First-Order Logic Theorem Proving and Model Building via Approximation and Instantiation

Andreas Teucke[1,2] and Christoph Weidenbach[1]

[1] Max-Planck Institut for Informatics, Campus E1 4 66123 Saarbrücken Germany
[2] Graduate School of Computer Science, Saarbrücken, Germany

Abstract. In this paper we consider first-order logic theorem proving and model building via approximation and instantiation. Given a clause set we propose its approximation into a simplified clause set where satisfiability is decidable. The approximation extends the signature and preserves unsatisfiability: if the simplified clause set is satisfiable in some model, so is the original clause set in the same model interpreted in the original signature. A refutation generated by a decision procedure on the simplified clause set can then either be lifted to a refutation in the original clause set, or it guides a refinement excluding the previously found unliftable refutation. This way the approach is refutationally complete. We do not step-wise lift refutations but conflicting cores, finite unsatisfiable clause sets representing at least one refutation. The approach is dual to many existing approaches in the literature because our approximation preserves unsatisfiability.

1 Introduction

The Inst-Gen calculus by Ganzinger and Korovin [5] and its implementation in iProver has shown to be very successfull. The calculus is based on a under-approximation - instantiation refinement loop. A given first-order clause set is under-approximated by finite grounding and afterwards a SAT-solver is used to test unsatisfiability. If the ground clause set is unsatisfiable then a refutation for the original clause set is found. If it is satisfiable, the model generated by the SAT-solver is typically not a model for the original clause set. If it is not, it is used to instantiate the original clause such that the found model is ruled out for the future.

In this paper we define a calculus that is dual to the Inst-Gen calculus. A given first-order clause set is over-approximated into a decidable fragment of first-order logic: a monadic, shallow, linear Horn (mslH) theory [12]. If the over-approximated clause set is satisfiable, so is the original clause set. If it is unsatisfiable, the found refutation is typically not a refutation for the original clause set. If it is not, the refutation is analyzed to instantiate the original clause set such that the found refutation is ruled out for the future. The mslH fragment properly include first-order ground logic, but is also expressive enough to represent minimal infinite models.

In addition to developing a new proof method for first-order logic this constitutes our second motivation for studying the new calculus and the particular mslH approximation. It is meanwhile accepted that a model-based guidence can significantly improve an automated reasoning calculus. The propositional CDCL calculus [8] is one prominent example for this insight. In first-order logic, (partial) model operators typically generate inductive models for which almost all interesting properties become undecidable, in general. One way out of this problem is to generate a model for an approximated clause set, such that important properties with respect to the original clause set are preserved. In the case of our calculus and approximation, a found model can be effectively translated into a model for the original clause set. So our result is also a first step towards model-based guidence in first-order logic automated reasoning.

For example, consider the first-order Horn clauses $S(x) \rightarrow P(x, g(x))$; $S(a)$; $S(b)$; $S(g(x))$; $\neg P(a, g(b))$; $\neg P(g(x), g(g(x)))$ that are approximated (Section 2) into the mslH theory $S(x), R(y) \rightarrow T(f_P(x, y))$; $S(x) \rightarrow R(g(x))$; $S(a)$; $S(b)$; $S(g(x))$; $\neg T(f_P(a, g(b)))$; $\neg T(f_P(g(x), g(g(x))))$ where the relation P is encoded by the function f_P and the non-linear occurrence of x in the first clause is approximated by the introduction of the additional variable y. The approximated clause set has two refutations: one using $\neg T(f_P(a, g(b)))$ and the second using $\neg T(f_P(g(x), g(g(x))))$ plus the rest of the clauses, respectively. While the first refutation cannot be lifted, the second one is liftable to a refutation of the original clause set (Section 3). Actually, we do not consider refutations, but conflicting cores (Definition 1). Conflicting cores are finite, unsatisfiable clause sets where variables are considered to be shared among clauses and rigid such that any instantiation preserves unsatisfiability. Conflicting cores can be effectively generated out of refutations via instantiation of (copies of) the input clauses involved in the refutation. For the above second refutation the conflicting core of the approximated clause set is $S(g(x)), R(g(g(x))) \rightarrow T(f_P(g(x), g(g(x))))$; $S(g(x)) \rightarrow R(g(g(x)))$; $S(g(x))$; $\neg T(f_P(g(x), g(g(x))))$.
In case the first refutation is selected for lifting, it fails, so the original clause set is refined (Section 4). The refinement replaces the first clause with
$S(a) \rightarrow P(a, g(a))$; $S(b) \rightarrow P(b, g(b))$ and $S(g(x)) \rightarrow P(g(x), g(g(x)))$.
The approximation of the resulting new clause set does no longer enable a refutation using $\neg T(f_P(a, g(b)))$. Therefore, the refutation using $\neg T(f_P(g(x), g(g(x))))$ is found after refinement. In case the original clause set contains a non-Horn clause, one positive literal is selected by the approximation.

The paper is now organized as follows. Section 2 introduces some basic notions and the approximation relation $\Rightarrow_{APR}$ that transforms any first-order clause set into an mslH theory. The lifting of conflicting cores is described in Section 3 and the respective abstraction refinement in Section 4 including soundness and completeness results. Missing proofs can be found in the appendix. The paper ends with Section 5 on future/related work and a conclusion.

2 Linear Shallow Monadic Horn Approximation

We consider a standard first-order language without equality where Σ denotes the set of function symbols. The symbols x, y denote variables, a, b constants, f, g, h are functions and s, t terms. Predicates are denoted by S, P, Q, R, literals by E, clauses by C, D, and sets of clauses by N, M. The term $t[s]_p$ denotes that the term t has the subterm s at position p. The notion is extended to atoms, clauses, and multiple positions. A predicate with at most one argument is called monadic. A literal is either an atom or an atom preceded by $\neg$ and it is then respectively called positive or negative. A term is shallow if it has at most depth one. It is called linear if there are no duplicate variable occurrences. A literal, where every term is shallow, is also called shallow. A clause is a multiset of literals which we write as an implication $\Gamma \rightarrow \Delta$ where the atoms in Δ denote the positive literals and the atoms in Γ the negative literals. If Γ is empty we omit $\rightarrow$, e.g., we write $P(x)$ instead of $\rightarrow P(x)$ whereas if Δ is empty $\rightarrow$ is always shown. If a clause has at most one positive literal, it is a Horn clause. If there are no variables, then terms, atoms and clauses are respectively called ground. A substitution σ is a mapping from variables into terms denoted by pairs $\{x \mapsto t\}$. If for some term (literal, clause) t, $t\sigma$ is ground, then σ is a grounding substitution.

A Herbrand interpretation I is a - possibly infinite - set of positive ground literals and I is said to satisfy a clause $C = \Gamma \rightarrow \Delta$, denoted by $I \vDash C$, if $\Delta\sigma \cap I \neq \emptyset$ or $\Gamma\sigma \nsubseteq I$ for every grounding substitution σ. An interpretation I is called a model of N if I satisfies N, $I \vDash N$, i.e., $I \vDash C$ for every $C \in N$. Models are considered *minimal* with respect to set inclusion. A set of clauses N is satisfiable, if there exists a model that satisfies N. Otherwise the set is unsatisfiable.

Definition 1 (Conflicting Core) *A finite clause set $N^\perp$ is a conflicting core if for all grounding substitutions τ the clause set $N^\perp\tau$ is unsatisfiable. $N^\perp$ is a conflicting core of N if $N^\perp$ is a conflicting core and for every clause $C \in N^\perp$ there exists a $C' \in N$ such that $C = C'\sigma$.*

Definition 2 (Specific Instances) *Let C be a clause and σ_1, σ_2 be two substitutions such that $C\sigma_1$ and $C\sigma_2$ have no common instances. Then the* specific instances *of C with respect to σ_1, σ_2 are clauses $C\tau_1, \ldots, C\tau_n$ such that (i) any ground instance of C is an instance of some $C\tau_i$, (ii) there is no $C\tau_i$ such that both $C\sigma_1$ and $C\sigma_2$ are instances of $C\tau_i$.*

The definition of specific instances can be extended to a single substitution σ. In this case we require C and σ to be linear, condition (i) from Definition 2 above, $C\sigma = C\tau_1$ and no $C\tau_i$, $i \neq 1$ has a common instance with $C\tau_1$. Note that under the above restrictions specific instances always exist [6].

Definition 3 (Approximation) *Given a clause set N and a relation $\Rightarrow$ on clause sets with $N \Rightarrow N'$ then (1) $\Rightarrow$ is called an* over-approximation *if satisfiability of N' implies satisfiability of N, (2) $\Rightarrow$ is called an* under-approximation *if unsatisfiability of N' implies unsatisfiability of N.*

Next we introduce our concrete over-approximation $\Rightarrow_{APR}$ that eventually maps a clause set N to an mslH clause set N'. Starting from a clause set N the transformation is parameterized by a single monadic projection predicate T, fresh to N and for each non-monadic predicate P a projection function f_P fresh to N. The approximation always applies to a single clause and we establish on the fly an ancestor relation between the approximated clause(s) and the parent clause. The ancestor relation is needed for lifting and refinement.

Monadic $N \cup \{\Gamma \rightarrow \Delta, P(t_1,\ldots,t_n)\} \ \Rightarrow_{\mathrm{MO}} \ N \cup \{\Gamma \rightarrow \Delta, T(f_P(t_1,\ldots,t_n))\}$
provided $n > 1$; $P(t_1,\ldots,t_n)$ is the ancestor of $T(f_P(t_1,\ldots,t_n))$

Horn $N \cup \{\Gamma \rightarrow E_1,\ldots,E_n\} \ \Rightarrow_{\mathrm{HO}} \ N \cup \{\Gamma \rightarrow E_i\}$
provided $n > 1$; $\Gamma \rightarrow E_1,\ldots,E_n$ is the ancestor of $\Gamma \rightarrow E_i$

Shallow $N \cup \{\Gamma \rightarrow E[s]_p\} \ \Rightarrow_{\mathrm{SH}} \ N \cup \{S(x), \Gamma_1 \rightarrow E[x]_p\} \cup \{\Gamma_2 \rightarrow S(s)\}$
provided s is a complex term, p not a top position, x and S fresh, and $\Gamma_1 \cup \Gamma_2 = \Gamma$; $\Gamma \rightarrow E[s]_p$ is the ancestor of $S(x), \Gamma_1 \rightarrow E[x]_p$ and $\Gamma_2 \rightarrow S(s)$

Linear $N \cup \{\Gamma \rightarrow E[x]_{p,q}\} \ \Rightarrow_{\mathrm{LI}} \ N \cup \{\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q\}$
provided x' is fresh, the positions p, q denote two different occurrences of x in E; $\Gamma \rightarrow E[x]_{p,q}$ is the ancestor of $\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q$

For the Horn transformation, the choice of the E_i is arbitrary. In the Shallow rule, Γ_1 and Γ_2 can be arbitrarily chosen as long as they "add up" to Γ. The goal, however, is to minimize the set of common variables $\mathrm{vars}(\Gamma_2, s) \cap \mathrm{vars}(\Gamma_1, E[x]_p)$. If this set is empty the Shallow transformation is satisfiability preserving. In rule Linear, the duplication of Γ is not needed if $x \notin \mathrm{vars}(\Gamma)$.

Definition 4 ($\Rightarrow_{APR}$) *The overall approximation $\Rightarrow_{APR}$ is given by $\Rightarrow_{APR} = \Rightarrow_{MO} \cup \Rightarrow_{HO} \cup \Rightarrow_{SH} \cup \Rightarrow_{LI}$ with a preference on the different rules where Monadic precede Horn precede Shallow precede Linear transformations.*

Definition 5 *Given a non-monadic n-ary predicate P, projection predicate T, and projection function f_P, define the injective function $\mu_P(P(t_1,\ldots,t_n)) := T(f_p(t_1,\ldots,t_n))$ and $\mu_P(Q(s_1,\ldots,s_m)) := Q(s_1,\ldots,s_m)$ for any atom with a predicate symbol different from P. The function is extended to clauses, clause sets and interpretations.*

Lemma 1 ($\Rightarrow_{APR}$ is sound and terminating) *The approximation rules are sound and terminating: (i) $\Rightarrow_{APR}$ terminates (ii) the Monadic transformation is an over-approximation (iii) the Horn transformation is an over-approximation (iv) the Shallow transformation is an over-approximation (v) the Linear transformation is an over-approximation*

Proof. (i) The transformations can be considered sequentially, because of the imposed rule preference (Definition 4). The monadic transformation strictly reduces the number of non-monadic atoms. The Horn transformation strictly reduces the number of non-Horn clauses. The Shallow transformation strictly reduces the multiset of term depths of the newly introduced clauses compared to the removed ancestor clause. The linear transformation strictly reduces the number of duplicate variables occurrences in positive literals. Hence $\Rightarrow_{APR}$ terminates.

(ii) Consider a transformation $N_k \Rightarrow^*_{\mathrm{MO}} N_{k+j}$ that exactly removes all occurrences of atoms $P(t_1, \dots, t_n)$ and replaces those by atoms $T(f_P(t_1, \dots, t_n))$. Then, $N_{k+j} = \mu_P(N_k)$ and $N_k = \mu_P^{-1}(N_{k+j})$. Let I be a model of N_{k+j} and $C \in N_k$. Since $\mu_P(C) \in N_{k+j}$, $I \vDash \mu_P(C)$ and thus, $\mu_P^{-1}(I) \vDash C$. Hence, $\mu_P^{-1}(I)$ is a model of N_k. Therefore, the Monadic transformation is an over-approximation.

(iii) Let $N \cup \{\Gamma \rightarrow E_1, \dots, E_n\} \Rightarrow_{\mathrm{HO}} N \cup \{\Gamma \rightarrow E_i\}$. The clause $\Gamma \rightarrow E_i$ subsumes the clause $\Gamma \rightarrow E_1, \dots, E_n$. Therefore, for any I if $I \models \Gamma \rightarrow E_i$ then $I \models \Gamma \rightarrow E_1, \dots, E_n$. Therefore, the Horn transformation is an over-approximation.

(iv) Let $N_k = N \cup \{\Gamma \rightarrow E[s]_p\} \Rightarrow_{\mathrm{SH}} N_{k+1} = N \cup \{S(x), \Gamma_1 \rightarrow E[x]_p\} \cup \{\Gamma_2 \rightarrow S(s)\}$. Let I be a model of N_{k+1} and $C \in N_k$ be a ground clause. If C is an instance of a clause in N, then $I \models C$. Otherwise $C = (\Gamma \rightarrow E[s]_p)\sigma$ for some ground substitution σ. Then $S(s)\sigma, \Gamma_1\sigma \rightarrow E[s]_p\sigma = (S(x), \Gamma_1 \rightarrow E[x]_p)\{x \mapsto s\}\sigma \in N_{k+1}$ and $\Gamma_2\sigma \rightarrow S(s)\sigma = (\Gamma_2 \rightarrow S(s))\sigma \in N_{k+1}$. Since $I \models N_{k+1}$, I also satisfies the resolvent $\Gamma_1\sigma, \Gamma_2\sigma \rightarrow E[s]\sigma = C$. Hence $I \models N_k$. Therefore, the Shallow transformation is an over-approximation.

(v) Let $N_k = N \cup \{\Gamma \rightarrow E[x]_{p,q}\} \Rightarrow_{\mathrm{LI}} N_{k+1} = N \cup \{\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q\}$. Let I be a model of N_{k+1} and $C \in N_k$ be a ground clause. If C is an instance of a clause in N, then $I \models C$. Otherwise $C = (\Gamma \rightarrow E[x]_{p,q})\sigma$ for some ground substitution σ. Then $(\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\{x' \mapsto x\}\sigma \in N_{k+1}$ and $I \models (\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\{x' \mapsto x\}\sigma = (\Gamma, \Gamma \rightarrow E[x]_q)\sigma \models C$. Hence $I \models N_k$. Therefore, Linear transformation is an over-approximation.

Corollary 2 *(i) $\Rightarrow_{APR}$ is an over-approximation. (ii) If $N \Rightarrow^*_{APR} N'$, $P_1, \dots, P_n$ are the non-monadic predicates in N and N' is satisfied by model I, then $\mu_{P_1}^{-1}(...(\mu_{P_n}^{-1}(I)))$ is a model of N.*

Proof. Follows from Lemma 1 (ii)-(v).

In addition to being an over-approximation, the minimal model (with respect to set inclusion) of the eventual approximation generated by $\Rightarrow_{APR}$ preserves the skeleton term structure of the original clause set, if it exists. The refinement introduced in Section 4 instantiates clauses. Thus it contributes to finding a model or a refutation.

Definition 6 (Term Skeleton) *The term skeleton of term t , skt(t), is defined as*
(1) skt$(x) = x'$, where x' is a fresh variable
(2) skt$(f(s_1, \dots, s_n)) = f(skt(s_1), \dots,$ skt$(s_n))$.

Lemma 3 *Let N_k be a monadic clause set and N_0 be its approximation via $\Rightarrow_{APR}$. Let N_0 be satisfiable and I be a minimal model for N_0. If $P(s) \in I$ and P is a predicate in N_k, then there exists a clause $C = \Gamma \rightarrow \Delta, P(t) \in N_k$ and a substitution σ such that $s = skt(t)\sigma$ and for each variable x and predicate S with $C = S(x), \Gamma' \rightarrow \Delta, P(t[x]_p)$, $S(s'') \in I$, where $s = s[s'']_p$.*

Proof. By induction on k.
For the base $N_k = N_0$, assume there is no $C \in N_0$ with $C\sigma = \Gamma \rightarrow \Delta, P(s)$ and $\Gamma \subseteq I$. Then $I \setminus \{P(s)\}$ is still a model of N_0 and therefore I is not minimal.
Let $N = N_k \Rightarrow_{APR} N_{k-1} \Rightarrow^*_{APR} N_0$, $P(s) \in I$ and P is a predicate in N_k and hence also in N_{k-1}. By the induction hypothesis, there exist a clause $C = \Gamma \rightarrow \Delta, P(t) \in N_{k-1}$ and a substitution σ such that $s = \text{skt}(t)\sigma$ and for each variable x and predicate S with $C = S(x), \Gamma' \rightarrow \Delta, P(t[x]_p)$, $S(s'') \in I$, where $s = s[s'']_p$. The first approximation rule application is either a Linear, a Shallow or a Horn transformation, considered below by case analysis.

Horn Case. Let $\Rightarrow_{APR}$ be a Horn transformation that replaces $\Gamma'' \rightarrow \Delta', Q(t')$ with $\Gamma'' \rightarrow Q(t')$. If $C \neq \Gamma'' \rightarrow Q(t')$, then $C \in N_k$ fulfills the claim. Otherwise, $\Gamma'' \rightarrow \Delta', Q(t) \in N_k$ fulfills the claim since $P = Q$ and $\Gamma' = \Gamma''$.

Linear Case. Let $\Rightarrow_{APR}$ be a linear transformation that replaces $C_k = \Gamma'' \rightarrow E[x]_{p,q}$ with $C_{k-1} = \Gamma'', \Gamma''\{x \mapsto x'\} \rightarrow E[x']_q$. If $C \neq C_{k-1}$, then $C \in N_k$ fulfills the claim. Otherwise, $C_k = \Gamma'' \rightarrow P(t)\{x' \mapsto x\} \in N_k$ fulfills the claim since $s = \text{skt}(t)\sigma = \text{skt}(t\{x' \mapsto x\})\sigma$ and $\Gamma'' \subseteq \Gamma'', \Gamma''\{x \mapsto x'\}$.

Shallow Case. Let $\Rightarrow_{APR}$ be a shallow transformation that replaces $C_k = \Gamma'' \rightarrow E[s']_p$ with $C_{k-1} = S(x), \Gamma_1 \rightarrow E[x]_p$ and $C'_{k-1} = \Gamma_2 \rightarrow S(s')$. Since S is fresh, $C \neq C'_{k-1}$. If $C \neq C_{k-1}$, then $C \in N_k$ fulfills the claim. Otherwise, $C = C_{k-1} = S(x), \Gamma_1 \rightarrow P(t[x]_p)$ and hence, $s = \text{skt}(t[x]_p)\sigma$ and $S(s'') \in I$ for $s = s[s'']_p$. Then by the induction hypothesis, there exist a clause $C_S = \Gamma_S \rightarrow \Delta_S, S(t_S) \in N_{k-1}$ and a substitution σ_S such that $s'' = \text{skt}(t_S)\sigma_S$ and for each variable x and predicate S' with $C_S = S'(x), \Gamma'_S \rightarrow \Delta_S, P(t_S[x]_q)$, $S'(s''') \in I$, where $s'' = s''[s''']_q$. By construction, $C_S = C'_{k-1}$. Thus, $s'' = \text{skt}(s')\sigma_S$ and $s = \text{skt}(t[x]_p)\sigma$ imply there exists a σ'' such that $s = \text{skt}(t[s']_p)\sigma$". Furthermore since $\Gamma_1 \cup \Gamma_2 = \Gamma''$, if $C_k = S'(x), \Gamma''' \rightarrow P(t[s']_p)[x]_q$, then either $S'(x) \in \Gamma_1$ and thus $S'(s'''') \in I$, where $s = s[s'''']_q$, or $S'(x) \in \Gamma_2$ and thus $S'(s'''') \in I$, where $s[s'']_p = (s[s'']_p)[s'''']_q$. Hence, $C_k \in N_k$ fulfills the claim.

Lemma 4 *Let N be a clause set and N' be its approximation via $\Rightarrow_{APR}$. Let N' be satisfiable and I be a minimal model for N'. If $P(s) \in I$ ($T(f_p(s_1, \ldots, s_n)) \in I$) and P is a predicate in N, then there exist a clause $\Gamma \rightarrow \Delta, P(t) \in N$ ($\Gamma \rightarrow \Delta, P(t_1, \ldots, t_n) \in N$) and a substitution σ such that $s = skt(t)\sigma$ ($s_i = skt(t_i)\sigma$ for all i).*

Proof. Let $P_1, \ldots, P_n$ be the non-monadic predicates in N and $N_{MO} = \mu_{P_1}(\ldots(\mu_{P_n}(N)))$. Then, N_{MO} is monadic and also has N' as its approximation via $\Rightarrow_{APR}$.

Let $P(s) \in I$ and P is a predicate in N. Since P is monadic, P is a predicate in N_{MO}. Hence by Lemma 3, there exists a clause $\Gamma \rightarrow \Delta, P(t) \in N_{MO}$ and a substitution σ such that $s = \text{skt}(t)\sigma$. Then, $\mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma \rightarrow \Delta, P(t)))\ldots) = \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma)\ldots) \rightarrow \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Delta)\ldots), P(t) \in N$ fulfills the claim.

Let $T(f_p(s_1, \ldots, s_n)) \in I$ and P is a predicate in N. T is monadic and a predicate in N_{MO}. Hence by Lemma 3, there exists a clause $\Gamma \rightarrow \Delta, T(t) \in N_{MO}$ and a substitution σ such that $f_p(s_1, \ldots, s_n) = \operatorname{skt}(t)\sigma$. Therefore, $t = f_p(t_1, \ldots, t_n)$ with $s_i = \operatorname{skt}(t_i)\sigma$ for all i. Then, $\mu_{P_1}^{-1}(\ldots(\mu_{P_n}^{-1}(\Gamma \rightarrow \Delta, T(f_p(t_1, \ldots, t_n))))\ldots) = \mu_{P_1}^{-1}(\ldots(\mu_{P_n}^{-1}(\Gamma)\ldots) \rightarrow \mu_{P_1}^{-1}(\ldots(\mu_{P_n}^{-1}(\Delta)\ldots), P(t_1, \ldots, t_n) \in N$ fulfills the claim.

The above lemma also holds if satisfiability of N' is dropped and I is replaced by the superposition partial minimal model operator [13].

3 Lifting the Conflicting Core

Given a monadic, linear, shallow, Horn approximation N_k of N and a conflicting core $N_k^\perp$ of N_k, using the transformations provided in this section we attempt to lift $N_k^\perp$ to a conflicting core $N^\perp$ of N. In case of success this shows the unsatisfiability of N. In case an approximation step cannot be lifted the original clause set is refined by instantiation, explained in the next section.

Let N_k be an unsatisfiable monadic, linear, shallow, Horn approximation. Since N_k belongs to a decidable first-order fragment, we expect an appropriate decision procedure to generate a proof of unsatisfiability for N_k, e.g., ordered resolution with selection [12]. A conflicting core can be straightforwardly generated out of a resolution refutation by applying the substitutions of the proof to the used input clauses.

Starting with a resolution refutation, in order to construct the conflicting core, we begin with the singleton set containing the pair of empty clause and the empty substitution. Furthermore, we assume that all input clauses from N_k used in the refutation are variable disjoint. Then we recursively choose a pair (C, σ) from the set where $C \notin N_k$. There exists a step in the refutation that generated this clause. In the case of a resolution inference, there are two parent clauses C_1 and C_2 in the refutation and two substitutions σ_1 and σ_2 such that C is the resolvent of $C_1\sigma_1$ and $C_2\sigma_2$. In the case of a factoring inference, there is one parent clause C' in the refutation and a substitution σ' such that C is the factor of $C'\sigma'$. Replace (C, σ) by $(C_1, \sigma_1\sigma)$ and $(C_2, \sigma_2\sigma)$ or by $(C', \sigma'\sigma)$ respectively. The procedure terminates in linear time in the size of the refutation. For each pair (C, σ), collect the clause $C\sigma$, resulting in a conflicting core $N_k^\perp$ of N_k.

Example 1 *Let $N = \{P(x, x'); P(y, a), P(z, b) \rightarrow\}$ with signature $\Sigma = a/0, b/0$. N is unsatisfiable and a possible resolution refutation is resolving $P(b, a)$ and $P(a, b)$ with $P(b, a), P(a, b) \rightarrow$. From this we get the conflicting core $N_{ba}^\perp = \{P(b, a); P(a, b); P(b, a), P(a, b) \rightarrow\}$.*

An alternative refutation is to resolve $P(x, x')$ and $P(y, a), P(z, b) \rightarrow$ with substitution $\{x \mapsto y; x' \mapsto a\}$ and then the resolvent and $P(x, x')$ with substitution $\{x \mapsto z; x' \mapsto b\}$. From this refutation we construct the conflicting core $N_{yz}^\perp = \{P(y, a);\ P(z, b);\ P(y, a), P(z, b) \rightarrow\}$.

Note that in Example 1 $N_{yz}^\perp$ is more general than $N_{ba}^\perp$ since $N_{yz}^\perp\{y \mapsto b; z \mapsto a\} = N_{ba}^\perp$. A conflicting core is minimal in that it represents the most general clauses corresponding to the refutation from that it is generated.

Lifting the Monadic Transformation. Since the Monadic transformation is satisfiability preserving, lifting always succeeds by replacing any $T(f_P(t_1,\ldots,t_n))$ atoms in the core by $P(t_1,\ldots,t_n)$.

Example 2 *Let $N_0 = \{P(x,x');\ P(y,a), P(z,b) \rightarrow\}$. Then $N_k = \{T(f_P(x,x'));\ T(f_P(y,a)), T(f_P(z,b)) \rightarrow\}$ is a Monadic transformation of N_0 and a conflicting core is $N_k^\perp = \{T(f_P(y,a));\ T(f_P(z,b));\ T(f_P(y,a)), T(f_P(z,b)) \rightarrow\}$. Reverting the atoms in $N_k^\perp$ gives $N^\perp = \{P(y,a);\ P(z,b);\ P(y,a), P(z,b) \rightarrow\}$ a conflicting core of N_0.*

Lemma 5 (Lifting the Monadic Transformation) *Let $N_k \Rightarrow^*_{MO} N_{k+l}$ be the transformation that exactly removes all occurrences of atoms $P(t_1,\ldots,t_n)$ and replaces those by atoms $T(f_P(t_1,\ldots,t_n))$. If $N_{k+l}^\perp$ is a conflicting core for N_{k+l} then there is a conflicting core $N_k^\perp$ of N_k.*

Proof. Since the Monadic transformation is satisfiability preserving, unsatisfiability of N_{k+l} directly implies unsatisfiability of N_k and the existence of a conflicting core of N_k.

Lifting the Horn Transformation. For a Horn transformation there are two ways for lifting. The first, directly lifting the core, only succeeds in special cases, where the original clause and its approximation are equivalent for the instantiations appearing in the core.

Example 3 *Let $N_0 = \{P(a,b) \rightarrow;\ P(x,b), P(a,y)\}$. Then $N_k = \{P(a,b) \rightarrow;\ P(x,b)\}$ is a Horn transformation of N_0 and a conflicting core is $N_k^\perp = \{P(a,b) \rightarrow;\ P(a,b)\}$. By substituting y with b, $N_k^\perp$ lifts to $N^\perp = \{P(a,b) \rightarrow;\ P(a,b), P(a,b)\}$ a conflicting core of N_0.*

Lemma 6 (Lifting the Horn Transformation (direct)) *Let $N_k \Rightarrow_{HO} N_{k+1}$ where $N_k = N \cup \{\Gamma \rightarrow E_1,\ldots,E_n\}$ and $N_{k+1} = N \cup \{\Gamma \rightarrow E_i\}$. Let $N_{k+1}^\perp$ be a conflicting core of N_{k+1}. If for all $(\Gamma \rightarrow E_i)\sigma_j \in N_{k+1}^\perp$, $1 \le j \le m$ there is a substitution σ'_j such that $N_k^j\tau_j \models (\Gamma \rightarrow E_1,\ldots,E_n)\sigma'_j \rightarrow (\Gamma \rightarrow E_i)\sigma_j$, such that $N_k^j \subseteq N_k$ and $N_k^j\tau_j \cup \{(\Gamma \rightarrow E_1,\ldots,E_n)\sigma'_j, \neg(\Gamma \rightarrow E_i)\sigma_j\}$ is a conflicting core, then $N_{k+1}^\perp \setminus \{(\Gamma \rightarrow E_i)\sigma_j \mid 1 \le j \le m\} \cup \{(\Gamma \rightarrow E_1,\ldots,E_n)\sigma'_j \mid 1 \le j \le m\} \cup \bigcup_j N_k^j\tau_j$ is a conflicting core of N_k.*

Proof. Let σ be a grounding substitution for $N_k^\perp$ and $N_{k+1}^\perp$. Since $N_k \models (\Gamma \rightarrow E_1,\ldots,E_n)\sigma'_j \rightarrow (\Gamma \rightarrow E_i)\sigma_j$, $N_k^\perp\sigma \models N_k^\perp\sigma \cup \{(\Gamma \rightarrow E_i)\sigma_j \mid 1 \le j \le m\}\sigma \models N_{k+1}^\perp\sigma$. Hence, $N_k^\perp\sigma$ is unsatisfiable because $N_{k+1}^\perp\sigma$ is unsatisfiable. Therefore, $N_k^\perp$ is an conflicting core of N_k.

Of course, the condition $N_k^j\tau_j \models (\Gamma \rightarrow E_1,\ldots,E_n)\sigma'_j \rightarrow (\Gamma \rightarrow E_i)\sigma_j$ itself is undecidable, in general. The above lemma is meant to be a justification for the cases where this relation can be decided, e.g, by reduction. In general, the next lemma applies. We assume any non-Horn clauses have exactly two positive literals. Otherwise, we would have first redefined pairs of positive literals using fresh predicates. Further assume w.l.o.g. that Horn transformation always chooses the first positive Literal of a non-Horn clause.

The indirect method uses the information from the conflicting core to replace the non-Horn clause with a satisfiable equivalent unit clause, which is then solved recursively. Since this unit clause is already Horn, we lifted one Horn approximation step.

Example 4 *Let $N_k = \{P(a), Q(a); P(x) \rightarrow\}$. The Horn transformation $N_k = \{P(a); P(x) \rightarrow\}$ has a conflicting core $N_k^\perp = \{P(a); P(a) \rightarrow\}$. $N_k^\perp$ abstracts a resolution refutation with $\perp$ as the result. If we replace $P(a)$ with $P(a), Q(a)$ in such a refutation, the result will be $Q(a)$ instead and hence $N_k \vDash Q(a)$ Since $Q(a)$ subsumes $P(a), Q(a)$,*
N_k is satisfiable if $N_k' = \{Q(a); P(x) \rightarrow\}$ is too.

Lemma 7 (Lifting the Horn Transformation (indirect)) *Let N be a set of variable disjoint clauses, $N \Rightarrow^*_{APR} N_k \Rightarrow_{HO} N_{k+1}$, $N_k = N \cup \{\Gamma \rightarrow E_1, E_2\}$ and $N_{k+1} = N \cup \{\Gamma \rightarrow E_1\}$ and $N_{k+1}^\perp$ be a conflicting core of N_{k+1} where Lemma 6 does not apply. Let $(\Gamma \rightarrow E_1)\sigma \in N_{k+1}^\perp$, where σ is a variable renaming and $N_k^j \tau_j \not\models (\Gamma \rightarrow E_1, E_2)\sigma_j' \rightarrow (\Gamma \rightarrow E_1)\sigma$ for any $N_k^j \subseteq N_k$,τ_j and σ_j'. If there exists a conflicting core $N^\perp$ of $N \cup \{E_2\}$, then a conflicting core of N_k exists.*

Proof. From the conflicting core $N_{k+1}^\perp$, we can conclude that there exists an unsatisfiability proof of N_{k+1} which derives $\perp$ and uses $(\Gamma \rightarrow E_1)\sigma$ as the only instance of $\Gamma \rightarrow E_1$. If we were to replace $(\Gamma \rightarrow E_1)\sigma$ by $(\Gamma \rightarrow E_1, E_2)\sigma$, the unsatisfiability proof's root clause would instead be $E_2\sigma$. Hence, we know that $N_k \models N_k \cup \{E_2\sigma\}$. Furthermore, $N_k \models N \cup \{E_2\sigma\}$ since $E_2\sigma$ subsumes $\Gamma \rightarrow E_1, E_2$.

Let $E_2\sigma_j \in N^\perp$ for $1 \leq j \leq m$ and $N_k^{E_2} = N_{k+1}^\perp \setminus \{(\Gamma \rightarrow E_1)\sigma\} \cup \{(\Gamma \rightarrow E_1, E_2)\sigma\}$ Then $N^\perp \setminus \{E_2\sigma_j \mid 1 \leq j \leq m\} \bigcup_j N_k^{E_2}\sigma_j$ is a conflict core of N_k.

Note that N_k now again contains the Non-Horn clause $\Gamma \rightarrow E_1, E_2$. Then, in a following indirect Horn lifting step $\Gamma \rightarrow E_1, E_2$ can not necessarily be again replaced by $E_2\sigma$. Hence, the indirect Horn lifting needs to be repeated.

Lifting the Shallow Transformation. A Shallow transformation introduces a new predicate S, which is removed in the lifting step. We take all clauses with S-atoms in the conflicting core and generate any possible resolutions on S-atoms. The resolvents, which don't contain S-atoms anymore, then replace their parent clauses in the core. Lifting succeeds if all introduced resolvents are instances of clauses before the shallow transformation.

Example 5 *Let $N_0 = \{P(x), Q(y) \rightarrow R(x, f(y)); P(a); Q(b); R(a, f(b)) \rightarrow\}$. Then $N_k = \{S(x'), P(x) \rightarrow R(x, x'); Q(y) \rightarrow S(f(y)); P(a); Q(b);\ R(a, f(b)) \rightarrow\}$*

is a Shallow transformation of N_0 *and a conflicting core is* $N_k^\perp = S(f(b)),$ $P(a) \to R(a, f(b)); Q(b) \to S(f(b)); P(a); Q(b); R(a, f(b)) \to$. *By replacing* $S(f(b)), P(a) \to R(a, f(b))$ *and* $Q(b) \to S(f(b))$ *with the resolvent,* $N_k^\perp$ *lifts to* $N^\perp = \{P(a), Q(b) \to R(a, f(b)); P(a); Q(b); R(a, f(b)) \to\}$ *a conflicting core of* N_0.

Lemma 8 (Lifting the Shallow Transformation) *Let* $N_k \Rightarrow_{SH} N_{k+1}$ *where* $N_k = N \cup \{\Gamma \to E[s]_p\}$ *and* $N_{k+1} = N \cup \{S(x), \Gamma_1 \to E[x]_p\} \cup \{\Gamma_2 \to S(s)\}$. *Let* $N_{k+1}^\perp$ *be a conflicting core of* N_{k+1}. *Let* N_S *be the set of all resolvents from clauses from* $N_{k+1}^\perp$ *on the* S *literal. If for all clauses* $C_j \in N_S$, $1 \leq j \leq m$ *there is a substitution* σ_j *such that* $C_j = (\Gamma \to E[s]_p)\sigma_j$ *then* $N_{k+1}^\perp \setminus \{C \mid C \in N_{k+1}^\perp$ *and contains an* S*-atom*$\} \cup \{(\Gamma \to E[s]_p)\sigma_j \mid 1 \leq j \leq m\}$ *is a conflicting core of* N_k.

Proof. Let σ be a grounding substitution for $N_k^\perp$ and $N_{k+1}^\perp$ and I be an interpretation. As $N_{k+1}^\perp\sigma$ is unsatisfiable, there is a clause $D \in N_{k+1}^\perp\sigma$ such that $I \not\models D$. If D does not contain an S-atom, then $D \in N_k^\perp\sigma$ and hence $I \not\models N_k^\perp\sigma$.
Now assume only clauses that contain S-atoms are false under I. By construction, any such clause is equal to either $(S(x), \Gamma_1 \to E[x]_p)\sigma' = C_1\sigma'$ or $(\Gamma_2 \to S(s))\sigma' = C_2\sigma'$ for some substitution σ'. Let $I' := \{S(s)\sigma' \mid C_2\sigma' \in N_{k+1}^\perp\sigma \text{ and } I \not\models C_2\sigma'\} \cup I \setminus \{S(x)\sigma' \mid C_1\sigma' \in N_{k+1}^\perp\sigma \text{ and } I \not\models C_1\sigma'\}$, i.e., we change the truth value for S-Literals such that the clauses unsatisfied under I are satisfied under I'.
Since I and I' only differ on literals with predicate S and $N_{k+1}^\perp\sigma$ is unsatisfiable, some clause C, containing an S-atom and satisfied under I, has to be false under I'.
Let $C = C_1\sigma_1$. Since $I \models C$, $S(x)\sigma_1$ was added to I' by some clause $D = C_2\sigma_2$, where $S(s)\sigma_2 = S(x)\sigma_1$. Hence, C and D can be resolved on their S-literals and the resolvent R is in $N_k^\perp\sigma$. Since $I \not\models D$, $I' \not\models C$ and R contains no S-atom, $I \not\models R$ and therefore $I \not\models N_k^\perp\sigma$.
For $C = C_2\sigma_2$ the proof is analogous.
Thus, for all interpretations I and grounding substitutions σ, $I \not\models N_k^\perp\sigma$ and hence $N_k^\perp\sigma$ is a conflicting core of N_k.

Lifting the Linear Transformation. In order to lift a Linear transformation the remaining and the newly introduced variable need to be instantiated the same term.

Example 6 *Let* $N_{k-1} = \{P(x, x);\ P(y, a), P(z, b) \to\}$. *Then* $N_k = \{P(x, x');$ $P(y, a), P(z, b) \to\}$ *is a Linear transformation of* N_{k-1} *and and* $N_k^\perp = \{P(a, a);$ $P(b, b);\ P(a, a), P(b, b) \to\}$ *is a conflicting core of* N_k. *Since* $P(a, a)$ *and* $P(b, b)$ *are instances of* $P(x, x)$ *lifting succeeds and* $N_k^\perp$ *is also a core of* N_{k-1}.

Lemma 9 (Lifting the Linear Transformation) *Let* $N_k \Rightarrow_{LI} N_{k+1}$ *where* $N_k = N \cup \{\Gamma \to E[x]_{p,q}\}$ *and* $N_{k+1} = N \cup \{\Gamma\{x \mapsto x'\}, \Gamma \to E[x']_q\}$. *Let* $N_{k+1}^\perp$ *be a conflicting core of* N_{k+1}. *If for all* $(\Gamma\{x \mapsto x'\}, \Gamma \to E[x']_q)\sigma_j \in N_{k+1}^\perp$,

$1 \leq j \leq m$ *we have* $x\sigma_j = x'\sigma_j$ *then* $N^{\perp}_{k+1} \setminus \{(\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\sigma_j \mid 1 \leq j \leq m\} \cup \{(\Gamma \rightarrow E[x]_{p,q})\sigma_j \mid 1 \leq j \leq m\}$ *is a conflicting core of* N_k.

Proof. Let σ be a grounding substitution for $N^{\perp}_k$ and $N^{\perp}_{k+1}$. As $x\sigma_j = x'\sigma_j$ for $1 \leq j \leq m$, $(\Gamma \rightarrow E[x]_{p,q})\sigma_j\sigma \models (\Gamma, \Gamma \rightarrow E[x]_{p,q})\sigma_j\sigma = (\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\sigma_j\sigma$. Hence, $N^{\perp}_k\sigma \models N^{\perp}_k\sigma \cup \{(\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\sigma_j\sigma \mid 1 \leq j \leq m\} \models N^{\perp}_{k+1}\sigma$. Since $N^{\perp}_{k+1}\sigma$ is unsatisfiable $N^{\perp}_k\sigma$ is unsatisfiable as well. Therefore, $N^{\perp}_k$ is a conflicting core of N_k.

Lifting with Instantiation. By definition, if $N^{\perp}$ is a conflicting core of N, then $N^{\perp}\tau$ is also a conflicting core of N for any τ. Example 7 shows it is sometimes possible to instantiate a conflicting core, where no lifting lemma applies, into a core, where one does. This then still implies a successful lifting.

Example 7 *Let* $N_{k-1} = \{P(x,x);\ P(y,a), P(z,b) \rightarrow\}$. *Then* $N_k = \{P(x,x');\ P(y,a), P(z,b) \rightarrow\}$ *is a Linear transformation of* N_{k-1} *and and* $N^{\perp}_k = \{P(y,a);\ P(b,b);\ P(y,a), P(b,b) \rightarrow\}$ *is a conflicting core of* N_k. *Since for* $P(y,a) = P(x,x')\sigma$ $x\sigma = y \neq a = x'\sigma$ *Lemma 9 is not applicable.*

However, Lemma 9 can be applied on $N^{\perp}_k\{y \mapsto a; z \mapsto b\} = \{P(a,a);\ P(b,b);\ P(a,a), P(b,b) \rightarrow\}$.

4 Approximation Refinement

In the previous section, we have presented the lifting process. If, however, in one of the lifting steps conditions of the lemma are not met, lifting fails and we now refine the original clause set in order to rule out the non-liftable conflicting core. Again, since lifting fails at one of the approximation steps, we consider the different approximation steps for refinement.

Linear Approximation Refinement. A Linear transformation enables further instantiations of the abstracted clause compared to the original, that is, two variables that were the same can now be instantiated differently. If the conflicting core of the approximation contains such instances the lifting fails.

Definition 7 (Linear Approximation Refinement) *Let* N *be a set of variable disjoint clauses,* $N \Rightarrow^{*}_{APR} N_k \Rightarrow_{LI} N_{k+1}$ *and* $N^{\perp}_{k+1}$ *be a conflicting core of* N_{k+1} *where Lemma 9 does not apply. Let* $C'\sigma = (\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\sigma \in N^{\perp}_{k+1}$ *such that* $x\sigma$ *and* $x'\sigma$ *have no common instances. Let* $C \in N$ *be the Ancestor of* $C' \in N_{k+1}$. *Then the* linear approximation refinement *of* N, C, x, x', σ *is the clause set* $N \setminus \{C\} \cup \{C\tau_1, \ldots, C\tau_n\}$ *where the* $C\tau_i$ *are the specific instances of* C *with respect to the substitutions* $\{x \mapsto x\sigma\}$ *and* $\{x \mapsto x'\sigma\}$.

Note that if there is no $C'\sigma$, where $x\sigma$ and $x'\sigma$ have no common instances, it implies that there is a substitution τ where Lemma 9 applies on $N^{\perp}_{k+1}\tau$. Hence, $N^{\perp}_{k+1}\tau$ is a liftable conflicting core.

Let $N_0 \Rightarrow^*_{APR} N_{k-1} = N \cup \{\Gamma \rightarrow E[x]_{p,q}\} \Rightarrow_{\text{LI}} N_k = N \cup \{\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q\}$ and the core $N_k^\perp$ of N_k contains the clause $C'\sigma = (\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\sigma$, where $x\sigma$ and $x'\sigma$ have no common instances. After applying Linear Approximation Refinement, there are $C\tau_i$ and $C\tau_j$ with $i \neq j$ such that $C\tau_i$ contains all instances where $\{x \mapsto x\sigma\}$ and $C\tau_j$ contains all instances where $\{x \mapsto x'\sigma\}$. Assume there is a C'' with an ancestor $C\tau$ such that $C'\sigma$ is an instance of C''. This would imply that $C\tau$ has instances, where $\{x \mapsto x\sigma\}$ and $\{x \mapsto x'\sigma\}$. Then $C\tau_i = C\tau = C\tau_j$, which is a contradiction to Definition 2.

Example 8 *Let $N_0 = \{P(x, x);\ P(y, a), P(z, b) \rightarrow\}$. Then $N_k = \{P(x, x');\ P(y, a), P(z, b) \rightarrow\}$ is a Linear transformation of N_0 and and $N_k^\perp = \{P(a, a);\ P(a, b);\ P(a, a), P(a, b) \rightarrow\}$ is a conflicting core of N_k.*
Due to $P(a, b) = P(x, x')\{x \mapsto a, x' \mapsto b\}$ lifting fails. The Linear Approximation Refinement replaces $P(x, x)$ in N_0 with $P(a, a)$ and $P(b, b)$. In the refined approximation $N_k' = \{P(a, a); P(b, b);\ P(y, a), P(z, b) \rightarrow\}$ the violating clause $P(a, b)$ is not an instance of N_k' and hence, the not-liftable conflicting core $N_k^\perp$ cannot be found again.

Shallow Approximation Refinement. The Shallow transformation is somewhat more complex than linear transformation, but the idea behind it is very similar to the linear case. As mentioned before, the Shallow transformation can always be lifted if the set of common variables $\text{vars}(\Gamma_2, s) \cap \text{vars}(\Gamma_1, E[x]_p)$ is empty. Otherwise, each such variable potentially introduces instantiations that are not liftable.

Definition 8 (Shallow Approximation Refinement) *Let N be a set of variable disjoint clauses, $N \Rightarrow^*_{APR} N_k \Rightarrow_{SH} N_{k+1}$ and $N_{k+1}^\perp$ be a conflicting core of N_{k+1} where Lemma 8 does not apply. Let C_R be the resolvent from the final Shallow rule application such that $C_R \neq (\Gamma \rightarrow E[s]_p)\sigma_R$ for any σ_R. Let $C_1\sigma_1 \in N_{k+1}^\perp$ and $C_2\sigma_2 \in N_{k+1}^\perp$ be the parent clauses of C_R. Let $y \in dom(\sigma_1) \cap dom(\sigma_2)$, where $y\sigma_1$ and $y\sigma_2$ have no common instances. Let $C \in N$ be the Ancestor of $C_1 \in N_{k+1}$. Then the* shallow approximation refinement *of N, C, x, σ_1, σ_2 is the clause set $N \setminus \{C\} \cup \{C\tau_1, \ldots, C\tau_n\}$ where the $C\tau_i$ are the specific instances of C with respect to the substitutions $\{x \mapsto x\sigma_1\}$ and $\{x \mapsto x\sigma_2\}$.*

As in Linear Approximation Refinement, if for every resolvent $C_R\sigma$ $y\sigma_1$ and $y\sigma_2$ have common instances, it implies that there is a substitution τ where Lemma 8 applies on $N_{k+1}^\perp\tau$. After applying Shallow Approximation Refinement, there are $C\tau_i$ and $C\tau_j$ with $i \neq j$ such that $C\tau_i$ contains all instances where $\{x \mapsto x\sigma_1\}$ and $C\tau_j$ contains all instances where $\{x \mapsto x\sigma_2\}$. Hence, $C\tau_i$ is now the ancestor of $C_1\sigma_1$, while $C\tau_j$ is the ancestor of $C_2\sigma_2$. Since they have different ancestors, they can no longer be resolved on their S-atoms which now have different predicates. Hence C_R is no longer a resolvent in the conflicting core.

Example 9 *Let $N_0 = \{P(f(x, g(x))); P(f(a, g(b)) \rightarrow\}$ with signature $\Sigma = a/0$, $b/0$, $g/1, f/2$. Then $N_k = \{S(z) \rightarrow P(f(x, z)); S(g(y));\ P(f(a, g(b)) \rightarrow\}$ is a*

Shallow transformation of N_0 *and and* $N_k^\perp = \{S(g(b)) \to P(f(a, g(b))); S(g(b));$ $P(f(a, g(b)) \to\}$ *is a conflicting core of* N_k.
The clauses $S(g(b)) \to P(f(a, g(b)))$ *and* $S(g(b))$ *have the resolvent* $P(f(a, g(b)))$, *which is not an instance of* $P(f(x, g(x)))$. *The Shallow Approximation Refinement replaces* $P(f(x, g(x)))$ *in* N_0 *with* $P(f(a, g(a)))$, $P(f(b, g(b)))$, $P(f(g(x), g(g(x))))$ *and* $P(f(f(x, y), g(f(x, y))))$.
The approximation of the refined N_0 *is now satisfiable.*

Horn Approximation Refinement. Lifting a core of a Horn transformation fails, if the positive literals removed by the Horn transformation are not dealt with in the approximated proof. Since Lemma 7 only handles cases where the approximated clause appears uninstantiated in the conflicting core, the Horn Approximation Refinement is used to ensure such a core exists.

Definition 9 (Horn Approximation Refinement) *Let* N *be a set of variable disjoint clauses,* $N \Rightarrow^*_{APR} N_k \Rightarrow_{HO} N_{k+1}$, $N_k = N \cup \{\Gamma \to E_1, E_2\}$ *and* $N_{k+1} = N \cup \{\Gamma \to E_1\}$ *and* $N_{k+1}^\perp$ *be a conflicting core of* N_{k+1} *where Lemmas 6 and 7 do not apply. Let* $(\Gamma \to E_1)\sigma \in N_{k+1}^\perp$ *be a clause from the final Horn rule application such that* σ *is not a variable renaming and* $N_k^j \tau_j \not\models (\Gamma \to E_1, E_2)\sigma'_j \to (\Gamma \to E_1)\sigma$ *for any* $N_k^j \subseteq N_k, \tau_j$ *and* σ'_j. *Let* $C \in N$ *be the Ancestor of* $\Gamma \to E_1 \in N_{k+1}$ *and* σ' *a substitution such that* $\sigma\sigma'$ *is linear for* C. *Then the* horn approximation refinement I *of* N, C, σ, σ' *is the clause set* $N \setminus \{C\} \cup \{C\sigma\sigma', C\tau_1, \dots, C\tau_n\}$ *where the* $C\tau_i$ *are the specific instances of* C *with respect to the substitutions* $\sigma\sigma'$.

Note that the condition for the extended version of specific instantiation to have a finite representation is not generally met by an arbitrary σ. Therefore, σ may need to be further instantiated or even made ground. After the Horn Approximation Refinement, Lemma 7 can be applied on the clause with ancestor $C\sigma\sigma'$.

Example 10 *Let* $N_0 = \{P(x), Q(x); P(a) \to\}$ *with signature* $\Sigma = a/0, f/1$. *The Horn transformation* $N_k = \{P(x); P(a) \to\}$ *has a conflicting core* $N_k^\perp = \{P(a); P(a) \to\}$. *We pick* $\to P(a)$ *as the instance of* $P(x) \in N_k^\perp$ *to use for the Horn Approximation Refinement. The result is* $N'_0 = \{P(a), Q(a); P(f(x)), Q(f(x));$ $P(a) \to\}$ *and its approximation also has* $N_k^\perp$ *as a conflicting core. However, now Lemma 7 applies.*

Lemma 10 (Completeness) *Let* N *be an unsatisfiable clause set and* N_k *its approximation. Then, there exists a conflicting core of* N_k *that can be lifted to* N.

Proof. by induction on the number k of approximation steps. The case $k = 0$ is obvious. For $k > 0$, let $N \Rightarrow^*_{APR} N_{k-1} \Rightarrow_{APR} N_k$. By the inductive hypothesis, there is a conflicting core $N_{k-1}^\perp$ of N_{k-1} which can be lifted to N.
The final approximation rule application is either a Linear, a Shallow, a Horn or a Monadic transformation, considered below by case analysis.

Linear Case. Let $N \Rightarrow^*_{APR} N_{k-1} = N' \cup \{\Gamma \rightarrow E[x]_{p,q}\} \Rightarrow_{\mathrm{LI}} N_k = N' \cup \{\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q\}$. For every $(\Gamma \rightarrow E[x]_{p,q})\sigma_j \in N^{\perp}_{k-1}$ $1 \leq j \leq m$, $(\Gamma \rightarrow E[x]_{p,q})\sigma_j \models (\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)(\{x' \mapsto x\}\sigma_j)$. Hence $N^{\perp}_k = N^{\perp}_{k-1} \setminus \{(\Gamma \rightarrow E[x]_{p,q})\sigma_j \mid 1 \leq j \leq m\} \cup \{(\Gamma\{x \mapsto x'\}, \Gamma \rightarrow E[x']_q)\{x' \mapsto x\}\sigma_j \mid 1 \leq j \leq m\}$ is a conflicting core of N_k. By Lemma 9 $N^{\perp}_k$ can be lifted back to $N^{\perp}_{k-1}$. Hence, the conflicting core $N^{\perp}_k$ can be lifted to N.

Shallow Case. Let $N \Rightarrow^*_{APR} N_{k-1} = N' \cup \{\Gamma \rightarrow E[s]_p\} \Rightarrow_{\mathrm{SH}} N_k = N' \cup \{S(x), \Gamma_1 \rightarrow E[x]_p\} \cup \{\Gamma_2 \rightarrow S(s)\}$. We construct $N^{\perp}_S$ from $N^{\perp}_{k-1}$ by replacing every $(\Gamma \rightarrow E[s]_p)\sigma_j \in N^{\perp}_{k-1}$ $1 \leq j \leq m$ with $(S_j(x), \Gamma_1 \rightarrow E[x]_p)\sigma_j$ and $(\Gamma_2 \rightarrow S_j(s))\sigma_j$. $N^{\perp}_S$ is a conflicting core, which by m applications of Lemma 8 on each S_j can be lifted to $N^{\perp}_{k-1}$. From $N^{\perp}_S$ we get $N^{\perp}_k$ by renaming every S_j into S, which is a conflicting core of N_k. The existence of $N^{\perp}_S$ shows that $N^{\perp}_k$ can be lifted to $N^{\perp}_{k-1}$.

Horn Case. W.l.o.g. let $N \Rightarrow^*_{APR} N_{k-1} = N' \cup \{\Gamma \rightarrow E_1, E_2\} \Rightarrow_{\mathrm{HO}} N_k = N' \cup \{\Gamma \rightarrow E_1\}$. Let $C = \Gamma \rightarrow E_1, E_2$ and $C' = \Gamma \rightarrow E_1$. If $C\sigma \in N^{\perp}_{k-1}$ holds for at most one σ, we construct $N^{\perp}_k$ from $N^{\perp}_{k-1}$ by replacing $C\sigma$ with $C'\sigma$ such that $N^{\perp}_k \subseteq N_k$. Since $C'\sigma$ subsumes $C\sigma$, $N^{\perp}_n \models N^{\perp}_n \cup \{C\sigma\}$. As $N^{\perp}_k \cup \{C\sigma\}$ is a superset of $N^{\perp}_{k-1}$, $N^{\perp}_k$ is therefore a ground conflicting core of N_k. If $C'\sigma$ and $C\sigma$ are already equivalent, $N^{\perp}_k$ can be lifted to $N^{\perp}_{k-1}$. Otherwise, let $N'^{\perp}_{k-1}$ be $N^{\perp}_{k-1}$ where $C\sigma$ is instead replaced by $E_2\sigma$. Again since $E_2\sigma$ subsumes $C\sigma$, $N'^{\perp}_{k-1}$ is a ground conflicting core. As shown before, $(N'^{\perp}_{k-1} \setminus \{E_2\sigma\}) \cup (N^{\perp}_k \setminus \{C'\sigma\}) = N^{\perp}_{k-1}$ is a lifting from N_k to N_{k-1}.
Assume $C\sigma_1 \in N^{\perp}_{k-1}$ and $C\sigma_2 \in N^{\perp}_{k-1}$ holds for $\sigma_1 \neq \sigma_2$. In this case the original clause C can be specifically instantiated in such a way that $C\sigma_1$ and $C\sigma_2$ are no longer instances of the same clause, while $N^{\perp}_{k-1}$ remains a conflicting core. Hence, after finitely many such partitions eventually the first case will hold.

Monadic Case. Let $N \Rightarrow^*_{APR} N_{k-j} \Rightarrow^*_{\mathrm{MO}} N_k$ where N_{k-j} has no occurrence of an atom $T(f_P(t_1, \ldots, t_n))$ and N_k no occurrence of an atom $P(t_1, \ldots, t_n)$ and all introduced atoms in the transformation are of the form $T(f_P(s_1, \ldots, s_n))$. By the inductive hypothesis, there is a ground conflicting core $N^{\perp}_{k-j}$ of N_{k-j} which can be lifted to N. By Lemma 1(ii) Monadic transformation preserves unsatisfiability and therefore $\mu_P(N^{\perp}_{k-j})$ is a ground conflicting core of N_k. $\mu_P(N^{\perp}_{k-j})$ can be lifted to $\mu_P^{-1}(\mu_P(N^{\perp}_{k-j})) = N^{\perp}_{k-j}$ a conflicting core of N_{k-j}.

The above lemma considers static completeness, i.e., it does not tell how the conflicting core that can eventually be lifted is found. One way is to enumerate all refutations of N_k in a fair way. A straightforward fairness criterion is to enumerate the refutations by increasing term depth of the clauses used in the refutation. Since the decision procedure on the mslH fragment [12] generates only finitely many different non-redundant clauses not exceeding a concrete term depth with respect to the renaming of variables, eventually the liftable refutation will be generated.

5 Future and Related Work

The condition for the lifing lemma for Shallow transformation (Lemma 8) is stronger than necessary, as the following example shows.

Example 11 *Let* $N_0 = \{P(x,z), Q(y,z) \rightarrow R(x, f(y)); P(a,a);\ P(a,b);$ $Q(b,a), Q(b,b);\ R(a, f(b)) \rightarrow\}$ *and* $N_k = \{S(y), P(x,z) \rightarrow R(x,y); Q(y,z) \rightarrow S(f(y)); P(a,a); P(a,b); Q(b,a), Q(b,b);\ R(a,f(b)) \rightarrow\}$ *is a Shallow transformation of* N_k*.* N_0 *and* N_k *are unsatisfiable and* $N_k^\perp = \{S(f(b)),\ P(a,a) \rightarrow R(a,f(b)); Q(b,a) \rightarrow S(f(b)); S(f(b)), P(a,b) \rightarrow R(a,f(b)); Q(b,b) \rightarrow S(f(b)); P(a,a); P(a,b); Q(b,a), Q(b,b); R(a,f(b)) \rightarrow\}$ *is a conflicting core of* N_k*. Lifting* $N_k^\perp$ *fails because the resolvent* $P(a,a), Q(b,b) \rightarrow R(a,f(b))$ *is not an instance of* $P(x,z), Q(y,z) \rightarrow R(x, f(y))$*. However, if we ignored the violating resolvents, it would result in the valid conflicting core* $N^\perp = \{P(a,a), Q(b,a) \rightarrow R(a,f(b)); P(a,b), Q(b,b) \rightarrow R(a,f(b)); P(a,a); P(a,b);\ Q(b,a), Q(b,b);$ $R(a,f(b)) \rightarrow\}$*.*

This does not break lifting. The shallow refinement will partition the clause in such a way that the resolvents that violate the lifting condition are one-by-one removed. In Example 11, the refinement would partition $P(x,z), Q(y,z) \rightarrow R(x,f(y))$ on the variable z. This will result in $S(f(b)),\ P(a,a) \rightarrow R(a,f(b))$ and $Q(b,b) \rightarrow S(f(b))$ containing different S-predicates and hence no longer being resolvable.

However, a refinement is not necessary to achieve this effect. The necessary information can be taken from the refutation and incorporated into the conflicting core during construction.

If a problem N is unsatisfiable, not only does there exist an unsatisfiability proof but one where S-literals only occur on leaves. Such a proof can be found by a ordered resolution calculus through selecting negative S-literals and an ordering where positive S-literals are strictly maximal. Given such a setting a solver will only resolve a clause $S(x), \Gamma_1 \rightarrow E[x]_{p_1,\ldots,p_n}$ with $\Gamma_2 \rightarrow S(s)$ on the S-atom and hence any S-atom will only appear at the leaves of the refutation.

In such a proof, we then uniquely rename the S-predicate in each pair of leaves. The conflicting core constructed from this proof then only allows resolutions on S-literals that also occur in the proof. On this core we can then check the lifting condition.

In example 11 the core would then instead be $\{S_1(f(b)), P(a,a) \rightarrow R(a,f(b));$ $Q(b,a) \rightarrow S_1(f(b)); S_2(f(b)), P(a,b) \rightarrow R(a,f(b)); Q(b,b) \rightarrow S_2(f(b));\ P(a,a);$ $P(a,b); Q(b,a), Q(b,b); R(a,f(b)) \rightarrow\}$. This core is liftable to $N^\perp$ by Lemma 8.

Related Work In "A theory of abstractions" [2] Giunchiglia and Walsh don't define an actual approximation but a general framework to classify and compare approximations, which are here called abstractions. They informally define abstractions as "the process of mapping a representations of a problem" that "helps deal with the problem in the original search space by preserving certain desirable properties“ and "is simpler to handle“.

In their framework an abstraction is a mapping between formal systems, i.e., a triple of a language, axioms and deduction rules, which satisfy one of the following conditions: An increasing abstraction (TI) f maps theorems only to theorems, i.e., if α is a theorem, then $f(\alpha)$ is also a theorem, while a decreasing abstraction (TD) maps only theorems to theorems, i.e., if $f(\alpha)$ is a theorem, then α was also a theorem.

Furthermore, they define dual definitions for refutations, where not theorems but formulas that make a formal system inconsistent are considered. An increasing abstraction (NTI) then maps inconsistent formulas only to inconsistent formulas and vice versa for decreasing abstractions (NTD).

They list several examples of abstractions such as ABSTRIPS by Sacerdoti [10], a GPS planning method by Newell and Simon [7], Plaisted's theory of abstractions [9], propositional abstractions exemplified by Giunchiglia [1], predicate abstractions by by Plaisted [9] and Tenenberg [11], domain abstractions by Hobbs [3] and Iemielinski [4] and ground abstractions introduced by Plaisted [9].

With respect to their notions the approximation described in this paper is an abstraction where the desirable property is the over-approximation and the decidability of the fragment makes it simpler to handle. More specifically in the context of [2] the approximation is an NTI abstraction for refutation systems, i.e., it is an abstraction that preserves inconsistency of the original.

In Plaisted [9] three classes of abstractions are defined. The first two are ordinary and weak abstractions, which share the condition that if C subsumes D then every abstraction of D is subsumed by some abstraction of C. However, our approximation falls in neither class as it violates this condition via the Horn approximation. For example Q subsumes P, Q, but the Horn approximation P of P, Q is not subsumed by any approximation of Q. The third class are generalization functions, which change not the problem but abstract the resolution rule of inference.

The theorem prover iProver uses the Inst-Gen [5] method, where a first-order problem is abstracted with a SAT problem by replacing every variable by the fresh constant $\perp$. The approximation is solved by a SAT solver and its answer is lifted to the original by equating abstracted terms with the set they represent, e.g., if $P(\perp)$ is true in a model returned by the SAT solver, then all instantiations of the original $P(x)$ are considered true as well. Inst-Gen abstracts using an under-approximation of the original clause set. In case the lifting of the satisfying model is inconsistent, the clash is resolved by appropriately instantiating the involved clauses, which mimics an inference step. This is the dual of our method with the roles of satisfiability and unsatisfiability switched. A further difference, however, is that Inst-Gen only finds finite models after approximation, while our approximation also discovers infinite models. For example the simple problem $\{P(a), \neg P(f(a)), P(x) \rightarrow P(f(f(x))), P(f(f(x))) \rightarrow P(x)\}$ has the satisfying model where P is the set of even numbers. However, iProver's approximation can never return such a model as any $P(f^n(\perp))$ will necessarily abstract both true and false atoms and therefore instantiate new clauses infinitely. Our method on the other hand will produce the approximation $\{P(a),$

$\neg P(f(a)),\ S(y) \rightarrow P(f(y)),\ P(x) \rightarrow S(f(x)),\ P(f(f(x))) \rightarrow P(x)\}$, which is saturated after inferring $P(x) \rightarrow P(f(f(x)))$ and $\neg S(f(a))$.

In summary, we have presented the first sound and complete calculus for first-order logic based on an over-approximation-refinement loop. There is no implementation so far, but the calculus will be practically useful if a problem is close to the mslH fragment in the sense that only a few refinement loops are needed for finding the model or a liftable refutation. The abstraction relation is already implemented and applying it to all satisfiable non-equality problems TPTP version 6.1 results in a success rate of 34%, i.e., for all these problems the approximation is not too crude and directly delivers the result.

It might be possible to apply our idea to other decidable fragments of first-order logic. However, then they have to support via approximation the presented lifting and refinement principle.

Our result is also a first step towards a model-based guidance of first-order reasoning. We proved that a model of the approximated clause set is also a model for the original clause set. For model guidance, we need this property also for partial models. For example, in the sense that if a clause is false with respect to a partial model operator on the original clause set, it is also false with respect to a partial model operator on the approximated clause set. This property does not hold for the standard superposition partial model operator and the mslH approximation suggested in this paper. It is subject to future research.

References

1. Fausto Giunchiglia and Enrico Giunchiglia. Building complex derived inference rules: A decider for the class of prenex universal-existential formulas. In *ECAI*, pages 607–609, 1988.
2. Fausto Giunchiglia and Toby Walsh. A theory of abstraction. *Artif. Intell.*, 57(2-3):323–389, October 1992.
3. Jerry R. Hobbs. Granularity. In *In Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 432–435. Morgan Kaufmann, 1985.
4. Tomasz Imielinski. Domain abstraction and limited reasoning. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'87, pages 997–1003, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.
5. Konstantin Korovin. Inst-gen - A modular approach to instantiation-based automated reasoning. In Andrei Voronkov and Christoph Weidenbach, editors, *Programming Logics - Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 239–270. Springer, 2013.
6. J.-L. Lassez and K. Marriott. Explicit representation of terms defined by counter examples. *J. Autom. Reason.*, 3(3):301–317, September 1987.
7. Allen Newell. *Human Problem Solving.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972.
8. Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving sat and sat modulo theories: From an abstract davis–putnam–logemann–loveland procedure to dpll(t). *Journal of the ACM*, 53:937–977, November 2006.
9. David A. Plaisted. Theorem proving with abstraction. *Artif. Intell.*, 16(1):47–108, 1981.

10. Earl D. Sacerdott. Planning in a hierarchy of abstraction spaces. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI'73, pages 412–422, San Francisco, CA, USA, 1973. Morgan Kaufmann Publishers Inc.
11. Josh Tenenberg. Preserving consistency across abstraction mappings. In *In Proceedings of the 10th IJCAI, pages 1011–1014. International Joint Conference on Artificial Intelligence*, 1987.
12. Christoph Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In Harald Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *LNAI*, pages 314–328. Springer, 1999.
13. Christoph Weidenbach. Combining superposition, sorts and splitting. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 27, pages 1965–2012. Elsevier, 2001.

A Skeleton and Partial Minimal Model Construction

As mentioned before, Lemma 4 also holds if satisfiability of N' is dropped and I is replaced by the superposition partial minimal model operator [13].

Definition 10 (Partial Minimal Model Construction) *Given the set of ground clauses N_g of N and an ordering $\prec$ we construct an interpretation $\mathcal{I}_N$ for N, called a partial model, inductively as follows:*

$$\mathcal{I}_C := \bigcup_{D \in N_g, D \prec C} \delta_D$$

$$\delta_D := \begin{cases} \{P\} & \text{if } D = D' \vee P,\ P \text{ strictly maximal and } \mathcal{I}_D \not\models D \\ \emptyset & \text{otherwise} \end{cases}$$

$$\mathcal{I}_N := \bigcup_{C \in N_g} \delta_C$$

Clauses C with $\delta_C \neq \emptyset$ are called productive.

Note that this construction doesn't terminate since the ground clause set of N is generally infinite.

Lemma 11 *Let N_k be a monadic clause set and N_0 be its approximation via $\Rightarrow_{APR}$. If $P(s) \in \mathcal{I}_{N_0}$ and P is a predicate in N_k, then there exists a clause $C = \Gamma \rightarrow \Delta, P(t) \in N_k$ and a substitution σ such that $s = skt(t)\sigma$ and for each variable x and predicate S with $C = S(x), \Gamma' \rightarrow \Delta, P(t[x]_p)$, $S(s'') \in \mathcal{I}_{N_0}$, where $s = s[s'']_p$.*

Proof. By induction on k.
The base $N_k = N_0$ holds by definition of the model operator $\mathcal{I}$.
Let $N = N_k \Rightarrow_{APR} N_{k-1} \Rightarrow^*_{APR} N_0$, $P(s) \in \mathcal{I}_{N_0}$ and P is a predicate in N_k and hence also in N_{k-1}. By the induction hypothesis, there exist a clause $C = \Gamma \rightarrow \Delta, P(t) \in N_{k-1}$ and a substitution σ such that $s = \mathrm{skt}(t)\sigma$ and for each variable x and predicate S with $C = S(x), \Gamma' \rightarrow \Delta, P(t[x]_p)$, $S(s'') \in \mathcal{I}_{N_0}$, where $s = s[s'']_p$.

Let $\Rightarrow_{APR}$ be a Horn transformation that replaces $\Gamma'' \rightarrow \Delta', Q(t')$ with $\Gamma'' \rightarrow Q(t')$. If $C \neq \Gamma'' \rightarrow Q(t')$, then $C \in N_k$ fulfills the claim. Otherwise, $\Gamma'' \rightarrow \Delta', Q(t) \in N_k$ fulfills the claim since $P = Q$ and $\Gamma' = \Gamma''$.

Let $\Rightarrow_{APR}$ be a linear transformation that replaces $C_k = \Gamma'' \to E[x]_{p,q}$ with $C_{k-1} = \Gamma'', \Gamma''\{x \mapsto x'\} \to E[x']_q$. If $C \neq C_{k-1}$, then $C \in N_k$ fulfills the claim. Otherwise, $C_k = \Gamma'' \to P(t)\{x' \mapsto x\} \in N_k$ fulfills the claim since $s = \text{skt}(t)\sigma = \text{skt}(t\{x' \mapsto x\})\sigma$ and $\Gamma'' \subseteq \Gamma'', \Gamma''\{x \mapsto x'\}$.

Let $\Rightarrow_{APR}$ be a shallow transformation that replaces $C_k = \Gamma'' \to E[s']_p$ with $C_{k-1} = S(x), \Gamma_1 \to E[x]_p$ and $C'_{k-1} = \Gamma_2 \to S(s')$. Since S is fresh, $C \neq C'_{k-1}$. If $C \neq C_{k-1}$, then $C \in N_k$ fulfills the claim. Otherwise, $C = C_{k-1} = S(x), \Gamma_1 \to P(t[x]_p)$ and hence, $s = \text{skt}(t[x]_p)\sigma$ and $S(s'') \in \mathcal{I}_{N_0}$ for $s = s[s'']_p$. Then by the induction hypothesis, there exist a clause $C_S = \Gamma_S \to \Delta_S, S(t_S) \in N_{k-1}$ and a substitution σ_S such that $s'' = \text{skt}(t_S)\sigma_S$ and for each variable x and predicate S' with $C_S = S'(x), \Gamma'_S \to \Delta_S, P(t_S[x]_q)$, $S'(s''') \in \mathcal{I}_{N_0}$, where $s'' = s''[s''']_q$. By construction, $C_S = C'_{k-1}$. Thus, $s'' = \text{skt}(s')\sigma_S$ and $s = \text{skt}(t[x]_p)\sigma$ imply there exists a σ'' such that $s = \text{skt}(t[s']_p)\sigma$". Furthermore since $\Gamma_1 \cup \Gamma_2 = \Gamma''$, if $C_k = S'(x), \Gamma''' \to P(t[s']_p)[x]_q$, then either $S'(x) \in \Gamma_1$ and thus $S'(s'''') \in \mathcal{I}_{N_0}$, where $s = s[s'''']_q$, or $S'(x) \in \Gamma_2$ and thus $S'(s'''') \in \mathcal{I}_{N_0}$, where $s[s'']_p = (s[s'']_p)[s'''']_q$. Hence, $C_k \in N_k$ fulfills the claim.

Lemma 12 *Let N be a clause set and N' be its approximation via $\Rightarrow_{APR}$. If $P(s) \in \mathcal{I}_{N'}$ ($T(f_p(s_1, \ldots, s_n)) \in \mathcal{I}_{N'}$) and P is a predicate in N, then there exist a clause $\Gamma \to \Delta, P(t) \in N$ ($\Gamma \to \Delta, P(t_1, \ldots, t_n) \in N$) and a substitution σ such that $s = skt(t)\sigma$ ($s_i = skt(t_i)\sigma$ for all i).*

Proof. Let $P_1, \ldots, P_n$ be the non-monadic predicates in N and $N_{MO} = \mu_{P_1}(\ldots(\mu_{P_n}(N)))$. Then, N_{MO} is monadic and also has N' as its approximation via $\Rightarrow_{APR}$.

Let $P(s) \in \mathcal{I}_{N'}$ and P is a predicate in N. Since P is monadic, P is a predicate in N_{MO}. Hence by Lemma 11, there exists a clause $\Gamma \to \Delta, P(t) \in N_{MO}$ and a substitution σ such that $s = \text{skt}(t)\sigma$. Then, $\mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma \to \Delta, P(t)))\ldots) = \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma)\ldots) \to \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Delta)\ldots), P(t) \in N$ fulfills the claim.

Let $T(f_p(s_1, \ldots, s_n)) \in \mathcal{I}_{N'}$ and P is a predicate in N. T is monadic and a predicate in N_{MO}. Hence by Lemma 11, there exists a clause $\Gamma \to \Delta, T(t) \in N_{MO}$ and a substitution σ such that $f_p(s_1, \ldots, s_n) = \text{skt}(t)\sigma$. Therefore, $t = f_p(t_1, \ldots, t_n)$ with $s_i = \text{skt}(t_i)\sigma$ for all i. Then, $\mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma \to \Delta, T(f_p(t_1, \ldots, t_n))))\ldots) = \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Gamma)\ldots) \to \mu^{-1}_{P_1}(\ldots(\mu^{-1}_{P_n}(\Delta)\ldots), P(t_1, \ldots, t_n) \in N$ fulfills the claim.