapter 2

(Higher-Order) Constrained Horn Problems and Clauses

In this chapter, we introduce (higher-order) constrained Horn clauses and (higher-order) constrained Horn problems. Besides, we define the corresponding satisfiability and solvability problems. Whilst the former have a simple syntax and thus give rise to a simple proof system, our completeness proof makes use of the latter. Furthermore, we introduce a normal form for higher-order constrained Horn problems, which is closer to higher-order constrained Horn clauses than general higher-order constrained Horn problems.

Moreover, we prove undecidability of the satisfiability problem for (higher-order) constrained Horn clauses in Section 2.4.

In this work, we fix a first-order signature $\Sigma$ and a $\Sigma$-structure $\mathcal{A}$ (e.g. linear integer arithmetic $\Sigma_{LIA}$ and its standard model $\mathcal{A}_{LIA}$). Note that this always allows us to include equality of the base type. Furthermore, we assume an extension $\Sigma' \supseteq \Sigma$ only extending $\Sigma$ with symbols of relational type, and a type environment $\Delta$.

For reasons of convenience, we defined first-order structures with respect to standard frames. However, in general there are many other Henkin frames $\mathcal{H}$ that have rich enough function spaces and thus allow us to regard $\mathcal{A}$ also as a $(\Sigma, \mathcal{H})$-structure. Furthermore, we will be interested in whether first-order structures can be expanded to larger (higher-order) signatures. This is made precise by the following:

**Definition 2.1.** (i) A Henkin frame $\mathcal{H}$ expands $\mathcal{A}$ if $\mathcal{H}[i] = \text{dom}(\mathcal{A})$ and $c^A \in \mathcal{H}[\sigma]$ for all $c : \sigma \in \Sigma$.

(ii) Suppose $\mathcal{H}$ expands $\mathcal{A}$. Then a $(\Sigma', \mathcal{H})$-structure $\mathcal{B}$ is a $(\Sigma', \mathcal{H})$-expansion of $\mathcal{A}$ if $c^A = c^B$ for all $c \in \Sigma$.

If no confusion arises, we omit explicit mentions of $\Sigma'$ and $\Delta$.

2.1 Syntax

We start with the following definition:

**Definition 2.2.** (i) If $F$ and $R \pi$, where $R \in \Sigma' \setminus \Sigma$ are positive existential formulas such that $\text{free}(F) \subseteq \text{free}(R \pi)$ and the variables of $\pi$ are distinct then $\neg F \lor R \pi$ is a definite formula.
(ii) A program is a set of definite formulas such that for each symbol \(R \in \Sigma' \setminus \Sigma\) there exists a unique definite formula \(\neg F_R \lor R \bar{x}_R\) in \(P\).

(iii) A (higher-order) constrained Horn problem in program form (HoCHP) is a pair \((P, F)\), where \(P\) is a program and \(F\) is a closed positive existential formula.

Example 2.3. Let \(\Sigma' = \Sigma \cup \{\text{Add} : \iota \rightarrow \iota \rightarrow \iota \rightarrow o, \text{Iter} : (\iota \rightarrow \iota \rightarrow \iota \rightarrow o) \rightarrow \iota \rightarrow \iota \rightarrow \iota \rightarrow o\}\) and let \(\Delta\) be a type environment satisfying 
\[
\begin{align*}
\Delta(x) = \Delta(y) = \Delta(z) = \Delta(n) = \Delta(s) &= \iota, \\
\Delta(f) &= \iota \rightarrow \iota \rightarrow \iota \rightarrow o.
\end{align*}
\]
Then \((P, F)\) defined by
\[
\begin{align*}
P &= \{\neg(z = x + y) \lor \text{Add } x y z, \\
&\quad \neg((n \leq 0 \land s = x) \lor (n > 0 \land \exists y. \text{Iter } f \ s \ (n - 1) y \land f \ n \ y \ x)) \lor \text{Iter } f \ s \ n \ x\}
\end{align*}
\]
\[
F = n \geq 1 \land \text{Iter Add } n \ n \ x \land x \leq n + n,
\]
is the program corresponding to the example from Section 1.1, where, for the sake of readability, we have renamed variables.

In the tradition of first-order logic and in order not to lose focus on the essentials, we develop our proof systems not for HoCHP but for problems in a closely related “clausal” form. In particular, we do not want to deal with terms of the form \(R \bar{M}, x \bar{M}\) or \((\lambda y. N)\bar{M}\), where \(\bar{M}\) or \(N\) contain logical symbols.

Definition 2.4. (i) An atom is a formula which does not contain a logical symbol.
(ii) An atom is a background atom if it is also a \(\Sigma\)-formula only containing variables of type \(\iota\). Otherwise it is a foreground atom.
(iii) A foreground atom of the form \(x \bar{M}\) is a simple foreground atom.

Note that a foreground atom necessarily has the form \(R \bar{M}\), where \(R \in \Sigma' \setminus \Sigma\), \(x \bar{M}\) or \((\lambda y. N)\bar{M}\). Furthermore, \(x\) in \(x \bar{M}\) is clearly not a first-order variable.

We use \(\phi\) and \(A\) (and variants thereof) to refer to background atoms and general atoms, respectively.

Definition 2.5 (Horn Clause). (i) A goal clause is a disjunction \(\neg A_1 \lor \cdots \lor \neg A_n\), where each \(A_i\) is an atom.
(ii) \(G\) is a simple goal clause if each \(A_i\) is a simple foreground atom.
(iii) If \(G\) is a goal clause, \(R \in \Sigma' \setminus \Sigma\) and the variables in \(\bar{x}\) are distinct then \(G \lor R \bar{x}\) is a definite clause.
(iv) A (higher-order) constrained Horn clause (HoCHC) is a goal or definite clause.

Furthermore, we use \(\bot\) to refer to the empty clause.

Example 2.6. The following set of HoCHCs (cf. Section 1.1) corresponds to the program of Example 2.3:

\[
\begin{align*}
\neg(z = x + y) & \lor \text{Add } x y z \\
\neg(n \leq 0) & \lor \neg(s = x) \lor \text{Iter } f \ s \ n \ x \\
\neg(n > 0) & \lor \neg \text{Iter } f \ s \ (n - 1) y \lor \neg(f \ n \ y \ x) \lor \text{Iter } f \ s \ n \ x \\
\neg(n \geq 1) & \lor \neg \text{Iter Add } n \ n \ x \lor (x \leq n + n).
\end{align*}
\]
2.2 Semantics

Definition 2.7. Let $S$ be a set of HoCHCs and let $\mathcal{H}$ be a Henkin frame expanding $\mathcal{A}$.

(i) A set $S$ of HoCHCs is $(\mathcal{A}, \mathcal{H})$-satisfiable if there exists a $(\Sigma', \mathcal{H})$-expansion $B$ of $\mathcal{A}$ satisfying $B \models_{\mathcal{H}} S$.
(ii) If $S$ is not $(\mathcal{A}, \mathcal{H})$-satisfiable then it is $(\mathcal{A}, \mathcal{H})$-unsatisfiable.

Furthermore, $S$ is $\mathcal{A}$-Henkin-satisfiable if it is $(\mathcal{A}, \mathcal{H})$-satisfiable for some $\mathcal{H}$ expanding $\mathcal{A}$ and it is $\mathcal{A}$-standard-satisfiable ($\mathcal{A}$-monotone-, $\mathcal{A}$-continuous-satisfiable) if it is $(\mathcal{A}, S)$-satisfiable ($((\mathcal{A}, \mathcal{M})$, $(\mathcal{A}, \mathcal{C})$-satisfiable).

Definition 2.8. Let $(P, F)$ and $(P', F')$ be HoCHPs and let $\mathcal{H}$ be a Henkin frame expanding $\mathcal{A}$. Then

(i) $(P, F)$ is $(\mathcal{A}, \mathcal{H})$-solvable if there exists a $(\Sigma', \mathcal{H})$-expansion $B$ of $\mathcal{A}$ satisfying $B \models_{\mathcal{H}} P$ and $B \not\models F$.
(ii) $(P, F)$ and $(P', F')$ are $(\mathcal{A}, \mathcal{H})$-equivalent if $(P, F)$ is $(\mathcal{A}, \mathcal{H})$-solvable if and only if $(P', F')$ is $(\mathcal{A}, \mathcal{H})$-solvable.

2.3 Normal Form and Translations

In this section, we begin establishing a connection between HoCHCs and HoCHPs.

First, let $S$ be a finite set of HoCHCs. We can assume that for each $R \in \Sigma' \setminus \Sigma$ there is at least one Horn clause $G \lor R \tau_R$ and each definite clause has this form. This is without loss of generality because otherwise we can take $\neg \phi \lor R \tau_R$ for some background atom $\phi$ such that $\mathcal{A}, \alpha \not\models \phi$ for arbitrary $\alpha$ (e.g. $y \neq y$), or rename variables.

For a goal clause $G = \neg A_1 \lor \cdots \lor \neg A_n$ let $\text{posex}(G, V) = \exists y_1, \ldots, y_m. A_1 \land \cdots \land A_n$, where $\{y_1, \ldots, y_m\} = \text{free}(G) \setminus V$ and $\text{posex}(G) = \text{posex}(G, \emptyset)$. Clearly, $\text{posex}(G)$ is a positive existential closed formula. Let $P(S)$ be the set of definite formulas

$$\neg (\text{posex}(G_{R,1}, \tau_R) \lor \cdots \lor \text{posex}(G_{R,n}, \tau_R)) \lor R \tau_R,$$

where $G_{R,1} \lor R \tau_R, \ldots, G_{R,n} \lor R \tau_R$ are the positive occurrences of $R \tau_R$ in $S$ and $R \in \Sigma' \setminus \Sigma$, and let

$$F(S) = \text{posex}(G_1) \lor \cdots \lor \text{posex}(G_m),$$

where $G_1, \ldots, G_m$ are the goal clauses in $F$.

Clearly, $(P(S), F(S))$ is a HoCHP. Furthermore, the following is obvious by definition:

Lemma 2.9. Let $\mathcal{H}$ be a Henkin frame and $B$ be a $(\Sigma', H)$-expansion of $\mathcal{A}$. Then

(i) $B \models_{\mathcal{H}} \{D \in S \mid D \text{ definite} \}$ if and only if $B \models_{\mathcal{H}} P(S)$ and
(ii) $B \models_{\mathcal{H}} \{G \in S \mid G \text{ goal} \}$ if and only if $B \not\models_{\mathcal{H}} F(S)$.

Therefore, we get:
Corollary 2.10. \( S \) is \( (\mathcal{A}, \mathcal{H}) \)-satisfiable if and only if \( (P(S), F(S)) \) is \( (\mathcal{A}, \mathcal{H}) \)-solvable.

This translation produces a very restricted form of HoCHP, which is captured by the following definition:

Definition 2.11. (i) A positive existential formula \( F \) is semi-normal if for all subterms \( \exists M \) of \( F \), \( M \) is a \( \lambda \)-abstraction.
(ii) A positive existential formula \( F \) is in normal form if it has the form \( F_1 \lor \cdots \lor F_n \), where each \( F_i \) has the form \( \exists y_1, \ldots, y_m.A_1 \land \cdots \land A_n \).
(iii) \( (P, F) \) is in normal form if for each \( \neg F_R \lor R \pi_R \) in \( P \), \( F_R \) and \( F \) are in normal form.

Obviously, every positive existential formula in normal form is also semi-normal. Note that the HoCHP from Example 2.3 is in fact in normal form.

The converse translation from HoCHPs to HoCHCs appears to be non-trivial. In fact, the correctness of our method given in Section 4.2.1 relies on a number of technical results developed in this work.

If we restrict attention to HoCHPs in normal form, the task becomes considerably easier. Let \( (P, F) \) be a HoCHP in normal form. We associate with \( (P, F) \) the set \( S(P, F) \) which is the union of

\[ \neg A_{1,1} \lor \cdots \lor \neg A_{1,n_1} \lor R \pi_R, \ldots, \neg A_{m,1} \lor \cdots \lor \neg A_{m,n_m} \lor R \pi_R \],

whenever \( \neg ((\exists \overline{y}_1, A_{1,1} \land \cdots \land A_{1,n_1}) \lor \cdots \lor (\exists \overline{y}_m, A_{m,1} \land \cdots \land A_{m,n_m})) \lor R \pi_R \) is in \( P \), and

\[ \neg A_{1,1} \lor \cdots \lor \neg A_{1,n_1}, \ldots, \neg A_{m,1} \lor \cdots \lor \neg A_{m,n_m} \],

where \( F = (\exists \overline{y}_1, A_{1,1} \land \cdots \land A_{1,n_1}) \lor \cdots \lor (\exists \overline{y}_m, A_{m,1} \land \cdots \land A_{m,n_m}) \). Clearly, \( S(P, F) \) is a set of HoCHC and we obtain the following

Lemma 2.12. Let \( \mathcal{H} \) be a Henkin frame and \( B \) be a \( (\Sigma', \mathcal{H}) \)-expansion of \( \mathcal{A} \). Then

(i) \( B \models_\mathcal{H} \{ D \in S(P, F) \mid D \ \text{definite} \} \) if and only if \( B \models_\mathcal{H} P \) and
(ii) \( B \models_\mathcal{H} \{ G \in S(P, F) \mid G \ \text{goal} \} \) if and only if \( B \not\models_\mathcal{H} F \).

Corollary 2.13. \( S(P, F) \) is \( (\mathcal{A}, \mathcal{H}) \)-satisfiable if and only if \( (P, F) \) is \( (\mathcal{A}, \mathcal{H}) \)-solvable.

2.4 Undecidability

In 1972, Peter Downey proved in a technical report [Downey, 1972] that the logic of Presburger arithmetic enriched with just a single monadic predicate is undecidable. His proof uses a reduction from two-counter machines, which are a very simple, yet Turing-complete model of computation introduced by Marvin Minsky [Minsky, 1967]. In this section, we sketch how to use Downey’s proof to show undecidability of the \( (\mathcal{A}_{LIA}, S) \)-satisfiability problem. There is a need for minimal changes because he uses a quantifier-alternation to express divisibility, which however can be avoided easily.

First, we give a formal definition of two-counter machines, which is a hybrid of the presentations in [Horbach et al., 2017b] and [Downey, 1972]:

---

1i.e. how to obtain a set of HoCHCs \( S \) given \( (P, F) \) which is \( (\mathcal{A}, \mathcal{H}) \)-satisfiable if and only if \( (P, F) \) is \( (\mathcal{A}, \mathcal{H}) \)-solvable
2.4. UNDECIDABILITY

Definition 2.14. A two-counter machine is a quadruple \((Q, q_0, q_f, \delta)\), where

(i) \(Q\) is a finite set of prime numbers greater than 3 (the set of states),
(ii) \(q_0 \in Q\) is the initial state,
(iii) \(q_f \in Q\) is the final state and
(iv) \(\delta : Q \to \{\text{inc}(q), \text{test}\&\text{dec}(q, q') \mid i \in \{1, 2\} \land q, q' \in Q\}\) is the transition function.

A configuration is a triple \((q, c_1, c_2) \in Q \times \mathbb{N} \times \mathbb{N}\) and a run is a finite sequence of configurations \((q_1, c_{1,1}, c_{2,1}), \ldots, (q_n, c_{1,n}, c_{2,n})\) such that \(q_1 = q_0\) (the initial state) and for each \(1 \leq i \leq 2, 1 \leq j < n\) and \(q, q' \in Q\),

(i) if \(\delta(q_j) = \text{inc}(q)\) then \(q_{j+1} = q, c_{i,j+1} = c_{i,j} + 1\) and \(c_{k,j+1} = c_{k,j}\), where \(k = (i + 1) \mod 2\), i.e. the \(i\)-th counter is increased and \(q\) is the next state,
(ii) if \(\delta(q_j) = \text{test}\&\text{dec}(q, q')\) and \(c_{i,j} > 0\), then \(q_{j+1} = q, c_{i,j+1} = c_{i,j} - 1\) and \(c_{k,j+1} = c_{k,j}\), where \(k = (i + 1) \mod 2\), i.e. if the \(i\)-th counter is positive it is decreased and \(q\) is the next state, and
(iii) if \(\delta(q_j) = \text{test}\&\text{dec}(q, q')\) and \(c_{i,j} = 0\), then \(q_{j+1} = q'\) and \(c_{k,j+1} = c_{k,j}\) for all \(k \in \{1, 2\}\), i.e. if the \(i\)-th counter is 0 both counters remain unchanged and \(q'\) is the next state.

\(m \in \mathbb{N}\) is accepted by a two-counter machine if there is a run \((q_1, m, 0), \ldots, (q_n, c_{1,n}, c_{2,n})\) such that \(q_n = q_f\) (the final state).

Note that runs are clearly unique. As we have already mentioned, two-counter machines are Turing-complete and hence the following holds:

Proposition 2.15 ([Minsky, 1967]). The problem whether \(m \in \mathbb{N}\) is accepted by a given two-counter machine is undecidable.

Next, we show how to reduce the acceptance problem for two-counter machines to the \((\mathcal{A}, \mathcal{S})\)-satisfiability problem for an appropriate set of HoCHCs. For that purpose, we fix a 2-counter machine \((Q, q_0, q_f, \delta)\) and a natural number \(m\).

For a natural number \(k\) and a variable \(x\) we use the abbreviations \(^{\text{t}}k\) and \(^{\text{t}}k \cdot x\) for the \(\Sigma_{\text{LIA}}\)-terms \(1 + \cdots + 1\) and \(x + \cdots + x\), respectively.

The proof exploits that we can establish an embedding of the set of configurations \(Q \times \mathbb{N} \times \mathbb{N}\) into the natural numbers by associating \((q, c_1, c_2)\) with \(q \cdot 2^{c_1} \cdot 3^{c_2}\) (because \(q > 3\) is a prime number). For each \(p \in Q\), we define a set \(S(p)\) of HoCHCs dependent on \(\delta(p)\).

(i) If \(\delta(p) = \text{inc}_2(q)\) then \(S(p) = \{-\mathcal{R}(\mathcal{R}p \cdot x) \lor \neg(y = \mathcal{R}3q \cdot x) \lor \mathcal{R}y\}\).
(ii) If \(\delta(p) = \text{test}\&\text{dec}_2(q, q')\) then \(S(p)\) is given by
\[
\begin{align*}
\neg\mathcal{R}(\mathcal{R}3p \cdot x) \lor \neg(w = \mathcal{R}q \cdot x) \lor \mathcal{R}w, & \quad "c_2 > 0", \\
\neg\mathcal{R}x \lor \neg(x = \mathcal{R}p \cdot y) \lor \neg(x = \mathcal{R}3 \cdot z) \lor \neg(w = \mathcal{R}q' \cdot y) \lor \mathcal{R}w, & \quad "c_2 = 0", \\
\neg\mathcal{R}x \lor \neg(x = \mathcal{R}p \cdot y) \lor \neg(x = \mathcal{R}3 \cdot z + 2) \lor \neg(w = \mathcal{R}q' \cdot y) \lor \mathcal{R}w, & \quad "c_2 = 0".
\end{align*}
\]
(iii) If \(\delta(p) = \text{inc}_1(q)\) or \(\delta(p) = \text{test}\&\text{dec}_1(q, q')\) then \(S(p)\) is defined very similarly\(^2\) to the case for \(\delta(p) = \text{inc}_2(q)\) or \(\delta(p) = \text{inc}_2(q, q')\), respectively.

\(^2\)Replace 3 with 2 and the second clause if the counter is zero is not needed.
Finally, we define
\[ S_0 = \{ \neg(x \approx q_0 \cdot 2^m) \lor Rx \}, \]
\[ S_f = \{ \neg R(\neg q_f \cdot x) \}, \]
\[ S' = S_0 \cup \bigcup_{q \in Q} S(q), \]
\[ S = S' \cup S_f. \]

The following lemma is the meat of the correctness proof of the reduction. We only give a proof sketch emphasising the main steps. The detailed proof can be found in [Downey, 1972].

**Lemma 2.16.** Let \((q, c_1, c_2)\) be a configuration. There exists a run \((q_0, m, 0), \ldots, (q, c_1, c_2)\) if and only if \(R^B(q \cdot 2^{c_1} \cdot 3^{c_2}) = 1\) for all \(\Sigma'\)-expansions \(B \models_H S'\) of \(A_{\text{LIA}}\).

**Proof sketch.** The “if”-direction is proven by induction on the length of the run.

For the “only if”-direction, suppose that there is no run \((q_0, m, 0), \ldots, (q, c_1, c_2)\). Let \(B\) be the \((A, H)\)-expansion of \(A_{\text{LIA}}\) defined by
\[ R^B(q \cdot 2^{c_1} \cdot 3^{c_2} | \text{there exists a run } (q_0, m, 0), \ldots, (q', c_1', c_2')}. \]

By definition, \(R^B(q \cdot 2^{c_1} \cdot 3^{c_2}) = 0\) and it is not difficult to verify that also \(B \models_H S'\) holds.

**Corollary 2.17.** \(n\) is accepted if and only if \(S\) is \((A_{\text{LIA}}, H)\)-unsatisfiable.

By Proposition 2.15 we immediately get:

**Theorem 2.18 (Undecidability).**

(i) The \((A, H)\)-satisfiability problem for HoCHC is undecidable in general.

(ii) The \((A_{\text{LIA}}, H)\)-satisfiability problem for HoCHC is undecidable even when \(\Sigma' \setminus \Sigma\) only contains \(R : i \rightarrow o\).

Next, we consider the \((A_{\text{Her}}, H)\)-satisfiability problem (cf. Example 1.8). The classic proof (cf. e.g. [Manna, 2003]) of the undecidability of first-order logic (without equality) is via a reduction from Post’s correspondence problem (PCP) [Post, 1946]. In fact the proof (with very minor tweaks) proves that first-order Horn-logic\(^3\) without equality and with just a single binary predicate \(R\), two unary function \(f_0\) and \(f_1\) and one constant symbol \(e\) is undecidable. Clearly, any such set of first-order Horn-clauses can be transformed into a set of HoCHCs by replacing clauses of the form \(G \lor RMN\) with \(G \lor \neg(x \approx M) \lor \neg(y \approx N) \lor Rx y\) (where \(x\) and \(y\) are fresh variables).

**Proposition 2.19.** For every PCP there exists a set of HoCHCs \(S\) such that

(i) \(S\) can be effectively obtained from the PCP and

(ii) \(S\) is \((A_{\text{Her}}, H)\)-unsatisfiable if and only if the PCP has a solution.

The following example illustrates how the reduction from PCP to HoCHC works:

\(^3\)In the first-order case usually no restrictions are imposed on the positive literal.
Example 2.20. Consider the PCP given by
\[
\begin{pmatrix}
10 \\
11
\end{pmatrix} \begin{pmatrix}
0 \\
00
\end{pmatrix},
\]
which has the (trivial) solution
\[
\begin{pmatrix}
10 \\
11
\end{pmatrix} \begin{pmatrix}
0 \\
00
\end{pmatrix}.
\]

Let \( \Sigma = \{ \approx : \iota \to \iota \to o, e : \iota, f_0, f_1 : \iota \to \iota \} \) and \( \Sigma' = \Sigma \cup \{ R : \iota \to \iota \to o \} \) and let \( S \) be the following set of HoCHCs
\[
\begin{align*}
\neg (x \approx e) \lor \neg (y \approx e) \lor Rx y & \quad \text{empty sequence} \quad C_1 \\
\neg R x' y' \lor \neg (x \approx f_0 (f_1 (f_1 x'))) \lor \neg (y \approx f_1 (f_1 y')) \lor Rx y & \quad \text{first “card”} \quad C_2 \\
\neg R x' y' \lor \neg (x \approx f_0 x') \lor \neg (y \approx f_0 (f_0 y')) \lor Rx y & \quad \text{second “card”} \quad C_3 \\
\neg R (f_0 x) (f_0 x) & \quad \text{non-empty solution} \quad C_4 \\
\neg R (f_1 x) (f_1 x) & \quad \text{non-empty solution} \quad C_5
\end{align*}
\]

To see that \( S \) is \((A_{\text{Her}}, H)\)-unsatisfiable, assume towards contradiction that there exists a \((\Sigma', H)\)-expansion \( B \) of \( A_{\text{Her}} \) satisfying \( B \models_H S \). Then by \( B \models_H C_1, B \models R e e \). Hence, due to \( B \models_H C_2, B \models R (f_0 (f_1 (f_1 e))) (f_1 (f_1 e)) \) and therefore by \( B \models_H C_3, B \models_R (f_0 (f_0 (f_1 (f_1 e)))) (f_0 (f_0 (f_1 (f_1 e)))) \),

which contradicts \( B \models_H C_4 \).

Hence, by the undecidability of PCP ([Post, 1946]), we get:

**Theorem 2.21** (Undecidability (cont.)). The \((A_{\text{Her}}, H)\)-satisfiability problem for HoCHC is undecidable even if \( \Sigma' \setminus \Sigma = \{ R : \iota \to \iota \to o \} \).

### 2.5 Discussion

Our formalisation of the satisfiability problem of HoCHCs and solvability problem of HoCHPs is related to but also quite different from the presentation in [Cathcart Burn et al., 2018] and [Charalambidis et al., 2013].

The latter consider a problem without a background theory and a signature which contains equality but no symbols of a higher-order type \( \sigma \to \iota \). Besides, they are not just interested in structures expanding a fixed standard model of a theory but only insist that equality (which only occurs negated) is interpreted by identity. In Section 4.3, we show that their problem is a special case of the HoCHC-satisfiability problem.

**Syntax** A key difference to [Cathcart Burn et al., 2018] is that they treat the symbols which extend the signature of the background theory and which represent the invariants as free variables. In this respect, our treatment of HoCHCs is much closer to the definition of a hierarchic specification (cf. Section 1.4.3.3 and [Bachmair et al., 1994, Althaus et al., 2009]).
Besides, [Cathcart Burn et al., 2018] and also [Charalambidis et al., 2013] allow an arbitrary nesting of conjunctions and disjunctions in what they call goal formulas, whilst our goal clauses are really clauses in the classical sense (i.e. conjunctions of atoms or the negation thereof), which is standard in the literature on automated theorem proving, in general. Therefore, our HoCHP and our solvability problem are closer to the constrained Horn clauses of [Cathcart Burn et al., 2018] and its solvability problem. HoCHPs also share the feature of their logic programs that for each new symbol/variable there is a unique “definition”, although our formalism remains purely logical. However, unlike [Cathcart Burn et al., 2018] but as in [Charalambidis et al., 2013] we allow nesting logical constants (other than \( \neg \)) completely arbitrarily inside terms (e.g. \((R \exists \iota) \lor (x = x)\)) in HoCHPs.

In principle, [Cathcart Burn et al., 2018] also allow arbitrary quantification of variables which only occur in background atoms. This has not been our focus and hence is not currently supported, although we believe this could be achieved with only minor modifications to our current presentation.

**Semantics** Apart from the differences to [Cathcart Burn et al., 2018] and [Charalambidis et al., 2013] that are due to the differences in the syntax, the solvability of (higher-order) Constrained Horn Clause Problem in [Cathcart Burn et al., 2018] is phrased with respect to standard semantics. In the course of the paper, it is shown to be equivalent to the corresponding problem for monotone semantics. Our HoCHCs satisfiability, on the other side, is parameterised by an arbitrary Henkin frame (e.g. the standard or monotone frame).

Moreover, [Charalambidis et al., 2013] consider a rather complicated semantics using ideas from domain theory [Abramsky and Jung, 1994], which does not fit in neatly into our framework of Henkin frames. However, it has been shown in a not yet published work [Jochems, 2018] that every set of HoCHC is satisfiable with respect to this semantics if and only if it is satisfiable with respect to continuous semantics.

Lastly, the “(higher-order) Constrained Horn Clause Problem” in [Cathcart Burn et al., 2018] is formulated relative to an arbitrary set of models of the background theory, whilst here, we are only interested in theories with one fixed (standard) model\(^4\). In Section 4.3 we give some evidence that our framework is also strong enough to deal with some theories with an infinite number of models. However, this generalised HoCHCs-satisfiability problem cannot be semi-decidable because it is not compact (see Example 5.1).

\(^4\) or finite sets of models
Chapter 3

Resolution and Model Theoretic Properties

Let $S$ be a finite set of HoCHCs and let $\mathcal{H}$ be a Henkin frame, which are fixed for the remainder of this section. (Hence, we omit the super- and subscripts from $B^\mathcal{H}[M](\alpha)$ and $B,\alpha \models \mathcal{H} M$.) Furthermore let $(P, F) = (P(S), F(S))$ be the HoCHP associated with $S$. In particular, $(P, F)$ is in normal form.

In this chapter, we propose a resolution proof system (Section 3.1) and prove that it is able to refute $S$ if and only if it is $(A, \mathcal{H})$-unsatisfiable (Corollary 3.8). Along the way, we prove a couple of model and proof theoretic properties (Sections 3.2 and 3.3, respectively), which are interesting in their own right.

3.1 Resolution Proof System

Our resolution proof system consists of the following three rules:

\[
\text{Resolution} \quad \frac{\neg R \overline{M} \lor G \quad G' \lor R \overline{x}_R}{G \lor G'[\overline{M}/\overline{x}_R]} \\
\text{\beta-Reduction} \quad \frac{\neg(\lambda x. L)M \overline{N} \lor G}{\neg L[\overline{M}/x]\overline{N} \lor G} \\
\text{Constraint Refutation} \quad \frac{G \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n}{\bot}
\]

provided that $G$ is simple, each $\phi_i$ is a background atom and there exists a valuation $\alpha$ such that $A, \alpha \models \phi_1 \land \cdots \land \phi_n$.

Note that resolution inferences are only possible between goal and definite clauses and that the rules $\beta$-Reduction and Constraint Refutation are only applicable to goal clauses. Besides, every (pre-)term obtained by any of the rules is clearly a goal clause.
Example 3.1. Consider again the set $S$ of HoCHCs from Example 2.6:

\[-(z = x + y) \lor \text{Add } xy z \quad D_1\]
\[-(n \geq 1) \lor -(n > 0) \lor -(\text{Iter } f s n x) \lor -(n = 1) \lor -(n = y) \lor -(x = n + y) \lor -(x \leq n + n) \quad D_2\]
\[-(n \geq 1) \lor -(n > 0) \lor -(n - 1 \leq 0) \lor -(n = y) \lor -(x = n + y) \lor -(x \leq n + n) \quad D_3\]

A refutation of $S$ is given in Fig. 3.1. The last inference is admissible because for any valuation satisfying $\alpha(n) = \alpha(y) = 1$ and $\alpha(x) = 2$,

$A_{\text{LIA}}, \alpha \models (n \geq 1) \land (n > 0) \land (n - 1 \leq 0) \land (n = y) \land (x = n + y) \land (x \leq n + n)$.

However, the following example illustrates that the rules as stated cannot refute every $(A_{\text{LIA}}, S)$-unsatisfiable set of HoCHC:

Example 3.2. Clearly $\{-(y = x_R) \lor R x R, \neg R z_1 \lor \neg R z_2 \lor -(z_1 \neq z_2)\}$ is unsatisfiable. However, the only three possible applications of the Resolution rule yield $\neg R z_1 \lor -(y = z_2) \lor -(z_1 \neq z_2)$, $-(y = z_1) \lor -R y z_2 \lor -(z_1 \neq z_2)$ and moreover $-(y = z_1) \lor -(y = z_2) \lor -(z_1 \neq z_2)$. Besides, for every valuation $\alpha$,

$A_{\text{LIA}}, \alpha \not\models (y = z_1) \land (y = z_2) \land (z_1 \neq z_2)$.

Rather, we have to apply the rules modulo renaming of (free) variables, in general:

Definition 3.3. Let $S \cup \{C\}$ be a set of HoCHCs. We write $S \Rightarrow_{\text{res}} S \cup \{C\}$ if there exist $C_1, C_2 \in S$ and HoCHCs $C'_1, C'_2$ satisfying

(i) $C'_1$ and $C'_2$ are obtained from $C_1$ and $C_2$, respectively, by renaming (free) variables and
(ii) $C$ is derived from $C'_1$ (and $C'_2$) by one of the rules Resolution, $\beta$-Reduction or Constraint Refutation.

Furthermore, let $\Rightarrow^*_\text{res}$ be the reflexive, transitive closure of $\Rightarrow_{\text{res}}$. 
Next, to justify the constraint refutation rule, we first prove the following:

**Lemma 3.4.** Let \( G \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n \) be goal clause, where \( G \) is simple and each \( \phi_i \) is a background atom. Then \( G \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n \models \neg \phi_1 \lor \cdots \lor \neg \phi_n \).

**Proof.** Let \( \mathcal{H} \) be a Henkin frame and suppose \( B \) is an arbitrary \( (\Sigma', \mathcal{H}) \)-structure satisfying

\[
B \models_{\mathcal{H}} G \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n. \tag{3.1}
\]

Let \( \alpha \) be an arbitrary \((\Delta, \mathcal{H})\)-valuation. We define another \((\Delta, \mathcal{H})\)-valuation \( \alpha' \) by

\[
\alpha'(x) = \begin{cases} 
\overline{\Delta} & \text{if } \Delta(x) = \rho \\
\alpha(x) & \text{otherwise } (\Delta(x) = \iota)
\end{cases}
\]

Because \( G \) is simple, \( B, \alpha' \not\models G \). Hence, by Eq. (3.1), \( B, \alpha' \models \neg \phi_i \) for some \( i \). Note that by Remark 1.2, \( \phi_i \) only contains variables of type \( \iota \). Hence, also \( B, \alpha \models \neg \phi_i \). This proves, \( B \models_{\mathcal{H}} \neg \phi_1 \lor \cdots \lor \neg \phi_n \). \( \square \)

Now, soundness of the proof system is rather straightforward:

**Proposition 3.5.** Let \( S \) be a set of HoCHCs and suppose that \( S \Rightarrow_{\text{Res}} S \cup \{C\} \). Then

(i) if \( C \neq \bot \) then \( S \models C \);
(ii) if \( B \) is an expansion of \( A \) and \( B \models S \) then \( B \models C \).

**Proof.**

(i) Note that by assumption the rule Constraint Refutation cannot have been applied. Besides, for \( \beta \)-Reduction this is a consequence of Lemma 1.12(i). Finally suppose that \( \neg R M \lor G \) and \( G' \lor R \tau R \) are in \( S \) (modulo renaming of variables). The proof for this case uses the same ideas as the classic one for first-order logic (see e.g. [Robinson, 1965, Fitting, 1996]):

Let \( \mathcal{H} \) be a Henkin frame, \( B \) be an arbitrary structure satisfying \( B \models_{\mathcal{H}} S \) and let \( \alpha \) be a valuation. Furthermore assume that \( B, \alpha \not\models G \). Due to \( B \models_{\mathcal{H}} S \) it must hold that \( B, \alpha \not\models R M \). Hence, \( B, \alpha[\tau R \mapsto B^H[M]](\alpha) \not\models R \tau R \) and since \( B \models_{\mathcal{H}} S \), \( B, \alpha[\tau R \mapsto B^H[M]](\alpha) \models G' \), which by the Substitution Lemma 1.9 implies \( B, \alpha \models G'[M/\tau R] \). Consequently, \( B, \alpha \models G \lor G'[M/\tau R] \).

(ii) Let \( B \) be an expansion of \( A \) satisfying \( B \models S \). By Part (i) it suffices to consider the case when the rule Constraint Refutation is applicable to some goal clause \( G \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n \), where \( G \) is simple, each \( \phi_i \) is a background atom and there exists a valuation \( \alpha \) such that \( A, \alpha \models \phi_1 \land \cdots \land \phi_n \). However, by Lemma 3.4, \( B, \alpha \models \neg \phi_1 \lor \cdots \lor \neg \phi_n \), which is clearly a contradiction to the fact that \( B \) is an expansion of \( A \). \( \square \)

**Corollary 3.6** (Soundness). Let \( S \) be a set of HoCHCs and suppose that \( S \Rightarrow^*_{\text{Res}} S' \). Then

(i) if \( \bot \not\models S' \) then \( S \models S' \);
(ii) if \( B \) is an expansion of \( A \) and \( B \models S \) then \( B \models S' \);
(iii) if \( \bot \in S' \) then \( S \) is \((A, \mathcal{H})\)-unsatisfiable.
The completeness theorem seems to be significantly more difficult. In fact, we will not prove it until the end of Section 3.3.

**Theorem 3.7** (Completeness). Let $S$ be a $\langle A, H \rangle$-unsatisfiable set of HoCHCs.

Then $S \Rightarrow_{\text{Res}}^* \{\bot\} \cup S'$ for some $S'$.

The following observation is a trivial consequence of the soundness and refutational completeness of the proof system:

**Corollary 3.8.** Let $S$ be a set of HoCHCs.

$S$ is $\langle A, H \rangle$-unsatisfiable if and only if there exists $S'$ such that $S \Rightarrow_{\text{Res}}^* \{\bot\} \cup S'$.

Consequently, the resolution proof system gives rise to a semi-decision procedure for the $\langle A, H \rangle$-satisfiability problem if it is (semi-)decidable whether conjunctions of background atoms are satisfiable in the background theory\(^1\).

Now, suppose that $S$ is a finite, $\langle A, H \rangle$-satisfiable set of HoCHCs that we can “saturation” in the following sense: It is possible to effectively obtain a set $S'$ of HoCHCs satisfying (i) $S \Rightarrow_{\text{Res}}^* S'$ and (ii) $\bot \in S'$ or for all $S''$ such that $S' \Rightarrow_{\text{Res}} S''$, $S'' = S'$. By Corollary 3.8, $S$ is $\langle A, H \rangle$-satisfiable if and only if $\bot \in S'$. Therefore, the resolution proof system induces a decision procedure for sets of HoCHCs satisfying these properties.

However, in accordance with the Undecidability Theorem 2.18(i) (and Completeness Theorem 3.7), it is impossible to saturate all satisfiable sets of HoCHCs, as the following example illustrates:

**Example 3.9** (Non-Termination). Consider the set of HoCHCs

$$S = \{\neg R(x_R - 1) \lor R x_R, \neg R x\}.$$  

$S$ is trivially $\langle A_{\text{LIA}}, H \rangle$-satisfiable. Furthermore, the resolution proof system cannot saturate $S$ (in particular not derive $\bot$ due to soundness) because new clauses can be derived indefinitely:

$$\begin{array}{c}
\text{Res} & \neg R x & \neg R(x_R - 1) \lor R x_R \\
\text{Res} & \neg R(x - 1) & \neg R(x - 1) \lor R x_R \\
\text{Res} & \neg R((x - 1) - 1) & \neg R(x - 1) \lor R x_R \\
\vdots & & \\
\end{array}$$

In fact, for any set $S' \supseteq S$ of HoCHCs it holds that for all $S' \Rightarrow_{\text{Res}} S''$, $S' = S''$, only if $S' \supseteq S \cup \{\neg R(x - 1 - \cdots - 1) | n \in \mathbb{N}\}$.

### 3.1.1 Outline of the Completeness Proof

Next, we give a brief outline of the completeness proof, which occupies the remainder of this chapter:

\(^1\)i.e. whether there exists a valuation $\alpha$ such that $A, \alpha \models \phi_1 \land \cdots \land \phi_n$
3.2. CANONICAL STRUCTURE

(S1) First, we show how to iteratively construct a canonical structure (Section 3.2), which is in fact the least model (in a non-standard sense) of the definite clauses in $S$ (Section 3.2.2).

(S2) Then, we prove that if a goal clause is falsified by the canonical structure then it is already falsified after a finite number of iterations (Section 3.2.3).

(S3) Afterwards, we use this insight to argue that the reason why the goal clause is falsified can be captured syntactically by essentially “unfolding definitions” (Section 3.3.1).

(S4) Finally, we prove that the “unfolding” actually only needs to take place at the leftmost positions of atoms, which can be captured by the resolution proof system (Sections 3.3.2 and 3.3.3).

Note that the character of Proof Steps (S1) and (S2) is model theoretic/semantic whilst the character of Proof Steps (S3) and (S4) is proof theoretic/syntactic.

3.2 Canonical Structure

In this section, we define a canonical structure for the program $P$ associated to $S$ (following e.g. [Charalambidis et al., 2013]) and prove various properties needed for the completeness proof.

3.2.1 Construction and Basic Properties

Let $\mathcal{B}$ be an expansion of $\mathcal{A}$. Given $\mathcal{B}$, the immediate consequence operator $T^H_P$ returns $T^H_P(\mathcal{B})$ defined by

$$R^{T^H_P(\mathcal{B})} = \mathcal{B}[\lambda R \cdot F_R](\top^H)$$

for relational symbols $R \in \Sigma' \setminus \Sigma$. (Recall that $F_R$ is the unique positive existential formula such that $\neg F_R \lor R\bar{x} \in P$.) Note that $\lambda \bar{x}. F_R$ does not contain free variables and by the definition of Henkin frames this is again an expansion of $\mathcal{A}$. If $\mathfrak{B}$ is a set of expansions of $\mathcal{A}$ then $\bigsqcup \mathfrak{B}$ is the expansion of $\mathcal{A}$ defined by

$$R^{\bigsqcup \mathfrak{B}} = \bigsqcup \{R^B \mid B \in \mathfrak{B}\},$$

for $R \in \Sigma' \setminus \Sigma$. Using the immediate consequence operator we can define

$$\begin{align*}
A^H_P,0 &= \perp^H, \\
A^H_P,\beta+1 &= T^H_P(A^H_P,\beta), & \text{for } \beta \in \text{On} \\
A^H_P,\gamma &= \bigsqcup_{\beta<\gamma} A^H_P,\beta, & \text{for } \gamma \in \text{Lim} \\
A^H_P &= \bigsqcup_{\beta \in \text{On}} A^H_P,\beta,
\end{align*}$$
where \( \perp_H^\Sigma \) is the \( \Sigma' \)-expansion of \( \mathcal{A} \) satisfying \( R^{\perp_H^\Sigma} = \perp_H^\Sigma \) for relational symbols \( R \in \Sigma' \setminus \Sigma \) of type \( \rho \). Note that by definition of (pre-)Henkin frames, all \( \mathcal{A}_{P,\beta}^H \) are expansions of \( \mathcal{A} \). If no confusion arises, we abbreviate \( \mathcal{A}_{P,\beta}^H \) as \( \mathcal{A}_\beta^H \) in the following.

Intuitively, we start from the \((\preceq_\Sigma)\)-minimal structure \( \perp_H^\Sigma \) and greedily enlarge the current structure to satisfy more of the program. In fact, we end up with a structure satisfying all of \( P \):

**Proposition 3.10 (Properties of \( \mathcal{A}_\beta^H \)).**

(i) There is an ordinal \( \gamma \) satisfying \( \mathcal{A}_\beta^H = \mathcal{A}_\beta^{\perp_H^\Sigma} \).

(ii) \( \mathcal{A}_\beta^H \models P \);

**Proof.** (i) First note that \( \mathcal{B} = \{ \mathcal{A}_\beta^H \mid \beta \in \text{On} \} \) is a set. For \( \mathcal{B} \in \mathcal{B} \) let \( \beta_\mathcal{B} \) be the minimal ordinal such that \( \mathcal{B} = \mathcal{A}_{\beta_\mathcal{B}}^H \). By the replacement axiom, \( \Delta = \{ \beta_\mathcal{B} \mid \mathcal{B} \in \mathcal{B} \} \) is a set of ordinals. Hence, \( \bigcup \Delta \) is an ordinal. Let \( \gamma \geq \bigcup \Delta \) be a limit ordinal. Then it holds

\[
\mathcal{A}_\beta^H = \bigsqcup \{ \mathcal{A}_\beta^H \mid \beta \in \Delta \} = \bigsqcup \mathcal{B} = \mathcal{A}_\beta^H.
\]

(ii) Assume towards contradiction that \( \mathcal{A}_\beta^H \not\models P \). Then there exists \( \neg F_R \lor R \pi R \in P \) and \( \pi \) satisfying \( \mathcal{A}_{\beta_\mathcal{B}}^H, \models \top_\Delta [\pi R \rightarrow \pi] \not\models \neg F_R \lor R \pi R \) (because free(\( F_R \)) = \( \pi R \)). By Part (i), there is an ordinal \( \beta \) such that \( \mathcal{A}_\beta^H = \mathcal{A}_{\beta_\mathcal{B}}^H \). Hence, \( \mathcal{A}_\beta^H, \models \top_\Delta [\pi R \rightarrow \pi] \models F_R \) and due to the fact that \( \mathcal{H} \) is a Henkin frame, \( \mathcal{A}_\beta^H, \models \top_\Delta [\pi R \rightarrow \pi] \models R A_{\beta_\mathcal{B}}^H (\pi) = 1 \). Consequently, \( R_{\mathcal{A}_{\beta_\mathcal{B}}^H} (\pi) = 1 \), which implies \( R A_{\beta_\mathcal{B}}^H (\pi) = 1 \). Clearly, this is a contradiction to \( \mathcal{A}_\beta^H, \models \top_\Delta [\pi R \rightarrow \pi] \not\models \neg F_R \lor R \pi R \). \( \square \)

Note that unlike the first-order case [Bjørner et al., 2015], stage \( \omega \) is not a fixed point of \( H^\Sigma_\beta \) in general as the following example illustrates:

**Example 3.11.** Consider the following program:

\[
\neg (x_R = 0 \lor R (x_R - 1)) \lor R x_R
\]

\[
\neg x_U R \lor U x_U
\]

where \( \Sigma' = \Sigma_{\text{LIA}} \cup \{ R : \iota \rightarrow o, U : ( (\iota \rightarrow o) \rightarrow o ) \rightarrow o \} \) and \( \Delta = \{ x_R : \iota, x_U : ( \iota \rightarrow o ) \rightarrow o \} \). Let \( \mathcal{A} \) be the standard model of linear integer arithmetic \( \mathcal{A}_{\text{LIA}} \) and let \( \mathcal{H} = \mathcal{S} \) be the standard frame. For ease of notation, we introduce the following functions

\[
r_\alpha : \mathcal{S}[\iota] \rightarrow \mathbb{B}
\]

\[
\delta_\alpha : \mathcal{S}[\iota \rightarrow o] \rightarrow \mathbb{B}
\]

\[
n \mapsto \begin{cases} 1 & \text{if } 0 \leq n < \alpha \\ 0 & \text{otherwise} \end{cases}
\]

\[
r \mapsto \begin{cases} 1 & \text{if } r = r_\alpha \\ 0 & \text{otherwise} \end{cases}
\]

where \( \alpha \in \omega \cup \{ \omega \} \). Then it holds \( R A^S = r_n, U A^S = \perp_{(\iota \rightarrow o) \rightarrow o} \) and \( U A^S (s) = s(r_{n-1}) \) for \( n > 0 \). Therefore

\[
R A^S = r_\omega
\]

\[
U A^S (s) = \begin{cases} 1 & \text{if there exists } n < \omega \text{ satisfying } s(r_n) = 1 \\ 0 & \text{otherwise} \end{cases}
\]

In particular, \( U A^S (\delta_\omega) = 0 \). On the other hand, \( \mathcal{A}_\omega^S, \models \top_S [x_U \rightarrow \delta_\omega] \models x_U R \) holds and therefore \( U A^{S+1} (\delta_\omega) = 1 \).
3.2. CANONICAL STRUCTURE

However, in Section 3.2.3, we will show that stage \( \omega \) is “essentially” a fixed point of \( T^H_P \).

3.2.2 Quasi-Monotonicity

In [Cathcart Burn et al., 2018], the authors show that sets of HoCHCs do not in general have least models with respect to the pointwise ordering \( \sqsubseteq \). In particular, \( \mathcal{A}^H_P \) is not \( \sqsubseteq \)-minimal.

Example 3.12. Consider the program \( P \)

\[
\neg x_R U \lor Rx_R \quad \neg x_U \neq x_U \lor Ux_U
\]

with signature \( \Sigma' = \Sigma \cup \{ R : (\tau \to o) \to o, U : \tau \to o \} \), a type environment \( \Delta \supseteq \{ x_R : (\tau \to o) \to o, x_U : \tau \to o \} \) taken from [Cathcart Burn et al., 2018]. Let \( \mathcal{H} = S \) be the standard frame and let \( \neg \mathfrak{g} \in S[\tau \to o] \) be such that \( \neg \mathfrak{g}(\bot^S_{\tau \to o}) = 1 \) and \( \neg \mathfrak{g}(\top^S_{\tau \to o}) = 0 \) (which clearly exists in \( S[\tau \to o] \)).

There are (at least) two expansions \( B_1 \) and \( B_2 \) defined by

\[
\begin{align*}
U^{B_1} &= \bot^S_{\tau \to o} \quad & U^{B_2} &= \top^S_{\tau \to o} \\
R^{B_1}(s) &= \begin{cases} 1 & \text{if } s(\bot^S_{\tau \to o}) = 1 \\ 0 & \text{otherwise} \end{cases} \\
R^{B_2}(s) &= \begin{cases} 1 & \text{if } s(\top^S_{\tau \to o}) = 1 \\ 0 & \text{otherwise} \end{cases}
\end{align*}
\]

Note that both \( B_1 \models P \) and \( B_2 \models P \) and there are no models smaller than any of these with respect to the pointwise ordering \( \sqsubseteq \). Furthermore, neither \( B_1 \sqsubseteq B_2 \) nor \( B_2 \sqsubseteq B_1 \) holds because \( R^{B_1}(\neg \mathfrak{g}) = 1 > 0 = R^{B_2}(\neg \mathfrak{g}) \) and for any \( n \in S[\iota] \), \( U^{B_2}(n) = 1 > 0 = U^{B_1}(n) \).

However, it turns out that \( \mathcal{A}^H_P \) is minimal with respect to a carefully chosen relation, which, in contrast to \( \sqsubseteq \), only takes elements into account which are already related:

Definition 3.13. We define a relation \( \sqsubseteq^H_{m,\pi} \subseteq \mathcal{H}[\sigma] \times \mathcal{H}[\sigma] \) in the following way by induction on the types \( \sigma \) by

\[
\begin{align*}
f \sqsubseteq^H_{m,\iota \to \tau} f' & \iff f = f' \quad & \text{for } f, f' \in \mathcal{H}[[\tau^n \to \iota]] \\
b \sqsubseteq^H_{m,\pi} b' & \iff b \leq b' \quad & \text{for } b, b' \in \mathcal{H}[\pi] \\
r \sqsubseteq^H_{m,\tau \to \rho} r' & \iff \forall s, s' \in \mathcal{H}[[\tau]], s \sqsubseteq^H_{m,\tau} s' \to r(s) \sqsubseteq^H_{m,\rho} r'(s') \quad & \text{for } r, r' \in \mathcal{H}[[\tau \to \rho]]
\end{align*}
\]

\( r \in \mathcal{H}[\rho] \) is quasi-monotone if \( r \sqsubseteq^H_{m,\rho} r' \).

In the following, we frequently omit the subscript (since it can be inferred) or the superscript. Note that \( \sqsubseteq^H_{m,\pi} \) is transitive (Lemma A.1) but neither reflexive nor antisymmetric (Examples 3.14(iii) and 3.14(iv)).

Example 3.14. (i) All of \( \bot^H_{\rho}, \top^H_{\rho}, \) or and and are trivially quasi-monotone.

(ii) Next, suppose \( r, r' \in \mathcal{H}[\tau \to o] \) are such that \( r \sqsubseteq^H_{m} r' \) and \( \exists s(r)(s) = 1 \). Hence, there exists \( s \in \mathcal{H}[[\tau]] \) satisfying \( r(s) = 1 \). If \( \tau = \iota \) then \( r'(s) = 1 \) and otherwise \( r'(\top^H_{\tau}) = 1 \) because \( s \sqsubseteq^H_{m} \top^H_{\tau} \). Consequently, \( \exists \mathcal{H}([r']) = 1 \) and \( \exists \mathcal{H} \) is quasi-monotone, too.
(iii) Let \( \mathcal{H} = \mathcal{S} \) be the standard frame and let \( \neg : \mathbb{B} \to \mathbb{B} \) be defined by \( \neg(b) = 1 - b \) for \( b \in \mathbb{B} \). Clearly, \( \neg \in \mathcal{S}[o \to o] \) and \( 0 \sqsubseteq_m 1 \). However, \( \neg(0) = 1 > 0 = \neg(1) \).
This shows that \( \sqsubseteq_m \) is not reflexive.
(iv) To also prove that \( \sqsubseteq_m \) is not antisymmetric we define

\[
r : \mathcal{S}[o \to o] \to \mathbb{B} \quad \quad \quad \quad \quad r' : \mathcal{S}[o \to o] \to \mathbb{B}
\]

\[
s \mapsto \begin{cases} 1 & \text{if } s = \top^S_{o \to o} \\ 0 & \text{otherwise} \end{cases} \quad \quad \quad \quad \quad s' \mapsto \begin{cases} 1 & \text{if } s \in \{ \neg, \top^S_{o \to o} \} \\ 0 & \text{otherwise} \end{cases}
\]

Clearly, \( r, r' \in \mathcal{S}[(o \to o) \to o] \) and \( r \sqsubseteq_m r' \). Furthermore, note that for all \( s, s' \in \mathcal{S}[o \to o] \), if \( r'(s') = 1 \) and \( s' \sqsubseteq_m s \) then \( s = \top^S_{o \to o} \). Consequently, \( r' \sqsubseteq_m r \)
holds, too, but obviously \( r \neq r' \).

**Remark 3.15.** Let \( r, r' \in \mathcal{H}[\tau \to \rho] \). Then \( r \sqsubseteq_m r' \) if and only if for all \( \tau, \tau' \in \mathcal{H}[\rho] \) such that \( \tau \sqsubseteq_m \tau' \), \( r(\tau) \leq r'(\tau') \).

**Lemma 3.16.** Let \( \rho \) a relational type and \( \mathcal{R}, \mathcal{R}' \subseteq \mathcal{H}[\rho] \) be sets such that for each \( r \in \mathcal{R} \) there exists \( r' \in \mathcal{R}' \) satisfying \( r \sqsubseteq_m r' \). Then \( \bigcup \mathcal{R} \sqsubseteq_m \bigcup \mathcal{R}' \).

**Proof.** We prove the lemma by induction on the relational type \( \rho \). For \( \rho = o \) this is obvious. Hence, suppose \( \rho = \tau \to \rho' \). To show \( \bigcup \mathcal{R} \sqsubseteq_m \bigcup \mathcal{R}' \), let \( s \sqsubseteq_m s' \in \mathcal{H}[\tau] \). We define \( \mathcal{S} = \{ r(s) \mid r \in \mathcal{R} \} \) and \( \mathcal{S}' = \{ r'(s') \mid r' \in \mathcal{R}' \} \). Let \( r \in \mathcal{R} \). By assumption, there exists \( r' \in \mathcal{R}' \) such that \( r \sqsubseteq_m r' \). Hence, due to \( s \sqsubseteq_m s' \), \( r(s) \sqsubseteq_m r'(s') \). Therefore, the inductive hypothesis is applicable to \( \mathcal{S} \) and \( \mathcal{S}' \), which yields \( (\bigcup \mathcal{R})(s) = \bigcup \mathcal{S} \sqsubseteq_m \bigcup \mathcal{S}' = (\bigcup \mathcal{R}')(s') \).

As a consequence, \( \bigcup \mathcal{R} \sqsubseteq_m \bigcup \mathcal{R}' \).

Next, we lift all notions to valuations and structures in a pointwise manner: For valuations \( \alpha \) and \( \alpha' \) and expansions \( \mathcal{B} \) and \( \mathcal{B}' \) of \( \mathcal{A} \),

\[
\alpha \sqsubseteq_m \alpha' \iff \forall x \in \text{dom}(\Delta). \alpha(x) \sqsubseteq_m \alpha'(x),
\]
\[
\mathcal{B} \sqsubseteq_m \mathcal{B}' \iff \forall c \in \Sigma'. \mathcal{B} \sqsubseteq_m \mathcal{B}'
\]

\( \alpha \) is quasi-monotone if \( \alpha \sqsubseteq_m \alpha \) and \( \mathcal{B} \) is quasi-monotone if \( \mathcal{B} \sqsubseteq_m \mathcal{B} \). As a consequence of Lemma 3.16, we get:

**Corollary 3.17.** Let \( \mathcal{B} \) be a set of quasi-monotone expansions of \( \mathcal{A} \). Then

(i) \( \mathcal{B} \in \mathcal{B} \) implies \( \mathcal{B} \sqsubseteq_m \bigcup \mathcal{B} \), and

(ii) \( \bigcup \mathcal{B} \) is quasi-monotone.

(iii) if \( \mathcal{B}' \) is an expansion of \( \mathcal{A} \) such that for all \( \mathcal{B} \in \mathcal{B} \), \( \mathcal{B} \sqsubseteq_m \mathcal{B}' \), then \( \bigcup \mathcal{B} \sqsubseteq_m \mathcal{B}' \).

**Example 3.18.** For the structures \( \mathcal{B}_1 \) and \( \mathcal{B}_2 \) of Example 3.12 it holds that \( \mathcal{B}_1 = \mathcal{A}_p^S \) and \( \mathcal{B}_1 \sqsubseteq_m \mathcal{B}_2 \) because due to \( \bot^S_{o \to o} \sqsubseteq_m \top^S_{o \to o} \), for any \( s \sqsubseteq_m s' \), \( s(\bot^S_{o \to o}) \leq s'(\top^S_{o \to o}) \) and therefore \( R^{\mathcal{B}_1}(s) \leq R^{\mathcal{B}_2}(s') \). In particular, \( \neg \) is not quasi-monotone and therefore the fact that \( R^{\mathcal{B}_1}(\neg) > R^{\mathcal{B}_2}(\neg) \) is not a concern.

All definitions have been set up in such a way that the denotation of positive existential terms is monotone with respect to \( \sqsubseteq_m \):
Proposition 3.19. Let \( \mathcal{B} \subseteq_m \mathcal{B}' \) be expansions of \( \mathcal{A} \), \( \alpha \subseteq_m \alpha' \) be valuations and let \( M \) be a positive existential term. Then \( \mathcal{B}[M](\alpha) \subseteq_m \mathcal{B}'[M](\alpha') \).

Proof. We prove the claim by induction on the structure of \( M \).

- If \( M \) is a variable \( x \) then \( \mathcal{B}[M](\alpha) = \alpha(x) \subseteq_m \alpha'(x) = \mathcal{B}'[M](\alpha') \) because of \( \alpha \subseteq_m \alpha' \).
- If \( M \) is a logical symbol (other than \( \neg \)) then this is a consequence of Examples 3.14(i) and 3.14(ii).
- If \( M \) is a (first-order) symbol \( c \in \Sigma \) then \( R^B = R^A = R^{B'} \). Otherwise, it is a (relational) symbol \( R \in \Sigma' \setminus \Sigma \) and \( \mathcal{B}[M](\alpha) = R^B \subseteq_m R^{B'} = \mathcal{B}'[M](\alpha') \) because of \( \mathcal{B} \subseteq_m \mathcal{B}' \).
- If \( M \) is an application \( N\ N' \) then by the inductive hypothesis \( \mathcal{B}[N](\alpha) \subseteq_m \mathcal{B}'[N](\alpha') \) and \( \mathcal{B}[N'](\alpha) \subseteq_m \mathcal{B}'[N'](\alpha') \). First, suppose that \( \Delta \vdash \tau : \nu^{n+1} \rightarrow \nu \). Then \( \mathcal{B}[N](\alpha) = \mathcal{B}'[N](\alpha') \) and \( \mathcal{B}[N'](\alpha) = \mathcal{B}'[N'](\alpha') \). Therefore,
  \[
  \mathcal{B}[M](\alpha) = \mathcal{B}[N](\alpha)(\mathcal{B}[N'](\alpha)) = \mathcal{B}'[N](\alpha')(\mathcal{B}'[N'](\alpha')) = \mathcal{B}'[M'](\alpha)
  \]
and thus \( \mathcal{B}[M](\alpha) \subseteq_m \mathcal{B}'[M'](\alpha) \). Otherwise, \( \Delta \vdash \tau \rightarrow \rho \) and hence,
  \[
  \mathcal{B}[M](\alpha) = \mathcal{B}[N](\alpha)(\mathcal{B}[N'](\alpha)) \subseteq_m \mathcal{B}'[N](\alpha')(\mathcal{B}'[N'](\alpha')) = \mathcal{B}'[M'](\alpha)
  \]
by definition of \( \subseteq_m \).
- Finally, suppose \( M \) is an abstraction \( \lambda x. \mathcal{N} \). Let \( s \subseteq_m s' \). By the inductive hypothesis \( \mathcal{B}[N](\alpha[x \mapsto s]) \subseteq_m \mathcal{B}'[N](\alpha'[x \mapsto s']) \) and hence,
  \[
  \mathcal{B}[M](\alpha)(s) = \mathcal{B}[N](\alpha[x \mapsto s]) \subseteq_m \mathcal{B}'[N](\alpha'[x \mapsto s']) = \mathcal{B}'[M](\alpha)(s')
  \]
because \( \mathcal{H} \) is a Henkin frame. Due to the fact that this holds for every \( s \subseteq_m s' \), \( \mathcal{B}[M](\alpha) \subseteq_m \mathcal{B}'[M'](\alpha') \).

Consequently, also the immediate consequence operator is monotone with respect to \( \subseteq_m \).

Corollary 3.20. If \( \mathcal{B} \subseteq_m \mathcal{B}' \) then \( T_\mathcal{H}^\mathcal{B}(\mathcal{B}) \subseteq_m T_\mathcal{H}^\mathcal{B}(\mathcal{B}') \). In particular, \( T_\mathcal{H}^\mathcal{B}(\mathcal{B}) \) is quasi-monotone if \( \mathcal{B} \) is quasi-monotone.

Proof. \( T_\mathcal{H}^\mathcal{B} \) is quasi-monotone (Example 3.14(i)). Hence, by Proposition 3.19, for every \( R \in \Sigma' \setminus \Sigma \),
  \[
  R^{T_\mathcal{H}^\mathcal{B}(\mathcal{B})} = \mathcal{B}[R](\lambda x. F R)(\lambda x. F R)(\mathcal{H})(\lambda x. F R)(\lambda x. F R) = R^{T_\mathcal{H}^\mathcal{B}(\mathcal{B}')}. \]

The straightforward proof (by transfinite induction) of the following lemma uses Corollaries 3.17(i), 3.17(iii) and 3.20 and can be found in Appendix A.1.

Lemma 3.21. Let \( \beta \) be an ordinal. Then

(i) \( A_\beta^\mathcal{H} \) is quasi-monotone and
(ii) for all ordinals \( \beta' \geq \beta \), \( A_{\beta'}^\mathcal{H} \subseteq_m A_\beta^\mathcal{H} \).
Next, we prove the key property to establish the \( \subseteq_m \)-leastness of the canonical structure \( A^H_P \).

**Proposition 3.22.** Let \( B \) be an expansion of \( A \) satisfying \( B \models P \) and let \( \beta \) be an ordinal. Then \( A^H_{\beta} \subseteq_m B \).

**Proof.** We prove the lemma by induction on \( \beta \).

- If \( \beta = 0 \) this is obvious because both \( A^H_0 \) and \( B \) are expansions of \( A \).
- Next, suppose \( \beta = \beta' + 1 \) is a successor ordinal. Let \( R : \tau \to o \in \Sigma' \setminus \Sigma \) and \( \pi, \pi' \in H[\tau] \) be such that \( \pi \subseteq_m \pi' \). Assume that \( R^A (\pi) = 1 \). Then \( A^H_B [F_R] (\top_A (\pi_R \to \pi)) = 1 \) (\( H \) is a Henkin frame). By the inductive hypothesis and Proposition 3.19,

\[
1 = A^H_B [F_R] (\top_A (\pi_R \to \pi)) \leq B [F_R] (\top_A (\pi_R \to \pi'))
\]

Consequently, due to \( B \models P \), \( R^B (\pi') = 1 \).
- Finally, if \( \beta \) is a limit ordinal, by the inductive hypothesis, \( A^H_{\beta'} \subseteq_m B \) for each \( \beta' < \beta \). Thus, by Corollary 3.17(iii), \( A^H_{\beta} \subseteq_m B \).

**Theorem 3.23** (Properties of \( A^H_B \) (cont.)). (i) \( A^H_B \) is quasi-monotone;

(ii) For each ordinal \( \beta \), \( A^H_{\beta} \subseteq_m A^H_P \).

(iii) If \( B \models P \) is an expansion of \( A \) then \( A^H_{\beta} \subseteq_m B \).

(iv) If \((P, F)\) is \((A, H)\)-solvable then \( A^H_P \models P \) and \( A^H_P \not\models F \).

**Proof.** (i) Proposition 3.10(i) and Lemma 3.21(i).

(ii) Corollary 3.17(i).

(iii) Propositions 3.10(i) and 3.22.

(iv) Let \( B \) be an expansion of \( A \) satisfying \( B \models P \) and \( B \not\models F \). By Proposition 3.19 and Part (iii), \( A^H_B [F] (\top_A (\pi)) \leq B [F] (\top_A (\pi)) \). Hence, by assumption and the fact that \( F \) is closed, \( A^H_B \not\models F \). Furthermore, by Proposition 3.10(ii) \( A^H_B \models P \).

Hence, we have established that \( A^H_B \) is the “canonical, least solution” to \((P, F)\) if it is \((A, H)\)-solvable and \( A^H_B \) is quasi-monotone.

### 3.2.3 Quasi-Continuity

Whilst in the previous section we have shown that \( A^H_P \not\models F \) (and \( A^H_P \models P \)) if \((P, F)\) is \((A, H)\)-solvable, we now turn to the question what we can tell if \( A^H_P \not\models F \) does not hold (i.e. \((P, F)\) is not \((A, H)\)-solvable). Although \( T^H_P \) is not continuous (Example 3.11), we show that there still exists some (finite) \( n \in \omega \) satisfying \( A^H_P \models F \) (Theorem 3.36).

To accomplish this, we define yet another relation:

**Definition 3.24.** We define \( \subseteq^H_{c,a} \subseteq H[\sigma] \times H[\sigma] \), the notion of \((\subseteq^H_{c,a})\)-directedness and quasi-continuity, as well as \( \text{dir}^H_{c,a} (-) \) simultaneously by induction on the type \( \sigma \):

\[
\begin{align*}
f^H_{c,n} \rightarrow f' & \iff f = f' & \text{for } f, f' \in H[\sigma^n] \rightarrow \delta \\
b^H_{c,o} b' & \iff b \leq b & \text{for } b, b' \in H[\sigma^o]
\end{align*}
\]
3.2. CANONICAL STRUCTURE

\[ r \subseteq_{c, \tau \rightarrow \rho} r' \iff \forall s \in \mathcal{H}[\tau], \mathcal{G}' \in \text{dir}^\mathcal{H}(s). \]

\[ r(s) \subseteq_{c, \rho} \bigcup \{ r'(s') \mid s' \in \mathcal{G}' \} \text{ for } r, r' \in \mathcal{H}[\tau \rightarrow \rho] \]

(i) A set \( \mathcal{R} \subseteq \mathcal{H}[\sigma] \) is \((\subseteq_{c, \sigma})\)-directed if for any \( r_1, r_2 \in \mathcal{R} \) there exists \( r' \in \mathcal{R} \) satisfying \( r_1, r_2 \subseteq_{c, \sigma} r' \).

(ii) \( \tau \in \mathcal{H}[\sigma] \) is quasi-continuous if \( r \subseteq_{c, \sigma} r \), and for a set \( \mathcal{R} \subseteq \mathcal{H}[\sigma] \), let \( C(\mathcal{R}) \) be the set of quasi-continuous elements of \( \mathcal{R} \).

(iii) Finally, for \( s \in \mathcal{H}[\sigma] \), \( \text{dir}^{\mathcal{H}}(s) \) is the set of non-empty directed sets \( \mathcal{G}' \subseteq C(\mathcal{H}[\sigma]) \) such that \( s \subseteq_{c, \sigma} \bigcup \mathcal{G}' \).

Note that for \( f \in \mathcal{H}[\nu \rightarrow \iota] \), \( \text{dir}^{\mathcal{H}}(f) = \{ \{ f \} \} \). In the following, we omit the subscript (since it can be inferred) or superscript from \( \subseteq_{c, \sigma} \) and \( \text{dir}^{\mathcal{H}}(s) \).

The relation \( \subseteq_c \) is transitive (Lemma A.2) but neither reflexive nor antisymmetric (Examples 3.25(iv) and 3.25(v)).

Example 3.25. (i) All of \( \bot_{\rho}^\mathcal{H}, \top_{\rho}^\mathcal{H} \) or and and are trivially quasi-continuous.

(ii) Next, suppose \( r \in \mathcal{H}[\tau \rightarrow \o], \mathcal{R}' \in \text{dir}(r) \) and \( s \in \mathcal{H}[\tau] \) are such that \( r(s) = 1 \).

If \( \tau = \iota \) then \( r'(s) = 1 \) for all \( r' \in \mathcal{R}' \). Otherwise, there exists \( r' \in \mathcal{R}' \) satisfying \( r'(\top_{\tau}^\mathcal{H}) \) because \( \{ \top_{\tau}^\mathcal{H} \} \in \text{dir}(s) \). Consequently, \( \mathcal{R} \) is quasi-continuous as well.

(iii) For every relational \( \rho \) and \( s \in \mathcal{H}[\rho], \bot_{\rho}^\mathcal{H} \subseteq_c \mathcal{S} \subset \mathcal{S} \top_{\rho}^\mathcal{H} \) and both \( \top_{\rho}^\mathcal{H} \) and \( \bot_{\rho}^\mathcal{H} \) are quasi-continuous.

(iv) \( \delta_\omega \) from Example 3.11 is not quasi-continuous because clearly \( \{ r_n \mid n \in \mathbb{N} \} \in \text{dir}(r_\omega) \) holds but \( \delta_\omega(r_\omega) = 1 > 0 = \max \{ \delta_\omega(r_n) \mid n \in \mathbb{N} \} \). This shows that \( \subseteq_c \) is not reflexive.

(v) Let \( \mathcal{H} = \mathcal{S} \) be the standard frame and consider again \( r, r' \in \mathcal{S}[\o \rightarrow \o] \).

From Example 3.14(iv). Suppose \( s \in \mathcal{S}[\o \rightarrow \o] \) and \( \mathcal{G}' \in \text{dir}(s) \). Note that if \( s \in \{ \neg, \top_{\o \rightarrow \o} \} \) then \( \top_{\o \rightarrow \o} \subseteq \mathcal{S} \). Hence, \( r \subseteq_c r' \) and \( r' \subseteq_c r \) but clearly \( r \neq r' \).

It appears like the relations \( \subseteq_m \) and \( \subseteq_c \) are quite similar: Both restrict attention in the recursive case only to certain elements that are already related (and not to all elements). In fact, we get analogues of many of the results of the previous section for \( \subseteq_c \). However, it seems impossible to obtain a similar result as Theorem 3.23(iii). In fact, it turns out that \( \subseteq_m \) and \( \subseteq_c \) are incomparable, in general, as the following examples illustrate:

Example 3.26. (i) First, let \( \chi \in \mathcal{S}[\iota \rightarrow \o] \) be defined by \( \chi(r) = 1 \) if and only if \( r \) is quasi-continuous, which resembles the Dirichlet function. \( \chi \) is trivially quasi-continuous because for all \( \emptyset \neq \mathcal{R} \subseteq C(\mathcal{S}[\iota \rightarrow \o]) \), \( \max \{ \chi(r') \mid r' \in \mathcal{R} \} = 1 \).

Furthermore note that negation is not quasi-continuous (since \( \{ \iota \} \in \text{dir}(0) \)). Hence, \( \chi(\top_{\o \rightarrow \o}) = 1 > 0 = \chi(\neg) \) but \( \top_{\o \rightarrow \o} \subseteq \mathcal{S} \). Consequently, \( \subseteq_c \chi \) and \( \chi \subseteq_m \chi \).

(ii) Next, consider the following modifications of \( \delta_\omega \) from Example 3.11

\[
\delta'_\omega : \mathcal{S}[\iota \rightarrow \o] \rightarrow \mathcal{B}
\]

\[
r \mapsto \begin{cases} 1 & \text{if } r_\omega \subseteq_m r \\ 0 & \text{otherwise.} \end{cases}
\]

\(^2\text{where again neg}(b) = 1 - b\)
Note that by transitivity of $\subseteq_m$, $\delta'_\omega \subseteq_m \delta'_\omega$, and note that $r_\omega$ is quasi-monotone. Hence, $\delta'_\omega(r_\omega) = 1 > 0 = \max \{\delta'_\omega(r_n) \mid n \in \mathbb{N}\}$ but as in Example 3.25(iv), it holds that $\{r_n \mid n \in \mathbb{N}\} \in \text{dir}(r_\omega)$. Consequently, $\delta'_\omega \subseteq_m \delta'_\omega$ but $\delta'_\omega \not\subseteq_c \delta'_\omega$.

**Remark 3.27.** Similarly as in Remark 3.15, for $r, r' \in \mathcal{H}[\tau_1 \to \cdots \to \tau_n \to \rho]$, $r \subseteq_c r'$ holds if and only if for all $s_1 \in \mathcal{H}[\tau_1], \ldots, s_n \in \mathcal{H}[\tau_n]$ and $\mathcal{G}'_1 \in \text{dir}(s_1), \ldots, \mathcal{G}'_n \in \text{dir}(s_n)$,

$$r(s_1) \cdots (s_n) \subseteq_c \bigsqcup \{r'(s'_1) \cdots (s'_n) \mid s'_i \in \mathcal{G}'_1 \land \cdots \land s'_n \in \mathcal{G}'_n\}.$$

The proof of the following useful lemma is similar to the proof of Lemma 3.16 and can be found in Appendix A.2.

**Lemma 3.28.** Let $\rho$ be a relational type and let $\mathcal{R}, \mathcal{R}' \subseteq \mathcal{H}[\rho]$ be sets such that for each $r \in \mathcal{R}$ there exists $r' \in \mathcal{R}'$ satisfying $r \subseteq_c r'$. Then $\bigcup \mathcal{R} \subseteq_c \bigcup \mathcal{R}'$.

Next, we lift all notions to valuations and structures in a pointwise manner: For valuations $\alpha$ and $\alpha'$ and expansions $\mathcal{B}$ and $\mathcal{B}'$ of $\mathcal{A}$,

$$\alpha \subseteq_c \alpha' \iff \forall x \in \text{dom}(\Delta). \alpha(x) \subseteq_c \alpha'(x),$$

$$\mathcal{B} \subseteq_c \mathcal{B}' \iff \forall R \in \Sigma' \setminus \Sigma. \mathcal{R}^\mathcal{B} \subseteq_c \mathcal{R}^\mathcal{B'}.$$

$\alpha$ is quasi-continuous if $\alpha(x)$ is quasi-continuous for every $x \in \text{dom}(\Delta)$ and $\mathcal{B}$ is quasi-continuous if $\mathcal{R}^\mathcal{B}$ is quasi-continuous for every $R \in \Sigma' \setminus \Sigma$. Besides, for a valuation $\alpha$, let $\text{dir}(\alpha)$ be the set of sets of valuations $\mathcal{B}'$ such that for every $x \in \text{dom}(\Delta)$, $\{\alpha'(x) \mid \alpha' \in \mathcal{B}'\} \in \text{dir}(\alpha(x))$. Similarly, for an expansion $\mathcal{B}$ of $\mathcal{A}$, let $\text{dir}(\mathcal{B})$ be the set of sets $\mathcal{B}'$ of expansions of $\mathcal{A}$ such that for each $R \in \Sigma' \setminus \Sigma$, $\{\mathcal{R}^\mathcal{B} \mid \mathcal{B}' \in \mathcal{B}'\} \in \text{dir}(\mathcal{R}^\mathcal{B})$.

By Lemma 3.28 we get:

**Corollary 3.29.** Let $\mathcal{B}$ be a set of quasi-continuous expansions of $\mathcal{A}$. Then

(i) $\mathcal{B} \in \mathcal{B}$ implies $\mathcal{B} \subseteq_c \bigcup \mathcal{B}$,

(ii) $\bigcup \mathcal{B}$ is quasi-continuous and

(iii) if $\mathcal{B}'$ is an expansion of $\mathcal{A}$ such that for all $\mathcal{B} \in \mathcal{B}, \mathcal{B} \subseteq_c \mathcal{B}'$, then $\bigcup \mathcal{B} \subseteq_c \mathcal{B}'$.

In a similar spirit as Proposition 3.19, $\subseteq_c$ turns out to be compatible with positive existential terms:

**Proposition 3.30.** Let $M$ be a positive existential term, $\mathcal{B}$ be expansions of $\mathcal{A}$, $\mathcal{B}' \in \text{dir}(\mathcal{B})$, $\alpha$ be a valuation and let $\mathcal{B}' \in \text{dir}(\alpha)$. Then

$$\mathcal{B}[M](\alpha) \subseteq_c \bigsqcup \{\mathcal{B}'[M](\alpha') \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathcal{A}'\} \quad (3.2)$$

and the expression on the right-hand side is well-defined.

The longish proof is presented in Appendix A.2.

**Corollary 3.31.** If $\mathcal{B} \subseteq_c \mathcal{B}' \subseteq_c \mathcal{B}'$ then $\mathcal{H}[\mathcal{B} \subseteq_c \mathcal{B}']$. In particular, $\mathcal{H}[\mathcal{B}]$ is quasi-continuous if $\mathcal{B}$ is quasi-continuous.

**Proof.** $\mathcal{H}[\mathcal{B}]$ is quasi-continuous (Example 3.25(iv)). Hence, $\{\mathcal{H}[\mathcal{B}]\} \in \text{dir}(\mathcal{H}[\mathcal{B}])$ and moreover $\{\mathcal{B}'\} \in \text{dir}(\mathcal{B})$. Hence, by Proposition 3.30 for every $R \in \Sigma' \setminus \Sigma$,

$$R^\mathcal{H}(\mathcal{B}) = \mathcal{B}[\mathcal{H}[\lambda F \mathcal{R}. F\mathcal{R}]](\mathcal{H}[\mathcal{B}]) \subseteq_c (\mathcal{B}'[\mathcal{H}[\lambda F \mathcal{R}. F\mathcal{R}]](\mathcal{H}[\mathcal{B}])) = R^\mathcal{H}(\mathcal{B}')$$.
3.2. CANONICAL STRUCTURE

The proof of the following lemma is analogous to the proof of Lemma 3.21 and can be found in Appendix A.2.

**Lemma 3.32.** Let $\beta$ be an ordinal. Then

(i) $A^H_\beta$ is quasi-continuous and
(ii) for all ordinals $\beta' \geq \beta$, $A^H_{\beta'} \sqsubseteq c A^H_\beta$.

It turns out that the immediate consequence operator does not have a great impact on $A^H_\omega$:

**Lemma 3.33.** $T^H_P(A^H_\omega) \subseteq c A^H_\omega$.

**Proof.** Let $R : \rho \in \Sigma' \setminus \Sigma$. Note that $\{\top^H_\Delta\} \in \text{dir}(\top^H_\Delta)$ and $\{A^H_n \mid n \in \omega\} \in \text{dir}(A^H_\omega)$ because of Lemma 3.32. Then it holds that

\[
R^{T^H_P(A^H_\omega)} = A^H_\omega \parallel \lambda R. F_R(\top^H_\Delta) \\
\subseteq c \bigsqcup \{A^H_n \parallel \lambda R. F_R(\top^H_\Delta) \mid n \in \omega\} \\
= \bigsqcup \{R^{A^H_{n+1}} \mid n \in \omega\} \\
= R^{A^H_\omega}.
\]

Hence, by transfinite induction (for the full proof see Appendix A.2) it follows that the structure corresponding to every stage remains below $A^H_\omega$ (w.r.t. $\subseteq c$):

**Proposition 3.34.** Let $\beta$ be an ordinal. Then $A^H_\beta \subseteq c A^H_\omega$.

Note that this does not contradict Example 3.11 because $\delta_\omega$ is not quasi-continuous (Example 3.25(iv)). Rather, by Theorem 3.23(iii) and the following it implies that $\delta_\omega$ is not definable in HoCHC.

**Theorem 3.35** (Properties of $A^H_\beta$ (cont.)).

(i) $A^H_\beta$ is quasi-continuous;
(ii) for each ordinal $\beta$, $A^H_\beta \subseteq c A^H_{\beta+1} \subseteq c A^H_\omega$;

**Proof.** (i) Proposition 3.10(i) and Lemma 3.32(i).
(ii) Propositions 3.10(i) and 3.34 and Corollary 3.29(i). \[\square\]

Furthermore, we get the following result, which is important for the refutational completeness of the proof system.

**Theorem 3.36.** Let $F$ be a closed positive existential formula such that $A^H_\beta \models F$.

Then there exists $n \in \omega$ such that $A^H_n \models F$.

**Proof.** By assumption $A^H_\beta, \top^H_\Delta \models F$. Note that $\{\top^H_\Delta\} \in \text{dir}(\top^H_\Delta)$ and because of Lemma 3.32 and Theorem 3.35(ii), $\{A^H_n \mid n \in \omega\} \in \text{dir}(A^H_\beta)$. Consequently, due to Proposition 3.30 there exists $n \in \omega$ such that $A^H_n, \top^H_\Delta \models F$. \[\square\]
3.3 Refutational Completeness of the Proof System

Having established Proof Steps (S1) and (S2) (there exists $n \in \omega$ such that $\mathcal{A}_n^H \models F$ if $(P, \mathcal{F})$ is not ($\mathcal{A}, \mathcal{H}$)-solvable, Theorems 3.36 and 3.23(iv)), we need to devise a method to capture (syntactically) that $\mathcal{A}_n^H \models F$ does indeed hold (Proof Step (S3)).

3.3.1 Syntactic Unfolding

Towards that aim we consider the relation inspired by the “parallel $\beta$-reduction” relation used in the proof of the Church-Rosser property of $\beta$-reduction due to Martin-Löf and Tait (for more details refer to [Barendregt, 2012] or [Ker, 2018]). The idea is that $M \vdash N$ if all occurrences of symbols $R \in \Sigma' \setminus \Sigma$ in $M$ have been replaced by $\lambda x. R \parallel F_R$ in $N$, which is used in the definition of the immediate consequence operator to define $R^T_{\parallel}^{H}(B)$. A similar idea is exploited in [Charalambidis et al., 2013]. Formally, this is captured by

\[
\frac{R \rightarrow \parallel \lambda x. R \parallel F_R, \ c \in \Sigma' \setminus \Sigma \quad c \rightarrow c \ \epsilon \in \Sigma \cup \{\land, \lor, \exists\} \quad \bar{x} \rightarrow \bar{x} \quad M \rightarrow_{\parallel} N}{M_1 \rightarrow_{\parallel} N_1 \quad M_2 \rightarrow_{\parallel} N_2 \quad M \rightarrow_{\parallel} N} \quad \frac{\lambda x. M \rightarrow_{\parallel} \lambda x. N}{M_1 M_2 \rightarrow_{\parallel} N_1 N_2}
\]

Furthermore, we write $M \vdash_{\parallel}^n N$ if $M \rightarrow_{\parallel}^n N$. Note that $\rightarrow_{\parallel}$ is functional on positive existential terms. The intuition that $\rightarrow_{\parallel}$ allows us to capture the effect of the immediate consequence operator syntactically is made precise by the following:

**Proposition 3.37.** Let $B$ be an expansion of $A$ and let $M$ and $N$ be positive existential terms satisfying $M \rightarrow_{\parallel} N$. Then for all valuations $\alpha$, $T^H_{\parallel}(B)[M](\alpha) = B\parallel[N](\alpha)$.

**Proof.** We prove the lemma by induction on $\rightarrow_{\parallel}$:

- For variables and logical constants (other than $\neg$) this is trivial.
- If $M$ is a symbol $R \in \Sigma' \setminus \Sigma$ then $T^H_{\parallel}(B)[R](\alpha) = B\parallel[\lambda x. R \parallel F_R](\alpha)$.
- Next, if $M$ is an application $M_1 M_2$, $M_1 \rightarrow_{\parallel} N_1$ and $M_2 \rightarrow_{\parallel} N_2$ then
  \[
  T^H_{\parallel}(B)[M_1 M_2](\alpha) = T^H_{\parallel}(B)[M_1](\alpha) T^H_{\parallel}(B)[M_2](\alpha)
  = B\parallel[N_1](\alpha) (B\parallel[N_2](\alpha)) \quad \text{inductive hypothesis}
  = B\parallel[N_1 N_2](\alpha).
  \]
- Finally, if $M$ is a $\lambda$-abstraction $\lambda x. M'$ and $M' \rightarrow_{\parallel} N'$ then
  \[
  T^H_{\parallel}(B)[\lambda x. M'](\alpha) = \lambda r \in \mathcal{H}\parallel[\Delta(x)]. T^H_{\parallel}(B)[M'](\alpha[x \rightarrow r]) \quad \mathcal{H} \text{ is Henkin frame}
  = \lambda r \in \mathcal{H}\parallel[\Delta(x)]. B\parallel[N'](\alpha[x \rightarrow r]) \quad \mathcal{H} \text{ is Henkin frame}
  = B\parallel[\lambda x. N'](\alpha).
  \]

\[\square\]

Nonetheless, there is a mismatch between $\rightarrow_{\parallel}$ and the resolution proof system in three respects:
(M1) →∥ replaces potentially many relational symbols by their “definitions” whilst the resolution proof systems just considers one at a time.

(M2) →∥ also acts “inside” atoms whilst the resolution proof system only takes relational symbols at the “leftmost” position of atoms into account.

(M3) →∥ is a relation on positive existential terms whilst the resolution proof system operates on clauses.

To remedy Mismatch (M1), let $υ = \{(R, \lambda x. F_R) \mid R ∈ Σ’ \setminus Σ\}$ and $βv = β ∪ v$. Besides, let $→_v$ and $→_βv$ be the compatible closures of $v$ and $βv$, respectively.

Example 3.38. Consider a program $P ⊇ \{¬(R_1 x R_2 x R) ∨ Rx R, ¬(∃y, y = y) ∨ U\}$ and the positive existential formula $RU$. Then $RU →∥ (λx R_1 x R_2 x R)(∃y, y = y)$ and

$$RU →_v R(∃y, y = y)$$
$$→_v (λx R_1 x R_2 x R)(∃y, y = y)$$
$$→_β R_1 (∃y, y = y) ∧ R_2 (∃y, y = y).$$

The following lemma states (amongst other properties) that this definition really does address Mismatch (M1) and its proof can be found in Appendix A.3.

Lemma 3.39 (Basic properties of $→_βv$). Suppose $M →_βv N$. Then

(i) $→_∥ ⊆ →_βv$.
(ii) $\text{free}(N) ⊆ \text{free}(M)$;
(iii) if $M$ is semi-normal then $N$ is semi-normal, too.

Lemma 3.40 (Subject Reduction). Let $Δ ⊨ M : σ$ be a term such that $M →_βv N$. Then

(i) $Δ ⊨ N : σ$ and
(ii) $σ$ is a relational type.

3.3.2 Leftmost Reduction

In this subsection, we deal with Mismatch (M2). We adapt the proof of the standardisation theorem in the λ-calculus as presented in [Kashima, 2000] to prove that if $M βv$-reduces to $N$ then there exists a $βv$-reduction sequence that first only applies to redexes in “leftmost positions” and then only in “non-leftmost positions”. Later (Propositions 3.49 and 3.50), we will prove that the latter reductions are not necessary to establish undecidability.
The notion of leftmost reduction is made precise by the following inductively defined relation:

\[
\begin{align*}
M &\xrightarrow{0 \ell} M \\
M_1 &\xrightarrow{m_1 \ell} N_1 \\
M_2 &\xrightarrow{m_2 \ell} N_2 \\
M_1 \circ M_2 &\xrightarrow{m_1 + m_2 \ell} N_1 \circ N_2
\end{align*}
\]

and we write \( M \xrightarrow{\ell} N \) if \( M \xrightarrow{m \ell} N \) for some \( m \). Intuitively, it holds that \( M \xrightarrow{\ell} N \) if \( m \)-many \( \beta v \)-reductions have been performed. The (straightforward inductive) proofs of the following basic properties of \( \xrightarrow{\ell} \) are given in Appendix A.4.

**Lemma 3.41** (Basic Properties of \( \xrightarrow{\ell} \)). Let \( L, M, N \) and \( Q \) be terms. Then

(i) \( \xrightarrow{\ell} \) is reflexive and transitive;

(ii) \( \xrightarrow{\ell} \subseteq \xrightarrow{\beta v} \).

(iii) If \( M \xrightarrow{0 \ell} N \) then \( M = N \);

(iv) If \( L \xrightarrow{m+1 \ell} N \) then there exists \( M \) satisfying \( L \xrightarrow{1 \ell} M \xrightarrow{m \ell} N \);

(v) If \( M \xrightarrow{m \ell} N \) and \( M \xrightarrow{m \ell} N \) then \( M \xrightarrow{m \ell} N Q \);

(vi) If \( M[Q/z] \) is well-typed and \( M \xrightarrow{\ell} N \) then \( M[Q/z] \xrightarrow{\ell} N[Q/z] \).

Besides, the following Inversion Lemma is immediate by definition.

**Lemma 3.42** (Inversion). (i) If \( \exists \bar{x}. E \xrightarrow{m \ell} F \) then there exists \( F' \) such that \( F' = \exists \bar{x}. F' \) and \( E \xrightarrow{m \ell} F' \).

(ii) If \( E_1 \circ \cdots \circ E_n \xrightarrow{m \ell} F \), where \( \circ \in \{\land, \lor\} \), then there exist \( F_1, \ldots, F_n \) and \( m_1, \ldots, m_n \) satisfying \( F = F_1 \circ \cdots \circ F_n \), \( m = \sum_{i=1}^{n} m_i \) and \( E_j \xrightarrow{m_j \ell} F_j \) for each \( 1 \leq j \leq n \).

(iii) If \( \exists \bar{x}. A_1 \land \cdots \land A_n \xrightarrow{m \ell} F \) then there exist \( F_1, \ldots, F_n \) and \( m_1, \ldots, m_n \) satisfying

\[
F = \exists \bar{x}. F_1 \land \cdots \land F_n, \quad m = \sum_{i=1}^{n} m_i \quad \text{and} \quad A_j \xrightarrow{m_j \ell} F_j \quad \text{for each} \quad 1 \leq j \leq n.
\]

(iv) If \( (\lambda x. K)L \xrightarrow{\ell} N \) then \( N = K[L/x]M \).

However, \( \xrightarrow{\ell} \) alone does not seem to solve Mismatch (M2) because a term of the form \( y R \) can be reduced by \( \xrightarrow{\|} \) but not by \( \xrightarrow{\ell} \). Hence, we also have to do \( \beta v \)-reductions at “non-leftmost positions” to mimic the effect of \( \xrightarrow{\|} \). Therefore, we define

\[
\begin{align*}
M' &\xrightarrow{\beta v} N' \\
\overline{M} &\xrightarrow{\ell} N \quad L \xrightarrow{\ell} cM, c \in \Sigma' \cup \{\land, \lor, \exists \bar{r}\} \\
\overline{M} &\xrightarrow{\ell} xN \quad L \xrightarrow{\ell} xM
\end{align*}
\]

\[
\begin{align*}
M' &\xrightarrow{\ell} N' \\
\overline{M} &\xrightarrow{\ell} N \quad L \xrightarrow{\ell} (\lambda x. N')M
\end{align*}
\]
By $L \rightarrow M$ we mean $L_j \rightarrow M_j$ for each $1 \leq j \leq n$ assuming $L = (L_1, \ldots, L_n)$ and $M = (M_1, \ldots, M_n)$, and similarly for $M \rightarrow N$.

The idea is that $L \rightarrow N$ if for some $M$, $L \rightarrow M$ and we can obtain $N$ from $M$ by performing $\rightarrow \beta_v$-reductions on “non-leftmost positions”. The proof of the following basic properties of $\rightarrow_\beta$ can be found in Appendix A.4.

Lemma 3.43 (Basic Properties of $\rightarrow_\beta$). (i) $\rightarrow_\beta$ is reflexive (on positive existential terms).

(ii) $\rightarrow_\beta \subseteq \rightarrow_{\beta_v}$.

(iii) If $L \rightarrow_\beta N$ and $\exists \rightarrow_\beta Q$ then $L \exists \rightarrow_\beta N \exists Q$.

(iv) If $K \rightarrow_\beta L \rightarrow_\beta N$ then $K \rightarrow_\beta N$.

(v) If $L \rightarrow_\beta N$ and $O \rightarrow_\beta Q$ then $L [O/z] \rightarrow_\beta N [Q/z]$.

Lemma 3.44 (Inversion). Let $E$ be a semi-normal formula.

(i) If $E \rightarrow_\beta xN$ then there exists $\overline{M}$ such that $E \rightarrow_\ell x \overline{M}$.

(ii) If $E \rightarrow_\beta cN$, where $c \in \Sigma' \cup \{\land, \lor, \exists_r\}$, then there exists $\overline{M}$ such that $E \rightarrow_\ell c \overline{M}$ and $\overline{M} \rightarrow_\beta N$.

(iii) If $E \rightarrow_\beta \exists N$ then there exist $x$, $N'$ and $M$ such that $N = (\lambda x. N')$, $E \rightarrow_\ell \exists x. M$ and $M \rightarrow_\beta N'$.

Indeed, it turns out that $M \rightarrow_{\beta_v} N$ implies $M \rightarrow_\beta N$ (Corollary 3.47), and in particular $M \rightarrow_{\beta_v} N$ implies $M \rightarrow_\beta N$. Before proving this formally, we consider an example to get some intuition why this holds:

Example 3.45. Consider again the program from Example 3.38 and recall the $\beta_v$-reduction sequence

$$RU \rightarrow_\beta R (\exists y. y = y)$$

$$\rightarrow_\beta (\lambda xR. R_1 x, R_2 x) (\exists y. y = y)$$

$$\rightarrow_\beta R_1 (\exists y. y = y) \land R_2 (\exists y. y = y).$$

It also holds that

$$RU \rightarrow_\beta (\lambda xR. R_1 x, R_2 x) U \rightarrow_\beta R_1 U \land R_2 U.$$  

Furthermore, $R_j U \rightarrow_\beta R_j (\exists y. y = y)$ for $j \in \{1, 2\}$ because $U \rightarrow_\beta (\exists y. y = y)$ and hence, $U \rightarrow_\beta (\exists y. y = y)$. Consequently, by definition of $\rightarrow_\beta$, $RU \rightarrow_\beta R_1 (\exists y. y = y) \land R_2 (\exists y. y = y)$.

Note however, that the “shape” of the reduction sequence has (necessarily) changed.

Proposition 3.46. If $K \rightarrow_\beta M \rightarrow_{\beta_v} N$ then $K \rightarrow_\beta N$.

Proof. We prove the lemma by induction on $K \rightarrow_\beta M$.

- First, suppose $K \rightarrow_\beta x M_1 \cdots M_n$ because for some $L_1, \ldots, L_n$, $K \rightarrow_\beta x L_1 \cdots L_n$ and $L_i \rightarrow_\beta M_i$ for each $i$. Clearly, $x M_1 \cdots M_n \rightarrow_{\beta_v} x N_1 \cdots N_n$, because of $M_j \rightarrow_{\beta_v} N_j$ for some $j$ and $M_i = N_i$ for $i \neq j$ are the only possible $\beta_v$-reductions. By the inductive hypothesis, $L_j \rightarrow_\beta N_j$ and therefore by definition, $L \rightarrow_\beta x N_1 \cdots N_n$. 


Next, suppose $K \rightarrow c\overline{M}$ because for some $L$, $K \rightarrow c\overline{L}$ and $L \rightarrow \overline{M}$. If $c = R \in \Sigma' \setminus \Sigma$ and $R\overline{M} \rightarrow_{\beta_v} (\lambda x.R.F_R)\overline{M}$ then $K \rightarrow R\overline{L} \frac{1}{\ell} (\lambda x.R.F_R)\overline{L}$. Therefore, by reflexivity of $\rightarrow$ (Lemma 3.43(i)), $K \rightarrow (\lambda x.R.F_R)\overline{M}$.

Otherwise, $\overline{M}$ is reduced and the argument is analogous to the case for $K \rightarrow x\overline{M}$.

Finally, suppose $K \rightarrow (\lambda x.M')\overline{M}$ because for some $L'$ and $\overline{L}$, $K \rightarrow (\lambda x.L')\overline{L}$, $L' \rightarrow M'$ and $\overline{L} \rightarrow \overline{M}$. Let $\overline{L} = (L_1, \ldots, L_n)$ and $\overline{M} = (M_1, \ldots, M_n)$.

First, suppose $(\lambda x.M')\overline{M} \rightarrow_{\beta_v} (\lambda x.N')\overline{M}$, where $M' \rightarrow_{\beta_v} N'$. By the inductive hypothesis, $L' \rightarrow N'$. Therefore, by definition, $(\lambda x.K')\overline{K} \rightarrow (\lambda x.N')\overline{M}$.

The argument for the case $(\lambda x.M')M_1 \cdots M_n \rightarrow_{\beta_v} (\lambda x.M')N_1 \cdots N_n$, where for some $j$, $M_j \rightarrow_{\beta_v} N_j$ and $M_k = N_k$ for all $k \neq j$, is very similar.

Finally, assume that $n \geq 1$ and $(\lambda x.M')M_1 \cdots M_n \rightarrow_{\beta_v} M'[M_1/x]M_2 \cdots M_n$. Then

$$L \rightarrow (\lambda x.L')L_1 \cdots L_n \frac{1}{\ell} L'[L_1/x]L_2 \cdots L_n \rightarrow M'[M_1/x]M_2 \cdots M_n,$$

which proves $L \rightarrow M'[M_1/x]M_2 \cdots M_n$ by Lemmas 3.41(i) and 3.43(iv).

Consequently, by Proposition 3.46 and Lemma 3.43(i) we obtain

**Corollary 3.47.** Let $M$ and $N$ be positive existential terms such that $M \rightarrow_{\beta_v} N$. Then $M \not\rightarrow N$.

So far, we have established that if $S$ is $(\mathcal{A}, \mathcal{H})$-unsatisfiable then $F(S) \rightarrow F'$ for some $F'$ and $\mathcal{A}_0^H \models F'$. Next, we prove that it is in fact sufficient to perform the corresponding leftmost reductions (which correspond to applications of the Resolution and $\beta$-Reduction rules) ending up with a formula $E$ such that $\mathcal{A}_0^H \models E$ holds “obviously”. To formalise the latter intuition we define a relation $\alpha \triangleright E$ inductively by:

$$
\begin{align*}
\alpha \triangleright x\overline{M} & \quad \frac{\alpha \triangleright c\overline{M}}{\alpha \triangleright \overline{A}, \alpha \models c\overline{M}}, & c \in \Sigma & \frac{\alpha[x \mapsto r] \triangleright M}{\alpha \triangleright \exists x.\overline{M}} & r \in \mathcal{H}[r] \\
\alpha \triangleright M_1 & \quad \frac{\alpha \triangleright M_2}{\alpha \triangleright M_1 \lor M_2} & \frac{\alpha \triangleright M_2}{\alpha \triangleright M_1 \lor M_2} & \frac{\alpha \triangleright M_1 \alpha \triangleright M_2}{\alpha \triangleright M_1 \land M_2}
\end{align*}
$$

and write $\triangleright F$ if for some valuation $\alpha$, $\alpha \triangleright F$. Note that by Remark 1.2, the condition for the case $\alpha \triangleright c\overline{M}$ is well-defined, i.e. $c\overline{M}$ is a $\Sigma$-formula.

The proof of the following is an induction on $\alpha \triangleright F$ and can be found in Appendix A.4.

**Lemma 3.48.** Let $\alpha, \alpha'$ be valuations and $F$ be positive existential formulas satisfying $\alpha \triangleright F$. If $\alpha(x) = \alpha'(x)$ for all $x \in \text{free}(F)$ then $\alpha' \triangleright F$.

Now, we prove that leftmost reductions are sufficient for our purposes:

**Proposition 3.49.** Let $E, F$ be positive existential formulas and $\alpha$ be a valuation such that $F$ is in $\beta$-normal form, $\mathcal{A}_0^H, \alpha \models F$ and $E \overset{\beta_v}{\rightarrow} F$.

Then there exists a positive existential formula $F'$ satisfying $E \overset{\beta_v}{\rightarrow} F'$ and $\alpha \triangleright F'$.
Proof. We prove the lemma by induction on the structure of $F$. Note that the case $L \not\vdash (\lambda x. N') N$ cannot occur for otherwise $F$ is not in $\beta$-normal form or does not have type $\alpha$. If $F$ has the form $x N$ then by the Inversion Lemma 3.44 there exists $\overline{M}$ such that $E \vdash x \overline{M}$, and clearly, $\alpha \triangleright x \overline{M}$.

Hence, the only remaining case is that $F$ has the form $c N$. By the Inversion Lemma 3.44 there exist $\overline{M}$ such that $E \vdash c \overline{M}$ and $\overline{M} \not\sigma N$. Note that $c \in \Sigma'$ implies $c : \ell^n \rightarrow o \in \Sigma$ for otherwise $A^H_0, \alpha \models c N$. By Lemmas 3.40(ii) and 3.43(ii), $M = N$ and thus $A^H_0, \alpha \models c \overline{M}$. Consequently, $\alpha \triangleright c \overline{M}$.

Next, suppose that $c$ is $\land$. Then $F$ is $N_1 \land N_2$ and $c \overline{M}$ has the form $M_1 \land M_2$. By Lemma 3.41(ii) and the Subject Reduction Lemma 3.40, $M_j$ is a positive existential formula and clearly, by assumption, $N_j$ is in $\beta$-normal form and $A^H_0, \alpha \models N_j$ for all $j \in \{1, 2\}$. By the inductive hypothesis, there are $N'_1$ and $N'_2$ satisfying $M_j \triangleright N'_j$ and $\alpha \triangleright N'_j$. Consequently, $\alpha \triangleright N'_1 \land N'_2$ and by definition, $E \vdash \ell c N'_1 \land N'_2$.

The case where $c$ is $\lor$ is very similar.

Finally, suppose that $c$ is $\exists r$ and that $F$ is $\exists r N_1$. By the Inversion Lemma 3.44 there exist $x, N'$ and $M$ such that $N_1 = (\lambda x. N')$, $E \vdash \exists r x$. $M$ and $M \not\sigma N'$. By Lemma 3.41(ii) and the Subject Reduction Lemma 3.40, $M$ is a positive existential formula and clearly $N'$ is in $\beta$-normal form and $A^H_0, \alpha \models N'$ for some $r \in H[\tau]$. By the inductive hypothesis, there exists $N''$ satisfying $M \triangleright N''$ and $\alpha[x \mapsto r] \triangleright N''$. Consequently, $\alpha \triangleright \exists r x. N''$ and by definition, $E \vdash \ell c N''$. \qed

3.3.3 Refutational Completeness

Finally, we remedy Mismatch (M3) by establishing a connection between the (abstract) $\rightarrow$-relation on positive existential terms and the resolution proof system on goal clauses. For a set $S' \supseteq S$ of HoCHCs we define a measure $\mu(S')$ by

$$\mu(S') = \min(\{\omega \cup \{m | G \in S, \text{exists } F \text{ s.t. } \text{posex}(G) \xrightarrow{\ell} F \text{ and } \triangleright F\})$$

We show that we can use the resolution proof system to derive a set of HoCHCs $S''$ with a strictly smaller measure by mimicking a $\rightarrow$-reduction step:

**Proposition 3.50.** Let $S' \supseteq S$ be a set of HoCHCs satisfying $0 < \mu(S') < \omega$.

Then there exists $S'' \supseteq S$ satisfying $S' \Rightarrow_{\text{Res}} S''$ and $\mu(S'') < \mu(S')$.

**Proof.** Let $G \in S$ be goal clause, $F$ be a (closed) positive existential formula and let $m = \mu(G) > 0$ be such that $\text{posex}(G) \xrightarrow{\ell} F$ and $\triangleright F$. Without loss of generality we can assume that

$$\text{free}(G) \cap \text{free}(C) = \emptyset$$

for all $C \in S$. \hspace{1cm} (3.3)

(Otherwise, rename all variables occurring in $G$ to obtain $\tilde{G}$ satisfying Eq. (3.3) and clearly, by definition of $\Rightarrow_{\text{Res}}$, $S \cup \{\tilde{G}\} \Rightarrow_{\text{Res}} S \cup \{G', G'\}$ implies $S \Rightarrow_{\text{Res}} S \cup \{G', G'\}$.)

Furthermore, suppose that $G = \neg A_1 \lor \cdots \lor \neg A_n$ and $\text{posex}(G) = \exists \tau. \bigwedge_{i=1}^n A_i$. By the Inversion Lemma 3.42, there exist $F_1, \ldots, F_n$ and $m_1, \ldots, m_n$ such that $F = \exists \tau. \bigwedge_{i=1}^n F_i$. 

Figure 3.2: Overview of Part (ii) of the proof of Proposition 3.50. The ingredients for $G', F'$ and $m'$ are highlighted in bold and red for the case $\alpha[\bar{y}_1 \mapsto \bar{v}] \triangleright F'_1$.

$m = \sum_{i=1}^{n} m_i$ and $A_j \xrightarrow{m_j} F_j$ for each $1 \leq j \leq m$. Note that due to $\triangleright F$ there exists a valuation $\alpha$ such that $\alpha \triangleright F_j$ for each $1 \leq j \leq n$.

We can assume without loss of generality that $m_1 > 0$. By Lemma 3.41(iv), there exists $E$ such that $A_1 \xrightarrow{1 \frac{m_1}{\ell}} E \xrightarrow{m_1-1} F_1$. Since $A_1$ is an atom there are exactly two cases:

(i) $A_1 = (\lambda y. L)M \overline{N}$ and $E = L[M/y]\overline{N}$ or

(ii) $A_1 = R \overline{M}$ and $E = (\lambda x. R. F_R) \overline{M}$.

The first case is easy because for $G' = \lnot L[M/y]\overline{N} \lor \bigvee_{i=2}^{n} \lnot A_i, S \cup \{G\} \Rightarrow \text{Res} \cup \{G, G'\}$ and $\text{posex}(G') = \exists x. L[M/y]\overline{N} \land \bigwedge_{i=2}^{n} A_i$.

In the second case, note that $1 \frac{m_1}{\ell}$ is functional on applied $\lambda$-abstractions (by the Inversion Lemma 3.42). Hence, we can assume that

$$(\lambda x. R. F_R)M \xrightarrow{\ell} F_R[M/x] \xrightarrow{m_1} F_1,$$

where $m_1^* \leq m_1 - 1$ for otherwise $\alpha \triangleright F_1$ would clearly not hold.

$F_R$ has the form $\text{posex}(G_{R,1}, \overline{x}) \lor \ldots \lor \text{posex}(G_{R,k}, \overline{x})$, where each $G_{R,j}$ is a goal clause and $G_{R,j} \lor R \overline{x} \in S$. Let $\overline{y}_1, \ldots, \overline{y}_k$ and $E'_1, \ldots, E'_k$ be such that for each $j$, $\text{posex}(G_{R,j}, \overline{x}) = \exists \overline{y}_j. E'_j$. Note that by Eq. (3.3), $\text{posex}(G_{R,j}, \overline{x})[\overline{M}/\overline{x}] = \exists \overline{y}_j. E'_j[\overline{M}/\overline{x}]$ for each $j$ and by the Inversion Lemma 3.42, there exist $F'_1, \ldots, F'_k$ and $m'_1, \ldots, m'_k$ such that $F_1 = \bigvee_{j=1}^{k} (\exists \overline{y}_j. F'_j), E'_j[\overline{M}/\overline{x}] \xrightarrow{m'_j} F'_j$ and $m'_j \leq m_1^*$ for each $j$.

Next, because of $\alpha \triangleright F_1$ there exists $1 \leq j \leq k$ and $\bar{v} \in \mathcal{H}[\Delta(\overline{y}_j)]$ satisfying $\alpha[\overline{y}_j \mapsto \bar{v}] \triangleright F'_j$. Furthermore, because of Eq. (3.3) and Lemmas 3.39(ii) and 3.48, $\alpha[\overline{y}_j \mapsto \bar{v}] \triangleright F_i$ for all $2 \leq i \leq n$. Therefore,

$$\alpha[\overline{y}_j \mapsto \bar{v}] \triangleright F'_j \land \bigwedge_{i=2}^{n} F_i. \quad (3.4)$$
3.3. REFUTATIONAL COMPLETENESS OF THE PROOF SYSTEM

Clearly, it holds that

\[
S \cup \{G\} \Rightarrow Res S \cup \left\{ G, G_{R,j}[M/\pi_R] \vee \bigvee_{i=2}^{n} \neg A_i \right\},
\]

(3.5)

\[
E'_j[M/\pi_R] \land \bigwedge_{i=2}^{n} A_2^{m_j+\sum_{i=2}^{n} m_i} F'_j \land \bigwedge_{i=2}^{n} F_i
\]

(3.6)

and free(G') ⊆ \pi ∪ \gamma. Let \pi' ⊆ \pi and \gamma' ⊆ \gamma be such that free(G') = \pi' ∪ \gamma'. Hence, posex(G') = \exists x', y'. E'_j[M/\pi_R] \land \bigwedge_{i=2}^{n} A^{m_j} F'_j \land \bigwedge_{i=2}^{n} F_i

By Eqs. (3.4) to (3.6), it holds that (i) \( S \Rightarrow Res S \cup \{G'\} \), (ii) posex(G') \( \Rightarrow F' \), (iii) \( \Rightarrow F' \) and (iv) \( m' \leq m^{*} + \sum_{i=2}^{n} m_i < \sum_{i=1}^{n} m_i = m \). Consequently, also \( \mu(S\cup\{G'\}) < \mu(S) \)

By the definition of \( \alpha \Rightarrow F \), it is very easy to refute sets of HoCHCs with measure 0. Therefore, we finally get:

**Theorem 3.7** (Completeness). Let \( S \) be a \((A, H)\)-unsatisfiable set of HoCHCs.

Then \( S \Rightarrow^* \{\bot\} = S' \) for some \( S' \).

**Proof.** By Lemma 2.9(i) and Proposition 3.10(ii), \( A^H_P \models D \) for all definite clauses \( D \in S \). Since \( S \) is \((A, H)\)-unsatisfiable there exists a goal clause \( G \in S \) satisfying \( A^H_P \not\models G \). By Theorem 3.36 there exists \( n \in \omega \) such that \( A^H_n \not\models G \). Let \( F_n \) be such that posex(G) \( \Rightarrow F_n \). By Proposition 3.37, \( A^H_n \models F_n \). Let \( F'_n \) be the \( \beta \)-normal form of \( F_n \). By Lemma 3.39(i), Corollary 3.47, and Proposition 3.49 there exists \( F' \) such that posex(G) \( \Rightarrow F' \) and \( \Rightarrow F' \). Consequently, \( \mu(S) < \omega \). By Proposition 3.50 there exists \( S' \supseteq S \) satisfying \( S \Rightarrow^{*} S' \) and \( \mu(S') = 0 \).

Hence, there exists \( G \in S' \) such that posex(G) \( \Rightarrow F' \) and \( \Rightarrow posex(F') \). Besides, by Lemma 3.41(iii), \( F' = posex(G) \) and therefore \( G \) has the form \( G' \lor \phi_1 \lor \cdots \lor \phi_n \), where \( G' \) is simple and the \( \phi_i \) are first-order such that there exists a valuation \( \alpha \) satisfying \( A, \alpha \models \phi_1 \land \cdots \land \phi_n \). Therefore, the rule Constraint Refutation is applicable to \( G \) and hence, \( S \Rightarrow^{*} S' \Rightarrow^{*} \{\bot\} = S' \).

\( \square \)
Chapter 4

Ramifications

In this chapter, we apply the results of the previous chapter to obtain some interesting corollaries. In Section 4.1, we prove that the satisfiability of higher-order constrained Horn clauses is independent of the choice of a Henkin frame. In particular, this re-proves the equivalence of standard, monotone and continuous semantics for HoCHCs.

Section 4.2 demonstrates that every program can indeed be transformed into our normal form and that λ-abstractions can be eliminated.

These insights are used in Section 4.3, where we relate our efforts to work on higher-order logic programming and show that we can also deal with some background theories with an infinite number of models.

Finally, Section 4.4 describes a reduction of HoCHCs to first-order logic with background theories. Fig. 4.1 summarises some of the results of this chapter.

4.1 Equivalence of Semantics

This section proves that the satisfiability problem for higher-order constrained Horn clauses coincides for all Henkin frames, which gives rise to an alternative proof of the equivalence of standard, monotone and continuous semantics for HoCHCs [Cathcart Burn et al., 2018, Jochems, 2018].

Theorem 4.1. Let $S$ be a set of HoCHCs and let $H$ and $H'$ be Henkin frames such that $S$ is $(\mathcal{A}, \mathcal{H})$-unsatisfiable. Then $S$ is also $(\mathcal{A}, \mathcal{H'})$-unsatisfiable.

Proof. By the Completeness Theorem 3.7, $S \Rightarrow_{\text{Res}} \{ \bot \} \cup S'$ for some $S'$ and by soundness of resolution (Corollary 3.6), $S$ is $(\mathcal{A}, \mathcal{H'})$-unsatisfiable. 

Next, note that $\subseteq_c$ and $\sqsubseteq$ coincide for the continuous frame. The straightforward proof can be found in Appendix B.1.

Lemma 4.2. $\sqsubseteq_c = \sqsubseteq^c$.

The proof of the following makes use of Lemma 4.2 and differs from the proof of Proposition 3.30 only in the case of λ-abstractions. The details can be found in Appendix B.1.
Lemma 4.3. Let $\Sigma$ be a signature, $\Delta$ be a type environment and $B$ be a $(\Sigma, C)$-structure.
Then for any positive existential term $M$, $(\Delta, C)$-valuation $\alpha$ and $A' \in \text{dir}(\alpha)$,

(i) if $M$ is a $\lambda$-abstraction then $B^C(M)(\alpha) = B^C[M](\alpha)$ and
(ii) $B^C[M](\biguplus A) \subseteq \biguplus \{B^C[M](\alpha) \mid \alpha \in A\}.$

Similarly, we get the following for the monotone frame

Lemma 4.4. (i) $\subseteq_m^M = \subseteq^M$;
(ii) If $\Sigma$ is a signature, $\Delta$ is a type environment, $A$ is a $(\Sigma, M)$-structure, $\alpha$ is a
$(\Delta, M)$-valuation, $\lambda x. M$ is a positive existential $\Sigma$-term then

$A^M(M)(\alpha) = A^M[M](\alpha).$

Lemmas 4.3 and 4.4 immediately imply:

Proposition 4.5. $S$, $M$ and $C$ are Henkin frames.

Finally, by Theorem 4.1 and Proposition 4.5, we get the main result of this section:

Theorem 4.6 (Equivalence of Semantics). Let $S$ be a set of HoCHCs. Then the following
are equivalent:

(i) $S$ is $A$-standard-satisfiable,
(ii) $S$ is $A$-Henkin-satisfiable,
(iii) $S$ is $A$-monotone-satisfiable,
(iv) $S$ is $A$-continuous-satisfiable,
(v) $S$ is $(A, H)$-satisfiable, where $H$ is a Henkin frame.

We call a set of $S$ of HoCHCs $A$-satisfiable if it satisfies any of the conditions of the
preceding theorem.
4.2 Elimination of Nested Terms

In this section, we show how to eliminate terms which are nested to a certain extend. We use this insight for a normal form transformation and to remove $\lambda$-abstractions.

Consider for example the rather peculiar formula $R \lor$, where $R : (o \rightarrow o \rightarrow o) \rightarrow o$. In full higher-order relational logic we could introduce a new symbol $U : o \rightarrow o \rightarrow o$, add the “definition” $U \equiv (x \lor y)$ and replace $R \lor$ with $RU$. These definitions cannot be transformed into HoCHCs because the “$\rightarrow$”-direction $\neg U \lor (x \lor y)$ clearly does not correspond to a HoCHC. Rather, we use ideas inspired by [Plaisted and Greenbaum, 1986] to show that adding the unproblematic direction $\neg x \lor U \lor (x \lor y)$ is sufficient. However, we have to greatly generalise their approach, which is for formulas of first-order logic, to arbitrary relational, positive existential higher-order terms.

Let $M$ be a positive existential $\Sigma$-term with free variables $\pi$ such that $\Delta \vdash M : \tau \rightarrow o$, let $P[-]$ be a set of terms with a hole of type $\tau \rightarrow o$ such that $P[M]$ is a program and let $F[-]$ be a term with a hole of type $\tau \rightarrow o$ such that $F[M]$ is a positive existential formula.

Assume that $\Delta(\pi) = \pi'$ and let $\bar{y}$ be distinct variables (different from $\pi$) satisfying $\Delta(\bar{y}) = \pi$. We define a signature $\Sigma'' = \Sigma' \cup \{R_M : \tau' \rightarrow \tau \rightarrow o\}$ and

$$P = \tilde{P}[M] \quad \quad F = \tilde{F}[M]$$
$$P' = \tilde{P}[R_M \pi] \cup \{\neg M \bar{y} \lor R_M \pi \bar{y}\} \quad \quad F' = \tilde{F}[R_M \pi].$$

These are clearly programs and positive existential formulas, respectively. In the following we prove that $(P, F)$ and $(P', F')$ are in fact $(A, \mathcal{H})$-equivalent for every Henkin frame $\mathcal{H}$. First, note that the proof of Proposition 3.19 can be adapted in a straightforward manner to obtain:

Lemma 4.7. Let $B$ be a $\Sigma'$- and $B'$ be a $\Sigma''$-structure, let $M[-]$ be a $\Sigma'$-term with a hole of type $\tau$, let $N$ be a $\Sigma'$ and $N'$ be a $\Sigma''$-formula satisfying

(i) $M[N]$ is a positive existential formula,
(ii) $\Delta \vdash N : \tau$ and $\Delta \vdash N' : \tau$,
(iii) for all $R \in \Sigma' \setminus \Sigma$, $R^B \subseteq_m R^{B'}$,
(iv) for all $(\Delta, \mathcal{H})$-valuations $\alpha \subseteq_m \alpha'$, $B^{\mathcal{H}[N]}(\alpha) \subseteq_m B^{\mathcal{H}[N']}(\alpha')$.

Then for all $(\Delta, \mathcal{H})$-valuations $\alpha \subseteq_m \alpha'$, $B^{\mathcal{H}[M[N]]}(\alpha) \subseteq_m B^{\mathcal{H}[M[N']]}(\alpha')$.

In the following lemma we prove that although $A_{P',n}^{H}$ “grows slower” (with increasing $n \in \omega$) than $A_{P,n}^{H}$, $A_{P',2n+1}^{H}$ still roughly “catches up” with $A_{P,n}^{H}$.

Lemma 4.8. For every $n \in \omega$,

(i) $R^{A_{P,n}^{H}} \subseteq_m R^{A_{P',2n+1}^{H}}$ for $R \in \Sigma' \setminus \Sigma$,
(ii) $R^{A_{P,n}^{H}} \subseteq_m R^{A_{P',2n+1}^{H}}$ for $R \in \Sigma' \setminus \Sigma$ and
(iii) $A_{P,n}^{H}(\lambda x. M)[\pi]\subseteq_m R^{A_{P',2n+1}^{H}}(M)\subseteq_m R^{A_{P',2n+1}^{H}}(M)$.

It is proven in detail in Appendix B.2. As a consequence we get:
Proposition 4.9. Let $\mathcal{H}$ be a Henkin frame. Then $(P, F)$ is $(A, \mathcal{H})$-equivalent to $(P', F')$.

Proof. First, suppose that there exists a $(\Sigma', \mathcal{H})$-expansion $B$ of $A$ satisfying $B \models_{\mathcal{H}} P$ and $B \not\models F$. We define a $(\Sigma'', \mathcal{H})$-expansion $B'$ of $A$ by setting $R_{\Sigma''}^{B'} = R_{\Sigma}^{B}$ for $R \in \Sigma' \setminus \Sigma$ and $R_{\Delta}^{B'} = B''[\lambda \bar{x}, \bar{y}, M \bar{y}]((\top)^{\mathcal{H}}_{\Delta})$. By definition, $B' \models_{\mathcal{H}} \neg M \bar{y} \lor R_{\Delta} \bar{x} \bar{y}$. Furthermore, for every positive existential $\Sigma'$-formula $E$ and $(\Delta, \mathcal{H})$-valuation $\alpha$, by Lemma 1.11, $B'[E[M]](\alpha) = B'[E[R_{\Delta} \bar{x}]](\alpha)$. Consequently, $B' \models_{\mathcal{H}} P'$ and $B' \not\models F'$.

Conversely, suppose that there exists a $(\Sigma'', \mathcal{H})$-expansion $B'$ of $A$ satisfying $B' \models_{\mathcal{H}} P'$ and $B' \not\models F'$. Assume towards contradiction that $A_{\mathcal{H}}^{\delta} \models F$. Then, by Theorem 3.36, $A_{F,n}^{\mathcal{H}} \models F$ for some $n \in \omega$. By Lemmas 4.7 and 4.8, $A_{F',2(n+1)}^{\mathcal{H}} \models F'$ and therefore, by Proposition 3.19 and Theorem 3.23(ii), $A_{F'}^{\mathcal{H}} \models F'$. Note that by Theorem 3.23(iii), $A_{F'}^{\mathcal{H}} \subseteq m B'$ and therefore, by Proposition 3.19, $B' \models F'$, which is clearly a contradiction. Consequently, $A_{\mathcal{H}}^{\delta} \not\models F$ and by Proposition 3.10(ii), $A_{\mathcal{H}}^{\delta} \models_{\mathcal{H}} P$. \hfill $\square$

4.2.1 Normal Form Transformation

In this subsection, we use the insights just gained to show how to obtain a HoCHC in normal form which is equivalent to a given HoCHP. This closes the gap in the translation of HoCHPs to HoCHCs from Section 2.3.

There are two important tasks to accomplish: (i) eliminating terms of the form $R \bar{M}$, $x \bar{M}$ or $(\lambda \bar{x}. N) \bar{M}$, where $\bar{M}$ or $N$ contain logical symbols, (ii) establishing the logical structure $\lor \exists \lambda \land$.

To achieve the first goal we make more precise which terms we need to eliminate.

Definition 4.10. (i) A term $c \bar{M}$ occurs impure in a formula $N_1 \cdots N_n$ if (i) $c \bar{M}$ is a subterm of $N_1 \cdots N_n$, (ii) $c$ is a logical symbol and (iii) $N_1$ is a variable or $N_1$ is a $\lambda$-abstraction.

(ii) A term $M$ occurs impure in $(P, F)$ if there exists a term $N$ such that $M$ occurs impure in $N$ and $N$ is a subterm of a term in $P \cup \{F\}$.

(iii) $(P, F)$ is pure if there are no term which occurs impure in $(P, F)$.

Example 4.11. Consider $\exists (\lambda x. R(x = x) \lor R(x \leq x \land x \geq x))$. $(x \leq x \land x \geq x)$, which is an abbreviation for $\land (x \leq x \land x \geq x)$, occurs impure in $R(x \leq x \land x \geq x)$. $R(x = x) \lor R(x \leq x \land x \geq x)$ does not occur impure in $\lambda x. R(x = x) \lor R(x \leq x \land x \geq x)$ because the latter is no formula.

Clearly, there is no term which occurs impure in an atom.

Proposition 4.12. Let $\mathcal{H}$ be a Henkin frame and $(P, F)$ be a HoCHP.

Then there exists a $(A, \mathcal{H})$-equivalent HoCHP which is pure.

Proof. We prove the lemma by induction on the number $m$ of distinct terms $M$ which occur impure in $(P, F)$. If $m = 0$ then $(P, F)$ is clearly pure.

Otherwise there exists a term $M$ which (i) occurs impure in $(P, F)$ and (ii) is not a subterm of another term which occurs impure in $(P, F)$. Suppose $\Delta \vdash M : \rho$. By
Lemma 1.10 there are (sets of) contexts with a $\rho$-hole $\tilde{P}[-]$ and $\tilde{F}[-]$ such that $\tilde{P}[M] = P$, $\tilde{F}[M] = F$ and $M$ occurs neither in $\tilde{P}[-]$ nor in $\tilde{F}[-]$. Let $P' = \tilde{P}[R_M \overline{x}] \cup \{\neg M \overline{y} \lor R_M \overline{x} \overline{y}\}$ and $F' = \tilde{F}[R_M \overline{x}]$. $(P, F)$ and $(P', F')$ are $(\mathcal{A}, \mathcal{H})$-equivalent by Proposition 4.9.

Note that $M$ does not occur impure in $(P', F')$. Furthermore, by Item (ii), each term which contains impure in $(P', F')$ also occurs impure in $(P, F)$. Hence, the number of distinct terms which occur impure in $(P', F')$ is strictly smaller than $m$. Consequently, by the inductive hypothesis, $(P', F')$ is $(\mathcal{A}, \mathcal{H})$-equivalent to some pure $(P'', F'')$. \qed

Therefore, we can assume that $(P, F)$ is pure. Note that by Lemma 1.12(ii) we can furthermore assume that for all occurrences of a term $\exists M$ in $P \cup \{F\}$, $M$ is a $\lambda$-abstraction. Hence, each positive existential formula occurring in $P \cup \{F\}$ has the form

$$F_1 := A \mid F_1 \land F_1 \mid F_1 \lor F_1 \mid \exists x. F_1,$$

where $A$ is an atom and $x$ is a variable. Clearly, we can transform each such positive existential formula into normal form by exploiting the distributivity law and pushing the existential quantifier inside or outside exactly as for the DNF-transformation (or dually CNF-transformation) for first-order logic (see e.g. [Nonnengart and Weidenbach, 2001]). Hence, we get:

**Theorem 4.13.** Let $\mathcal{H}$ be a Henkin frame and $(P, F)$ be a HoCHP.

Then there exists a $(\mathcal{A}, \mathcal{H})$-equivalent HoCHP in normal form.

**Example 4.14.** Consider the program $\{\neg (\exists x. (R \lor (x \land (\lambda y. y \land y)y))) \lor R z\}$ and the positive existential formula $\exists x. x$. The elimination of impure terms results in the program

$$\{\neg (\exists x. R R \land (x \lor (\lambda y. R_{y/y} y)y))) \lor R z,\
\neg (x \lor y) \lor R_{x/y} x y,\
\neg (y \land y) \lor R_{y/y} y\}.$$ 

After establishing the required logical structure we get:

$$\{\neg ((\exists x. R R \lor x) \lor (\exists x. R R \land (\lambda y. R_{y/y} y)y))) \lor R z,\
\neg (x \lor y) \lor R_{x/y} x y,\
\neg (y \land y) \lor R_{y/y} y\}.$$ 

Note that using distributivity exhaustively may result in an exponential blow-up in the size of the HoCHP. To counteract this, we could introduce “renamings” exactly as in first-order logic, exploiting the insights gained in the previous section. However, this is not the focus of the present work and the interested reader should refer to e.g. [Plaisted and Greenbaum, 1986] or [Nonnengart and Weidenbach, 2001] for a good exposition.

### 4.2.2 Elimination of $\lambda$-Abstractions

Next, we show that for each set $S$ of HoCHCs there exists a set $S'$ of HoCHCs which does not contain $\lambda$-abstractions and which is $(\mathcal{A}, \mathcal{H})$-satisfiable if and only if $S$ is $(\mathcal{A}, \mathcal{H})$-satisfiable. We use the results from Section 4.2.1, which are stated for HoCHPs. Since HoCHPs may contain $\lambda$-abstractions (implicitly) below existential quantifiers, we first have to make precise when a HoCHC “essentially” does not contain $\lambda$-abstractions:
Definition 4.15. (i) A positive existential term \( M \) is \( \lambda \)-free if it has the form
\[
M_\lambda ::= x | c | \wedge | \vee | \exists \tau (\lambda x. M_\lambda) | M_\lambda M_\lambda
\]
(ii) A HoCHP \((P, F)\) is \( \lambda \)-free if all positive existential subformulas of formulas in \( P \cup \{F\} \) are \( \lambda \)-free.

Analogously to Proposition 4.12 we get (the details are presented in Appendix B.2):

Proposition 4.16. Let \( H \) be a Henkin frame and \((P, F)\) be a HoCHP in normal form.
Then there exists a \( \lambda \)-free HoCHP in normal form which is \((A, H)\)-equivalent to \((P, F)\).

Hence, using the translations from HoCHCs to HoCHP in normal form (Corollary 2.10), the elimination of \( \lambda \)-abstractions (Proposition 4.16) and the translation back to HoCHCs (Corollary 2.13) we get:

Theorem 4.17. Let \( H \) be a Henkin frame and \( S \) be a set of HoCHCs.
Then there exists a set of HoCHCs which does not contain \( \lambda \)-abstractions and which is \((A, H)\)-satisfiable if and only if \( S \) is \((A, H)\)-satisfiable.

Note that in the proof of Theorem 4.17 we eliminate more \( \lambda \)-abstractions than strictly necessary, some of which we have to reintroduce to obtain a normal form:

Example 4.18. Consider an arbitrary program \( P \) and the positive existential formula \( R (\lambda x. x) \lor \exists x. x \), which is an abbreviation for \( R (\lambda x. x) \lor \exists (\lambda x. x) \). The proof of Proposition 4.16 constructs a HoCHP \((P' \cup \{-x \lor R_{\lambda x.x} x\}, R R_{\lambda x.x} x \lor \exists R_{\lambda x.x})\). However, there is no need to eliminate the \( \lambda \)-abstraction in \( \exists (\lambda x. x) \) and we could have taken \((P' \cup \{-x \lor R_{\lambda x.x} x\}, R R_{\lambda x.x} x \lor \exists x. x)\), which is \( \lambda \)-free and in normal form, too.

4.3 Resolution for the Theory of Positive Equality and Uninterpreted Functions

In this section, we show how to use our framework to solve the problem considered by [Charalambidis et al., 2013] (cf. the discussion in Sections 2.5 and 5.1) with respect to arbitrary Henkin frames. In Section 4.2.1, we bridged the difference in terms of syntax. Therefore, it only remains to demonstrate how we can determine satisfiability of unconstrained Horn clauses.

Formally, let \( \Sigma' \) be a signature containing the equality symbol \( \approx : \tau \rightarrow \tau \rightarrow o \). Then, there exists a (unique) \( \Sigma \subseteq \Sigma' \) such that \( \approx \in \Sigma, \Sigma \setminus \{\approx\} \) contains only symbols of type \( \tau^n \rightarrow \tau \) and \( \Sigma' \setminus \Sigma \) is purely relational. Furthermore, let \( \mathcal{F} \) be a non-empty class of Henkin frames (e.g. \( \{C_D \mid D \text{ a set}\} \) or \( \{H \mid H \text{ is a Henkin frame}\} \)).

Definition 4.19. A set \( S \) of \((\Sigma, \Sigma')\)-HoCHC is \( \mathcal{F} \)-satisfiable if there exists \( H \in \mathcal{F} \) and a \((\Sigma', \mathcal{H})\)-structure \( B \) such that \( B \models_H S \) and \( \approx_B (n)(m) = 1 \) if and only if \( n = m \).

Proposition 4.20. Let \( S \) be a set of HoCHCs. \( S \) is \( \mathcal{F} \)-satisfiable if and only if there exists some \( H \in \mathcal{F} \) such that \( S \) is \((A_{\text{Her}}, \mathcal{H})\)-satisfiable.
4.4. APPLICATIVE ENCODING

Proof. If $S$ is $(\mathcal{A}_{\text{Her}}, \mathcal{H})$-satisfiable for some $\mathcal{H} \in \mathfrak{S}$ then $S$ is clearly $\mathfrak{S}$-satisfiable.

Conversely, suppose $S$ is $\mathfrak{S}$-satisfiable. Then, there exists $\mathcal{H} \in \mathfrak{S}$ and a $(\Sigma', \mathcal{H})$-structure $\mathcal{B}$ such that $\mathcal{B} \models_{\mathcal{H}} S$ and $\models^B (n)(m) = 1$ if and only if $n = m$. Assume towards contradiction that $S$ is not $(\mathcal{A}_{\text{Her}}, \mathcal{H})$-satisfiable. Then by the Completeness Theorem 3.7 there exists $S'$ and $G \lor \neg (M_1 \approx N_1) \land \cdots \land \neg (M_n \approx N_n)$ in $S'$ such that $\bot \notin S'$, $S \Rightarrow^\ast_{\text{Res}} S'$, $G$ is simple and there exists a valuation $\alpha$ such that $\mathcal{A}_{\text{Her}}, \alpha \models M_1 \approx N_1 \land \cdots \land M_n \approx N_n$.

Let $\alpha'(x) = \mathcal{B}^H[[\alpha(x)]](\top^H)$ (the valuation $\top^H$ is clearly not of importance). Note that for each $\Sigma$-term $L$,

$$\mathcal{B}^H[[L]](\alpha') = \mathcal{B}^H[[\mathcal{A}_{\text{Her}}[[L]](\alpha)]](\top^H).$$

Therefore, $\mathcal{B}, \alpha' \models M_1 \approx N_1 \land \cdots \land M_n \approx N_n$. This constitutes a contradiction because due to Corollary 3.6 and Lemma 3.4, $\mathcal{B}, \alpha' \models \neg (M_1 \approx N_1) \lor \cdots \lor \neg (M_n \approx N_n)$.

Consequently, by the Equivalence of Semantics Theorem 4.6, it suffices to check whether $S$ is $(\mathcal{A}_{\text{Her}}, \mathcal{H})$-satisfiable for a fixed $\mathcal{H} \in \mathfrak{S}$ using the resolution proof system. Clearly, this gives rise to a semi-decision procedure because checking whether there exists a valuation such that $\mathcal{A}_{\text{Her}}, \alpha \models M_1 \approx N_1 \land \cdots \land M_n \approx N_n$ just amounts to the problem of deciding whether $(M_1, \ldots, M_n)$ and $(N_1, \ldots, N_n)$ are unifiable.

4.4 Applicative Encoding

Finally, we present a way to reduce the $\mathcal{A}$-satisfiability problem for HoCHCs to the satisfiability problem of first-order logic modulo a theory by using an applicative encoding in the spirit of e.g. [Van Benthem and Doets, 1983, Kerber, 1991, Blanchette et al., 2016]. We prove that this translation is sound and complete, and demonstrate that the target logic still admits refutational complete proof systems. As a consequence of Theorem 4.17, we do not need to consider sets of HoCHCs containing $\lambda$-abstractions.

Clearly, the signature $\Sigma$ and the $\Sigma$-structure $\mathcal{A}$ can be regarded as a many-sorted signature and structure, respectively (with only one sort $\iota$). Let $[\Sigma']$ be the many-sorted first-order signature, which extends $\Sigma$ with

(i) the (foreground) sorts $[\rho]$ for relational $\rho$ (and we set $[\iota^n \rightarrow \iota] = [\iota^n \rightarrow \iota])$,

(ii) constants $c_R : [\rho]$ for $R : \rho \in \Sigma' \setminus \Sigma$ and $c_\rho : [\rho]$ for relational types $\rho$,

(iii) a binary function symbol $@_{\tau, \rho} : [\tau \rightarrow \rho] \rightarrow [\tau] \rightarrow [\rho]$ for each relational type $\tau \rightarrow \rho$ and

(iv) a monadic relation symbol $H : [o] \rightarrow o$.

To enhance readability we frequently omit the subscript from $@$ in what follows. The following observation is trivial:

Lemma 4.21. $(\Sigma, \{A\}, [\Sigma'])$ is a hierarchic specification and $[\Sigma']$ does not contain a function symbol $f : [\tau_1] \rightarrow \cdots \rightarrow [\tau_n] \rightarrow \iota$.

Let $[\Delta]$ be the type environment consisting of $x : [\tau]$ whenever $x : \tau \in \Delta$ and variables $x^{(i)}_{\tau_i}$ for relational $\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow o$. For a $\Sigma'$-term $M$ containing neither logical symbols nor $\lambda$-abstractions, we define $[M]$ by structural induction:

$$[x] = x.$$
Clearly, due to \( \alpha \)-valuation \( \rho \) and relational \( \tau \rightarrow \rho \), \( \rho \) is satisfiable. We proceed by introducing a proof system for the hierarchic

\[
| R | = c_R, \\
| c N | = c N, \\
[ M \overline{N} N' ] = @ [ M \overline{N} ] [ N' ],
\]

if \( R \in \Sigma' \setminus \Sigma \), if \( c \in \Sigma \), if \( M \not\in \Sigma \). Note that because of Remark 1.2, for each \( \Sigma' \)-term \( M \), \( [ M ] \) is a \( \Sigma' \)-term, and \( \Delta \vdash M : \sigma \) if and only if \( \Delta \vdash [ M ] : [ \sigma ] \). For higher-order constrained Horn clauses we define

\[
\neg A_1 \lor \cdots \lor \neg A_n = \neg [ A_1 ] \lor \cdots \lor \neg [ A_n ], \\
\neg A_1 \lor \cdots \lor \neg A_n \lor R R_R = \neg [ A_1 ] \lor \cdots \lor A_n \lor [ R R_R ].
\]

Furthermore, for relational \( \rho = \tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow o \), we define

\[
\text{Comp}_\rho = H ( ( \cdots ( ( @ @ c_\rho x^{(1)}_1 x^{(2)}_2 ) \cdots ) x^{(n)}_n )
\]

and for a set \( S \) of HoCHCs which does not contain \( \lambda \)-abstractions, let

\[
[S] = \{ [C] \mid C \in S \} \cup \{ \text{Comp}_\rho \mid \rho \in \Delta \text{ occurs in } S \}.
\]

Note that \( [ S ] \) is a set of \( \Sigma' \)-Horn clauses.

**Example 4.22.** Consider again the set \( S \) of HoCHCs from Example 2.6. Applying the encoding \([S]\) to \( S \) get:

\[
[D_1] = \neg(z = x + y) \lor H ( ( @ ( @ Add ) x ) y ) z \\
[D_2] = \neg(n \leq 0) \lor \neg(s = x) \lor H ( ( @ ( @ Iter f ) s ) n ) x \\
[D_3] = \neg(n > 0) \lor \neg H ( ( @ ( @ Iter f ) s ) (n - 1) ) y ) \lor \neg H ( ( @ ( @ f N ) y ) x ) \\
\lor H ( ( @ ( @ Iter f ) ) ( n ) n ) x \\
[G] = \neg(n \geq 1) \lor \neg H ( ( @ ( @ ( @ Add ) n ) n ) ) x \lor \neg(x \leq n + n) \\
\text{Comp}_{3 \rightarrow o} = H ( ( @ ( @ c_{3 \rightarrow o} x_1 ) x_2 ) x_3 )
\]

**Proposition 4.23 (Soundness of Encoding).** If \( S \) is satisfiable then \( [ S ] \) is satisfiable.

**Proof.** First suppose \( S \) is satisfiable. Then, there exists a \( ( \Sigma', S ) \)-expansion \( B \) of \( A \) satisfying \( B \models_S S \). We define a many-sorted first-order \( \Sigma' \)-expansion \( [ B ] \) of \( A \) by setting

1. \([ B ] [ [ \rho ] ] = H [ [ \rho ] ] \) for relational \( \rho \),
2. \([ R ] [ B ] = R B \) for \( R \in \Sigma' \setminus \Sigma \) and \([ c_p ] [ B ] = \tau_p \) for relational \( \rho \),
3. \( @_\tau c_\rho ( r ) = r ( s ) \) for relational \( \tau \rightarrow \rho \), \( r \in [ B ] [ [ \tau \rightarrow \rho ] ] \) and \( s \in [ B ] [ [ \tau ] ] \),
4. \( H [ [ B ] ] ( b ) = b \) for \( b \in [ [ B ] ] ( [ \theta ] ) = B \).

Clearly, \( B \models \) for each relational \( \rho \). Furthermore, for each \( \Sigma' \)-term \( M \), \( ( \Delta, H ) \)-valuation \( \alpha \) and \( \Delta \)-valuation \( [ \alpha ] \), \( B [ [ M ] ] ( [ \alpha ] ) = [ B ] [ [ [ M ] ] ([ [ \alpha ] ] ] \) if for all \( x \in \text{dom}(\Delta) \), \( \alpha(x) = [ \alpha ] ( x ) \). Furthermore, each \( C \in S \) only contains variables from \( \Delta \). Consequently, due to \( B \models_S S \), \( [ B ] \models [ S ] \).

Next, we prove that the encoding is in fact also complete, i.e. if \( S \) is unsatisfiable then also \( [ S ] \) is unsatisfiable. We proceed by introducing a proof system for the hierarchic
4.4. APPLICATIVE ENCODING

specification (similar to the one in [Bachmair et al., 1994]) and showing that it can simulate refutations of $S$ on $\lfloor S \rfloor$.

**FO-Resolution**

$$\frac{\neg H M \lor G}{(G \lor G') \theta} G' \lor H M'$$

provided $G$ is a unifier of $M$ and $M'$.

**FO-Constraint Refutation**

$$\frac{\neg \phi_1 \lor \cdots \lor \neg \phi_n}{\perp}$$

provided each $\phi_i$ is a $\Sigma$-term and there exists a valuation $\alpha$ such that $\mathcal{A}, \alpha \models \phi_1 \land \cdots \land \phi_n$.

Similarly as for $\Rightarrow_{\text{Res}}$, we write $S \Rightarrow_{\text{FO-Res}} S \cup \{C\}$ if $C$ can be derived from clauses in $S$ using any of the above two rules and $\Rightarrow_{\text{FO-Res}}^*$ for the reflexive, transitive closure of $\Rightarrow_{\text{FO-Res}}$. Variables are again silently renamed, where necessary.

**Lemma 4.24.** Let $B$ be a $[\Sigma']$-expansion of $A$ and suppose $S \Rightarrow_{\text{FO-Res}} S'$.

Then $B \models S$ implies $B \models S'$.

**Proof.** Soundness of first-order resolution is a classic result (e.g. [Robinson, 1965, Fitting, 1996]) and if there exists a clause $\neg \phi_1 \lor \cdots \lor \neg \phi_n$ in $S$ and a valuation $\alpha$ such that $\mathcal{A}, \alpha \models \phi_1 \land \cdots \land \phi_n$ then $S$ is clearly unsatisfiable. $\square$

**Corollary 4.25** (Soundness of $\Rightarrow_{\text{FO-Res}}^*$). Let $B$ be a $[\Sigma']$-expansion of $A$ and suppose $S \Rightarrow_{\text{FO-Res}}^* S' \cup \{\perp\}$. Then $S$ is unsatisfiable.

It turns out that we can mimic inferences from $\Rightarrow_{\text{Res}}$ in $\Rightarrow_{\text{FO-Res}}^*$:

**Lemma 4.26** (Lifting). Let $S$ be a set of HoCHCs not containing $\lambda$-abstractions.

If $S \Rightarrow_{\text{Res}} S \cup \{G\}$ and $S' \supseteq \lfloor S \rfloor$ then there exists $S''$ such that (i) $S' \Rightarrow_{\text{FO-Res}}^* S''$, (ii) $\lfloor G \rfloor \in S''$ and (iii) if $x : \rho$ occurs in $S''$ then some $y : \rho$ occurs in $S$.

The full proof of Lemma 4.26 can be found in Appendix B.3. The idea how the lifting works is illustrated in the following example:

**Example 4.27.** Consider the set of HoCHCs

$$\neg(z = x + y) \lor \text{Add } x y z \quad \neg \text{Add } n n \ (n + n) \lor \neg w$$

and its encoding

$$\neg(z = x + y) \lor H (((@ (@ (Add x)) y) z)) \quad \neg H (((@ (@ (Add n)) n) (n + n)) \lor \neg H w)$$

We can refute $S$ by using the rules Resolution and Constraint Refutation in the following way:

$$\begin{align*}
\text{Res} & \quad \neg(z = x + y) \lor \text{Add } x y z \quad \neg \text{Add } n n \ (n + n) \lor \neg w \\
\text{Cons. Ref} & \quad \neg(n + n = n + n) \lor \neg w \\
\text{\perp} &
\end{align*}$$
There is a refutation of $\lfloor S \rfloor$ using the rules FO-Resolution and FO-Constraint Refutation of a similar structure:

\[
\begin{array}{c}
\text{FO-Res} & \frac{- (z = x + y) \lor H (\@ (@ (\@ Add x) y) z)}{
\text{FO-Res} & \frac{- (n + n = n + n) \lor H w}{
\text{FO-Cons. Ref} & \frac{- (n + n = n + n)}{\bot}
\end{array}
\]

using the unifier $[n/x, n/y, (n + n)/z]$ and $[c_0/w]$, respectively.

**Corollary 4.28** (Completeness of Encoding). *If $S$ is unsatisfiable then $\lfloor S \rfloor$ is unsatisfiable.*

**Proof.** If $S$ is unsatisfiable then by Theorems 3.7 and 4.6, $S \Rightarrow_{\text{Res}} S' \cup \{\bot\}$ (for some $S'$). By the Lifting Lemma 4.26 $S \Rightarrow_{\text{FO-Res}} S'' \cup \{\bot\}$ therefore by Corollary 4.25 $\lfloor S \rfloor$ is unsatisfiable.

Consequently, we get (also using Proposition 4.23):

**Theorem 4.29.** $S$ is satisfiable if and only if $\lfloor S \rfloor$ is satisfiable.

From this, Theorem 1.13 and Lemma 4.21 we infer:

**Corollary 4.30.** $S$ is unsatisfiable if and only if hierarchic superposition refutes $\lfloor S \rfloor$.

Note that for the theory of positive equality and uninterpreted functions (cf. Section 4.3) as considered in [Charalambidis et al., 2013] this encoding results in a set of first-order Horn clauses effectively *without* a background theory.
Chapter 5

Related Work, Conclusions and Future Directions

5.1 Related Work and Discussion

**Higher-Order Automated Theorem Proving** There has been a long history of resolution procedures for higher-order logic which are refutationally complete with respect to Henkin semantics [Andrews, 1971, Huet, 1972, Benzmüller and Kohlhase, 1998]. They mostly differ in their treatment of unification (which is undecidable for higher-order logic [Lucchesi, 1972, Huet, 1973, Goldfarb, 1981]) and extensionality\(^1\) [Benzmüller, 2002]. Whilst in [Andrews, 1971] variables are instantiated completely blindly, the proof systems in [Huet, 1972, Benzmüller and Kohlhase, 1998] encode unification problems in clauses. Furthermore, in [Andrews, 1971, Huet, 1972] the infinitely many extensionality axioms (see discussion and Eq. (5.1) below) have to be added explicitly whilst in [Benzmüller and Kohlhase, 1998] the proof system takes care of guaranteeing extensionality.

Recently, [Bentkamp et al., 2018] began extending superposition (cf. Section 1.4.3.3) to higher-order logic. Being work in progress, it is currently necessary to add extensionality and comprehension axioms explicitly. Their approach seems to be very promising given the fact that Superposition is the state-of-the-art in first-order theorem proving.

Furthermore, a tableau-style proof system has been proposed and implemented, which delegates subtasks to a propositional SAT solver [Brown, 2011].

All of the completeness proofs of the above proof system construct Henkin models out of terms in case the proof system is unable to refute a problem. In particular, the interpretation of all types is countable if the signature is countable. Hence, these proofs do not seem to be extendable to provide standard models. Furthermore, the procedures are designed for full higher-order logic without background theories.

**Translation to First-Order Logic** It is well-known that higher-order logic with Henkin semantics can be regarded as a (many-sorted) first-order theory [Van Benthem and Doets, 1983, Kerber, 1991], which has successfully been exploited in interactive theorem provers [Blanchette et al., 2016].

\(^1\)The latter is no concern for us because our approach is not based on building structures out of terms.
A significant advantage of our encoding over approaches for general (unconstrained) higher-order logic is that we only need to introduce a very small (finite) number of comprehension axioms corresponding to the fact that Henkin frames contain $\top^H$ for relational $\rho$. Furthermore, we do not need extensionality axioms at all. They have the form
\begin{equation}
 x (\text{diff}_{r,\rho} x y) \neq y (\text{diff}_{r,\rho} x y) \lor x = y \tag{5.1}
\end{equation}
for a function symbol $\text{diff}_{\iota,\rho} : [\iota \rightarrow \rho] \rightarrow [\iota \rightarrow \rho] \rightarrow \iota$ (cf. [Bentkamp et al., 2018]). Note that by introducing these axioms we have entered the realm of first-order logic with equality (which is generally more complicated and difficult to solve) although $\lfloor C \rfloor$ only contains equalities in the background theory for each $C \in S$. Furthermore, sufficient completeness with respect to simple instances is not guaranteed any more by the simple criterion (cf. Section 1.4.3). This means that it is not obvious that the target fragment of the encoding admits refutationally complete proof systems.

**Defunctionalisation**  The translation to first-order logic resembles a technique from the theory of programming languages called defunctionalisation [Reynolds, 1972]. This is a whole-program transformation, which reduces higher-order functional programs to first-order ones. It eliminates higher-order features, such as partial applications and $\lambda$-abstractions, by storing arguments in data types and recovering them in an application function, which performs a matching on the data type.

Recently, [Pham, 2018] has adapted the approach to solve the satisfiability problem for HoCHCs: given a set of HoCHCs, it generates an equi-satisfiable set of first-order Horn clauses over the original background theory and additionally the theory of data types by taking a detour via programs and defunctionalisation.

By contrast, our translation is purely logical and directly results in first-order Horn clauses. Besides, it avoids using inductive data types.

**Hierarchic Refutational First-Order Theorem Proving**  In an abstract sense, the proof system presented in this work is very similar to the one in [Bachmair et al., 1994, Althaus et al., 2009]: there is a clear separation between logical/foreground reasoning and reasoning in the background theory. Moreover, the search is directed purely by the former whilst the latter is only used in a final step to check satisfiability of a conjunction of theory atoms.

However, since they consider full first-order logic with equality they have to do equational reasoning and factoring, whilst we have to deal with $\beta$-conversion. Furthermore, our Constraint Refutation rule implicitly contains a weak form of comprehension in addition.

**Extensional Higher-Order Logic Programming**  In contrast to our work, the aim of higher-order logic programming is not only to establish satisfiability of a set of Horn clauses but also to find (representatives of) “answers to queries”, i.e. witnesses that goal clauses are falsified in every model of the definite clauses. Therefore, [Charalambidis et al., 2013] propose a rather complicated semantics using ideas from domain theory [Abramsky and Jung, 1994]. They design a resolution-based proof system that supports...
a strong notion of completeness ([Charalambidis et al., 2013, Theorem 7.38]) with respect to this semantics.

Their proof system is much more complicated because it operates on positive expressions/positive existential formulas instead of clauses. Furthermore, it requires the instantiation of variables with certain terms. We avoid this by implicitly instantiating all remaining relational variables with $\top^H_\rho$ in the Constraint Refutation rule.

The general idea of our completeness proof is quite similar to the one they propose: By their choice of semantics our Proof Steps (S1) and (S2) are trivial. Proof Step (S3) is used in both approaches in a similar fashion. [Charalambidis et al., 2013] take a more direct route to prove completeness without taking the slight detour of explicitly doing Proof Step (S4). However, we believe that this obscures the essence of the proof idea (partially because they also want to get the stronger notion of completeness).

**Equivalence of Monotone, Continuous and Standard Semantics for HoCHCs**

[Catcart Burn et al., 2018] present an explicit translation of models of a set of HoCHCs with respect to the monotone frame into a model with respect to the standard frame and vice versa using Galois connections. In as yet unpublished work [Jochems, 2018], this result was extended to continuous semantics and the semantics considered by [Charalambidis et al., 2013] in the context of logic programming (see previous paragraph).

**Refinement Type Assignments**

[Cathcart Burn et al., 2018] additionally introduce a refinement type system, the aim of which is to automate the search for models. In this respect, the approach is orthogonal to our resolution proof system, which can be used to refute all unsatisfiable problems (but might fail on satisfiable instances). However, for satisfiable clause sets the method by [Cathcart Burn et al., 2018] may also be unable to generate models.

### 5.2 Conclusions

In this work, we have provided further evidence that higher-order constrained Horn clauses are an attractive basis for the verification of higher-order programs: they are not only a programming-language-independent description of invariants but are also robustly semi-decidable regardless of the choice of semantics (if the background theory is decidable).

Focussing on background theories with a single (standard) model, we have presented a simple resolution proof system which is sound and refutationally complete for all Henkin frames. Our completeness proof establishes and exploits a number of model theoretic results, which suggest that higher-order constrained Horn clauses inherit more properties from the first-order counterpart than it seems at first glance.

Moreover, we have re-proven known equivalence results of standard, monotone and continuous semantics for higher-order constrained Horn clauses employing the (properties of the) proof system. Remarkably, this additionally establishes the equivalence of standard semantics to Henkin semantics (i.e. equi-satisfiability for the fragment).

Given the latter result, it does not come as a big surprise that there is a reduction of higher-order constrained Horn clauses to first-order logic with background theories which
is sound and complete for standard semantics. To support the practicality of this insight we have demonstrated that the target logic admits refutationally complete proof systems leveraging work on automated theorem proving for (hierarchic) first-order theories.

Furthermore, we have described how our approach can also be used to determine (un-)satisfiability of unconstrained Horn clauses as considered in extensional higher-order logic programming.

5.3 Future Directions

In this final section we point out directions for interesting future work.

5.3.1 Implementation

Our work naturally suggests two implementations to solve the satisfiability problem for HoCHC: a direct implementation of our resolution proof system calling existing theory solvers in a modular way or a tool which performs the first-order translation and then calls a theorem prover supporting background theories.

Obviously, the former is a much more involved task. Resolution-based provers are generally extremely complex pieces of software [Weidenbach et al., 2009, Riazanov and Voronkov, 2001] and it is of utmost importance to implement every subroutine and all data structures in a careful and highly efficient way. However, in contrast to first-order theorem proving we do not need to compute (most general) unifiers and we only need to deal with Horn clauses, which potentially results in a comparatively simple implementation.

Furthermore, as in first-order theorem proving it would probably be highly important to remove as much redundancy (e.g. \( \neg U \lor R \) is redundant in the set \( \{ R, \neg U \lor R \} \)) in the working clause set as possible. Moreover, it might be helpful to strengthen the proof system by restricting the inferences necessary for completeness as it is done in e.g. superposition (cf. Sections 1.4.3.3 and 5.1). However, this seems to be a highly non-trivial task because the completeness arguments rely on (countable) first-order term models.

On the other hand, although the theoretical foundation has been laid for theorem proving for (hierarchic) first-order theories, its implementation does not seem to have attracted lots of attention (in stark contrast to SMT solving). Hence, it is not predictable how useful this translation will be in practice.

5.3.2 Background Theories with Infinitely Many Models

Our resolution proof system is designed for background theories with a single (standard) model such as linear arithmetic. Although we have given evidence that we can also deal with some theories with an infinite number of models, our approach cannot deal with arbitrary background theories.

As a matter of fact it is impossible to devise proof systems for this general case because the logic is not compact any more as the following example illustrates:
Example 5.1. Consider the signature $\Sigma = \Sigma_{\text{LIA}} \cup \{c: \iota\}$ and the (infinite) set of HoCHCs

$$S = \{\neg(c = 1 + \cdots + 1) \mid n \in \mathbb{N}\}.$$ 

Furthermore, for $n \in \mathbb{N}$ let $A_n$ be the expansion of $A_{\text{LIA}}$ defined by $c^{A_n} = n$, and consider the infinite set of background theories $S = \{A_n \mid n \in \mathbb{N}\}$. Clearly, for every finite $S' \subseteq S$ there exists $B \in \mathcal{B}$ satisfying $B \models S'$ but for every $B \in \mathcal{B}$, $B \not\models S$.

Consequently, it is necessary to find appropriate restrictions on the background theory which admit refutationally complete calculi. A good starting point is presumably the work on theorem proving for hierarchic first-order theories [Bachmair et al., 1994, Althaus et al., 2009], which addresses similar problems.

5.3.3 Extensions of the Fragment

It turns out that only seemingly minor relaxations of the restrictions on the syntax of HoCHCs allow us to recover full relational clausal higher-order logic. Possible natural extensions of the fragment allow clauses of the form $G \lor R M$, where $G$ is a goal clause and the following may occur in $M$ apart from distinct variables:

(i) a logical symbol $\text{false}$ (with the obvious interpretation),
(ii) relational symbols from $\Sigma' \setminus \Sigma$,
(iii) non-distinct variables.

Furthermore we may think of also allowing

(iv) more than one positive literal of the form $R x$, where all variables in $x$ are distinct.

Let $\text{HoCC}$ be the clausal, relational fragment of higher-order logic, i.e. in contrast to HoCHC we allow clauses of the form $A_1 \lor \cdots \lor A_n \lor G$, where $G$ is a goal clause, each $A_i$ is an (arbitrary) atom and $n$ may be greater than 1. We call the fragments of HoCC defined in Items (i) to (iv), $\text{HoCHC}_{\text{false}}$, $\text{HoCHC}_{\Sigma'}$, $\text{HoCHC}_{\text{vars}}$ and $\text{HoCC}^-$, respectively.

It turns out that all these fragments coincide in terms of expressiveness because in all of them it is possible to define the negation function $\neg: \mathcal{B} \to \mathcal{B}$, i.e. $\neg(b) = 1 - b$ for $b \in \mathcal{B}$. To see how this works, let $S$ be a set of arbitrary HoCC. Consider the signature

$$\Sigma'' = \Sigma' \cup \{T, F : o, \text{Neg} : o \to o, \text{Imp} : o \to o \to o\}$$

(assuming without loss of generality that the new symbols do not already occur in $\Sigma'$). Let $S'$ and $S''$ be the sets of HoCHC defined by

$$S' = \{\neg \text{Neg} A_1 \lor \cdots \lor \neg \text{Neg} A_n \lor G \mid (A_1 \lor \cdots \lor A_n \lor G) \in S\}$$

$$S'' = \{T, \neg F, \neg \text{Neg} T\}$$

and let

$$S_{\text{false}} = \{\text{Neg false}\}$$

$$S_{\Sigma'} = \{\text{Neg } F\}$$
\[ S_{\text{vars}} = \{ \text{Imp } x x, \neg \text{Imp } T F, \neg y \lor \text{Imp } x y, \neg \text{Imp } F x \lor \neg x \} \]
\[ S^- = \{ \text{Imp } x y \lor \text{Imp } y x, \neg \text{Imp } T F, \neg \text{Imp } F x \lor \neg x \} \]

\( S_{\text{false}} \) is clearly a set of HoCHC\(_{\text{false}}\) and similarly for the other fragments. Note that for any \( \Sigma''\)-structure \( B, B \models S'' \cup S_{\text{false}} \) if and only if \( \text{Neg}\_B = \text{neg} \) and similarly for the other fragments. Consequently, all of the following are equivalent:

(i) \( S' \cup S'' \cup S_{\text{false}} \) is \((A, S)\)-satisfiable,

(ii) \( S' \cup S'' \cup S_{\text{vars}} \) is \((A, S)\)-satisfiable,

(iii) \( S' \cup S'' \cup S_{\text{false}} \) is \((A, S)\)-satisfiable,

(iv) \( S' \cup S'' \cup S^- \) is \((A, S)\)-satisfiable,

(v) \( S \) is \((A, S)\)-satisfiable.

The essence of the argument is that by either nesting symbols from \( \Sigma' \setminus \Sigma \) into unnegated atoms or using variables multiple times in unnegated atoms it is possible to define a negation function. Therefore, it is natural to consider the following type of clauses, which renders these constructions impossible:

**Definition 5.2.** A \((\text{higher-order})\) **constrained positive-linear clause** is a term

\[ M_1 \lor \cdots \lor M_m \lor R_1 \overline{N}_1 \lor \cdots \lor R_n \overline{N}_n \lor G, \]

where

(i) \( G \) is a goal clause,

(ii) \( R_i \in \Sigma' \setminus \Sigma \) for each \( i \),

(iii) each variable occurs at most once free in any of the \( M_i \) and \( \overline{N}_j \),

(iv) each \( M_i \) and \( \overline{N}_j \) neither contains logical symbols nor symbols from \( \Sigma' \).

It seems to be impossible to define functions which are not monotone in this fragment. Besides, it appears to be likely that the techniques of Section 3.2.2 can be extended to prove that every satisfiable set of such clauses also has a model which is monotone (in an appropriate sense) as in Theorem 3.23(iii). However, the canonical structure construction would probably be dependent on a given (higher-order) model of the clauses. Hence, the construction does not seem to be as useful as in the case for HoCHC when it comes to proving completeness.

It may also be possible to split (negated) atoms in which variables from multiple positive atoms occur. This would imply that the (higher-order) constrained positive-linear clause satisfiability problem could be reduced to the HoCHC satisfiability problem.

Researchers in the logic programming community investigated non-Horn extensions (dually allowing negations in the body of rules) [Charalambidis et al., 2014, Charalambidis and Rondogiannis, 2014] but as far as we are aware, no complete proof system is known for this (unconstrained) fragment of higher-order logic.

### 5.3.4 Decidable Fragments

Although the combination of even first-order Horn logic with linear integer arithmetic is highly undecidable (as we have seen in Section 2.4), there has recently been some...
investigation into decidable fragments. [Horbach et al., 2017a] consider the Bernays-Schönfinkel-Ramsay fragment extended with a restricted form of linear integer arithmetic, i.e. the fragment of first-order logic without function symbols (but constant symbols) plus arithmetic constraints of the form (i) $M \succcurlyeq N$, (ii) $x \succcurlyeq M$, where $\succcurlyeq \in \{<, \leq, =, \neq, \geq, >\}$ and $M, N$ are ground, or (iii) $x \succcurlyeq y$, where $\succcurlyeq \in \{\leq, =, \geq\}$. Using ideas from quantifier-elimination, they obtain decidability by showing that for every finite set of such clauses there exists a finite, equi-satisfiable set of ground instance.

We conjecture that a similar decidability result can be obtained for HoCHC. The challenge seems to be to permit constraints of the form (iii) because for the other two forms it is easy to lift the equivalence classes induced by the constraints to higher types.

Moreover, we are interested in finding decidable fragments of unconstrained higher-order Horn clauses (cf. Section 4.3). A promising candidate is the monadic shallow linear Horn-fragment [Weidenbach, 1999, Teucke and Weidenbach, 2017]. In this context the first-order translation of HoCHC might turn out to be a valuable tool to transfer decidability results from first-order to higher-order logic, although the deep nesting of terms generated by the encoding might constitute an obstacle.
Appendix A

Remaining Proofs for Chapter 3

A.1 Remaining Proofs for Section 3.2.2

**Lemma A.1.**  
(i) $\sqsubseteq_m$ is transitive;  
(ii) if $r, r', \in A[J][\rho]$ and $r \sqsubseteq_m r'$ then there exists $r^* \in A[J][\rho]$ such that $r \sqsubseteq_m r^* \sqsubseteq_m r'$.

**Proof.** We prove both parts of the lemma simultaneously by induction on types. For $\iota^* \to \iota$ and $\iota$ this is obvious. Hence, suppose a relational type $\tau \to \rho$ and $r, r', r'' \in H[J][\tau \to o]$ such that $r \sqsubseteq_m r' \sqsubseteq_m r''$.

(i) For the first part, let $\vec{s}, \vec{s}' \in H[J][\tau]$ be arbitrary such that $\vec{s} \sqsubseteq_m \vec{s}'$. By Part (ii) of the inductive hypothesis there exists $\vec{s}^* \sqsubseteq_m \vec{s}' \sqsubseteq_m \vec{s}''$. Clearly, $r(\vec{s}) \leq r'(\vec{s}^*)$ and $r'(s^*) \leq r''(s'')$ and hence $r(s) \leq r''(s'')$. This proves $r \sqsubseteq_m r''$.

(ii) For the second part we define

$$r^* : H[J][\tau] \to B$$

$$\vec{t} \mapsto \begin{cases} 1 & \text{if there exists } \vec{s} \sqsubseteq_m \vec{t} \text{ such that } r(\vec{s}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

To show that $r \sqsubseteq_m r^* \sqsubseteq_m r'$ let $\vec{s} \sqsubseteq_m \vec{s}'$ be arbitrary such that $r^*(\vec{s}) = 1$. Clearly, by definition of $r^*$, $r(\vec{s}) = 1$ implies $r^*(\vec{s}') = 1$. Hence, $r \sqsubseteq_m r^*$. Furthermore, if $r^*(\vec{s}) = 1$ then there exists $\vec{s}''$ such that $\vec{s}'' \sqsubseteq_m \vec{s}$ and $r(\vec{s}'') = 1$. By Part (i) of the inductive hypothesis, $\vec{s}'' \sqsubseteq_m \vec{s}'$ and due to $r \sqsubseteq_m r'$, $r(\vec{s}') = 1$. Consequently, $r^* \sqsubseteq_m r'$.

**Lemma 3.21.** Let $\beta$ be an ordinal. Then

(i) $A^H_\beta$ is quasi-monotone and

(ii) for all ordinals $\beta' \geq \beta$, $A^H_{\beta'} \sqsubseteq_m A^H_\beta$.

**Proof.** (i) The first part is proven by an easy induction using Corollary 3.20 and Corollary 3.17(i).

(ii) By the previous part it suffices to show the claim for $\beta' > \beta$. We proceed by (transfinite) induction on $\beta$. 

61
Claim 1. First, we prove the following:

Proof.

Lemma A.2. Remaining Proofs for Section 3.2.3

\[ A_{\beta}^H = T_p^H (A_{\beta}^H) \preceq_m T_p^H (A_{\beta}^H) = A_{\beta}^H \]

using the inductive hypothesis and the quasi-monotonicity of \( T_p^H \) (Corollary 3.20).

If \( \beta' \) is a limit ordinal then by Corollary 3.17(i), \( A_{\beta}^H \preceq_m A_{\beta'}^H \).

Finally, suppose \( \beta \) is a limit ordinal. By the inductive hypothesis, for each \( \beta < \beta, A_{\beta}^H \preceq_m A_{\beta'}^H \).

Therefore, by Corollary 3.17(iii), \( A_{\beta}^H \preceq_m A_{\beta'}^H \).

A.2 Remaining Proofs for Section 3.2.3

Lemma 3.28. Let \( \rho \) be a relational type and let \( \mathcal{R}, \mathcal{R}' \subseteq \mathcal{H}[\rho] \) be sets such that for each \( r \in \mathcal{R} \) there exists \( r' \in \mathcal{R}' \) satisfying \( r \sqsubseteq c r' \). Then \( \bigcup \mathcal{R} \sqsubseteq c \bigcup \mathcal{R}' \).

Proof. We prove the lemma by induction on the relational type \( \rho' \). For \( o \) this is obvious.

Hence, suppose \( \rho = \tau \to \rho' \). To show \( \bigcup \mathcal{R} \sqsubseteq c \bigcup \mathcal{R}' \), let \( s \in \mathcal{H}[\tau] \) and \( \mathcal{S} \in \text{dir}(s) \).

We define \( \mathcal{T} = \{r(s) \mid r \in \mathcal{R}\} \) and \( \mathcal{T}' = \bigcup \{r'(s') \mid s' \in \mathcal{S}' \} \mid r' \in \mathcal{R}' \} \). Note that for each \( r \in \mathcal{R} \) there exists \( r' \in \mathcal{R}' \) such that \( r \sqsubseteq c r' \) and hence due to \( \mathcal{S} \in \text{dir}(s) \), \( r(s) \sqsubseteq c \bigcup \{r'(s') \mid s' \in \mathcal{S}' \} \). Consequently, the inductive hypothesis is applicable to \( \mathcal{T} \) and \( \mathcal{T}' \), which yields

\[
\left( \bigcup \mathcal{R} \right)(s) = \bigcup \mathcal{T} \subseteq c \bigcup \mathcal{T}' = \bigcup \{r'(s') \mid r' \in \mathcal{R}' \land s' \in \mathcal{S}' \} = \bigcup \left\{ \left( \bigcup \mathcal{R}' \right) s' \mid s' \in \mathcal{S}' \right\}
\]

using Lemma 1.4.

Lemma A.2. \( \sqsubseteq_c \) is transitive.

Proof. First, we prove the following:

Claim 1. If \( r \in \mathcal{H}[\sigma] \), \( \mathcal{R}' \in \text{dir}(r) \) and \( r' \in \mathcal{R}' \in \text{dir}(r') \)

Proof. Clearly, it suffices to prove \( r' \sqsubseteq c \bigcup \mathcal{R}' \). But by Lemma 3.28 and the fact that \( r' \in \mathcal{R}' \) is quasi-continuous this is obvious.

Now, we prove the lemma by induction on the type. For \( o \) and \( \iota^n \to \iota \) this is trivial. Hence, let \( r, r', r'' \in \mathcal{H}[\tau \to \rho] \) be such that \( r \sqsubseteq c r' \sqsubseteq c r'' \). Furthermore, let \( s \in \mathcal{H}[\tau] \) and \( \mathcal{S}'' \in \text{dir}(s) \) be arbitrary. Then it holds that

\[
\begin{align*}
\bigcup \{r'(s') \mid s' \in \mathcal{S}'' \} &= \bigcup \{r''(s'') \mid s'' \in \mathcal{S}'' \} & & \text{Lemma 3.28 and Claim 1} \\
\bigcup \{r''(s') \mid s' \in \mathcal{S}'' \} &= \bigcup \{r''(s') \mid s' \in \mathcal{S}'' \} & & \text{Lemma 1.4}
\end{align*}
\]

and therefore by the inductive hypothesis, \( r(s) \sqsubseteq c \bigcup \{r''(s') \mid s' \in \mathcal{S}'' \} \), which proves \( r \sqsubseteq c r'' \).
Proposition 3.30. Let $M$ be a positive existential term, $\mathcal{B}$ be expansions of $\mathcal{A}$, $\mathcal{B}' \in \text{dir}(\mathcal{B})$, $\alpha$ be a valuation and let $\mathfrak{A}' \in \text{dir}(\alpha)$. Then

$$\mathcal{B}[M](\alpha) \sqsubseteq_c \bigsqcup \{\mathcal{B}'[M](\alpha') \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'\}$$

(A.2)

and the expression on the right-hand side is well-defined.

Proof. We prove that for all positive existential terms $M$, expansions $\mathcal{B}$ of $\mathcal{A}$, $\mathcal{B}' \in \text{dir}(\mathcal{B})$, valuations $\alpha$ and $\mathfrak{A}' \in \text{dir}(\alpha)$ Eq. (A.2) holds by induction on the structure of $M$.

- If $M$ is a logical constant (other than $\neg$) then this is due to Examples 3.25(i) and 3.25(ii).
- If $M$ is a symbol $R \in \Sigma'$ then

$$\mathcal{B}[M](\alpha) = R^\mathcal{B} \sqsubseteq_c \bigsqcup \{R^{\mathcal{B}'} \mid \mathcal{B}' \in \mathcal{B}'\} = \bigsqcup \{\mathcal{B}'[M](\alpha') \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'\}$$

because $\mathcal{B}' \in \text{dir}(\mathcal{B})$ and clearly the expression on the right-hand side is well-defined.

- If $M$ is a variable $x$ then

$$\mathcal{B}[M](\alpha) = \alpha(x) \sqsubseteq_c \bigsqcup \{\alpha'(x) \mid \alpha' \in \mathfrak{A}'\} = \bigsqcup \{\mathcal{B}'[M](\alpha') \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'\}$$

because $\mathfrak{A}' \in \text{dir}(\alpha)$ and clearly the expression on the right-hand side is well-defined.

- Next, suppose $M$ is an application $M_1 M_2$. By the inductive hypothesis,

$$\mathcal{B}[M_1](\alpha) \sqsubseteq_c \bigsqcup \{\mathcal{B}_1[M_1](\alpha_1) \mid \mathcal{B}_1 \in \mathcal{B}' \land \alpha_1 \in \mathfrak{A}'\}$$

(A.1)

$$\mathcal{B}[M_2](\alpha) \sqsubseteq_c \bigsqcup \{\mathcal{B}_2[M_2](\alpha_2) \mid \mathcal{B}_2 \in \mathcal{B}' \land \alpha_2 \in \mathfrak{A}'\}$$

(A.2)

and the expressions on the right-hand sides are well-defined. Therefore, if $M_1$ has type $\iota^{n+1} \rightarrow \iota$ then by Eqs. (A.1) and (A.2), $\mathcal{B}[M_1](\alpha) = \mathcal{B}'[M_1](\alpha')$ and $\mathcal{B}[M_2](\alpha) = \mathcal{B}'[M_2](\alpha')$ for every $\mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'$. Consequently,

$$\mathcal{B}[M](\alpha) = \mathcal{B}'[M_1](\alpha')(\mathcal{B}'[M_2](\alpha'))$$

$$= \bigsqcup \{\mathcal{B}'[M_1](\alpha')(\mathcal{B}'[M_2](\alpha')) \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'\}$$

and therefore $\mathcal{B}[M](\alpha) \sqsubseteq_c \bigsqcup \{\mathcal{B}'[M](\alpha') \mid \mathcal{B}' \in \mathcal{B}' \land \alpha' \in \mathfrak{A}'\}$.

Otherwise it holds that $\Delta \vdash M_1 : \tau \rightarrow \rho$. We define $s = \mathcal{B}[M_2](\alpha)$ and the set $\mathcal{S}' = \{B_2[M_2](\alpha_2) \mid B_2 \in \mathcal{B}' \land \alpha_2 \in \mathfrak{A}'\}$.

Claim 1. $\mathcal{S}' \in \text{dir}(s)$.

Proof. By Eq. (A.2), $s \sqsubseteq_c \bigsqcup \mathcal{S}'$. Next, let $B_2 \in \mathcal{B}'$ and $\alpha_2 \in \mathfrak{A}'$. Note that $\{B_2\} \in \text{dir}(B_2)$ and $\{\alpha_2\} \in \text{dir}(\alpha_2)$ because both $B_2$ and $\alpha_2$ are quasi-continuous. Therefore, by the inductive hypothesis, $B_2[M_2](\alpha_2) \sqsubseteq_c B_2[M_2](\alpha_2)$. This proves $\mathcal{S}' \subseteq C(\mathcal{H}[\tau'])$. 


Finally, to prove that $\mathcal{G}'$ is directed, let $\mathcal{B}'^{(1)}, \mathcal{B}'^{(2)} \in \mathcal{B}'$ and $\alpha^{(1)}, \alpha^{(2)} \in \mathcal{A}'$. Since $\mathcal{B}'$ and $\mathcal{A}'$ are directed there are $\mathcal{B}' \in \mathcal{B}'$ and $\alpha' \in \mathcal{B}'$ satisfying $\mathcal{B}'^{(j)} \sqsubseteq_c \mathcal{B}'$ and $\alpha^{(j)} \sqsubseteq_c \alpha'$ for $j \in \{1, 2\}$. Note that $\mathcal{B}' \in \text{dir}(\mathcal{B}'^{(1)}) \cap \text{dir}(\mathcal{B}'^{(2)})$ and moreover $\{\alpha'\} \in \text{dir}(\alpha^{(1)}) \cap \text{dir}(\alpha^{(2)})$. Therefore by the inductive hypothesis for $j \in \{1, 2\}$, $\mathcal{B}'^{(j)} \sqcup \mathcal{M}[\alpha^{(j)}] \sqsubseteq \mathcal{B}' \sqcup \mathcal{M}[\alpha']$.

Next, we define

$$
\mathcal{I} = \{ \mathcal{B}' \sqcup \mathcal{M}[\alpha] \mid \alpha \in \mathcal{A}' \} \\
\mathcal{I}' = \{ \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \mid \alpha' \in \mathcal{A}' \}.
$$

**Claim 2.** $\bigcup \mathcal{I} \sqsubseteq_c \bigcup \mathcal{I}'$.

**Proof.** We prove the claim using Lemma 3.28. To show that the lemma is applicable, let $\mathcal{B}_1, \mathcal{B}_2 \in \mathcal{B}'$ and $\alpha, \alpha_2 \in \mathcal{A}'$ be arbitrary. By directedness of $\mathcal{B}'$ and $\mathcal{A}'$, there are $\mathcal{B}' \in \mathcal{B}'$ and $\alpha' \in \mathcal{A}'$ such that $\mathcal{B}_j \sqsubseteq_c \mathcal{B}'$ and $\alpha_j \sqsubseteq_c \alpha'$ for $j \in \{1, 2\}$. Note that $\{\mathcal{B}'\} \in \text{dir}(\mathcal{B}_1) \cap \text{dir}(\mathcal{B}_2)$, $\{\alpha'\} \in \text{dir}(\alpha_1) \cap \text{dir}(\alpha_2)$. Therefore, again by the inductive hypothesis,

$$
\mathcal{B}_1 \sqcup \mathcal{M}[\alpha] \sqsubseteq \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \quad \text{(A.3)} \\
\mathcal{B}_2 \sqcup \mathcal{M}[\alpha] \sqsubseteq \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \quad \text{(A.4)}
$$

Furthermore, due to $\{\mathcal{B}'\} \in \text{dir}(\mathcal{B}')$ and $\{\alpha'\} \in \text{dir}(\alpha')$ and the inductive hypothesis, $\mathcal{B}' \sqcup \mathcal{M}[\alpha'] \sqsubseteq \mathcal{B}' \sqcup \mathcal{M}[\alpha']$. Hence (using Eq. (A.4)), $\{\mathcal{B}' \sqcup \mathcal{M}[\alpha']\} \in \text{dir}(\mathcal{B}_2 \sqcup \mathcal{M}[\alpha_2])$. Therefore by Eq. (A.3),

$$
\mathcal{B}_1 \sqcup \mathcal{M}[\alpha] \sqcup \mathcal{B}_2 \sqcup \mathcal{M}[\alpha] \sqsubseteq \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \sqcup \mathcal{B}' \sqcup \mathcal{M}[\alpha'].
$$

Consequently, Lemma 3.28 is applicable to $\mathcal{I}$ and $\mathcal{I}'$ yielding $\bigcup \mathcal{I} \sqsubseteq_c \bigcup \mathcal{I}'$.

Combining everything we get

$$
\mathcal{B}[\mathcal{M}](\alpha) = \mathcal{B}[\mathcal{M}_1](\alpha)(s) \\
\sqsubseteq_c \bigcup \left\{ \left( \mathcal{B}_1 \sqcup \mathcal{M}[\alpha] \mid \mathcal{B}_1 \in \mathcal{B}' \land \alpha_1 \in \mathcal{A}' \right) \mid s' \in \mathcal{G}' \right\} \\
\sqsubseteq_c \bigcup \mathcal{I} \\
\sqsubseteq_c \bigcup \mathcal{I}' \\
= \bigcup \{ \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \mid \alpha' \in \mathcal{A}' \}.
$$

This concludes the proof of $\mathcal{B}[\mathcal{M}_1 \mathcal{M}_2](\alpha) \sqsubseteq \bigcup \{ \mathcal{B}' \sqcup \mathcal{M}[\alpha'] \mid \alpha' \in \mathcal{A}' \}$.

Finally, suppose $M$ is $\lambda x. M'$. and assume $\Delta \vdash M : \tau \rightarrow \rho$. Let $s \in \mathcal{H}[\tau]$, $\mathcal{G}' \in \text{dir}(s)$. Note that $\{\alpha'[x \mapsto s'] \mid \alpha' \in \mathcal{A}' \land s' \in \mathcal{G}'\} \in \text{dir}(\alpha[x \mapsto s])$. Therefore by the inductive hypothesis,

$$
\mathcal{B}[\mathcal{M}'](\alpha[x \mapsto s]) \sqsubseteq \bigcup \{ \mathcal{B}' \sqcup \mathcal{M}'[\alpha'[x \mapsto s']] \mid \alpha' \in \mathcal{A}' \land s' \in \mathcal{G}' \}. 
$$

(A.5)
Consequently,
\[
\mathcal{B}[M](\alpha)(s) = \mathcal{B}[M'](\alpha[x \mapsto s])
\]

\vspace{1ex}

\[\mathcal{B}[M] \subseteq c \{ \text{Lemma 1.4} \}\]

\vspace{1ex}

\[\mathcal{B}[M] \subseteq c \{ \text{Lemma 1.4} \}\]

\vspace{1ex}

This concludes the proof.

**Lemma 3.32.** Let \( \beta \) be an ordinal. Then

(i) \( A^H_\beta \) is quasi-continuous and

(ii) for all ordinals \( \beta' \geq \beta \), \( A^H_\beta \subseteq c A^H_{\beta'} \).

**Proof.**

(i) For \( \beta = 0 \) this is obvious, for successor ordinals this is Corollary 3.31 and for limit ordinals this is Corollary 3.29(ii).

(ii) By the previous part it suffices to show the claim for \( \beta' > \beta \). We proceed by induction on \( \beta \).

- If \( \beta = 0 \) this is obvious.

- Next, suppose \( \beta = \tilde{\beta} + 1 \) is a successor ordinal. Note that \( \beta' = 0 \) is impossible. If \( \beta' = \beta' + 1 \) then

\[ A^H_{\beta'} = T^H_P(A^H_{\beta'}) \subseteq c T^H_P(A^H_{\beta}) = A^H_{\beta} \]

using the inductive hypothesis and Corollary 3.31. Otherwise \( \beta' \) is a limit ordinal and then by Corollary 3.29(i), \( A^H_{\beta} \subseteq c A^H_{\beta'} \).

- Finally, suppose \( \beta \) is a limit ordinal. By the inductive hypothesis, for each \( \tilde{\beta} < \beta \), \( A^H_{\tilde{\beta}} \subseteq c A^H_{\tilde{\beta}} \). Therefore, by Corollary 3.29(iii), \( A^H_{\beta} \subseteq c A^H_{\beta} \).

**Proposition 3.34.** Let \( \beta \) be an ordinal. Then \( A^H_{\beta} \subseteq c A^H_{\omega} \).

**Proof.** We prove the lemma by induction on \( \beta \).

- If \( \beta = 0 \) this is trivial.

- Next, suppose \( \beta = \beta' + 1 \) is successor ordinal. By the inductive hypothesis, \( A^H_{\beta'} \subseteq c A^H_{\beta} \), by Corollary 3.31 and Lemma 3.33, \( A^H_{\beta} = T^H_P(A^H_{\beta'}) \subseteq c T^H_P(A^H_{\omega}) \subseteq c A^H_{\beta} \).

- Finally, suppose \( \beta \) is a limit ordinal. By the inductive hypothesis, for every \( \beta' < \beta \), \( A^H_{\beta'} \subseteq c A^H_{\beta} \). Therefore, by Corollary 3.29(iii), \( A^H_{\beta} \subseteq c A^H_{\omega} \).

\section*{A.3 Remaining Proofs for Section 3.3.1}

**Lemma 3.39 (Basic properties of \( \rightarrow_{B^v} \)).** Suppose \( M \rightarrow_{B^v} N \). Then

(i) \( \rightarrow \subseteq \rightarrow_{B^v} \).
(ii) \(\text{free}(N) \subseteq \text{free}(M)\);
(iii) if \(M\) is semi-normal then \(N\) is semi-normal, too.

\textbf{Proof.} (i) Straightforward induction on the definition of \(\rightarrow_\|\).
(ii) We prove the first part of the lemma by induction on the compatible closure of \(\rightarrow_\beta\). If \((M, N) \in \beta\) this is a standard fact of \(\beta\)-reduction. If \((M, N) \in \nu\) then \(\text{free}(M) = \text{free}(N) = \emptyset\). In the inductive cases the claim immediately follows from the inductive hypothesis.
(iii) We prove the claim by induction on the compatible closure of \(\rightarrow_\beta\).

- First, suppose that \((R, \lambda\overline{x}_R, F_R) \in \nu\). Since \(F_R\) is in normal form it is in particular semi-normal. Hence, \(\lambda\overline{x}_R, F_R\) is semi-normal, too.
- Next, suppose that \(((\lambda x. M)M', M[M'/x]) \in \beta\). Clearly, \(M\) and \(M'\) must be semi-normal. We prove by induction on \(M\) that \(M[M'/x]\) is semi-normal. If \(M\) is a variable this is obvious (because \(M'\) is semi-normal). The cases for (logical) constants and \(\lambda\)-abstractions are straightforward.

Finally, suppose that \(M\) is an application and let \(\exists L\) be a subterm of

\[M_1[M'/x]M_2[M'/x]\]

By the inductive hypothesis, both \(M_1[M'/x]\) and \(M_2[M'/x]\) are semi-normal. Hence, if \(\exists K\) is a subterm of either \(M_1[M'/x]\) or \(M_2[M'/x]\) then \(K\) must be a \(\lambda\)-abstraction. Otherwise \(M_1 = \exists\) and \(K = M_2[M'/x]\). Then by assumption \(M_2\) is a \(\lambda\)-abstraction and clearly, \(M_2[M'/x]\) is a \(\lambda\)-abstraction, too.

- Next, suppose that \(M_1M_2 \rightarrow_\beta N_1N_2\) because \(M_1 \rightarrow_\beta N_1\). Clearly, \(M_1\) is semi-normal. Therefore, by the inductive hypothesis, \(N_1\) is semi-normal. Note that \(N_1 = \exists\) is impossible. Therefore, any subterm \(\exists L\) of \(N_1\) is either a subterm of \(N_1\) or \(M_2\), which are both semi-normal. Hence, \(L\) is a \(\lambda\)-abstraction.

- Suppose \(M_1M_2 \rightarrow_\beta M_1N_2\) because \(M_2 \rightarrow_\beta N_2\). Clearly, \(M_2\) is semi-normal. Therefore, by the inductive hypothesis, \(N_2\) is semi-normal. Let \(\exists L\) be a subterm of \(M_1N_2\). If \(\exists L\) is a subterm of \(M_1\) or \(N_2\) the argument is as in the previous case. Hence, suppose \(M_1 = \exists\) and \(L = N_2[M'/x]\). By assumption \(M_2\) is a \(\lambda\)-abstraction. Due to \(M_2 \rightarrow_\beta L\), \(L\) is a \(\lambda\)-abstraction, too.

- Finally, suppose that \(\lambda x. M \rightarrow_\beta \lambda x. N\) because \(M \rightarrow_\beta N\). Clearly, \(M\) is semi-normal and hence by the inductive hypothesis \(N\) is semi-normal. Let \(\exists L\) be a subterm of \(\lambda x. N\). Obviously, \(\exists L\) must be a subterm of \(M\), which is semi-normal. Hence, \(L\) is a \(\lambda\)-abstraction. \(\square\)

\textbf{Lemma 3.40 (Subject Reduction).} Let \(\Delta \vdash M : \sigma\) be a term such that \(M \rightarrow_\beta N\). Then

(i) \(\Delta \vdash N : \sigma\) and
(ii) \(\sigma\) is a relational type.

\textbf{Proof.} (i) We prove the lemma by induction on the compatible closure of \(\beta\). For \((M, N) \in \beta\) this is [Barendregt et al., 2013, Proposition 1.2.6]. If \((R, \lambda\overline{x}_R, F_R) \in \nu\) and \(R: \overline{\tau} \rightarrow o \in \Sigma'\setminus\Sigma\) then by convention \(\Delta(\overline{x}_R) = \overline{\tau}\) and hence, \(\Delta \vdash \lambda\overline{x}_R. F_R : \overline{\tau} \rightarrow o\),
too. The proofs for the recursive cases are exactly as in the proof of [Barendregt et al., 2013, Proposition 1.2.6].

(ii) Clearly, it suffices to prove by induction on the compatible closure of $\beta v$ that $M \to_{\beta v} N$ implies $\Delta \nvdash M : \ell \to \nu$ for all $\ell \in \mathbb{N}$.

- If $(M, N) \in \beta v$ then clearly $\Delta \nvdash M : \ell \to \nu$ for all $\ell \in \mathbb{N}$.

- Next, suppose $M_1M_2 \to_{\beta v} N_1N_2$ due to $M_1 \to_{\beta v} N_1$. Then by the inductive hypothesis $\Delta \nvdash M_1 : \ell \to \nu$ for all $\ell \in \mathbb{N}$. Hence, clearly $\Delta \nvdash M_1M_2 : \ell \to \nu$ for all $\ell \in \mathbb{N}$.

- Suppose $M_1M_2 \to_{\beta v} N_1N_2$ due to $M_2 \to_{\beta v} N_2$ and assume towards contradiction that $\Delta \vdash M_1M_2 : \ell \to \nu$. Then $\Delta \vdash M_1 : \sigma \to \ell \to \nu$ and $\Delta \vdash M_2 : \sigma$ for some $\sigma$. However, by the definition of types this implies $\sigma = \nu$, which contradicts the inductive hypothesis.

- Finally, if $\lambda x. M' \to_{\beta v} \lambda x. N$ then clearly $\Delta \nvdash \lambda x. M' : \ell \to \nu$ for all $\ell \in \mathbb{N}$.

\[\square\]

### A.4 Remaining Proofs for Section 3.3.2

**Lemma 3.41** (Basic Properties of $\Rightarrow^\ell$). Let $L, M, N$ and $Q$ be terms. Then

(i) $\Rightarrow^\ell$ is reflexive and transitive;

(ii) $\Rightarrow^\ell_0 \subseteq \Rightarrow^\ell_\beta v$.

(iii) If $M \not\Rightarrow^\ell_0 N$ then $M = N$;

(iv) If $L \not\Rightarrow^\ell_{m+1} N$ then there exists $M$ satisfying $L \not\Rightarrow^\ell_{m+1} M \not\Rightarrow^\ell_0 N$;

(v) If $M \not\Rightarrow^\ell Q$ is well-typed and $M \Rightarrow^\ell N$ then $M \not\Rightarrow^\ell Q$;

(vi) If $M[Q/z]$ is well-typed and $M \Rightarrow^\ell N$ then $M[Q/z] \Rightarrow^\ell N[Q/z]$.

**Proof.** (i) Completely trivial.

(ii) Straightforward induction on the definition of $\Rightarrow^\ell_0$

(iii) Straightforward induction on the definition of $\Rightarrow^\ell_0$

(iv) Straightforward induction on the definition of $\Rightarrow^\ell_{m+1}$

(v) Straightforward induction on the definition of $M \not\Rightarrow^\ell_{m+1} N$ noting that for $\circ \in \{\land, \lor\}$ the cases $M_1 \circ M_2 \not\Rightarrow^\ell_{m+1} N_1 \circ N_2$ and $\exists x. M' \not\Rightarrow^\ell_{m+1} N'$ cannot occur because $(M_1 \circ M_2)Q$ and $(\exists x. M')Q$ are not well-typed.

(vi) We prove by induction on $M \not\Rightarrow^\ell_{m+1} N$ that $M[Q/z] \not\Rightarrow^\ell_{m+1} N[Q/z]$.

- If $M = N$ and $m = 0$ then also $M[Q/z] \not\Rightarrow^\ell_0 N[Q/z]$.

- If there exist $L, m_1$ and $m_2$ such that $M \not\Rightarrow^\ell_{m_1} L \not\Rightarrow^\ell_{m_2} N$ and $m = m_1 + m_2$ then by the inductive hypothesis $M[Q/z] \not\Rightarrow^\ell_{m_1} N[Q/z]$. Consequently, $M[Q/z] \not\Rightarrow^\ell_{m_1 + m_2} N[Q/z]$. 


Next, suppose that \( M \) is \( M_1 \circ M_2 \) for \( \circ \in \{\land, \lor\} \) and that there exist \( m_1 \) and \( m_2 \) such that \( m = m_1 + m_2 \) and \( M_j \xrightarrow[\ell]{m_j} N_j \) for \( j \in \{1, 2\} \). By the inductive hypothesis, \( M_j[Q/z] \xrightarrow[\ell]{m_j} N_j[Q/z] \). Consequently,

\[
(M_1 \circ M_2)[Q/z] = (M_1[Q/z] \circ M_2[Q/z]) \xrightarrow[\ell]{m_1+m_2} (N_1[Q/z] \circ N_2[Q/z]) = (N_1 \circ N_2)[Q/z].
\]

Suppose that \( M \) is \( \exists x. M' \) and that \( \exists x. M' \xrightarrow[\ell]{m} \exists x. N' \). By the inductive hypothesis, \( M'[Q/z] \xrightarrow[\ell]{m} N'[Q/z] \). By the variable convention (Convention 1.1), \( x \neq z \). Hence,

\[
(\exists x. M')[Q/z] = \exists x. M'[Q/z] \xrightarrow[\ell]{m} \exists x. N'[Q/z] = (\exists x. N')[Q/z].
\]

Suppose that \( M \) is \( R \overline{M} \) for \( R \in \Sigma' \setminus \Sigma \) and that \( R \overline{M} \xrightarrow[\ell]{1} (\lambda x. R.F_R)\overline{M} \).

Clearly,

\[
(R \overline{M})[Q/z] = R \overline{M}[Q/z] \xrightarrow[\ell]{1} (\lambda x. R.F_R)\overline{M}[Q/z] = ((\lambda x. R.F_R)\overline{M})[Q/z]
\]

using the variable convention (Convention 1.1) and the fact that \( \lambda x. R.F_R \) is closed.

Finally, suppose that \( M \) is \( (\lambda x. M')M''\overline{M}''' \) and that

\[
(\lambda x. M')M''\overline{M}''' \xrightarrow[\ell]{1} M'[M''/x]\overline{M}'''.
\]

By the variable convention \( x \neq z \). Hence

\[
(\lambda x. M')M''\overline{M}'''[Q/z] = (\lambda x. M'[Q/z])M''[Q/z]\overline{M}'''[Q/z]
\]

\[
\xrightarrow[\ell]{1} M'[Q/z][M''[Q/z]/x]\overline{M}'''[Q/z] = (M'[M''/x]\overline{M}''')[Q/z]
\]

using the Nested Substitution Lemma from [Barendregt, 2012, 2.1.16. Substitution Lemma].

\[\square\]

**Lemma 3.43** (Basic Properties of \( \xrightarrow{\ell} \)). (i) \( \xrightarrow{\ell} \) is reflexive (on positive existential terms).

(ii) \( \xrightarrow{\ell} \subseteq \xrightarrow{\beta_v} \).

(iii) If \( L \xrightarrow{\ell} N \) and \( \overline{O} \xrightarrow{\ell} \overline{Q} \) then \( L\overline{O} \xrightarrow{\ell} N\overline{Q} \).

(iv) If \( K \xrightarrow{\ell} L \xrightarrow{\ell} N \) then \( K \xrightarrow{\ell} N \).

(v) If \( L \xrightarrow{\ell} N \) and \( O \xrightarrow{\ell} Q \) then \( L[O/z] \xrightarrow{\ell} N[Q/z] \).

**Proof.** (i) We prove by structural induction on \( M \) that \( M \xrightarrow{\ell} M \). \( M \) has the form \( M_1 \cdots M_n \), where \( M_1 \) is either a variable, a symbol from \( \Sigma' \cup \{\land, \lor, \exists\} \) or a \( \lambda \)-abstraction. In any case the inductive hypothesis and the reflexivity of \( \xrightarrow{\ell} \) immediately yield that \( M_1 \cdots M_n \xrightarrow{\ell} M_1 \cdots M_n \).

We prove the remaining four parts by induction on the definition of \( \xrightarrow{\ell} \). We only show the detailed proof for the case \( L \xrightarrow{\ell} x \overline{N} \) due to \( L \xrightarrow{\ell} x \overline{M} \) and \( \overline{M} \xrightarrow{\ell} \overline{N} \) for some \( \overline{M} \) (the other cases are analogous).
(ii) By Lemma 3.41(ii) and the inductive hypothesis, \( L \rightarrow \beta_{x \in \Sigma}, xM \) and \( M \rightarrow \beta_{x \in \Sigma} N \). Therefore clearly, \( L \rightarrow \beta_{x \in \Sigma}, xM \rightarrow \beta_{x \in \Sigma} N \) and hence also, \( L \rightarrow \beta_{x \in \Sigma}, xN \).

(iii) By Lemma 3.41(v), \( LO \rightarrow xM \overline{O} \) and hence by definition \( L \rightarrow xN \overline{Q} \).

(iv) By transitivity of \( \rightarrow p \) (Lemma 3.41(i)), \( K \rightarrow xM \) and hence by definition \( K \rightarrow xN \).

(v) We prove the lemma by induction on \( L \rightarrow N \). By the inductive hypothesis, \( M(O/z) \rightarrow N(Q/z) \) and by assumption or Part (i), \( x(O/z) \rightarrow x(Q/z) \). Therefore by Part (iii) and Lemma 3.41(iv),

\[
L(O/z) \rightarrow x(O/z)M(O/z) \rightarrow x(Q/z)N(Q/z),
\]

which proves \( L(O/z) \rightarrow (xN)(Q/z) \) by Part (iv).

\[\square\]

**Lemma 3.44** (Inversion). Let \( E \) be a semi-normal formula.

(i) If \( E \rightarrow xN \) then there exists \( M \) such that \( E \rightarrow xM \).

(ii) If \( E \rightarrow cN \), where \( c \in \Sigma' \cup \{\land, \lor, \exists_{x}\} \), then there exists \( M \) such that \( E \rightarrow cM \) and \( M \rightarrow \overline{N} \).

(iii) If \( E \rightarrow \exists_{x}N \) then there exist \( x \), \( N' \) and \( M \) such that \( N = (\lambda x. N') \), \( E \rightarrow \exists_{x}x. M \) and \( M \rightarrow N' \).

**Proof.** The first two parts are obvious by definition of \( \rightarrow \). Hence, suppose that \( E \rightarrow \exists_{x}N \).

By Lemmas 3.43(ii) and 3.39(iii), \( N \) has the form \( \lambda x. N' \) for some \( N' \). Furthermore, by Part (ii) there exists \( L \) such that \( E \rightarrow \exists_{x}L \) and \( L \rightarrow \lambda x. N' \). Again, by Lemmas 3.41(ii) and 3.39(iii), \( L \) has the form \( \lambda y. L' \). By definition of \( \rightarrow \), \( (\lambda y. L') \rightarrow (\lambda x. N') \) implies that there exists \( M \) such that \( (\lambda y. L') \rightarrow (\lambda x. M) \) and \( M \rightarrow N' \). However, \((\lambda y. L') \rightarrow (\lambda x. M)\) clearly implies \( y = x \) and \( L' = M \). Consequently, \( E \rightarrow \exists_{x}x. M \) and \( M \rightarrow N' \).

\[\square\]

**Lemma 3.48.** Let \( \alpha, \alpha' \) be valuations and \( F \) be positive existential formulas satisfying \( \alpha \triangleright F \). If \( \alpha(x) = \alpha'(x) \) for all \( x \in \text{free}(F) \) then \( \alpha' \triangleright F \).

**Proof.** We prove the lemma by induction on the definition of \( \alpha \triangleright F \).

- If \( \alpha \triangleright xM \) then \( \alpha' \triangleright F \) is trivial.
- If \( \alpha \triangleright cM \) for \( c \in \Sigma \) because \( A, \alpha \models cM \) then by Lemma 1.7, \( A, \alpha' \models cM \) and therefore \( \alpha' \triangleright cM \).
- Next, suppose that \( \alpha \triangleright \exists_{x}x. M \) because \( \alpha[x \mapsto r] \triangleright M \) for some \( r \in H[\tau] \). By the inductive hypothesis \( \alpha'[x \mapsto r] \triangleright M \) and therefore \( \alpha' \triangleright \exists_{x}x. M \).
- If \( \alpha \triangleright M_{1} \lor M_{2} \) because \( \alpha \triangleright M_{j} \) for some \( j \in \{1, 2\} \) then by the inductive hypothesis \( \alpha' \triangleright M_{j} \) and therefore \( \alpha' \triangleright M_{1} \lor M_{2} \).
- Finally, if \( \alpha \triangleright M_{1} \land M_{2} \) because \( \alpha \triangleright M_{j} \) for all \( j \in \{1, 2\} \) then by the inductive hypothesis \( \alpha' \triangleright M_{j} \) for all \( j \in \{1, 2\} \) and therefore \( \alpha' \triangleright M_{1} \land M_{2} \).
Appendix B

Remaining Proofs for Chapter 4

B.1 Remaining Proofs for Section 4.1

Lemma 4.2. \( \Box C = \Box C \).

Proof. We prove by induction on \( \tau \) that \( \Box C = \Box C \). For \( \iota'^m \rightarrow \iota \) and \( o \) this is obvious. Hence, suppose \( \tau = \tau' \rightarrow \rho \) and let \( r, r' \in M[\tau' \rightarrow \rho] \).

First, suppose \( r \subseteq^C_{c, \tau' \rightarrow \rho} r' \). Let \( s \in C[\tau'] \) be arbitrary. By the inductive hypothesis, \( s \subseteq^C_{c, \tau' \rightarrow \rho} s \) (\( \subseteq \) is reflexive). Thus, \( \{s\} \in \text{dir}(s) \) and therefore, \( r(s) \subseteq^C_{c, \tau' \rightarrow \rho} r'(s) \). Again by the inductive hypothesis, \( r(s) \subseteq^C_{c, \tau' \rightarrow \rho} r'(s) \). Consequently, \( r(s) \subseteq^C_{c, \tau' \rightarrow \rho} r'(s) \).

Conversely, suppose \( r \subseteq^C_{c, \tau' \rightarrow \rho} r' \). Let \( s, s' \in C[\tau'] \) be arbitrary. By the inductive hypothesis, \( s \subseteq^C_{c, \tau' \rightarrow \rho} s' \) and \( s' \) is \( \subseteq \)-directed. Therefore

\[
\begin{align*}
\text{r monotone} & \quad r(s) \subseteq^C_{\tau'} r \left( \bigcup_{s' \in \mathcal{G}'} s' \right) \\
\text{r continuous} & \quad r(s) \subseteq^C_{\tau'} \bigcup \{r'(s') \mid s' \in \mathcal{G}'\} \subseteq^C_{\tau'} \bigcup \{B^C_{c, \tau'} M' \mid \alpha \in \mathcal{A}\}.
\end{align*}
\]

Again by the inductive hypothesis, \( r(s) \subseteq^C_{c, \tau' \rightarrow \rho} \bigcup \{r'(s') \mid s' \in \mathcal{G}'\} \). Hence, \( r \subseteq^C_{c, \tau' \rightarrow \rho} r' \). \( \Box \)

Lemma 4.3. Let \( \Sigma \) be a signature, \( \Delta \) be a type environment and \( B \) be a \((\Sigma, C)\)-structure.

Then for any positive existential term \( M \), \((\Delta, C)\)-valuation \( \alpha \) and \( \mathcal{A} \in \text{dir}(\alpha) \),

(i) if \( M \) is a \( \lambda \)-abstraction then \( B^C_{c, \tau'} (M)(\alpha) = B^C_{\tau'} M(\alpha) \) and

(ii) \( B^C_{\tau'} \bigcup \mathcal{A} \subseteq^C_{\tau'} \bigcup \{B^C_{\tau'} M(\alpha) \mid \alpha \in \mathcal{A}\} \).

Proof. We prove both parts of the lemma simultaneously by induction on the structure of \( M \). For all cases except \( \lambda \)-abstractions, Part (i) is trivially true and Part (ii) is proven as in Proposition 3.30.

Hence, suppose \( M \) is a \( \lambda \)-abstraction \( \Delta \vdash \lambda x. M' : \tau \rightarrow \rho \).

Claim 1. \( B^C_{c, \tau'} (M)(\alpha) \in C[\tau \rightarrow \rho] \).

Proof. First, let \( s, s' \in C[\tau] \) be such that \( s \subseteq^C s' \). Note that by Lemma 4.2 and the reflexivity of \( \subseteq^C \), \( \{s'\} \in \text{dir}(s) \) and \( \{\alpha[x \mapsto s']\} \in \text{dir}(\alpha) \). Hence, by the inductive hypothesis,

\[
B^C_{c, \tau'} (M)(\alpha)(s) = B^C_{\tau'} M'(\alpha[x \mapsto s]) \subseteq^C B^C_{\tau'} M'(\alpha[x \mapsto s]) = B^C_{\tau'} (M)(\alpha)(s')
\]

71
Consequently by Lemma 4.2, $\mathcal{B}_C^{C}(M)(\alpha)(s) \sqsubseteq C \mathcal{B}_C^{C}(M)(\alpha)(s')$ and $\mathcal{B}_C^{C}(M)(\alpha)$ is monotone.

Next, suppose that $\mathcal{G} \subseteq C[\mathcal{F}]$ is $(\subseteq^C)$-directed. Note that by Lemma 4.2, $\mathcal{G} \in \text{dir}(\bigcup \mathcal{G})$ and $\{\alpha[x \mapsto s] \mid s \in \mathcal{G}\} \in \text{dir}(\alpha[x \mapsto \bigcup \mathcal{G}])$. Therefore, by the inductive hypothesis

$$\mathcal{B}_C^{C}(M)(\alpha)(\bigcup \mathcal{G}) = \mathcal{B}_C^{C}[\alpha[x \mapsto \bigcup \mathcal{G}]](\alpha[x \mapsto \bigcup \mathcal{G}])$$

$$\sqsubseteq C \bigcup \{\mathcal{B}_C^{C}(M)(\alpha)(s) \mid s \in \mathcal{G}\}$$

Again by Lemma 4.2, $\mathcal{B}_C^{C}(M)(\alpha)(\bigcup \mathcal{G}) \sqsubseteq C \bigcup \{\mathcal{B}_C^{C}(M)(\alpha)(s) \mid s \in \mathcal{G}\}$.

Furthermore, let $s \in \mathcal{G}$. Clearly, $s \sqsubseteq C \bigcup \mathcal{G}$. Since we have already proven that $\mathcal{B}_C^{C}(M)(\alpha)$ is monotone, $\mathcal{B}_C^{C}(M)(\alpha)(s) \sqsubseteq C \mathcal{B}_C^{C}(M)(\alpha)(\bigcup \mathcal{G})$. Due to the fact that this holds for every $s \in \mathcal{G}$ and $C$ induces a lattice structure,

$$\bigcup \{\mathcal{B}_C^{C}(M)(\alpha)(s) \mid s \in \mathcal{G}\} \sqsubseteq C \mathcal{B}_C^{C}(M)(\alpha)(\bigcup \mathcal{G})$$

Consequently, by the antisymmetry of $\subseteq^C$, $\mathcal{B}_C^{C}(M)(\alpha)(\bigcup \mathcal{G}) = \bigcup \{\mathcal{B}_C^{C}(M)(\alpha)(s) \mid s \in \mathcal{G}\}$.

This concludes the proof of the claim that $\mathcal{B}_C^{C}(M)(\alpha)$ is continuous. □

As a consequence of Claim 1, $\mathcal{B}_C^{C}(\lambda x. M')(\alpha) = \mathcal{B}_C^{C}[\lambda x. M'](\alpha)$. Therefore, the same argument as in the proof of Proposition 3.30 can be used to demonstrate Part (ii). □

### B.2 Remaining Proofs for Section 4.2

**Lemma 4.8.** For every $n \in \omega$,

(i) $R^{A^H_{p,n}} \sqsubseteq_m R^{A^H_{p,2n}}$ for $R \in \Sigma' \setminus \Sigma$,

(ii) $R^{A^H_{p,n}} \sqsubseteq_m R^{A^H_{p,2n+1}}$ for $R \in \Sigma' \setminus \Sigma$ and

(iii) $A^{H}_{p,n}[\lambda \mathfrak{f}. M](\top^H_\Delta) \sqsubseteq_m R^{A^H_{p,2n+1}}_{M}$.

**Proof.** We prove the lemma by induction on $n \in \omega$.

- If $n = 0$ then the first two parts are obvious. Furthermore,

  $$A^{H}_{p,0}[\lambda \mathfrak{f}. M](\top^H_\Delta) \sqsubseteq_m A^{H}_{p,0}[\lambda \mathfrak{f}. \overline{M} \overline{\mathfrak{f}}](\top^H_\Delta) = A^{H}_{p,0}[\lambda \mathfrak{f}. \overline{M} \overline{\mathfrak{f}}](\top^H_\Delta) = R^{A^H_{p,1}}_{M}$$

  because $M$ is a $\Sigma'$-term.

- Next, to show the induction step from $n$ to $n + 1$ suppose that $n \geq 0$. By the inductive hypothesis, (i) $R^{A^H_{p,n}} \sqsubseteq_m R^{A^H_{p,2n}}$ for $R \in \Sigma' \setminus \Sigma$, (ii) $R^{A^H_{p,n}} \sqsubseteq_m R^{A^H_{p,2n+1}}$ for $R \in \Sigma' \setminus \Sigma$ and (iii) $A^{H}_{p,n}[\lambda \mathfrak{f}. M](\top^H_\Delta) \sqsubseteq_m R^{A^H_{p,2n+1}}_{M}$.

  (i) Therefore, by Parts (ii) and (iii) of the inductive hypothesis and Lemma 4.7,

  $$R^{A^H_{p,n+1}}_{M} = A^{H}_{p,n}[\lambda \mathfrak{f}_{R} F_{R}[M]](\top^H_\Delta)$$
\[ \subseteq_m A^H_{P',2n+1}[[\lambda \bar{x}. F_R[R_M \bar{x}]]]((\top_K^H) \Delta) \]

\[ = R^{A^H_{P',2n+2}} \]

for \( R \in \Sigma' \setminus \Sigma \).

(ii) By Part (i) and Lemma 3.21(ii), \( R^{A^H_{P,n+1}} \subseteq_m R^{A^H_{P',2n+2}} \subseteq_m R^{A^H_{P',2(n+1)+1}} \) for \( R \in \Sigma' \setminus \Sigma \).

(iii) Using Part (i) for \( n + 1 \) we obtain

\[ \subseteq_m A^H_{R_{n+1},[[\lambda \bar{x}. M]]((\top_K^H) \Delta) \subseteq_m A^H_{P',2(n+1)}[[\lambda \bar{x}. M]]((\top_K^H) \Delta) \]

Proposition 3.19, Part (i)

\[ = A^H_{P',2(n+1)}[[\lambda \bar{x}. \bar{y}. M \bar{y}]]((\top_K^H) \Delta) \]

Lemma 1.12(ii)

\[ = R^{A^H_{P',2(n+1)+1}}. \]

\[ \square \]

**Proposition 4.16.** Let \( H \) be a Henkin frame and \((P, F)\) be a HoCHP in normal form. Then there exists a \( \lambda \)-free HoCHP in normal form which is \((A, H)\)-equivalent to \((P, F)\).

**Proof.** We prove by induction on the number \( m \) of \( \lambda \)-abstractions that occur as subterms of atoms that \((P, F)\) is equivalent to a \( \lambda \)-free HoCHP (similar to the proof of Proposition 4.12). If \( m = 0 \) then \((P, F)\) is clearly \( \lambda \)-free.

Otherwise there exists a \( \lambda \)-abstraction \( \lambda \bar{z}. M \) which (i) occurs as a subterm of an atom in \((P, F)\) and (ii) is not a subterm of another \( \lambda \)-abstraction occurring as a subterm of an atom in \((P, F)\). By Lemma 1.10 there are (sets of) contexts with a hole \( \bar{P}[-] \) and \( F[-] \) such that \( \bar{P}[\lambda \bar{z}. M] = P \) and \( F[\lambda \bar{z}. M] = F \) and \( \lambda \bar{z}. M \) occurs neither in \( \bar{P}[-] \) nor in \( F[-] \). Let

\[ P' = \bar{P}[R_{\lambda \bar{z}. M} \bar{x}] \cup \{ \neg(\lambda \bar{z}. M)z' \bar{y} \lor R_{\lambda \bar{z}. M} \bar{x} z' \bar{y} \} \]

\[ P'' = \bar{P}[R_{\lambda \bar{z}. M} \bar{x}] \cup \{ \neg M \bar{y} \lor R_{\lambda \bar{z}. M} \bar{x} z \bar{y} \}. \]

Furthermore, let \((P'', F'')\) be the HoCHP (obtained by \( \eta \)-expansion) in which all occurrences of a term \( \exists M \) in \( P \cup \{ F \} \), \( M \) is a \( \lambda \)-abstraction. By Proposition 4.9 and Lemmas 1.12(i) and 1.12(ii) all of \((P, F), (P', F'), (P'', F'')\) and \((P''', F''')\) are \((A, H)\)-equivalent.

Since \( M \) is a subterm of an atom it cannot contain logical symbols. Hence, \( M \bar{y} \) is an atom and \((P'', F'')\) is in normal form.

Again, the number of distinct \( \lambda \)-abstractions occurring as subterms of atoms in \((P'', F'')\) is strictly smaller than \( m \). Hence, the same holds for \((P''', F''')\), which is in normal form, because only \( \lambda \)-abstractions immediately below existential quantifiers are introduced. Consequently, by the inductive hypothesis, \((P''', F''')\) is \((A, H)\)-equivalent to some \( \lambda \)-free HoCHP.

\[ \square \]

**B.3 Remaining Proofs for Section 4.4**

Before turning to Lemma 4.26, we prove the following auxiliary lemma:

**Lemma B.1.** (i) If \( R \in \Sigma' \setminus \Sigma, R M_1 \cdots M_n \) and \( R x_1 \cdots x_n \) are terms such that 

\[ \text{free}(R M_1 \cdots M_n) \cap \text{free}(R x_1 \cdots x_n) = \emptyset \] then 

\[ [[M_1]/x_1, \ldots, [M_n]/x_n] \]
is a unifier of \( RM_1 \cdots M_n \) and \( Rx_1 \cdots x_n \).

(ii) If \( \Delta(y) = \rho = \tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow o \) and \( x M_1 \cdots M_n \) is a formula then

\[
[c_\rho/x, M_1/x_{\tau_1}, \ldots, M_n/x_{\tau_n}]
\]

is a unifier of \( \text{Comp}_\rho \) and \( y M_1 \cdots M_n \).

(iii) If \( M \) is a term neither containing logical symbols nor \( \lambda \)-abstractions then

\[
[M[\bar{L}/\bar{x}]] = [M][\bar{L}/\bar{x}].
\]

(iv) If \( G \) is a goal clause then \( [G[M_1/x_1, \ldots, M_n/x_n]] = [G][[M_1]/x_1, \ldots, [M_n]/x_n] \).

Proof. (i) We prove Part (i) by induction on \( n \). For \( n = 0 \) this is trivial. Hence, suppose \( n \geq 0 \). By the inductive hypothesis,

\[
[RM_1 \cdots M_n][[M_1]/x_1, \ldots, [M_n]/x_n]
= [Rx_1 \cdots x_n][[M_1]/x_1, \ldots, [M_n]/x_n]. \tag{B.1}
\]

Consequently,

\[
[RM_1 \cdots M_{n+1}][[M_1]/x_1, \ldots, [M_{n+1}]/x_{n+1}]
= (@[RM_1 \cdots M_n][M_{n+1}][[M_1]/x_1, \ldots, [M_{n+1}]/x_{n+1}]
= (@[RM_1 \cdots M_n][[M_1]/x_1, \ldots, [M_n]/x_n]M_{n+1} \quad x_i \notin \text{free}(M_j)
= (@[Rx_1 \cdots x_n][x_{n+1}][[M_1]/x_1, \ldots, [M_{n+1}]/x_{n+1}] \quad \text{Eq. (B.1)}
= [Rx_n \cdots x_1][[M_1]/x_1, \ldots, [M_{n+1}]/x_{n+1}].
\]

(ii) Similar to Part (i).

(iii) We prove the claim by structural induction. For variables, and symbols from \( \Sigma' \setminus \Sigma \) this is obvious. Next, consider a term of the form \( cN \), where \( c \in \Sigma \). By Remark 1.2, \( cN \) only contains variables \( y : \iota^n \rightarrow \iota \) and for each term \( \Delta \vdash K : \iota^n \rightarrow \iota \), \( |K| = K \). Hence,

\[
[[cN][L/\bar{x}]] = (cN)[L/\bar{x}] = [cN][[L]/\bar{x}].
\]

Finally, consider a term of the form \( MNN' \), where \( M \notin \Sigma \). Then,

\[
[(MNN')[[L]/\bar{x}]] = [(M)[[L]/\bar{x}]]N'[[L]/\bar{x}]]
= (@[[M][[L]/\bar{x}]]N'[[[L]/\bar{x}]]
= (@[[M][[N]']][[L]/\bar{x}]] \quad \text{inductive hypothesis}
= ([[M]][[N']][[L]/\bar{x}]]
= [[M][N'][[[L]/\bar{x}]].
\]

(iv) Immediate from Part (iii). \( \square \)

Lemma 4.26 (Lifting). Let \( S \) be a set of HoCHC not containing \( \lambda \)-abstractions.

If \( S \Rightarrow_{\text{Res}} S \cup \{G\} \) and \( S' \supseteq |S| \) then there exists \( S'' \) such that (i) \( S' \Rightarrow_{\text{FO-Res}} S'' \),

(ii) \( |G| \in S'' \) and (iii) if \( x : \rho \) occurs in \( S'' \) then some \( y : \rho \) occurs in \( S \).
Proof. First note that by assumption the rule $\beta$-Reduction is not applicable. Next, let $\neg R\overrightarrow{M} \lor G$ and $G' \lor RxR$ be clauses in $S$ and suppose $S \Rightarrow_{\text{Res}} S \cup \{G \lor G'[\overrightarrow{M}/\overrightarrow{xR}]\}$. By Lemmas B.1(i) and B.1(iv),

$$S' \Rightarrow_{\text{FO-Res}} S' \cup \{([G] \lor [G'])[[\overrightarrow{M}]/\overrightarrow{xR}]\} = S' \cup \{G \lor G'[\overrightarrow{M}/\overrightarrow{xR}]\}$$

(renameing the variables in $\neg R\overrightarrow{M} \lor G$ if necessary). Furthermore,

$$\text{vars}(([G] \lor [G'])[[\overrightarrow{M}/\overrightarrow{xR}]] \subseteq \text{vars}(\neg R\overrightarrow{M} \lor G) \cup \text{vars}(RxR)$$

and therefore the additional condition is satisfied.

Finally, suppose $S \Rightarrow_{\text{Res}} S \cup \{\bot\}$ because there exists a higher-order constrained Horn clause $\neg x_1 \overrightarrow{M}_1 \lor \cdots \lor \neg x_m \overrightarrow{M}_m \lor \neg \phi_1 \lor \cdots \lor \neg \phi_n$ in $S$ and a valuation $\alpha$ such that each $\phi_i$ is background atom and $A, \alpha \models \phi_1 \land \cdots \land \phi_n$. Note that each $x_i$ does not occur in any of the $\phi_j$. Therefore, by Lemma B.1(ii) (applying the rule FO-Resolution $m$ times),

$$S' \Rightarrow_{\text{FO-Res}} S'' \cup \{\neg \phi_1 \lor \cdots \lor \neg \phi_n\}$$

and by assumption $S' \Rightarrow_{\text{FO-Res}} m' \cup \{\bot, \neg \phi_1 \lor \cdots \lor \neg \phi_n\}$ for some $S'' \supseteq S'$. \qed
Bibliography


Index

λ-abstraction, 5
atom, 14
  background atom, 14
  foreground atom, 14
  simple foreground atom, 14
compatible closure, 10
constrained Horn clause, 14
  continuous-satisfiable, 15
  Henkin-satisfiable, 15
  monotone-satisfiable, 15
  satisfiable, 15
  standard-satisfiable, 15
  unsatisfiable, 15
constrained Horn problem in program form, 14
  λ-free, 48
  equivalent, 15
  normal form, 16
  pure, 46
  solvable, 15
goal clause, 14
  simple goal clause, 14
Henkin frame, 8
  continuous Henkin frame, 8
  monotone Henkin frame, 8
  pre-Henkin frame, 5
  standard Henkin frame, 8
hierarchic specification, 11
  compact hierarchic specification, 11
  term-generated hierarchic specification, 11
higher-order constrained Horn clause,
  see constrained Horn clause
higher-order constrained Horn problem in program form,
  see constrained Horn problem in program form
HoCHC, see constrained Horn clause
HoCHP, see constrained Horn problem in program form
immediate consequence operator, 25
occur impure, 46
program, 14
quasi-continuous, 31
quasi-monotone, 27
signature, 4
  first-order signature, 4
  many-sorted (first-order) signature, 11
structure, 7
  first-order structure, 8
term-generated structure, 11

83
symbol, 4
  logical symbol, 4

term, 5
  λ-free positive existential term, 48
  first-order term, 5
  ground term, 5
  positive existential term, 5
  pre-term, 4

theorem proving for hierarchic first-order theories, 11

two-counter machine, 17
  accepted, 17
  configuration, 17

final state, 17
  initial state, 17
  run, 17
  state, 17
  transition function, 17

type, 4
  argument type, 4
  first-order type, 4
  relational type, 4

type environment, 4

unifier, 9

well-typed, 5

well-typed substitution, 9
## Conventions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathfrak{A}$</td>
<td>Sets of valuations</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Valuations</td>
</tr>
<tr>
<td>$A, B$</td>
<td>Structures</td>
</tr>
<tr>
<td>$\mathfrak{B}$</td>
<td>Sets of structures</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Type environments</td>
</tr>
<tr>
<td>$\mathfrak{H}, \mathfrak{S}, \mathfrak{M}, \mathfrak{C}$</td>
<td>Sets of Henkin frames</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Background atoms</td>
</tr>
<tr>
<td>$\mathfrak{R}, \mathfrak{S}$</td>
<td>Subsets of $\mathfrak{H}[\tau]$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Relational types</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>Signatures</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Types</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Argument types</td>
</tr>
<tr>
<td>$A$</td>
<td>Atoms</td>
</tr>
<tr>
<td>$\mathcal{C}$</td>
<td>Clauses</td>
</tr>
<tr>
<td>$c$</td>
<td>Symbols from the signature</td>
</tr>
<tr>
<td>$D$</td>
<td>Definite clauses</td>
</tr>
<tr>
<td>$E, F$</td>
<td>Positive existential formulas</td>
</tr>
<tr>
<td>$G$</td>
<td>Goal clauses</td>
</tr>
<tr>
<td>$K, L, M, N, O, Q$</td>
<td>Terms</td>
</tr>
<tr>
<td>$P$</td>
<td>Programs</td>
</tr>
<tr>
<td>$r, s$</td>
<td>Elements of $\mathfrak{H}[\tau]$</td>
</tr>
<tr>
<td>$R, U$</td>
<td>Relational symbols from $\Sigma \setminus \Sigma$</td>
</tr>
<tr>
<td>$S$</td>
<td>Sets of HoCHCs</td>
</tr>
<tr>
<td>$x, y, z, w$</td>
<td>Variables</td>
</tr>
</tbody>
</table>