A fixed-point theorem for Horn formula equations

Stefan Hetzl   Johannes Kloibhofer[*]

Institute of Discrete Mathematics and Geometry

TU Wien
Vienna, Austria

stefan.hetzl@tuwien.ac.at   j.kloibhofer@gmx.net

We consider constrained Horn clause solving from the more general point of view of solving formula equations. Constrained Horn clauses correspond to the subclass of Horn formula equations. We state and prove a fixed-point theorem for Horn formula equations which is based on expressing the fixed-point computation of a minimal model of a set of Horn clauses on the object level as a formula in first-order logic with a least fixed point operator. We describe several corollaries of this fixed-point theorem, in particular concerning the logical foundations of program verification, and sketch how to generalise it to incorporate abstract interpretations.

1 Introduction

Constrained Horn clauses are a versatile and practical formalism for representing and solving a variety of problems in program verification and model checking [4, 16]. In this paper we approach constrained Horn clause solving from a theoretical point of view. In logic, related problems have a long history: solving Boolean equations goes back to the 19th century and was already investigated in [31], see [29] for a comprehensive textbook. Solving Boolean equations is closely related to Boolean unification, a subject of thorough study in computer science, see, e.g., [25] for a survey. The generalisation of this problem from propositional to first-order logic has been made explicit as early as [2, 3]. Solving a formula equation in first-order logic is closely related to second-order quantifier elimination, a problem with applications in a variety of areas in computer science, e.g., databases or common-sense reasoning [13]. A seminal work on second-order quantifier elimination and basis of several algorithms still in use today is Ackermann's [1]. See [34] for a recent survey of this area of related problems.

Solving constrained Horn clauses is closely related to solving formula equations. In fact, constrained Horn clauses correspond to a natural class of formula equations which we will call Horn formula equations. This relationship allows for an elegant theoretical description of the connections between problems considered from Ackermann to contemporary verification.

Fixed-point theorems play an important role for solving equations in many areas of mathematics. In recursion theory, a subject with close ties to verification, the recursion theorem guarantees the existence of a solution of a system of recursion equations by computing a fixed point. But also in areas quite remote from verification similar constructions can be found, as in the use of Banach's fixed-point theorem in the proof of the Picard-Lindelöf theorem on the unique solvability of ordinary differential equations. In constrained Horn clause solving we have a comparable situation: it is a well-known result from logic programming and constraint logic programming that every set of Horn clauses has a unique minimal model (in the sense of identifying a model with the ground atoms true in it) and that this minimal model can be computed as the fixed-point of an operator induced by the clause set [32, 21].

In this paper we formulate and prove a fixed-point theorem for Horn formula equations which (essentially) makes the construction of the minimal model explicit *in the logic*. Expressing this construction will be achieved by using FO[LFP], first-order logic with a least fixed-point operator, thus providing a canonical solution for any Horn formula equation. The fixed-point theorem has a number of applications: it helps to explain, at least from a theoretical point of view, the efficacy of interpolation for Horn clause solving and invariant generation. Moreover, as a simple corollary one can obtain the expressibility of the weakest precondition and the strongest postcondition, and thus the partial correctness of an imperative program in FO[LFP]. As another corollary it allows a generalisation of a result by Ackermann [1] on second-order quantifier-elimination in a direction different from the recent generalisation [33]. A result from a recently introduced approach to inductive theorem proving with tree grammars described in [10] on generating a proof with induction based on partial information about that proof can be obtained from our fixed-point theorem as another straightforward corollary. Last, but not least, an abstract form of the fixed-point theorem, stated here as Conjecture 24, would permit to considerably simplify the proof of the decidability of affine formula equations [18].

In Section 2 we relate constrained Horn clause solving with solving formula equations. The fixed-point theorem is stated and proved in detail in Section 3. Section 4 describes some of its applications to the foundations of program verification. In Section 5 we sketch how to generalise our fixed-point theorem to accommodate abstract interpretation based on Galois connections. This paper is an improved presentation of some of the main results of the second authors master's thesis [24].

2   Constrained Horn clauses and formula equations

We use standard notation from logic. In particular, for a first-order language $\mathscr{L}$, an $\mathscr{L}$-structure $\mathscr{M}$, an $\mathscr{L}$-formula φ, and an interpretation I of the free variables of φ in $\mathscr{M}$ we write $\mathscr{M}, I \models \varphi$ to express that φ is true in $\mathscr{M}$ under the interpretation I (in the usual sense of Tarski semantics). Occasionally we will, in a slight abuse of notation, also allow elements of $\mathscr{M}$ or relations over $\mathscr{M}$ to appear on the right-hand side of $\models$ by which we intend to denote truth in $\mathscr{M}$ under an accordingly modified interpretation I. We write $\models \varphi$ to express that φ is a valid formula. Individual variables will be denoted by lowercase Latin letters $x, y, z, u, v \ldots$. Predicate variables will be denoted by uppercase Latin letters $X, Y, \ldots$. If X is a k-ary predicate variable and χ is a first-order formula with the free variables $v_1, \ldots, v_k$ we write $[X \backslash \chi]$ for the substitution of X by χ inserting the i-th argument of an X-atom for v_i. We write $[X_1 \backslash \chi_1, \ldots, X_n \backslash \chi_n]$ for the simultaneous substitution of X_i by χ_i for $i = 1, \ldots, n$. A substitution $[X_1 \backslash \chi_1, \ldots, X_n \backslash \chi_n]$ is called *first-order substitution* if $\chi_1, \ldots, \chi_n$ are first-order formulas. The logical symbol $\bot$ is a nullary predicate constant which is false in all structures.

Let $\mathscr{L}$ be a first-order language and let T be an $\mathscr{L}$-theory. A *constrained Horn clause* is an $\mathscr{L}$-formula of the form $\varphi \wedge X_1(\overline{t_1}) \wedge \cdots \wedge X_n(\overline{t_n}) \rightarrow Y(\overline{s})$ or $\varphi \wedge X_1(\overline{t_1}) \wedge \cdots \wedge X_n(\overline{t_n}) \rightarrow \bot$ where $\overline{t_1}, \ldots, \overline{t_n}, \overline{s}$ are tuples of first-order terms of appropriate arity and φ is a first-order formula, i.e., a formula not containing predicate variables. Note that a constrained Horn clause is allowed to (and typically does) contain free individual variables which, as usual in clause logic, are treated as universally quantified. A finite set S of constrained Horn clauses is considered as the conjunction of these clauses and is thus logically equivalent to a formula of the form $\forall^* \bigwedge_{C \in S} C$ where $\forall^*$ denotes the universal closure w.r.t. individual variables. We are interested in solving a given finite set of constrained Horn clauses. There are different notions of solvability in the literature which we will discuss in detail below.

In this paper we embed constrained Horn clauses in the more general framework of formula equa-

tions. In the context of logical formulas, we consider an equation to be

$$\text{an } \mathscr{L}\text{-formula } \varphi_1 \leftrightarrow \varphi_2 \text{ containing predicate variables } \overline{X} = X_1, \ldots, X_n. \tag{1}$$

A solution of (1) is a first-order substitution $[\overline{X}\backslash\overline{\chi}]$ s.t. $\models (\varphi_1 \leftrightarrow \varphi_2)[\overline{X}\backslash\overline{\chi}]$. A solution in T has to satisfy $T \models (\varphi_1 \leftrightarrow \varphi_2)[\overline{X}\backslash\overline{\chi}]$ instead. Since the concept of solution in T subsumes that of solution (by setting $T = \emptyset$) we will only use the former. We can simplify equations to instead considering

$$\text{an } \mathscr{L}\text{-formula } \varphi \text{ containing predicate variables } \overline{X} = X_1, \ldots, X_n. \tag{2}$$

Then, as a solution in T, we ask for a first-order substitution $[\overline{X}\backslash\overline{\chi}]$ s.t. $T \models \varphi[\overline{X}\backslash\overline{\chi}]$. Note that every instance of (1) is an instance of (2) by letting φ be $\varphi_1 \leftrightarrow \varphi_2$ and every instance of (2) is an instance of (1) by letting φ_1 be φ and φ_2 be $\top$. Moreover, it will be notationally useful to explicitly indicate the predicate variables by existential quantifiers. Consequently we define:

Definition 1. A *formula equation* is a closed $\mathscr{L}$-formula $\exists \overline{X} \, \varphi$ where φ contains only first-order quantifiers. A *solution* of $\exists \overline{X} \, \varphi$ in T is a first-order substitution $[\overline{X}\backslash\overline{\chi}]$ s.t. $T \models \varphi[\overline{X}\backslash\overline{\chi}]$.

The problem of computing a solution to a formula equation given as input will be denoted as FEQ in the sequel. A formula equation $\exists \overline{X} \, \varphi$ is called *valid* if it is a valid second-order formula and *satisfiable* if it is a satisfiable second-order formula. If S is a set of constrained Horn clauses in the predicate variables $\overline{X} = X_1, \ldots, X_n$, then $\exists \overline{X} \forall^* \bigwedge_{C \in S} C$ will be called *Horn formula equation*. Thus constrained Horn clauses correspond to existential second order Horn logic, which also plays a significant role in finite model theory, see [14].

There are different notions of solvability for constrained Horn clauses in the literature: *satisfiability* of [16] is satisfiability of a Horn formula equation, *semantic solvability* of [30] is validity of a Horn formula equation, and *syntactic solvability* of [30] is solvability of a Horn formula equation. In this paper we will primarily be interested in this last notion: solvability of a Horn formula equation. Every solvable formula equation is valid and every valid formula equation is satisfiable but neither of the converse implications are true as the following example shows.

Example 2. Let τ be the formula $\exists x_1 \exists x_2 \, x_1 \neq x_2$ in the empty language in first-order logic with equality and let $\sigma(X)$ be the formula $\exists u \, X(u) \wedge \exists v \, \neg X(v)$.

The formula equation $\exists X \, \sigma(X)$ is satisfiable but not valid. For satisfiability take, e.g., the two-element structure and pick any singleton set for X. As a countermodel to validity take the one-element structure and observe that there is no subset S of the domain which satisfies $\sigma(S)$.

The formula equation $\exists X \, (\tau \to \sigma(X))$ is valid but not solvable. For validity note that every model $\mathscr{M}$ of τ has cardinality at least 2 and thus there is a set different from $\emptyset$ and different from $\mathscr{M}$. However, the only parameter-free first-order definable sets in the empty language in a structure $\mathscr{M}$ are $\emptyset$ and $\mathscr{M}$ itself. As a countermodel to solvability it suffices to consider the two-element structure $\mathscr{M}_2$: any solution $\psi(u)$ would satisfy $\mathscr{M}_2 \models \exists u \, \psi(u) \wedge \exists v \, \neg \psi(v)$ thus providing a first-order definition of a set in $\mathscr{M}_2$ which is neither $\emptyset$ nor $\mathscr{M}_2$. Contradiction.

Solving formula equations (FEQ) is closely related to the problem of second-order quantifier elimination (SOQE): given a formula $\exists \overline{X} \, \varphi$ where φ contains only first-order quantifiers find a first-order formula ψ s.t. $\models \exists \overline{X} \, \varphi \leftrightarrow \psi$. The relationship between FEQ and SOQE often hinges on a third problem: second-order quantifier elimination by a witness (WSOQE): given a formula $\exists \overline{X} \, \varphi$ where φ contains only first-order quantifiers find a first-order substitution $[\overline{X}\backslash\overline{\chi}]$ s.t. $\models \exists \overline{X} \, \varphi \leftrightarrow \varphi[\overline{X}\backslash\overline{\chi}]$, see [34] for more details. This complex of problems has a long history in logic and a wealth of applications in computer science, see the textbook [13] on second-order quantifier elimination. A number of algorithms for

second-order quantifier elimination have been developed, for example: The SCAN algorithm introduced in [12] (tries to) compute(s) a first-order formula equivalent to $\exists X \varphi$ for a conjunctive normal form φ by forming the closure of φ under constraint resolution which only resolves on X-literals. The DLS algorithm has been introduced in [8] and consists essentially of formula rewriting steps tailored to allow application of Ackermann's lemma (which instantiates a predicate variable provided some conditions on the polarity of its occurrences are met).

3   The fixed-point theorem

It is well-known that a set of Horn clauses has a minimal model and that it can be obtained as least fixed point of an operator induced by the clause set. In order to integrate this insight seamlessly into the framework of formula equations we will express it on the object level by representing this least fixed-point by means of an explicit least fixed-point operator. An adequate tool to that end is first-order logic with least fixed points, FO[LFP], which plays an important role in finite model theory and descriptive complexity [20].

In order to introduce FO[LFP] we first define, by induction on a formula φ, what it means for a predicate variable X to occur positively (negatively) in φ:

1. If φ is an atom of the form $X(\bar{t})$ then X occurs positively in φ.

2. X occurs positively (negatively) in $\neg\varphi$ iff X occurs negatively (positively) in φ.

3. For $\circ \in \{\wedge, \vee\}$, X occurs positively (negatively) in $\varphi \circ \psi$ iff X occurs positively (negatively) in φ or X occurs postively (negatively) in ψ.

4. X occurs positively (negatively) in $\varphi \to \psi$ iff X occurs positively (negatively) in ψ or X occurs negatively (positively) in φ.

5. For $Q \in \{\forall, \exists\}$, X occurs positively (negatively) in $Qx\,\varphi$ iff X occurs positively (negatively) in φ.

Example 3. X occurs positively but not negatively in $\forall u\,(P(u) \to X(u))$. X occurs both positively and negatively in $\forall u\,(X(u) \to X(s(u)))$.

FO[LFP] is first-order logic, augmented with a least fixed point operator lfp which allows to add new predicates to the logic that have the form $[\mathrm{lfp}_X\,\varphi(X, \bar{u})]$ where φ is a formula in which X occurs only positively and the length of $\bar{u}$ is the arity of X. Then φ defines a monotone function $F_\varphi : \mathscr{P}(M)^k \to \mathscr{P}(M)^k$ and since the power set lattice $\mathscr{P}(M)^k$ is complete, the Knaster-Tarski theorem applies and the least fixed point of F_φ is well-defined. The predicate $[\mathrm{lfp}_X\,\varphi(X, \bar{u})]$ is interpreted as that least fixed point.

Example 4. Working in the language $\{E/2\}$ of graphs, let X be a binary predicate variable and define

$$\varphi(X, u, v) \quad \equiv \quad u = v \vee \exists w(X(u,w) \wedge E(w,v)).$$

As X occurs only positively in φ we can form $[\mathrm{lfp}_X\,\varphi(X, u, v)]$ and observe that $[\mathrm{lfp}_X\,\varphi(X, u, v)](a, b)$ is true iff there is a path from a to b.

In this paper we will consider first-order logic with an operator for simultaneous least fixed points (which corresponds to introducing mutual recursion). Then we require a tuple $\Phi = (\varphi_i(X_1, \ldots, X_n, \bar{u_i}))_{i=1}^n$ of formulas containing the X_i only positively where the length of $\bar{u_i}$ is the arity k_i of X_i. For a structure $\mathscr{M}$ with domain M we define

$$
\begin{aligned}
F_i : \quad & M^{k_1} \times \cdots \times M^{k_n} \to M^{k_i}, \\
& (R_1, \ldots, R_n) \mapsto \{\bar{a} \in M^{k_i} \mid \mathscr{M} \models \varphi_i(R_1, \ldots, R_n, \bar{a})\}.
\end{aligned}
$$

and the operator $F_\Phi = (F_1, ..., F_n) : M^{k_1} \times \cdots \times M^{k_n} \to M^{k_1} \times \cdots \times M^{k_n}$. Since the X_i occur only positively in the φ_j, the operator F_Φ is monotone and, again, the Knaster-Tarski theorem applies. Thus we obtain new predicates $[\text{lfp}_{X_i}\Phi]$ for $i = 1, \ldots, n$ which are interpreted by the i-th component of the least fixed point of the operator F_Φ. For more details, the reader is referred to [7].

Let $\exists X_1 \cdots \exists X_n \psi$ be a Horn formula equation. We distinguish three different types of clauses in ψ:

$$
\begin{array}{lrl}
(B) & \varphi & \to X_{i_0}(\bar{s}), \\
(I) & \varphi \wedge X_{i_1}(\bar{t_1}) \wedge \cdots \wedge X_{i_m}(\bar{t_m}) & \to X_{i_0}(\bar{s}), \\
(E) & \varphi \wedge X_{i_1}(\bar{t_1}) \wedge \cdots \wedge X_{i_m}(\bar{t_m}) & \to \bot,
\end{array}
$$

where the constraint φ is a formula in $\mathscr{L}$ not containing a predicate variable, $m \geq 1, \bar{t_1}, .., \bar{t_m}, \bar{s}$ are tuples of first-order terms in $\mathscr{L}$ of appropriate arity and $i_0, i_1, \ldots, i_m \in \{1, \ldots, n\}$. Note that free variables $\bar{y}$ may occur in the formulas φ and the terms $\bar{s}, \bar{t_1}, \ldots, \bar{t_m}$. We call the first base clauses, the second induction clauses, and the third end clauses. The idea now is to build an inductive relation from the base and induction clauses for every formula variable. For $j = 1, \ldots, n$ let B_j and I_j be the sets of clauses of the form (B) and (I), respectively, where $i_0 = j$. In order to abbreviate notation we write $\iota := (i_1, \ldots, i_m)$ and $\tau := (\bar{t_1}, \ldots, \bar{t_m})$. A clause in I_j is determined by the tuple $(\varphi, \iota, \tau, \bar{s})$, thus we write $(\varphi, \iota, \tau, \bar{s})$ for the clause $\varphi \wedge X_{i_1}(\bar{t_1}) \wedge \cdots \wedge X_{i_m}(\bar{t_m}) \to X_j(\bar{s})$ in I_j. Analogously we write $(\varphi, \bar{s})$ for the clause $\varphi \to X_j(\bar{s})$ in B_j.

Definition 5. Let $\exists X_1 \cdots \exists X_n \psi$ be a Horn formula equation. Define the n-tuple $\Phi_\psi = (\varphi_j(X_1, \ldots, X_n, \overline{x_j}))_{j=1}^n$ where, for $j = 1, \ldots, n$,

$$
\varphi_j(X_1, \ldots, X_n, \overline{x_j}) \quad \text{is} \quad \exists \bar{y} \left(\bigvee_{(\varphi, \bar{s}) \in B_j} (\varphi \wedge \overline{x_j} = \bar{s}) \vee \bigvee_{(\varphi, \iota, \tau, \bar{s}) \in I_j} \left(\varphi \wedge \bigwedge_{k=1}^m X_{i_k}(\bar{t_k}) \wedge \overline{x_j} = \bar{s} \right) \right)
$$

where $\bar{y}$ are the free variables of the clauses in $B_j \cup I_j$ and $\overline{x_j}$ is a tuple of variables s.t. $|\overline{x_j}|$ equals the arity of X_j.

Note that $X_1, \ldots, X_n$ only occur positively in Φ_ψ, hence we can introduce the simultaneous fixed-point formulas $[\text{lfp}_{X_j} \Phi_\psi]$ for $j \in \{1, \ldots, n\}$.

Lemma 6. *Let $\exists X_1 \cdots \exists X_n \psi$ be a Horn formula equation and $\mu_j := [\text{lfp}_{X_j} \Phi_\psi]$ for $j \in \{1, \ldots, n\}$, then*

1. $\models \exists \overline{X} \psi \leftrightarrow \psi[\overline{X} \backslash \overline{\mu}]$ *and*

2. *if $\mathscr{M} \models \psi[\overline{X} \backslash \overline{R}]$ for some structure $\mathscr{M}$ and relations $R_1, \ldots, R_n$ in $\mathscr{M}$, then $\mathscr{M} \models \bigwedge_{j=1}^n (\mu_j \to R_j)$.*

Proof. The right-to-left direction of 1 is clear. For the left-to-right direction we first observe that the formulas $\mu_1, \ldots, \mu_n$ satisfy all clauses in (B) and (I), i.e., for all $j \in \{1, \ldots, n\}$ we have

$$
\begin{array}{ll}
\models \forall \bar{y} (\varphi \to \mu_j(\bar{s})), & \forall (\varphi, \bar{s}) \in B_j, \\
\models \forall \bar{y} (\varphi \wedge \mu_{i_1}(\bar{t_1}) \wedge \cdots \wedge \mu_{i_m}(\bar{t_m}) \to \mu_j(\bar{s})), & \forall (\varphi, \iota, \tau, \bar{s}) \in I_j.
\end{array}
$$

To see this let $\mathscr{M}$ be a structure and $\bar{a}$ s.t. $\mathscr{M}, [\bar{y} := \bar{a}] \models \varphi \wedge \mu_{i_1}(\bar{t_1}) \wedge \cdots \wedge \mu_{i_m}(\bar{t_m})$, then, as $(\mu_1, \ldots, \mu_n)$ is a fixed point of F_Φ, we have $\mathscr{M}, [\bar{y} := \bar{a}] \models \mu_j(\bar{s})$. The argumentation is analogous for clauses of the form (B).

Now let $\mathscr{M}$ be a structure s.t. $\mathscr{M} \models \exists \overline{X} \psi$. Let $R_1, \ldots, R_n$ be relations in $\mathscr{M}$ s.t. $\mathscr{M} \models \psi[\overline{X} \backslash \overline{R}]$. Then for all $j \in \{1, \ldots, n\}$:

$$
\begin{array}{lll}
\mathscr{M} \models \forall \bar{y} (\varphi \to R_j(\bar{s})), & \forall (\varphi, \bar{s}) \in B_j, & (3) \\
\mathscr{M} \models \forall \bar{y} (\varphi \wedge R_{i_1}(\bar{t_1}) \wedge \cdots \wedge R_{i_m}(\bar{t_m}) \to R_j(\bar{s})), & \forall (\varphi, \iota, \tau, \bar{s}) \in I_j. & (4)
\end{array}
$$

Assume $\overline{a_j} \in F_\Phi(R_1,\ldots,R_n)_j$. Then there either exists $(\varphi,\overline{s}) \in B_j$ s.t. $\mathscr{M} \models \exists \overline{y}\,(\varphi \wedge \overline{a_j} = \overline{s})$ or there exists $(\varphi,\iota,\tau,\overline{s}) \in I_j$ s.t. $\mathscr{M},[\overline{X} := \overline{R}] \models \exists \overline{y}\,(\varphi \wedge \bigwedge_{k=1}^{m} X_{i_k}(\overline{t_k}) \wedge \overline{a_j} = \overline{s})$. We assume the latter, the proof for the former is analogous. Thus let $\overline{a}$ be s.t.

$$\mathscr{M},[\overline{X} := \overline{R}, \overline{y} := \overline{a}] \models \varphi \wedge \bigwedge_{k=1}^{m} X_{i_k}(\overline{t_k}) \wedge \overline{x_j} = \overline{s}.$$

From (4) we obtain $\mathscr{M},[\overline{y} := \overline{a}] \models \chi_j(\overline{s})$ and thus $\mathscr{M} \models \chi_j(\overline{a_j})$.

Hence $F_\Phi(R_1,\ldots,R_n) \subseteq (R_1,\ldots,R_n)$ and as $(\mu_1,\ldots,\mu_n)$ is the least fixed point of F_Φ we obtain $\mathscr{M} \models \bigwedge_{j=1}^{n}(\mu_j \to R_j)$.

For all clauses in (E) we have

$$\mathscr{M},[\overline{X} := \overline{R}] \models \forall \overline{y}\,(\varphi \wedge X_{i_1}(\overline{t_1}) \wedge \cdots \wedge X_{i_m}(\overline{t_m}) \to \bot),$$

and therefore, as $X_1,\ldots,X_n$ only occur positively in Φ_ψ we obtain

$$\mathscr{M},[\overline{X} := \overline{\mu}] \models \forall \overline{y}\,(\varphi \wedge X_{i_1}(\overline{t_1}) \wedge \cdots \wedge X_{i_m}(\overline{t_m}) \to \bot).$$

Thus $\mathscr{M},[\overline{X} := \overline{\mu}]$ satisfies all clauses in ψ and we conclude that $\mathscr{M} \models \psi[\overline{X}\backslash\overline{\mu}]$. For 2. we get $\mathscr{M} \models \bigwedge_{j=1}^{n}(\mu_j \to R_j)$ analogously to the proof of 1. $\qquad\square$

Theorem 7 (Fixed-Point theorem). *Let $\exists X_1 \cdots \exists X_n\,\psi$ be a valid Horn formula equation and let $\mu_j = [\mathrm{lfp}_{X_j}\,\Phi_\psi]$ for $j \in \{1,\ldots,n\}$. Then:*

1. $\models \psi[\overline{X}\backslash\overline{\mu}]$,
2. *If $\models \psi[\overline{X}\backslash\overline{\chi}]$ for FO[LFP]-formulas $\chi_1,\ldots,\chi_n$, then $\models \bigwedge_{j=1}^{n}(\mu_j \to \chi_j)$.*

Proof. Follows immediately from Lemma 6. $\qquad\square$

We now turn to dual and linear Horn formula equations. A dual constrained Horn clause is an L-formula of the form $\varphi \wedge X(\overline{t}) \to Y_1(\overline{s_1}) \vee \cdots \vee Y_n(\overline{s_n})$ or $\varphi \to Y_1(\overline{s_1}) \vee \cdots \vee Y_n(\overline{s_n})$ where $\overline{t},\overline{s_1},\ldots,\overline{s_n}$ are tuples of first-order terms of appropriate arity and φ is a first-order formula, i.e., a formula not containing predicate variables. A *dual Horn formula equation* is a formula equation of the form $\exists \overline{X}\forall \overline{y}\,\bigwedge_{i=1}^{m} H_i$, where H_i is a constrained dual Horn clause for $i \in \{1,\ldots,m\}$. A *linear Horn formula equation* is a formula equation that is both Horn and dual Horn.

For a formula ψ we define ψ^D as $\psi[X_1\backslash\neg X_1,\ldots,X_n\backslash\neg X_n]$ where $X_1,\ldots,X_n$ are all predicate variables occurring in ψ. Note that $\models \psi \leftrightarrow (\psi^D)^D$ for all formulas ψ. Moreover, note that $\models \exists \overline{X}\,\psi \leftrightarrow \exists \overline{X}\,\psi^D$ where $\overline{X} = X_1,\ldots,X_n$ are all predicate variables occurring in ψ. If $\exists \overline{X}\,\psi$ is a Horn formula equation, then $\exists \overline{X}\,\psi^D$ is logically equivalent to a dual Horn formula equation and if $\exists \overline{X}\,\varphi$ is a dual Horn formula equation, then $\exists \overline{X}\,\varphi^D$ is logicall equivalent to a Horn formula equation. Note that dualisation of a (dual) Horn formula equation interchanges (B)- and (E)-clauses.

Example 8. Consider the constrained Horn clauses

$$\psi \quad\equiv\quad X(a) \wedge (X(u) \wedge X(v) \to Y(f(u,v))) \wedge (Y(w) \to \bot).$$

The dualisation of ψ is

$$\psi^D \quad\equiv\quad \neg X(a) \wedge (\neg X(u) \wedge \neg X(v) \to \neg Y(f(u,v))) \wedge (\neg Y(w) \to \bot)$$

which is logically equivalent to the dual constrained Horn clauses

$$(X(a) \to \bot) \wedge (Y(f(u,v)) \to X(u) \vee X(v)) \wedge Y(w).$$

We can now prove the following result which is dual to Lemma 6.

Lemma 9. *Let $\exists X_1 \cdots \exists X_n \, \psi$ be a dual Horn formula equation and $v_j := \neg[\mathrm{lfp}_{X_j} \, \Phi_{\psi^D}]$ for $j \in \{1,\ldots,n\}$, then*

1. $\models \exists \overline{X} \, \psi \leftrightarrow \psi[\overline{X} \backslash \overline{v}]$ *and*

2. *if $\mathscr{M} \models \psi[\overline{X} \backslash \overline{R}]$, for a structure $\mathscr{M}$ and relations $R_1,\ldots,R_n$ in $\mathscr{M}$, then $\mathscr{M} \models \bigwedge_{j=1}^{n}(R_j \to v_j)$.*

Proof. Let $\mu_j := [\mathrm{lfp}_{X_j} \, \Phi_{\psi^D}]$ for $j = 1,\ldots,n$. For 1. note that, since $\exists \overline{X} \, \psi$ is a dual Horn formula equation, $\exists \overline{X} \, \psi^D$ is logically equivalent to a Horn formula equation. An application of Lemma 6/1. yields $\models \exists \overline{X} \, \psi^D \leftrightarrow \psi^D[\overline{X} \backslash \overline{\mu}]$. Since $\psi^D[\overline{X} \backslash \overline{\mu}]$ is syntactically equal to $\psi[\overline{X} \backslash \overline{v}]$ we obtain $\models \exists \overline{X} \, \psi \leftrightarrow \exists \overline{X} \, \psi^D \leftrightarrow \psi^D[\overline{X} \backslash \overline{\mu}] \leftrightarrow \psi[\overline{X} \backslash \overline{v}]$.

For 2. assume that $\mathscr{M} \models \psi[\overline{X} \backslash \overline{R}]$ for a structure $\mathscr{M}$ and relations $R_1,\ldots,R_n$ in $\mathscr{M}$. Then $\mathscr{M} \models \psi^D[X_1 \backslash R_1^{\mathrm{c}},\ldots,X_n \backslash R_n^{\mathrm{c}}]$, so, by Lemma 6/2., $\mathscr{M} \models \bigwedge_{j=1}^{n}(\mu_j \to R_j^{\mathrm{c}})$ which yields $\mathscr{M} \models \bigwedge_{j=1}^{n}(R_j \to v_j)$ by contraposition. $\qquad\square$

Theorem 10 (Dual Horn fixed-point theorem)**.** *Let $\exists X_1 \cdots \exists X_n \psi$ be a valid dual Horn formula equation and let $v_j = \neg[\mathrm{lfp}_{X_j} \, \Phi_{\psi^D}]$ for $j = 1,\ldots,n$, then*

1. $\models \psi[\overline{X} \backslash \overline{v}]$ *and*

2. *if $\models \psi[\overline{X} \backslash \overline{\chi}]$ for FO[LFP]-formulas $\chi_1,\ldots,\chi_n$, then $\models \bigwedge_{j=1}^{n}(\chi_j \to v_j)$.*

Proof. Follows immediately from Lemma 9. $\qquad\square$

Note that the operator induced by Φ_{ψ^D} is not the dual operator of the one induced by Φ_ψ in the sense of [11] because Φ_{ψ^D} is not the (pointwise) negation of Φ_ψ. Therefore v is not the greatest fixed point of Φ_ψ. The question whether v permits a sensible definition as a greatest fixed point is left as future work by this paper. For the case of linear Horn formula equations we obtain:

Theorem 11 (Linear Horn fixed-point theorem)**.** *Let $\exists X_1 \cdots \exists X_n \, \psi$ be a valid linear Horn formula equation, let $\mu_j = [\mathrm{lfp}_{X_j} \, \Phi_\psi]$ and $v_j = \neg[\mathrm{lfp}_{Y_j} \, \Phi_{\psi^D}]$ for $j = 1,\ldots,n$, then*

1. $\models \psi[\overline{X} \backslash \overline{\mu}]$ *and* $\models \psi[\overline{X} \backslash \overline{v}]$ *and*

2. *if $\models \psi[\overline{X} \backslash \overline{\chi}]$ for FO[LFP]-formulas $\chi_1,\ldots,\chi_n$, then $\models \bigwedge_{j=1}^{n}((\mu_j \to \chi_j) \wedge (\chi_j \to v_j))$.*

Theorem 11/2. shows that solving a linear Horn formula equation is equivalent to solving an interpolation problem in FO[LFP] in the sense that, given two tuples of FO[LFP]-formulas we seek to find a tuple of first-order, i.e., *fixed-point free*, formulas which is between them in the implication ordering. At least from a theoretical point of view this result helps to explain the efficacy of interpolation-based methods for solving constrained Horn clauses, see, e.g., [27]. The relationship between interpolation and Horn clauses has also been studied by encoding interpolation problems with a language condition on the constant symbols as Horn clause sets [30, 15].

4   Applications to program verification

In this section we will describe some direct applications of our fixed-point theorem to the foundations of program verification. As an exemplary framework we will consider the Hoare calculus for

a simple imperative programming language as in [35]. We fix the first-order language of arithmetic $\mathscr{L} = \{0,1,+,-,\cdot,\leq\}$. The set of programs is defined by

$$p ::= \mathbf{skip} \mid x := t \mid p_0; p_1 \mid \mathbf{if}\ B\ \mathbf{then}\ p_0\ \mathbf{else}\ p_1 \mid \mathbf{while}\ B\ \mathbf{do}\ p_0,$$

where t is an $\mathscr{L}$-term, B a quantifier-free first-order formula in $\mathscr{L}$ and x is a program variable.

The denotational semantics of programs is defined as usual based on a set of states Σ: we write $C(p)$ for the partial function from Σ to Σ that is the denotational semantics of the program p. A Hoare triple is written as $\{\varphi\}p\{\psi\}$. For the purposes of this paper we fix the program variables to taking values in the integers and thus we can work in the standard model $\mathbb{Z}$. We write $\sigma \models \{\varphi\}p\{\psi\}$ if $\mathbb{Z},\sigma \models \varphi$ implies that $\mathbb{Z},C(p)(\sigma) \models \psi$ and $\models \{\varphi\}p\{\psi\}$ if $\sigma \models \{\varphi\}p\{\psi\}$ for all $\sigma \in \Sigma$. The Hoare calculus can be defined as usual, see, e.g. [35]. We write $\vdash \{\varphi\}p\{\psi\}$ if $\{\varphi\}p\{\psi\}$ is provable in the Hoare calculus. We can then consider the verification condition of a Hoare triple as a Horn formula equation as follows:

Definition 12. The *verification condition* of a Hoare triple $\{\varphi\}p\{\psi\}$, written $\mathrm{vc}(\{\varphi\}p\{\psi\})$, is a formula equation $\exists \bar{I} \forall^* \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$, where $\tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$ is defined by structural induction on p as follows:

$$\tilde{\mathrm{vc}}(\{\varphi\}\mathbf{skip}\{\psi\}) = \varphi \to \psi \tag{1}$$

$$\tilde{\mathrm{vc}}(\{\varphi\}x_j := t\{\psi\}) = \varphi \to \psi[x_j \backslash t] \tag{2}$$

$$\tilde{\mathrm{vc}}(\{\varphi\}p_0; p_1\{\psi\}) = \tilde{\mathrm{vc}}(\{\varphi\}p_0\{I\}) \wedge \tilde{\mathrm{vc}}(\{I\}p_1\{\psi\}) \tag{3}$$

$$\tilde{\mathrm{vc}}(\{\varphi\}\mathbf{if}\ B\ \mathbf{then}\ p_0\ \mathbf{else}\ p_1\{\psi\}) = \tilde{\mathrm{vc}}(\{\varphi \wedge B\}p_0\{\psi\}) \wedge \tilde{\mathrm{vc}}(\{\varphi \wedge \neg B\}p_1\{\psi\}) \tag{4}$$

$$\tilde{\mathrm{vc}}(\{\varphi\}\mathbf{while}\ B\ \mathbf{do}\ p_0\{\psi\}) = \tilde{\mathrm{vc}}(\{I \wedge B\}p_0\{I\}) \wedge (\varphi \to I) \wedge (I \wedge \neg B \to \psi) \tag{5}$$

where, in clauses (3) and (5), I is a fresh predicate variable which does not appear in φ nor in ψ. Then $\mathrm{vc}(\{\varphi\}p\{\psi\}) = \exists \bar{I} \forall^* \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$ is obtained from $\tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$ by existential quantification of all predicate variables in $\forall^* \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$.

Note that this is an purely syntactic definition, thus we can define $\mathrm{vc}(\{\varphi\}p\{\psi\})$ analogously for a program p and second-order formulas φ, ψ. We then obtain the following completeness result which characterises Hoare provability by truth of a formula equation.

Theorem 13. *Let $\{\varphi\}p\{\psi\}$ be a Hoare triple. Then $\vdash \{\varphi\}p\{\psi\}$ iff $\mathbb{Z} \models \{\varphi\}p\{\psi\}$ iff $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi\})$.*

Proof Sketch. $\vdash \{\varphi\}p\{\psi\}$ iff $\mathbb{Z} \models \{\varphi\}p\{\psi\}$ is soundness and completeness of the Hoare calculus. The implication from $\vdash \{\varphi\}p\{\psi\}$ to $\Rightarrow \quad \mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi\})$ is proved by a straightforward induction on the structure of the Hoare proof of $\{\varphi\}p\{\psi\}$ The implication from $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi\})$ to $\mathbb{Z} \models \{\varphi\}p\{\psi\}$ is proved by translating the semantics. □

As a corollary we can now obtain the statement that a partial correctness assertion of an imperative program is expressible as a formula in FO[LFP], a point also made in [5] for existential least fixed-point logic.

Corollary 14. *Let $\{\varphi\}p\{\psi\}$ be a Hoare triple and let $\overline{\mu}$ be the solution of $\mathrm{vc}(\{\varphi\}p\{\psi\})$ obtained from Theorem 7, then $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi\})$ iff $\mathbb{Z} \models \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})[\bar{I} \backslash \overline{\mu}]$.*

A second corollary is based on the observation that $\mathrm{vc}(\{\varphi\}p\{\psi\})$ is a linear Horn formula equation (which can be shown by a straightforward induction). Therefore we can apply Theorem 11 to the verification condition and obtain:

Corollary 15. *Let $\{\varphi\}p\{\psi\}$ be a Hoare triple, then $\exists I_1 \cdots \exists I_n\ \pi \equiv \mathrm{vc}(\{\varphi\}p\{\psi\})$ is a linear Horn formula equation. Let π^D be the dual formula of π with predicate variables $K_1, \ldots, K_n$. Assume $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi\})$ and let $\mu_j = [\mathrm{lfp}_{I_j}\ \Phi_\pi]$ and $v_j = \neg[\mathrm{lfp}_{K_j}\ \Phi_{\pi^D}]$ for $j = 1, \ldots, n$. Then*

1. $\mathbb{Z} \models \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})[\bar{I}\backslash\bar{\mu}]$ *and* $\mathbb{Z} \models \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})[\bar{I}\backslash\bar{v}]$.

2. *If* $\mathbb{Z},[\bar{I}:=\bar{R}] \models \tilde{\mathrm{vc}}(\{\varphi\}p\{\psi\})$ *for relations* $R_1,\ldots,R_n$, *then* $\mathbb{Z} \models \bigwedge_{j=1}^{n}((\mu_j \to R_j) \wedge (R_j \to v_j))$.

This corollary shows that finding first-order formulas as loop invariants is equivalent to an interpolation problem in the sense of finding a fixed-point free interpolant. Just as Theorem 11 does for linear Horn clauses, this corollary contributes to explaining the efficacy of interpolation-based methods for loop invariant generation, see, e.g. [26].

As a third corollary we will show that the weakest precondition and the strongest postcondition are expressed by the least and greatest solutions μ and v of linear Horn formula equations based on the verification condition.

Definition 16. Let p be a program and φ, ψ be first-order formulas in $\mathscr{L}$. The *weakest precondition*[1] of p and ψ, written $\mathrm{wp}(p,\psi)$, is defined as

$$\mathrm{wp}(p,\psi) = \{\sigma \in \Sigma \mid \mathbb{Z}, C(p)(\sigma) \models \psi\}.$$

The *strongest postcondition* of p and φ, written $\mathrm{sp}(p,\varphi)$, is defined as

$$\mathrm{sp}(p,\varphi) = \{\sigma \in \Sigma \mid \exists \sigma' \in \Sigma : \mathbb{Z}, \sigma' \models \varphi \text{ and } C(p)(\sigma') = \sigma\}.$$

The following property of the weakest precondition and the strongest postcondition justifies the terminology and is of fundamental importance.

Lemma 17. *Let* $\{\varphi\}p\{\psi\}$ *be a Hoare triple, then* $\models \{\varphi\}p\{\psi\}$ *iff* $[\varphi] \subseteq \mathrm{wp}(p,\psi)$ *iff* $[\psi] \supseteq \mathrm{sp}(p,\varphi)$.

For a formula φ all of whose free variables are program variables we define $[\varphi] = \{\sigma \in \Sigma \mid \mathbb{Z}, \sigma \models \varphi\}$, the set of states defined by φ. It is well-known that for any program p and any formula ψ there is a first-order formula φ_{wp} which defines $\mathrm{wp}(p,\psi)$, i.e., $[\varphi_{\mathrm{wp}}] = \mathrm{wp}(p,\psi)$ and, symmetrically, for any program p and any formula φ there is a first-order formula ψ_{sp} which defines $\mathrm{sp}(p,\varphi)$, i.e., $[\psi_{\mathrm{sp}}] = \mathrm{sp}(p,\varphi)$. Note that these formulas rely on the expressivity of the assertion language, i.e, in our setting, on an encoding of finite sequences in $\mathbb{Z}$.

We will consider the formula equation $\exists X_0 \mathrm{vc}(\{\varphi\}p\{X_0\})$, which asks for a formula X_0 s.t. all states satisfying φ satisfy X_0 after running the program p. Symmetrically we will consider the formula equation $\exists X_0 \mathrm{vc}(\{X_0\}p\{\psi\})$. Note that these are linear Horn formula equations and therefore we can apply Theorem 11. In general there also occur predicate variables in $\mathrm{vc}(\{\varphi\}p\{\psi\})$, yet here we will only be interested in the solution for the outermost predicate variable.

Corollary 18. *Let* p *be a program, let* φ *be a formula, let* $\exists \bar{X} \pi \equiv \exists X_0 \mathrm{vc}(\{\varphi\}p\{X_0\})$, *and let* $\mu = [\mathrm{lfp}_{X_0} \Phi_\pi]$, *then* $[\mu] = \mathrm{sp}(p,\varphi)$.

Proof. Since $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\top\})$ we have $\mathbb{Z} \models \exists X \mathrm{vc}(\{\varphi\}p\{X_0\})$. From Lemma 6/1. we obtain $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\mu\})$ which, by Theorem 13, is equivalent to $\models \{\varphi\}p\{\mu\}$. By Lemma 17 we obtain $[\mu] \supseteq \mathrm{sp}(p,\varphi)$.

For the other direction let ψ_{sp} be a first-order formula with $[\psi_{\mathrm{sp}}] = \mathrm{sp}(p,\varphi)$. By Lemma 17 we have $\models \{\varphi\}p\{\psi_{\mathrm{sp}}\}$, which is equivalent to $\mathbb{Z} \models \mathrm{vc}(\{\varphi\}p\{\psi_{\mathrm{sp}}\})$. By Lemma 6/2. we have $\mathbb{Z} \models \forall \bar{x}(\mu(\bar{x}) \to \psi_{\mathrm{sp}}(\bar{x}))$ and thus $[\mu] \subseteq [\psi_{\mathrm{sp}}] = \mathrm{sp}(p,\varphi)$. $\square$

Corollary 19. *Let* p *be a program and let* ψ *be a formula, let* $\exists \bar{X} \pi \equiv \exists X_0 \mathrm{vc}(\{X_0\}p\{\psi\})$, *and let* $v = \neg[\mathrm{lfp}_{X_0} \Phi_{\pi^D}]$, *then* $[v] = \mathrm{wp}(p,\psi)$.

---

[1]In the literature this is mostly called weakest liberal precondition and the term weakest precondition is reserved for the context of total correctness. As we only talk about partial correctness of programs there is no need for us to do so.

Proof. Symmetric to that of Corollary 18   □

Note that the formulas μ and ν thus obtained do not rely on an expressivity hypothesis anymore. The encoding of sequences is replaced by the least fixed-point operator.

5   Towards an abstract fixed-point theorem

Abstract interpretation, originally introduced in [6], is one of the most important techniques in static analysis and software verification. Since many verification techniques could successfully be generalised from programs to the logical level of constrained Horn clauses it is also natural to expect this possibility for abstract interpretation. And indeed, abstract interpretations have been used in tools for solving Horn clauses [19, 22]. In this section we briefly outline how we expect this generalisation to apply to our fixed-point theorem. Proving the main statement of this section, Conjecture 24, is currently work-in-progress.

An application we have in mind is the following: in the recent article [18] the decidability of the existence of affine invariants for programs with affine assignments (essentially due to Karr [23]) has been generalised to formula equations of the form $\exists \overline{X} \forall^* \varphi$ for φ being a quantifier-free formula. The essential difference between the proof in [18] and Theorem 7 is that in [18] the fixed point is formed in the lattice of affine subspaces and not in the power set lattice. We can allow for this possibility as follows.

Definition 20. Let $\mathscr{A} = (A, \subseteq)$ and $\mathscr{B} = (B, \sqsubseteq)$ be two partially ordered sets. A *Galois connection* between $\mathscr{A}$ and $\mathscr{B}$ consists of two functions $\alpha : A \to B$ and $\gamma : B \to A$, s.t. for all $X \in A$ and $Y \in B$:

$$X \subseteq \gamma(Y) \quad \Leftrightarrow \quad \alpha(X) \sqsubseteq Y.$$

Definition 21. Let $\mathscr{L}$ be a language. A *model abstraction* is a pair $(\mathscr{M}, G)$, where $\mathscr{M}$ is an $\mathscr{L}$-structure and $G = (\mathscr{V}_k, \alpha_k, \gamma_k)_{k \in \mathbb{N}}$ is a sequence of triples, s.t. for all $k \in \mathbb{N} : \mathscr{V}_k = (V_k, \sqsubseteq)$ is a complete lattice and $\alpha_k : \mathscr{P}(M^k) \to V_k$ and $\gamma_k : V_k \to \mathscr{P}(M^k)$ form a Galois-connection between $(\mathscr{P}(M^k), \subseteq)$ and $\mathscr{V}_k$.

Example 22. Consider the language $\mathscr{L}_{\mathrm{aff}} = (0, 1, +, (c)_{c \in \mathbb{Q}})$ where the intended interpretation of the unary function symbol c for $c \in \mathbb{Q}$ is multiplication with c. Let $G_{\mathrm{aff}} = ((\mathrm{Aff}\, \mathbb{Q}^k, \subseteq), \mathrm{aff}_k, \mathrm{id}_k)_{k \in \mathbb{N}}$, where $\mathrm{Aff}\, \mathbb{Q}^k$ is the set of all affine subspaces of $\mathbb{Q}^k$, aff_k maps every subset of $\mathbb{Q}^k$ to its affine hull and id_k is the embedding of $\mathrm{Aff}\, \mathbb{Q}^k$ in $\mathscr{P}(\mathbb{Q}^k)$. Then $(\mathbb{Q}, G_{\mathrm{aff}})$ is a model abstraction.

We will now introduce the semantics of model abstractions. We will interpret FO[LFP]- and second-order-formulas φ by defining a satisfaction relation $(\mathscr{M}, G) \models_a \varphi$. The crucial difference between $\models_a$ and standard Tarski semantics $\models$ will be that second-order quantifiers and the least fixed-point operator will not be interpreted in the power set of the domain but in G instead (for the appropriate arity). This restriction of the domain of second-order quantification is reminiscent of (but different from) Henkin semantics of second- and higher-order logic [17].

Definition 23. The defining clauses for first-order atoms, propositional connectives, and first-order quantifiers for $\models_a$ are identical to those for $\models$. For formulas of the form $\exists X \psi$, where X is a k-ary predicate variable, we define

$$(\mathscr{M}, G) \models_a \exists X \psi \quad \Leftrightarrow \quad \exists S \in V_k : (\mathscr{M}, G) \models_a \psi[X \backslash \gamma_k(S)],$$

and analogously for formulas of the form $\forall X \psi$. The semantics of the lfp-operator is defined as follows. Let $X_1, \ldots, X_n$ be formula variables with X_i having arity k_i, let $\Phi = (\varphi_i(X_1, \ldots, X_n, \overline{u_i}))_{i=1}^n$ be a tuple of

formulas s.t. $|\overline{u_i}| = k_i$ and $X_1, \ldots, X_n$ occur only positively in φ_i for all $i \in \{1, \ldots, n\}$. Let

$$F_i^\# : V_{k_1} \times \cdots \times V_{k_n} \to V_{k_i}$$
$$(Y_1, \ldots, Y_n) \mapsto \alpha_{k_i} \circ F_i(\gamma_{k_1}(Y_1), \ldots, \gamma_{k_n}(Y_n)),$$

where F_i is defined as in Section 3 and let $F_\Phi^\# = (F_1^\#, \ldots, F_n^\#)$. Then

$$(\mathscr{M}, G) \models_a [\mathrm{lfp}_{X_i} \Phi](\overline{a}) \quad \Leftrightarrow \quad \overline{a} \in \gamma_{k_i}(\mathrm{lfp}(F_\Phi^\#)_i).$$

Note that the semantics of the least fixed-point operator is well-defined: We already know that F_i is a monotone operator and α_{k_i} and γ_{k_i} are monotone as they form a Galois connection for $i \in \{1, \ldots, n\}$. Thus $F_i^\#$ is monotone for all $i \in \{1, \ldots, n\}$ and therefore $F_\Phi^\#$ is monotone as well. As $\mathscr{V}_k$ is a complete lattice for all $k \in \mathbb{N}$, we can use the Knaster-Tarski theorem to obtain the least fixed point of $F_\Phi^\#$.

Conjecture 24 (Abstract fixed-point theorem). *Let $\mathscr{L}$ be a language and $(\mathscr{M}, G)$ be a model abstraction. Let $\exists X_1 \cdots \exists X_n \psi$ be a Horn formula equation and let $\mu_j = [\mathrm{lfp}_{X_j} \Phi_\psi]$ for $j = 1, \ldots, n$. Then:*

1. $(\mathscr{M}, G) \models_a \exists \overline{X} \psi \leftrightarrow \psi[\overline{X} \backslash \overline{\mu}]$ *and*

2. *if* $(\mathscr{M}, G) \models_a \psi[\overline{X} \backslash \overline{\chi}]$ *for FO[LFP]-formulas* $\chi_1, \ldots, \chi_n$, *then* $(\mathscr{M}, G) \models_a \bigwedge_{j=1}^n (\alpha_j \to \chi_j)$.

Example 25. In continuation of Example 22 consider an $\mathscr{L}_{\mathrm{aff}}$ formula equation $\exists \overline{X} \forall^* \psi$ where ψ is a quantifier-free formula. We want to decide whether there are formulas $\overline{\chi}$ s.t. $\mathbb{Q} \models \psi[\overline{X} \backslash \overline{\chi}]$ and $\chi_1, \ldots, \chi_n$ are conjunctions of affine equations, i.e., they define affine subspaces of $\mathbb{Q}^k$ in the model $\mathbb{Q}$. As in [18] we reduce the solvability of $\exists \overline{X} \forall^* \psi$ to solvability of one of its finitely many projections which are Horn formula equations. Let $\exists \overline{X} \forall^* \varphi$ be one of them, then an application of Conjecture 24 yields a tuple of FO[LFP]-formulas $\overline{\mu}$ s.t. $(\mathbb{Q}, G_{\mathrm{aff}}) \models_a \exists \overline{X} \varphi \leftrightarrow \varphi[\overline{X} \backslash \overline{\mu}]$. Since all lattices Aff $\mathbb{Q}^k$ have finite height we can compute fixed-point free formulas $\overline{\chi}$ equivalent to $\overline{\mu}$ from $\overline{\mu}$ and therefore $(\mathbb{Q}, G_{\mathrm{aff}}) \models_a \exists \overline{X} \varphi \leftrightarrow \varphi[\overline{X} \backslash \overline{\chi}]$. Now $\varphi[\overline{X} \backslash \overline{\chi}]$ is a first-order formula and hence $(\mathbb{Q}, G_{\mathrm{aff}}) \models_a \varphi[\overline{X} \backslash \overline{\chi}]$ iff $\mathbb{Q} \models \varphi[\overline{X} \backslash \overline{\chi}]$. The latter statement can now be checked by a decision procedure for linear arithmetic.

6 Conclusion

We have shown a fixed-point theorem for Horn formula equations and applied it to dual Horn formula equations and linear Horn formula equations. The proof of this result essentially consists of expressing the construction of a minimal model of a set of Horn clauses, which is familiar from logic programming, on the object level as a formula in first-order logic with least fixed points, thus providing a canonical solution to a Horn formula equation in FO[LFP].

Note that Theorem 7 applies equally to constraints φ being FO[LFP]-formulas. It therefore shows that FO[LFP], in contrast to first-order logic, has the property of being closed under solving Horn formula equations. It thus shows that in FO[LFP] validity and solvability of Horn formula equations coincide. This is in contrast to formula equations in first-order logic, cf. Example 2.

Explicit fixed-point operators have been employed for second-order quantifier elimination in the DLS* algorithm introduced in [28, 9]. In this algorithm Ackermann's lemma is generalised to a fixed-point lemma that covers additional situations. While the DLS* algorithm as such will fail on Horn formula equations due to its priority of shifting universal quantifiers inwards, the fixed-point lemma of the DLS* algorithm could be used for solving Horn formula equations with a single variable when combined with a different preprocessing. However, for Horn formula equations with more than one

predicate variable it would result in solutions with iterated fixed points. In this sense our fixed-point theorem can be considered a generalisation of the fixed-point lemma of [28]. On the other hand the fixed-point lemma of [28] is not restricted to Horn formula equations.

These fixed-point theorems contribute to our theoretical understanding of the logical foundations of constrained Horn clause solving and software verification: Theorem 11 and Corollary 15 help to explain, from a theoretical point of view, the efficacy of interpolation for Horn clause solving and invariant generation respectively. Moreover, as corollary to our fixed point theorem we have obtained the expressibility of the weakest precondition and the strongest postcondition, and thus the partial correctness of an imperative program in FO[LFP].

As shown in [24], our fixed-point theorem has a number of further corollaries in a variety of application areas: it allows a generalisation of a result by Ackermann [1] on second-order quantifier-elimination in a direction different from the recent generalisation [33] of that result. It allows to obtain a result on the generation of a proof with induction based on partial information about that proof shown in [10] as straightforward corollary. Last, but not least, an abstract form of the fixed-point theorem, Conjecture 24, would permit to considerably simplify the proof of the decidability of affine formula equations given in [18].

In conclusion, we believe that it is fruitful to consider constrained Horn clause solving from the more general point of view of solving formula equations. On the theoretical level this perspective uncovers connections to a number of topics such as second-order quantifier elimination and results such as Ackermann's [1]. On the practical level it suggests to study the applicability of algorithms such as DLS and SCAN for constrained Horn clauses and vice versa, that of algorithms for constrained Horn clause solving for applications of second-order quantifier elimination.

References

[1] Wilhelm Ackermann (1935): *Untersuchungen über das Eliminationsproblem der mathematischen Logik.* Mathematische Annalen 110(1), pp. 390–413, doi:10.1007/BF01448035.

[2] Heinrich Behmann (1950): *Das Auflösungsproblem in der Klassenlogik.* Archiv für mathematische Logik und Grundlagenforschung 1(1), pp. 17–29. First of two parts.

[3] Heinrich Behmann (1951): *Das Auflösungsproblem in der Klassenlogik.* Archiv für mathematische Logik und Grundlagenforschung 1(2), pp. 33–51. Second of two parts.

[4] Nikolaj Bjørner, Arie Gurfinkel, Kenneth L. McMillan & Andrey Rybalchenko (2015): *Horn Clause Solvers for Program Verification.* In Lev D. Beklemishev, Andreas Blass, Nachum Dershowitz, Bernd Finkbeiner & Wolfram Schulte, editors: Fields of Logic and Computation II - Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday, Lecture Notes in Computer Science 9300, Springer, pp. 24–51, doi:10.1007/978-3-319-23534-9_2.

[5] Andreas Blass & Yuri Gurevich (1987): *Existential Fixed-Point Logic.* In Egon Börger, editor: Computation Theory and Logic, In Memory of Dieter Rödding, Lecture Notes in Computer Science 270, Springer, pp. 20–36, doi:10.1007/3-540-18170-9_151.

[6] Patrick Cousot & Radhia Cousot (1977): *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.* In Robert M. Graham, Michael A. Harrison & Ravi Sethi, editors: 4th ACM Symposium on Principles of Programming Languages, ACM, pp. 238–252, doi:10.1145/512950.512973.

 [7] Anuj Dawar & Yuri Gurevich (2002): *Fixed point logics*. Bulletin of Symbolic Logic 8(1), pp. 65–88, doi:10.2178/bsl/1182353853.

 [8] Patrick Doherty, Witold Lukaszewicz & Andrzej Szalas (1997): *Computing Circumscription Revisited: A Reduction Algorithm*. Journal of Automated Reasoning 18(3), pp. 297–336.

 [9] Patrick Doherty, Witold Łukaszewicz & Andrzej Szałas (1998): *General Domain Circumscription and its Effective Reductions*. Fundamenta Informaticae 36(1), pp. 23–55, doi:10.3233/FI-1998-3612.

[10] Sebastian Eberhard & Stefan Hetzl (2015): *Inductive theorem proving based on tree grammars*. Annals of Pure and Applied Logic 166(6), pp. 665–700, doi:10.1016/j.apal.2015.01.002.

[11] Carsten Fritz (2001): *Some Fixed Point Basics*. In Erich Grädel, Wolfgang Thomas & Thomas Wilke, editors: *Automata, Logics, and Infinite Games: A Guide to Current Research*, Lecture Notes in Computer Science 2500, Springer, pp. 359–364, doi:10.1007/3-540-36387-4_20.

[12] Dov Gabbay & Hans Jürgen Ohlbach (1992): *Quantifier Elimination in Second Order Predicate Logic*. South African Computer Journal 7, pp. 35–43.

[13] Dov M. Gabbay, Renate A. Schmidt & Andrzej Szałas (2008): *Second-Order Quantifier Elimination*. College Publications.

[14] Erich Grädel (1991): *The Expressive Power of Second Order Horn Logic*. In Christian Choffrut & Matthias Jantzen, editors: *8th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science 480, Springer, pp. 466–477, doi:10.1007/BFb0020821.

[15] Ashutosh Gupta, Corneliu Popeea & Andrey Rybalchenko (2014): *Generalised Interpolation by Solving Recursion-Free Horn Clauses*. In Nikolaj Bjørner, Fabio Fioravanti, Andrey Rybalchenko & Valerio Senni, editors: *Proceedings First Workshop on Horn Clauses for Verification and Synthesis (HCVS)*, EPTCS 169, pp. 31–38, doi:10.4204/EPTCS.169.5.

[16] Arie Gurfinkel & Nikolaj Bjørner (2019): *The Science, Art, and Magic of Constrained Horn Clauses*. In: *21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing SYNASC*, IEEE, pp. 6–10, doi:10.1109/SYNASC49474.2019.00010.

[17] Leon Henkin (1950): *Completeness in the Theory of Types*. Journal of Symbolic Logic 15(2), pp. 81–91, doi:10.2307/2266967.

[18] Stefan Hetzl & Sebastian Zivota (2020): *Decidability of affine solution problems*. Journal of Logic and Computation 30(3), pp. 697–714, doi:10.1093/logcom/exz033.

[19] Krystof Hoder, Nikolaj Bjørner & Leonardo Mendonça de Moura (2011): *μZ - An Efficient Engine for Fixed Points with Constraints*. In Ganesh Gopalakrishnan & Shaz Qadeer, editors: *23rd International Conference on Computer-Aided Verification (CAV)*, Lecture Notes in Computer Science 6806, Springer, pp. 457–462, doi:10.1007/978-3-642-22110-1_36.

[20] Neil Immerman (1999): *Descriptive complexity*. Springer, doi:10.1007/978-1-4612-0539-5.

[21] Joxan Jaffar & Michael J. Maher (1994): *Constraint Logic Programming: A Survey*. The Journal of Logic Programming 19/20, pp. 503–581, doi:10.1016/0743-1066(94)90033-7.

[22] Bishoksan Kafle, John P. Gallagher & José F. Morales (2016): *Rahft: A Tool for Verifying Horn Clauses Using Abstract Interpretation and Finite Tree Automata*. In Swarat Chaudhuri & Azadeh Farzan, editors: *28th International Conference on Computer Aided Verification*, Lecture Notes in Computer Science 9779, Springer, pp. 261–268, doi:10.1007/978-3-319-41528-4_14.

[23] Michael Karr (1976): *Affine Relationships Among Variables of a Program*. Acta Informatica 6, pp. 133–151, doi:10.1007/BF00268497.

[24] Johannes Kloibhofer (2020): *A fixed-point theorem for Horn formula equations*. Master's thesis, TU Wien, Austria.

[25] Ursula Martin & Tobias Nipkow (1989): *Boolean Unification – The Story So Far*. Journal of Symbolic Computation 7(3-4), pp. 275–293, doi:10.1016/S0747-7171(89)80013-6.

[26] Kenneth L. McMillan (2003): *Interpolation and SAT-Based Model Checking*. In Warren A. Hunt Jr. & Fabio Somenzi, editors: *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings, Lecture Notes in Computer Science* 2725, Springer, pp. 1–13.

[27] Kenneth L. McMillan & Andrey Rybalchenko (2012): *Solving Constrained Horn Clauses using Interpolation*. Technical Report, Microsoft Research. MSR-TR-2013-06.

[28] Andreas Nonnengart & Andrzej Szałas (1998): *A Fixpoint Approach to Second-Order Quantifier Elimination with Applications to Correspondence Theory*, pp. 307–328. Studies in Fuzziness and Soft Computing 24, Springer.

[29] Sergiu Rudeanu (1974): *Boolean Functions and Equations*. North-Holland.

[30] Philipp Rümmer, Hossein Hojjat & Viktor Kuncak (2013): *Classifying and Solving Horn Clauses for Verification*. In Ernie Cohen & Andrey Rybalchenko, editors: *5th International Conference on Verified Software: Theories, Tools, Experiments (VSTTE), Lecture Notes in Computer Science* 8164, Springer, pp. 1–21, doi:10.1007/978-3-642-54108-7_1.

[31] Ernst Schröder (1890): *Vorlesungen über die Algebra der Logik*. 1, Teubner.

[32] M. H. Van Emden & R. A. Kowalski (1976): *The Semantics of Predicate Logic as a Programming Language*. Journal of the ACM 23(4), p. 733–742, doi:10.1145/321978.321991.

[33] Christoph Wernhard: *Approximating Resultants of Existential Second-Order Quantifier Elimination upon Universal Relational First-Order Formulas*. In: *Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2017)*.

[34] Christoph Wernhard (2017): *The Boolean Solution Problem from the Perspective of Predicate Logic*. In: *11th International Symposium on Frontiers of Combining Systems (FroCoS), Lecture Notes in Computer Science* 10483, Springer, pp. 333–350, doi:10.1007/978-3-319-66167-4_19.

[35] Glynn Winskel (1993): *The Formal Semantics of Programming Languages – An Introduction*. Foundation of computing series, MIT Press.