Exploring Partial Models with SCL

Martin Bromberger[1], Simon Schwarz[1,2] and Christoph Weidenbach[1]

[1] *Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany*
[2] *Graduate School of Computer Science, Saarbrücken, Germany*

Abstract
The family of SCL (Clause Learning from Simple Models) calculi learns clauses with respect to a partial model assumption, similar to CDCL (Conflict Driven Clause Learning). The partial model always consists of ground first-order literals and is built by decisions and propagations. In contrast to propositional logic where propagation chains are limited, in first-order logic they can become infinite. Therefore, the SCL family does not require exhaustive propagation and the size of the partial model is always finitely bounded. Any partial model not leading to a conflict constitutes a model for the respective finitely bounded ground clause set. We show that all potential partial models can be explored as part of the SCL calculus for first-order logic without equality and that any overall model is an extension of a partial model considered.

Keywords
first-order model building, first-order reasoning, non-redundant learning, SCL

1. Introduction

There are meanwhile three instances of the SCL calculus family: SCL for first-order logic without equality [1], SCL for first-order logic over theories [2], and SCL for first-order logic with equality [3]. They share: (i) an explicit trail (partial model assumption) built from ground literals, (ii) a finite limit to the potential size of trails and hence considered ground instances, and (iii) non-redundant clause learning. The finite limit to the trails size is a way to deal with potentially infinite propagations in first-order logic. For example from a trail $[P(a)]$ and a single clause $\neg P(x) \lor P(g(x))$ already infinitely many ground literals $P(g^i(a))$ can be propagated. Posing a finite limit on trail size let the SCL calculi run into *stuck* states. In a stuck state the partial model assumption is a model for the finitely considered ground instances of a clause set, but not necessarily for the clause set in general. In this paper we show that the search for a refutation as considered in previous work [1, 2, 3] can be combined with an exhaustive search for all partial ground models under the current finite limit. We finally prove that in fact for any model of the overall clause set, if it exists, our exhaustive search will yield the restriction of this model to the current finite limit, Theorem 16. Therefore, the SCL family enables simultaneous

PAAR'22: 8th Workshop on Practical Aspects of Automated Reasoning, August 11–12, 2022, Haifa, Israel
EMAIL: mbromber@mpi-inf.mpg.de (M. Bromberger); sschwarz@mpi-inf.mpg.de (S. Schwarz); weidenbach@mpi-inf.mpg.de (C. Weidenbach)
ORCID: 0000-0001-7256-2190 (M. Bromberger); 0000-0002-0064-2922 (S. Schwarz); 0000-0001-6002-0458 (C. Weidenbach)

search for a refutation and a model in a controlled way. The general idea of the new HSCL calculus is to learn a new clause from any stuck state, that prevents the repetition of the stuck state. Since such clauses are not logically implied by the initial clause set, they are treated separately.

The family of propositional CDCL calculi uses similar ideas if extended to optimization [4]. For example, if a satisfying assignment with "minimal weight" should be computed, already found assignments not improving the weight are ruled out by learning respective clauses, e.g., a clause consisting of the negation of all decisions leading to the assignment. In first-order logic there have been calculi developed that built an explicit model assumption, e.g. [5, 6, 7, 8, 9, 10], but to the best of our knowledge there is no calculus that learns new non-redundant clauses simultaneously towards a refutation and exhaustive model exploration.

The paper is now organized as follows. After clarifying some notions, Section 2, the HSCL calculus is introduced in Section 3. The HSCL calculus is an extension and unification of the already existing calculi for first-order logic without equality [1, 2]. The paper ends with a short discussion.

2. Preliminaries

We assume a first-order language without equality where N denotes a clause set; C, D denote clauses; L, K, H denote literals; A, B denote atoms; P, Q, R denote predicates; t, s terms; f, g, h function symbols; a, b, c constants; and x, y, z variables. Atoms, literals, clauses and clause sets are considered as usual, where in particular clauses are identified both with their disjunction and multiset of literals. The complement of a literal is denoted by the function comp. Semantic entailment $\models$ is defined as usual where variables in clauses are assumed to be universally quantified. Substitutions σ, τ are total mappings from variables to terms, where $\operatorname{dom}(\sigma) := \{x \mid x\sigma \neq x\}$ is finite and $\operatorname{codom}(\sigma) := \{t \mid x\sigma = t, x \in \operatorname{dom}(\sigma)\}$. Their application is extended to literals, clauses, and sets of such objects in the usual way. A term, atom, clause, or a set of these objects is *ground* if it does not contain any variable. A substitution σ is *ground* if $\operatorname{codom}(\sigma)$ is ground. A substitution σ is *grounding* for a term t, literal L, clause C if $t\sigma$, $L\sigma$, $C\sigma$ is ground, respectively. The function mgu denotes the *most general unifier* of two terms, atoms, literals. We assume that any mgu of two terms or literals does not introduce any fresh variables and is idempotent. A *closure* is denoted as $C \cdot \sigma$ and is a pair of a clause C and a grounding substitution σ. The function gnd returns the set of all ground instances of a literal, clause, or clause set with respect to the signature of the respective clause set.

A *partial model* M for a clause set N is a satisfiable set of ground literals. A ground clause C is true in M, denoted $M \models C$, if $C \cap M \neq \emptyset$, and false otherwise. A ground clause set N is true in M, denoted $M \models N$ if all clauses from N are true in M.

Let $\prec$ denote a well-founded, total, strict ordering on ground literals. This ordering is then lifted to clauses and clause sets by its respective multiset extension. We overload $\prec$ for literals, clauses, clause sets if the meaning is clear from the context. The ordering is lifted to the non-ground case via instantiation: we define $C \prec D$ if for all grounding substitutions σ it holds $C\sigma \prec D\sigma$. We define $\preceq$ as the reflexive closure of $\prec$ and $N^{\preceq C} := \{D \mid D \in N \text{ and } D \preceq C\}$.

Definition 1 (Clause Redundancy)**.** *A ground clause C is* redundant *with respect to a ground clause set N and an order $\prec$ if $N^{\preceq C} \models C$. A clause C is* redundant *with respect to a clause set N and an order $\prec$ if for all $C' \in \text{gnd}(C)$ it holds that C' is redundant with respect to* $\text{gnd}(N)$.

Let $\prec_B$ denote a well-founded, total, strict ordering on ground atoms such that for any ground atom A there are only finitely many ground atoms B with $B \prec_B A$. For example, an instance of such an ordering could be KBO without zero-weight symbols. The ordering $\prec_B$ is lifted to literals by comparing the respective atoms. It is lifted to clauses by a multiset extension. Given an ordering $\prec_B$ and a ground literal β, the function $\text{gnd}^{\prec_B \beta}$ computes the set of all ground instances of a literal, clause, or clause set where the grounding is restricted to produce literals L with $L \prec_B \beta$.

3. Exhaustive Partial Model Exploration with SCL

In this section, we restrict model exploration to finite ground models. Hence, models are build with respect to a maximal literal β and a literal ordering $\prec_B$. For fixed β and $\prec_B$, the proposed calculus HSCL always terminates: Either by finding a contradiction, or by exploring *all* partial models that are smaller than β with respect to $\prec_B$. Of course, those (finite) models are in general not extendable to a complete model of the clause set. If a clause set can be refuted by instantiating to ground literals smaller β, such a refutation will be found by HSCL. Thus, HSCL is complete for first-order logic when run with a sufficiently large β. Even in cases where no refutation exists, enumerating partial models yields information about the overall structure of complete models. Furthermore, clauses learned from conflicts during this process can be re-used in later runs to speed up exploration of the original problem.

The HSCL Rules

The inference rules of HSCL are represented by an abstract rewrite system. They operate on a problem state, a seven-tuple $(\Gamma; N; U; \beta; N'; k; D)$ where Γ is a sequence of annotated ground literals, the *trail*; N and U are the sets of *initial* and *learned* clauses; β is a ground literal limiting the literals considered for instantiation; N' is a set of clauses that excludes all already seen partial models; k counts the number of decisions; and D is a status that is either true $\top$, false $(\bot)_R$, finished with exploration $(\bot)_{\text{E}}$, or an annotated closure $((C \cdot \sigma))_u$, where $u \in \{\text{E}, R\}$

Literals in Γ have the form $X{:}L^r$, where $X \in \{\text{D}, \text{E}, \text{P}\}$ is the *type* of the literal and r its justification. The justification r is a level, a closure, or the combination of a level and a closure. We often omit irrelevant parts of the justification in specific contexts. The type can either be a Decision, a Propagation from N, or an Exclusion from N'. As we do not want to explore partial models twice, decisions are no longer completely arbitrary.

Instead, if a decision of a ground literal $L\sigma$ would lead to visiting a partial model again, HSCL will *exclude* this possibility by appending $\text{comp}(L\sigma)$ to the trail instead. This mechanism works similar to propagation. However, instead of propagating from $N \cup U$, excluded literals are propagated from N' and are therefore decision literals with respect to $N \cup U$. This mechanism is similar for conflict detection: *Regular* conflicts can be detected against clauses from $N \cup U$. In contrast, *excluded* conflicts are to a clause in N'. These two kinds of conflicts are distinguished by their respective annotation $u \in \{\text{E}, R\}$.

In the trail, furthermore, decided and excluded literals are annotated with a numerical level k, meaning that L is the k-th decided or excluded literal. Lastly, propagated and excluded literals are annotated with a closure that propagated the literal to become true.

A ground literal L is of *level* i with respect to a problem state $(\Gamma; N; U; \beta; N'; k; D)$ if L or $\text{comp}(L)$ occurs in Γ and the first decision or exclusion literal left from L ($\text{comp}(L)$) in Γ, including L, is annotated with i. If there is no such decision literal then its level is zero. A ground clause D is of *level* i with respect to a problem state $(\Gamma; N; U; \beta; N'; k; D)$ if i is the maximal level of a literal in D; the level of the empty clause $\bot$ is 0. A literal L is *undefined* in Γ if neither L nor $\text{comp}(L)$ occur in Γ. The initial state for a first-order clause set N is $(\epsilon; N; \emptyset; \beta; \emptyset; 0; \top)$, where β is an arbitrary but fixed literal.

The basic rules for trail building, Propagate and Decide, are left unmodified compared to the original SCL calculus, except for the difference that literals on the trail are now annotated with their respective source. However, the trail building rules are supplemented with the Exclude rule.

Propagate $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\Gamma, \text{P}{:}L\sigma^{(C_0 \vee L)\delta\cdot\sigma}; N; U; \beta; N'; k; \top)$
provided $C \vee L \in (N \cup U)$, $C = C_0 \vee C_1$, $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, $C_0\sigma$ does not contain $L\sigma$, δ is the mgu of the literals in C_1 and L, $(C \vee L)\sigma$ is ground, $(C \vee L)\sigma \prec_B \{\beta\}$, $C_0\sigma$ is false under Γ, and $L\sigma$ is undefined in Γ

The rule Propagate applies exhaustive factoring to the propagated literal with respect to the grounding substitution σ and annotates the factored clause to the propagation literal on the trail.

Decide $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\Gamma, \text{D}{:}L\sigma^{k+1}; N; U; \beta; N'; k+1; \top)$
provided $L \in C$ for a $C \in (N \cup U)$, $L\sigma$ is a ground literal undefined in Γ, and $L\sigma \prec_B \beta$

Exclude $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\Gamma, \text{E}{:}L\sigma^{k+1:(C_0 \vee L)\delta\cdot\sigma}; N; U; \beta; N'; k+1; \top)$
provided $C \vee L \in N'$, $C = C_0 \vee C_1$, $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, $C_0\sigma$ does not contain $L\sigma$, δ is the mgu of the literals in C_1 and L, $C_0\sigma$ is false under Γ, $(C \vee L)\sigma$ is ground, $(C \vee L)\sigma \prec_B \{\beta\}$, and $L\sigma$ is undefined in Γ

The rule Exclude works like Propagate, except it uses learned information from N' instead of $N \cup U$. Thus, the inferred literal is not necessarily entailed by $\Gamma \cup N$, but it prevents the generation of an already visited partial model (*stuck* state, see Definition 2 below). However, in combination with N', the literal must always be entailed by the clause set and the trail, i.e. $\Gamma \cup N \cup N' \models L\sigma$. Hence, when only considering $N \cup U$, the excluded literal will be treated as a *decision*, but when considering $N \cup U \cup N'$, the excluded literal can be treated as *propagated*. This difference will be respected in all rules below. Most rules, in their basic form, are left essentially unmodified, but have a dual version added that treats excluded literals and information from N' accordingly.

ConflictR $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\Gamma; N; U; \beta; N'; k; (D \cdot \sigma)_R)$
provided $D \in (N \cup U)$, $D\sigma$ false in Γ

ConflictE $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\Gamma; N; U; \beta; N'; k; (D \cdot \sigma)_{\text{E}})$
provided $D \in N'$, $D\sigma$ false in Γ

The classical rules construct a (partial) model via the trail Γ for $N \cup U$ until a conflict, i.e., a false clause with respect to Γ is found. In HSCL, we also allow a conflict to a clause in N', meaning that all partial models that could be built under this trail have already been discovered. Thus, ConflictE signals the rewrite system that all further attempts with trail Γ should be excluded from future searches.

Clearly, these two kinds of conflicts need to be separated. A conflict is annotated $(D \cdot \sigma)_R$ if it was a regular conflict to $N \cup U$, i.e., learning clauses from $N \cup U$ and searching for a refutation. In contrast, $(D \cdot \sigma)_{\text{E}}$ denotes a conflict to a clause from N', i.e., the current state was already visited, resulted in a partial model and can therefore be excluded.

If a conflict is found, it is resolved by the conflict resolution rules below. Before any conflict resolution step, we assume that the respective clauses are renamed such that they do not share any variables and that the grounding substitutions of closures are adjusted accordingly.

Skip $(\Gamma, X{:}L; N; U; \beta; N'; k; (D \cdot \sigma)_u) \Rightarrow_{\text{HSCL}} (\Gamma; N; U; \beta; N'; k - i; (D \cdot \sigma)_u)$
provided $\text{comp}(L)$ does not occur in $D\sigma$, and if $X \in \{D, E\}$, i.e. L is a decision or exclusion literal, then $i = 1$, else $i = 0$

Factorize $(\Gamma; N; U; \beta; N'; k; ((D \lor L \lor L') \cdot \sigma)_u) \Rightarrow_{\text{HSCL}} (\Gamma; N; U; \beta; N'; k; ((D \lor L)\eta \cdot \sigma)_u)$
provided $L\sigma = L'\sigma$, $\eta = \text{mgu}(L, L')$

ResolveR $(\Gamma, \text{P}{:}L\delta^{(C \lor L) \cdot \delta}; N; U; \beta; N'; k; ((D \lor L') \cdot \sigma)_R) \Rightarrow_{\text{HSCL}}$
$(\Gamma, \text{P}{:}L\delta^{(C \lor L) \cdot \delta}; N; U; \beta; N'; k; ((D \lor C)\eta \cdot \sigma\delta)_R)$
provided $L\delta = \text{comp}(L'\sigma)$, $\eta = \text{mgu}(L, \text{comp}(L'))$

Note that ResolveR strongly resembles the original SCL resolve. In particular, it does not remove the literal $L\delta$ from the trail. This is needed if the clause $D\sigma$ contains further literals complementary of $L\delta$ that have not been factorized.

ResolveE $(\Gamma, X{:}L\delta^{(C \lor L) \cdot \delta}; N; U; \beta; N'; k; ((D \lor L') \cdot \sigma)_{\text{E}}) \Rightarrow_{\text{HSCL}}$
$(\Gamma, X{:}L\delta^{(C \lor L) \cdot \delta}; N; U; \beta; N'; k; ((D \lor C)\eta \cdot \sigma\delta)_{\text{E}})$
provided $X \in \{\text{E}, \text{P}\}$, $L\delta = \text{comp}(L'\sigma)$, $\eta = \text{mgu}(L, \text{comp}(L'))$

ResolveE takes all information from $N \cup U \cup N'$. Since excluded literals act like propagations with respect to N', they can be resolved with during applications of ResolveE. In contrast, this is not possible in ResolveR, and excluded literals must be treated like decisions while resolving regular conflicts.

BacktrackR $(\Gamma_0, K, \Gamma_1, X : \text{comp}(L\sigma)^k; N; U; \beta; N'; k; ((D \vee L) \cdot \sigma)_R) \Rightarrow_{\text{HSCL}} (\Gamma_0; N; U \cup \{D \vee L\}; \beta; N'; j - i; \top)$
provided $X \in \{\text{E}, \text{D}\}$ and $D\sigma$ is of level $i' < k$, and Γ_0, K is the minimal trail subsequence such that there is a grounding substitution τ with $(D \vee L)\tau$ is false in Γ_0, K but not in Γ_0, the literal K is of level j, if K is a decision or an exclusion literal then $i = 1$, otherwise $i = 0$

BacktrackE $(\Gamma_0, K, \Gamma_1, D : \text{comp}(L\sigma)^k; N; U; \beta; N'; k; ((D \vee L) \cdot \sigma)_{\text{E}})$
$\Rightarrow_{\text{HSCL}} (\Gamma_0; N; U; \beta; N' \cup \{D \vee L\}; j - i; \top)$
provided $D\sigma$ is of level $i' < k$, and Γ_0, K is the minimal trail subsequence such that there is a grounding substitution τ with $(D \vee L)\tau$ is false in Γ_0, K but not in Γ_0, the literal K is of level j, if K is a decision or an exclusion literal then $i = 1$, otherwise $i = 0$

Please note that the corner case $j + 1 = k$ and $\tau = \sigma$ is also part of both backtrack rules. The rules backtrack to the minimal trail where the clause $D \vee L$ propagates. Also, note that the existence of the literal K is guaranteed if $(D \vee L)\sigma \neq \bot$ and all other preconditions of Backtrack are met. Then, $(D \vee L)\sigma$ is false under $\Gamma_0, K, \Gamma_1, \text{comp}(L\sigma)$ by soundness (see Definition 3). However, $(D \vee L)\sigma$ must be undefined and hence not false under the empty trail. Thus, there must be an intermediate literal K on the trail where the demanded property holds.

While BacktrackR learns to the clause set U, BacktrackE learns clauses to N'. Here, BacktrackR can also jump back to excluded literals since they are treated like decisions w.r.t. $N \cup U$. In contrast, excluded literals are propagations for conflicts with $N \cup U \cup N'$ and, thus, cannot be backtracked to. The clause $D \vee L$ added by the rule BacktrackR to U is called a *learned clause*. Similarly, clauses added by backtrack to N' are called *excluded clauses*.

Definition 2 (Stuck State)**.** *A state $(M; N; U; \beta; N'; k; D)$ is called* stuck *if $D \neq (\bot \cdot \sigma)_u$ and none of the above rules are applicable.*

Unstuck $(\Gamma; N; U; \beta; N'; k; \top) \Rightarrow_{\text{HSCL}} (\epsilon; N; U; \beta; N' \cup \{C\}; 0; \top)$
provided $(\Gamma; N; U; \beta; N'; k; \top)$ is a stuck state and $C = \bigvee_{L_i \in \text{decision}(\Gamma)} \text{comp}(L_i)$

In this rule, the function decision(Γ) collects all decided literals D:L_k that have been introduced by applications of the rule Decide. If no such literal exists in Γ, then $C = \bot$. This rule could be further refined by considering the effect of decisions on satisfying clauses from $\text{gnd}^{\prec_B \beta}(N)$.

Finish $(\Gamma; N; U; \beta; N'; k; (\bot)_{\text{E}}) \Rightarrow_{\text{HSCL}}$

The last two rules allow us to explore stuck states. Whenever a stuck state is found, the trail is reset, and this particular stuck state will be prevented from being explored again by adding the clause consisting of the complement of all decisions to the exclusion set N'. Notably, stuck states directly correspond to partial models with literals $\prec_B \beta$, see Definition 15. Thus, exploring stuck states is a way to get insights on the literal structure of partial models. In the following, we will construct regularity rules such that a complete run will eventually explore all stuck states. This run also enumerates all partial models w.r.t. $\prec_B$ and β, as shown in Theorem 16.

A first simple example showing the application of the calculus rules is as follows. Consider $N = \{C_1 = P(x) \vee Q(x), C_2 = \neg P(x) \vee \neg Q(x), C_3 = P(x) \vee \neg Q(x)\}$. Let σ denote the substitution $\{x \mapsto a\}$. Choose β and $\prec_B$ in a way that only $\{P(a), \neg P(a), Q(a), \neg Q(a)\} \prec_B \{\beta\}$. Then, the following HSCL run explores all models for N:

$$
\begin{array}{ll}
 & (\varepsilon; N; \emptyset; \beta; \emptyset; 0; \top) \\
\Rightarrow^{\text{Decide}}_{\text{HSCL}} & (\text{D}{:}P(a)^1; N; \emptyset; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{D}{:}P(a)^1, \text{P}{:}\neg Q(a)^{C_2\cdot\sigma}; N; \emptyset; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{Unstuck}}_{\text{HSCL}} & (\varepsilon; N; \emptyset; \beta; N' = \{\neg P(a)\}; 0; \top) \\
\Rightarrow^{\text{Exclude}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}; N; \emptyset; \beta; N'; 1; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}, \text{P}{:}Q(a)^{C_1\cdot\sigma}; N; \emptyset; \beta; N'; 1; \top) \\
\Rightarrow^{\text{ConflictR}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}, \text{P}{:}Q(a)^{C_1\cdot\sigma}; N; \emptyset; \beta; N'; 1; (C_3\cdot\sigma)_R) \\
\Rightarrow^{\text{ResolveR}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}, \text{P}{:}Q(a)^{C_1\cdot\sigma}; N; \emptyset; \beta; N'; 1; (P(x) \vee P(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{Factorize}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}, \text{P}{:}Q(a)^{C_1\cdot\sigma}; N; \emptyset; \beta; N'; 1; (P(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{Skip}}_{\text{HSCL}} & (\text{E}{:}\neg P(a)^{1:\neg P(a)}; N; \emptyset; \beta; N'; 1; (P(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{BacktrackR}}_{\text{HSCL}} & (\varepsilon; N; \{P(x)\}; \beta; N'; 0; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P}{:}P(a)^{P(x)\cdot\sigma}; N; \{P(x)\}; \beta; N'; 0; \top) \\
\Rightarrow^{\text{ConflictE}}_{\text{HSCL}} & (\text{P}{:}P(a)^{P(x)\cdot\sigma}; N; \{P(x)\}; \beta; N'; 0; (\neg P(a))_{\text{E}}) \\
\Rightarrow^{\text{ResolveE}}_{\text{HSCL}} & (\varepsilon; N; \{P(x)\}; \beta; N'; 0; (\bot)_{\text{E}}) \\
\Rightarrow^{\text{Finish}}_{\text{HSCL}} &
\end{array}
$$

In this example, there is only one partial model $\{P(a), \neg Q(a)\}$. Hence, Unstuck is only applied once in the overall HSCL run. A more complex example will be presented at the end of this section.

Definition 3 (Sound States)**.** *A state* $(\Gamma; N; U; \beta; N'; k; D)$ *is* sound *if the following conditions hold:*

1. Γ *is a consistent sequence of annotated ground literals,*
2. *for each decomposition* $\Gamma = \Gamma_1, P{:}L\sigma^{C\vee L\cdot\sigma}, \Gamma_2$ *we have that* $C\sigma$ *is false under* Γ_1 *and* $L\sigma$ *is undefined under* Γ_1*,* $N \cup U \models C \vee L$

3. *for each decomposition* $\Gamma = \Gamma_1, E{:}L\sigma^{k:C\vee L\cdot\sigma}, \Gamma_2$ *we have that* $C\sigma$ *is false under* Γ_1 *and* $L\sigma$ *is undefined under* Γ_1*,* $N \cup N' \models C \vee L$
4. *for each decomposition* $\Gamma = \Gamma_1, X{:}L, \Gamma_2$ *we have that* L *is undefined in* Γ_1*,*
5. $N \models U$*,*
6. *if* $D = (C \cdot \sigma)_R$ *then* $C\sigma$ *is false under* Γ *and* $N \models C$*. In particular,* $\mathrm{gnd}^{\prec_B \beta}(N) \models C\sigma$*.*
7. *if* $D = (C \cdot \sigma)_E$ *then* $C\sigma$ *is false under* Γ *and* $N \cup N' \models C$*. In particular,* $\mathrm{gnd}^{\prec_B \beta}(N \cup N') \models C\sigma$*.*
8. *for any* $L\sigma \in \Gamma$ *we have* $L\sigma \prec_B \beta$ *and there is a* $C \in (N \cup U)$ *such that* $L \in C$*.*

Lemma 4 (Soundness of the initial state)**.** *The initial state* $(\epsilon; N; \emptyset; \beta; \emptyset; 0; \top)$ *is sound.*

Proof. Criteria 1–4 and 8 are trivially satisfied by $\Gamma = \epsilon$. Furthermore, $N \models \emptyset$, fulfilling criterion 5. Lastly, criteria 6 and 7 are trivially fulfilled for $D = \top$. □

Theorem 5 (Soundness of HSCL)**.** *All HSCL rules preserve sound states.*

Proof. Assume a state $(\Gamma; N; U; \beta; N'; k; D)$ is sound. We show that any application of a rule results again in a sound state. For the conflict, resolve and backtrack rules we only show the extended versions, the original versions are similar.

$\Rightarrow^{\text{Decide}}_{\text{HSCL}}$. Assume Decide is applicable to $(\Gamma; N; U; \beta; N'; k; D)$, yielding a resulting state $(\Gamma, \mathrm{D}{:}L\sigma^{k+1}; N; U; \beta; N'; k+1; D)$. Then there is a $L \in C$ for $C \in N \cup U$, $L\sigma$ is ground and undefined in Γ, and $L\sigma \prec_B \beta$. Also, there can be no active conflict, i.e. $D = \top$.

- 1, 4 By the precondition, $L\sigma$ is undefined in Γ (4). Hence, adding $\mathrm{D}{:}L\sigma$ does not make Γ inconsistent (1).
- 2, 3, 5 Trivially fulfilled by hypothesis.
- 6, 7 Since $D = \top$, the rules are trivially satisfied.
- 8 For all literals $L'\sigma' \in \Gamma$, this holds by hypothesis. For $L\sigma$ this follows directly from the preconditions of the rule.

$\Rightarrow^{\text{Propagate}}_{\text{HSCL}}$. Assume Propagate is applicable to $(\Gamma; N; U; \beta; N'; k; D)$, yielding a resulting state $(\Gamma, \mathrm{P}{:}L\sigma^{(C_0 \vee L)\delta\cdot\sigma}; N; U; \beta; N'; k; D)$. Then, there is a $C \vee L \in (N \cup U)$ such that $C = C_0 \vee C_1$, $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, $C_0\sigma$ does not contain $L\sigma$, δ is the mgu of the literals in C_1 and L, $(C \vee L)\sigma$ is ground, $(C \vee L)\sigma \prec_B \{\beta\}$, $C_0\sigma$ is false under Γ, and $L\sigma$ is undefined in Γ Also, there can be no active conflict, i.e. $D = \top$.

- 1, 4 By the precondition, $L\sigma$ is undefined in Γ (4). Hence, adding $\mathrm{P}{:}L\sigma$ does not make Γ inconsistent (1).
- 2 Consider any decomposition $\Gamma, \mathrm{P}{:}L\sigma^{(C_0 \vee L)\delta\cdot\sigma} = \Gamma_1, \mathrm{P}{:}L'\sigma'^{C_0' \vee L'\cdot\sigma'}, \Gamma_2$. In the case of $L'\sigma \neq L\sigma$, we can apply the hypothesis for the state $(\Gamma; N; U; \beta; N'; k; D)$. Hence, only the case $\Gamma_1 = \Gamma$, $L'\sigma' = L\sigma$, and $C_0'\sigma = C_0\sigma$ is left to prove.
  First, note that $C_0\sigma$ is false under $\Gamma_1 = \Gamma$ by the preconditions. Also, $L\sigma$ must be undefined in Γ by the preconditions. Lastly, it needs to be shown that $N \cup U \models (C_0 \vee L)\delta$. Clearly, since $C \vee L \in (N \cup U)$, it holds that $N \cup U \models C \vee L$. Since $C = C_0 \vee C_1$ and $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$ it follows from the soundness of Factorization that $C \models (C_0 \vee L)$ and by this $N \cup U \models C_0 \vee L$.

3, 5 Follows trivially from the induction hypothesis.

6, 7 Since $D = \top$, the rules are trivially satisfied.

8 For all literals $L'\sigma' \in \Gamma$, this holds by hypothesis. For $L\sigma$, consider the precondition that $(C \vee L)\sigma \prec_B \{\beta\}$. By the definition of the multiset extension of $\prec_B$, it follows that $L\sigma \prec_B \beta$ must hold as well.

$\Rightarrow^{\text{Exclude}}_{\text{HSCL}}$. Assume Exclude is applicable to $(\Gamma; N; U; \beta; N'; k; D)$, yielding a resulting state $(\Gamma, \text{E}{:}L\sigma^{k+1:(C_0 \vee L)\delta\cdot\sigma}; N; U; \beta; N'; k+1; \top)$.

Then, there is a $C \vee L \in N'$, $C = C_0 \vee C_1$, $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, $C_0\sigma$ does not contain $L\sigma$, δ is the mgu of the literals in C_1 and L, $C_0\sigma$ is false under Γ, $(C \vee L)\sigma$ is ground, $(C \vee L)\sigma \prec_B \{\beta\}$, and $L\sigma$ is undefined in Γ. Also, there can be no active conflict, i.e. $D = \top$.

1, 4 By the precondition, $L\sigma$ is undefined in Γ (4). Hence, adding E:$L\sigma$ does not make Γ inconsistent (1).

2, 5 Follows trivially from the induction hypothesis.

3 Consider any decomposition $\Gamma, \text{E}{:}L\sigma^{k+1:(C_0 \vee L)\delta\cdot\sigma} = \Gamma_1, \text{E}{:}L'\sigma'^{k':C_0' \vee L'\cdot\sigma'}, \Gamma_2$. In the case of $L'\sigma \neq L\sigma$, we can apply the hypothesis for the state $(\Gamma; N; U; \beta; N'; k; D)$. Hence, only the case $\Gamma_1 = \Gamma$, $L'\sigma' = L\sigma$, and $C_0'\sigma = C_0\sigma$ is left to prove.
First, note that $C_0\sigma$ is false under $\Gamma_1 = \Gamma$ by the preconditions. Also, $L\sigma$ must be undefined in Γ by the preconditions. Lastly, it needs to be shown that $N \cup N' \models (C_0 \vee L)\sigma$. Clearly, since $C \vee L \in N'$, it holds that $N \cup N' \models C \vee L$. $(C_0 \vee C_1 \vee L)\sigma$ is an instance of $C \vee L$. By the preconditions of Propagate, $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$. Hence, $C \models (C_0 \vee L)\sigma$ and by this $N \cup N' \models (C_0 \vee L)\sigma$.

6, 7 Since $D = \top$, the rules are trivially satisfied.

8 For all literals $L'\sigma' \in \Gamma$, this holds by hypothesis. For $L\sigma$, consider the precondition that $(C \vee L)\sigma \prec_B \{\beta\}$. By the definition of the multiset extension of $\prec_B$, it follows that $L\sigma \prec_B \beta$ must hold as well.

$\Rightarrow^{\text{ConflictE}}_{\text{HSCL}}$. Assume ConflictE is applicable to $(\Gamma; N; U; \beta; N'; k; D)$, yielding a resulting state $(\Gamma; N; U; \beta; N'; k; (C \cdot \sigma)_{\text{E}})$. Then, there is a $C \in N'$ such that $C\sigma$ false in Γ.

1-4, 8 Trivially fulfilled by hypothesis, as the trail Γ is not modified.

5 Follows trivially from the induction hypothesis, as U is not modified.

6 Since $D = (C \cdot \sigma)_{\text{E}}$, this rules is trivially satisfied.

7 It holds that $D = (C \cdot \sigma)_{\text{E}}$. By the preconditions of ConflictE, $C\sigma$ must be false under Γ. Furthermore, since $C \in N'$ it holds that $N' \models C$. Hence, clearly it is also the case that $N \cup N' \models C$. Lastly, it remains to show that $\text{gnd}^{\prec_B \beta}(N \cup N') \models C\sigma$. By soundness (8), we know that for all literals $L\mu \in \Gamma$ it holds that $L\mu \prec_B \beta$. Since $C\sigma$ is false in Γ, it must hold that all literals in $C\sigma$ are also $\prec_B \beta$. Combined with $N \cup N' \models C$, this yields that $\text{gnd}^{\prec_B \beta}(N \cup N') \models C\sigma$.

$\Rightarrow^{\text{Skip}}_{\text{HSCL}}$. Assume Skip is applicable to $(\Gamma = \Gamma', X{:}L; N; U; \beta; N'; k; (D \cdot \sigma)_u)$, yielding a resulting state $(\Gamma'; N; U; \beta; N'; k - i; (D \cdot \sigma)_u)$. By the preconditions of skip, it must hold that comp(L) does not occur in $D\sigma$, and if $X \in \{D, E\}$, i.e. L is a decision or exclusion literal, then $i = 1$, else $i = 0$

1-4, 8 Directly fulfilled by hypothesis, as all prefixes of Γ still fulfil all properties. In particular, this holds for the prefix Γ' of Γ.

5 Follows trivially from the induction hypothesis, as U is not modified.

6, 7 After the application of Skip, $(D \cdot \sigma)_u$ is the current conflict. Since D is not modified, $N \models D$ (resp. $N \cup N' \models D$) and $\text{gnd}^{\prec_B \beta}(N) \models D\sigma$ (resp. $\text{gnd}^{\prec_B \beta}(N \cup N') \models D\sigma$) still hold by hypothesis. It is left to shot that $D\sigma$ is false under the resulting Γ' under the assumption that $D\sigma$ is false under Γ. However, since $\text{comp}(L) \notin D\sigma$, this is trivially fulfilled, as the removal of $\text{comp}(L)$ from the trail Γ cannot make $D\sigma$ undefined. Hence, $D\sigma$ must be false under Γ' as well.

$\Rightarrow^{\text{Factorize}}_{\text{HSCL}}$. Assume Factorize is applicable to $(\Gamma; N; U; \beta; N'; k; ((D \vee L \vee L') \cdot \sigma)_u)$, yielding a resulting state $(\Gamma; N; U; \beta; N'; k; ((D \vee L)\eta \cdot \sigma)_u)$. Then, $L\sigma = L'\sigma$ and $\eta = \text{mgu}(L, L')$.

1-4, 8 Trivially fulfilled by hypothesis, as the trail Γ is not modified.

5 Follows trivially from the induction hypothesis, as U is not modified.

6, 7 After the application of Factorize, $((D \vee L)\eta \cdot \sigma)_u$ is the current conflict. W.l.o.g. assume we are in the $((D \vee L)\eta \cdot \sigma)_R$ case, i.e. the factorized clause is a regular conflict. By the hypothesis $N \models (D \vee L \vee L')$. From the preconditions of Factorize, $L\sigma = L'\sigma$ and $\eta = \text{mgu}(L, L')$. Thus, $(D \vee L \vee L')\eta$ is an instance of $(D \vee L \vee L')$ and $N \models (D \vee L \vee L')\eta$. Since $L\eta = L'\eta$, $(D \vee L \vee L')\eta \models (D \vee L')\eta$. Thus, $N \models (D \vee L)\eta$. By the preconditions, $\text{gnd}^{\prec_B \beta}(N) \models \text{gnd}^{\prec_B \beta}((L \vee L \vee L')\sigma)$. Hence, $(D \vee L \vee L')\sigma \prec_B \{\beta\}$. Thus, $(D \vee L)\eta\sigma = (D \vee L)\sigma \prec_B \{\beta\}$. From this, it follows that $\text{gnd}^{\prec_B \beta}(N) \models \text{gnd}^{\prec_B \beta}((D \vee L)\sigma)$.
Furthermore, $(D \vee L)\eta\sigma$ is false under Γ, since $(D \vee L)\eta\sigma = (D \vee L)\sigma$ by the definition of an mgu, and $(D \vee L \vee L')\sigma$ is already false under Γ.

$\Rightarrow^{\text{ResolveE}}_{\text{HSCL}}$. Assume ResolveE is applicable to $(\Gamma', X{:}L\delta^{(C \vee L) \cdot \delta}; N; U; \beta; N'; k; ((D \vee L') \cdot \sigma)_{\text{E}})$ yielding a resulting state $(\Gamma', X{:}L\delta^{(C \vee L) \cdot \delta}; N; U; \beta; N'; k; ((D \vee C)\eta \cdot \sigma\delta)_{\text{E}})$

By the preconditions of ResolveE, it holds that $X \in \{\text{E}, \text{P}\}$.

1-4, 8 Trivially fulfilled by hypothesis, as the trail Γ is not modified.

5 Follows trivially from the induction hypothesis, as U is not modified.

6 Since $D = (C \cdot \sigma)_{\text{E}}$, this rule is trivially satisfied.

7 After the application of ResolveE, $((D \vee C)\eta \cdot \sigma\delta)_R$ is the current conflict.
By the hypothesis, $(D \vee L')\sigma$ is false under Γ. In particular, $D\sigma$ is false under Γ. By soundness (2), we know that $C\delta$ must be false under Γ as well. Hence, $(D \vee L)\eta\sigma\delta$ is false under Γ.
By the hypothesis, $N \cup N' \models (D \vee L')$. Since $(D \vee L')\eta$ is an instance of $(D \vee L')$, it holds that $N \cup N' \models (D \vee L')\eta$. Furthermore, by soundness (3) we know that $N \cup N' \models (C \vee L)$ By instantiation with η, it holds that $N \cup N' \models (C \vee L)\eta$. By the soundness of resolution, this implies $N \cup N' \models (D \vee C)\eta$.
Lastly, since $(D \vee L')\sigma$ is false in Γ, all occuring literals in $\{(D \vee L')\sigma\} \prec_B \{\beta\}$. With similar argumentation, $\{(C \vee L)\delta\} \prec_B \{\beta\}$. Hence, in particular, $(D \vee C)\eta\sigma\delta \prec_B \{\beta\}$ and, thus, $\text{gnd}^{\prec_B \beta}(N \cup N') \models \text{gnd}^{\prec_B \beta}((D \vee C)\eta\sigma\delta)$.

$\Rightarrow_{\text{HSCL}}^{\text{BacktrackE}}$. Assume BacktrackE is applicable to $(\Gamma = \Gamma', \Gamma''; N; U; \beta; N'; k; ((D \vee L) \cdot \sigma)_{\text{E}})$, yielding a resulting state $(\Gamma'; N; U; \beta; N' \cup \{D \vee L\}; k'; \top)$.

1-4, 8 Directly fulfilled by hypothesis, as all prefixes of Γ still fulfil all properties. In particular, this holds for the prefix Γ' of Γ.

5 Follows trivially from the hypothesis, as neither N nor U are modified.

6, 7 Since after an application of BacktrackR the conflict is resolved, i.e. $D = \top$, the rules are trivially satisfied.

$\Rightarrow_{\text{HSCL}}^{\text{Unstuck}}$. Assume Unstuck is applicable to $(\Gamma; N; U; \beta; N'; k; \top)$, yielding a resulting state $(\epsilon; N; U; \beta; N' \cup \{C\}; 0; \top)$.

1-4, 8 For the empty trail $\Gamma = \epsilon$, all properties follow directly as in Lemma 4.

5 Follows trivially from the hypothesis, as neither N nor U are modified.

6, 7 Since $D = \top$ the rules are trivially satisfied.

(Idea) By induction and case analysis for the different rules.

The proofs for the classical rules (Propagate, Decide, ConflictR, Skip, Factorize, ResolveR, and BacktrackR) are similar to classical SCL, as shown in [11, 1, 2]. The new trail building rule Exclude preserves soundness: Only criteria 3, 4 and 8 are relevant, which are both directly fulfilled by the preconditions of Exclude.

The ConflictE rule detects a clause $D = (C \cdot \sigma)_{\text{E}}$ only if $C \in N'$. Hence, $N' \models C$ (7) by definition. Since $C\sigma$ is false in Γ, it must consist only of literals $\prec_B \beta$. Hence, $\text{gnd}^{\prec_B \beta}(N \cup N') \models C\sigma$. The rules ResolveE, Skip, BacktrackE preserve soundness by similar argumentation as classical SCL, only with respect to a clause set $N \cup N'$. □

Corollary 6. *Assume a state* $(\Gamma; N; U; \beta; N'; k; D)$ *resulting from a run. Then,* $(\Gamma; N; U; \beta; N'; k; D)$ *is sound.*

Proof. Follows with induction over the size of the run. The base case is handled by Lemma 4, the induction step is contained in Theorem 5. □

Definition 7 (Reasonable Runs). *A sequence of HSCL rule applications is called a* reasonable run *if the rule Decide does not enable an immediate application of rule ConflictR or ConflictE.*

Definition 8 (Regular HSCL Runs). *A sequence of HSCL rule applications is called a* regular run *if it is a reasonable run and the following hierarchy of rule preferences is respected:*

- *ConflictR, ConflictE and Finish always have priority over every other rule.*
- *Propagate always has priority over Exclude.*
- *Exclude always has priority over Decide.*

Lemma 9 (Correct Termination without Unstuck and Finish). *If in a regular run no rule except Unstuck or Finish is applicable to a state* $(\Gamma; N; U; \beta; N'; k; D)$*, then either* $D = (\bot)_R$*, or* $D = (\bot)_E$*, or* $D = \top$ *and* $\Gamma \models \text{gnd}^{\prec_B \beta}(N)$.

Proof. Consider a state $(\Gamma; N; U; \beta; N'; k; D)$ where $D \notin \{(\bot)_R, (\bot)_\mathrm{E}\}$.

Then, D can have one of the following shapes

(Case $D = (C\sigma)_R$) then one of the rules ResolveR, Skip, Factorize or BacktrackR is applicable. First, consider the case of $\Gamma = \varepsilon$. By soundness, $C\sigma$ must be false under Γ. However, the only false clause under ε is $\bot$, a contradiction to $D \notin \{(\bot)_R, (\bot)_\mathrm{E}\}$. Thus, there is at least one literal on the trail. We split $\Gamma = \Gamma', X{:}L$ and distinguish on the source X of L:
If $X = P$, i.e. the top level literal is a propagated literal, then either ResolveR or Skip are applicable. In the case that $\mathrm{comp}(L)$ occurs in $C\sigma$, ResolveR is applicable. If $\mathrm{comp}(L) \notin C\sigma$, Skip is applicable.
For $X \in \{E, D\}$, i.e. the top level literal is a decision or exclusion literal, one of the rules Skip, BacktrackR, or Factorize is applicable. If $\mathrm{comp}(L)$ does not occur in $C\sigma$, then Skip can be applied. BacktrackR can be applied in all other cases if $C = (C' \vee \mathrm{comp}(L))$, where C' is of level $i' < k$. Note that for BacktrackR there must be a level j that is backtracked to. This level j always exists if all other preconditions are met. Hence, if Skip is not applicable, C is of the shape $C' \vee \mathrm{comp}(L)$. If C' is of level k, then Factorize can be applied instead, as C' must contain another instance of $\mathrm{comp}(L)$. Otherwise, C' is of level $i' < k$ and BacktrackR can be applied.

(Case $D = (C\sigma)_E$), then one of ResolveE, Skip, Factorize, or BacktrackE is applicable. This follows similarly to the previous case.

(Case $D = \top$) i.e. there is no conflict. Assume there are no undefined ground literals $L \prec_B \beta$ for $L \in C$, $C \in N \cup U$ in Γ. Now, either $\Gamma \models \mathrm{gnd}^{\prec_B \beta}(N)$ and thus Γ is already a partial model for N w.r.t. $\prec_B$ and β. Otherwise, if $\Gamma \not\models \mathrm{gnd}^{\prec_B \beta}(N)$ but all literals are defined, there must be a false clause $C \in \mathrm{gnd}^{\prec_B \beta}(N)$ which can be chosen as a ConflictR instance.
If there is at least one undefined ground literal $L \prec_B \beta$ occuring in $N \cup U$, one of the trail building rules Propagate, Decide, Exclude, ConflictR or ConflictE are applicable. Decide on the undefined ground literal L is always possible, as we only consider literals $L \in C$ for a $C \in (N \cup U)$. The application of Decide can, however, be restricted by reasonability or regularity.
If Decide on L is not applicable by reasonability, then $\Gamma, \mathrm{D}{:}L$ must lead to a direct application of ConflictR or ConflictE. Thus, there is a clause $D \in N \cup U \cup N'$ such that $D\sigma$ is false under $\Gamma, \mathrm{D}{:}L$. If $D\sigma$ is already false under Γ, then either ConflictR or ConflictE are applicable, depending on if $D \in N \cup U$ or $D \in N'$. Otherwise, D has the shape $D_0 \vee D_1$ where D_0 is false under Γ, and $D_1\sigma = \mathrm{comp}(L) \vee \cdots \vee \mathrm{comp}(L)$. Since D_0 is false under Γ, also $D_0 \prec_B \{\beta\}$ and since $L \prec_B \beta$ it holds that $D_0 \vee D_1 \prec_B \{\beta\}$. Hence, depending on if $D \in N \cup U$ or $D \in N'$, either Propagate or Exclude can be applied.
If Decide is not applicable by regularity, another rule of Propagate, Exclude, ConflictR or ConflictE must directly be applicable, since regularity only priorizes rule applications. □

Corollary 10 (Correct Termination of HSCL)**.** *If in a run no rules are applicable to a state $(\Gamma; N; U; \beta; N'; k; D)$, then either $D = (\bot \cdot \sigma)_R$ and N is unsatisfiable, or the run has ended with $\Rightarrow^{Finish}_{HSCL}$.*

Proof. We instantiate Lemma 9. If $D = (\bot \cdot \sigma)_R$, we are done. Similarly, if $D = (\bot \cdot \sigma)_{\mathrm{E}}$ then Finish can be applied. Otherwise, if $D = \top$, Unstuck can be applied to our state by definition. □

Lemma 11 (Regular Conflict Resolution in HSCL)**.** *Consider a HSCL conflict state* $(\Gamma, X{:}L; N; U;\ \beta; N'; k; (D)_u)$. *In a regular run, during conflict resolution, at least the rightmost literal* L *is resolved with.*

Proof. To prove the above claim, we distinguish the two cases how a conflict can be detected in HSCL. For the resulting conflict state, only the six conflict resolution rules Skip, Factorize, ResolveR, ResolveE, BacktrackR and BacktrackE can be applicable. To prove the claim of a resolution happening, we show that only Factorize and Resolve can be applied in a regular run to the resulting conflict state. Since Factorize does not remove literals from the trail and, hence, does not enable the application of any other rule, this shows that a Resolve step must happen at least once before any further conflict resolution rules are applied.

The conflict D was either detected by the ConflictR rule. Then, it is of shape $(D)_R$. Otherwise, the conflict was detected by the ConflictE rule and is of shape $(D)_{\mathrm{E}}$.

(Case ConflictR) In a reasonable run, if the rule Decide produced the state $(\Gamma, \mathrm{D}{:}L; N; U; \beta; k; \top)$, ConflictR is not immediately applicable. In case Backtrack produced the state $(\Gamma, X{:}L; N; U; \beta; k; \top)$, i.e., there is the sequence of rule applications

$(\Gamma_0', L, K, \Gamma_1, X : \mathrm{comp}(L''\sigma)^{k'}; N; U'; \beta; k'; ((D \vee L'') \cdot \sigma)_R) \Rightarrow_{\mathrm{Backtrack}} (\Gamma, L; N; U' \cup (D \vee L''); \beta; k; \top)$ then by the definition of Backtrack, the newly learned clause $(D \vee L'')$ cannot be false with respect to Γ_0', L. Thus, ConflictR is not applicable to $(D \vee L'')$. Furthermore, if there is a conflict to any other clause from $N \cup U$, by regularity, ConflictR must have been applied earlier in the run. In summary, L must be either a propagated or excluded literal.

Note that it, also, is not possible for L to be an exclusion literal. If $(\Gamma, \mathrm{E}{:}L; N; U; \beta; k; \top)$ is a state where ConflictR is applicable, then there is also an application of the rule Propagate which produces the state $(\Gamma, \mathrm{P}{:}\mathrm{comp}(L); N; U; \beta; k; \top)$. By regularity, Propagate has priority over Exclude and thus, no exclusion can lead to a direct application of ConflictR. Overall, L can neither be a decision nor an exclusion literal.

Then, BacktrackR is not applicable to $(\Gamma, \mathrm{D}{:}L; N; U; \beta; k; (D)_R)$, as it requires L to be a decision or exclusion. Furthermore, L must occur in the conflict clause D. Otherwise, ConflictR could have been applied earlier to $(\Gamma; N; U; \beta; k; \top)$, contradicting regularity. Hence, Skip is not applicable to our state. Overall, only Factorize and Resolve can possibly be applied to our state. Factorize does not modify the trail, and, thus, cannot enable any of the rules Skip or Backtrack. Following from that, at least one application of Resolve must take place in conflict resolution.

(Case ConflictE) This case works similar to the previous case. However, BacktrackE cannot be applied from a state $(\Gamma, \mathrm{E}{:}L; N; U; \beta; k; \top)$. Hence, it is allowed for an Exclusion to lead to an application of ConflictE.

□

Definition 12 (State Induced Ordering)**.** *Let $(L_1, L_2, \ldots, L_n; N; U; N'; \beta; k; D)$ be a sound state of HSCL. The trail induces a total well-founded strict order on the defined literals by*

$$L_1 \prec_\Gamma \mathrm{comp}(L_1) \prec_\Gamma L_2 \prec_\Gamma \mathrm{comp}(L_2) \prec_\Gamma \cdots \prec_\Gamma L_n \prec_\Gamma \mathrm{comp}(L_n).$$

We extend $\prec_\Gamma$ to a strict total order on all literals where all undefined literals are larger than $\mathrm{comp}(L_n)$. We also extend $\prec_\Gamma$ to a strict total order on ground clauses by multiset extension and also on multisets of ground clauses and overload $\prec_\Gamma$ for all these cases. With $\preceq_\Gamma$ we denote the reflexive closure of $\prec_\Gamma$.

Theorem 13 (Non-redundant Learning in HSCL)**.** *Let $(\Gamma; N; U; \beta; N'; k; (C_0 \cdot \sigma_0)_u)$ be the state after an application of ConflictR (resp. ConflictE) in a regular run and let C be the clause learned at the end of the conflict resolution, then C is not redundant with respect to $N \cup U$ (resp. $N \cup N' \cup U$) and $\prec_\Gamma$.*

Proof. Consider the following fragment of a derivation learning a clause implied by N:

$$\begin{array}{lll} \Rightarrow^{\text{ConflictR}}_{\text{HSCL}} & (\Gamma; N; U; N'; \beta; k; (C_0 \cdot \sigma_0)_R) & \\ \Rightarrow^{\{\text{Skip, Fact., Res.}\}^*}_{\text{HSCL}} & (\Gamma'; N; U; N'; \beta; k; (C \cdot \sigma)_R) & \Rightarrow^{\text{BacktrackR}}_{\text{HSCL}} \end{array}$$

By soundness $N \cup U \models C$ and $C\sigma$ is false under both Γ and Γ'. We prove that $C\sigma$ is non-redundant to $N \cup U$ with respect to $\prec_\Gamma$.

Assume there is an $S \subseteq \mathrm{gnd}(N \cup U)^{\preceq_\Gamma C\sigma}$ s.t. $S \models C\sigma$. There must be a clause $D \in S$ false under Γ, since all clauses in S have a defined truth value (as all undefined literals are greater in $\prec_\Gamma$ than all defined literals) and if $\Gamma \models S$ then $\Gamma \models C\sigma$ by transitivity of entailment, a contradiction.

By regularity, Γ must be of the shape $\Gamma = \Gamma'', L\delta^{C \vee L \cdot \delta}$, since no application of Decide can lead to an application of the rule ConflictR. Thus, the last applied rule must have been PropagateR. Furthermore, by Lemma 11, Resolve must have resolved at least the rightmost literal $L\delta$ from Γ. Thus, $L\delta \notin C\sigma$ and $\mathrm{comp}(L\delta) \notin C\sigma$. Since $D \prec_\Gamma C\sigma$, neither $L\delta$ nor $\mathrm{comp}(L\delta)$ may occur in D. However, this is a contradiction, since D is then already false under Γ'' and, thus, must have been chosen as a Conflict instance earlier in a regular run. Overall, there can be no $S \subseteq \mathrm{gnd}(N \cup U)^{\preceq_\Gamma C\sigma}$ with $S \models C\sigma$. Hence, $C\sigma$ is non-redundant to $N \cup U$ with respect to $\prec_\Gamma$.

Similarly, this result can be proven for learned clauses to N'. In contrast, a derivation learning a clause to N' with BacktrackE learns only non-redundant clauses with respect to $N \cup U \cup N'$ and $\prec_\Gamma$.

□

Of course, in a regular run, the ordering of foreground literals on the trail will change, i.e., the ordering of Definition 12 will change as well. Thus the non-redundancy property of Lemma 13 reflects the situation at the time of creation of the learned clause. A non-redundancy property holding for an overall run must be invariant against changes on the ordering. However, the ordering of Definition 12 also entails a fixed subset ordering $\prec_\subseteq$ that is invariant against changes on the overall ordering. This means that our dynamic ordering entails non-redundancy criteria based on subset relations including forward redundancy. From an implementation perspective, this means that learned clauses need not to be tested for forward redundancy. Current resolution or superposition based provers spent a reasonable portion of their time in testing forward redundancy of newly generated clauses. In addition, also tests for backward reduction can be restricted knowing that learned clauses are not redundant.

Theorem 14 (Termination of HSCL). *All regular HSCL runs terminate.*

Proof. In Theorem 9, we proved that all regular runs which do not use Unstuck terminate. Thus, it is left to show that Unstuck cannot be used infinitely often. Note that every regular use of Unstuck on a state $(\Gamma; N; U; \beta; N'; k; D)$ adds a clause C to N'. However, by Theorem 13, C is not redundant to $N \cup U \cup N'$ under $\prec_{\subseteq}$, which is well-founded. Due to the restriction of all learned clauses to be smaller than $\{\beta\}$, the number of non-redundant ground clauses is finite. Thus, Unstuck cannot be applied infinitely often, and all regular HSCL runs must terminate. □

Definition 15 (Stuck States correspond to Partial Models). *Let M be a partial model for N under $\prec_B$ and β, i.e., $M \models \text{gnd}^{\prec_B \beta}(N)$ and all ground literals $L\sigma$ with $L\sigma \prec_B \{\beta\}$ and $L \in C$ for a $C \in N$ are defined in M. We call a stuck state $(\Gamma; N; U; \beta; N'; k; D)$ corresponding to M if $\Gamma \models M$.*

Theorem 16 (Exhaustive Stuck State Exploration). *Consider a HSCL run that ends with the rule $(\Gamma; N; U; \beta; N'; k; (\bot)_E) \Rightarrow^{Finish}_{HSCL}$. For all partial models M of N under $\prec_B$ and β, a stuck state corresponding to M is eventually explored in such a run.*

Proof. Assume there is a partial model M such that no stuck state corresponding to M was visited. For M, by definition $M \models \text{gnd}^{\prec_B \beta}(N)$ and $M \not\models \bot$. Since our run ended with $D = (\bot)_\text{E}$, by soundness of the calculus it follows that $\text{gnd}^{\prec_B \beta}(N \cup N') \models \bot$ in the final state. However, it cannot be the case that $\text{gnd}^{\prec_B \beta}(N) \models \bot$, since otherwise by transitivity $M \models \text{gnd}^{\prec_B \beta}(N) \models \bot$. Since initially, $N' = \emptyset$, there must be a HSCL rule application

$$(\Gamma; N; U; \beta; N'; k; D) \Rightarrow_{\text{HSCL}} (\Gamma; N; U; \beta; N' \cup \{C\}; k; D)$$

such that $M \models \text{gnd}^{\prec_B \beta}(N \cup N')$, but $M \not\models \text{gnd}^{\prec_B \beta}(N \cup N' \cup \{C\})$. This clause C can be added by two rules to N', $\Rightarrow^{\text{BacktrackE}}_{\text{HSCL}}$ or $\Rightarrow^{\text{Unstuck}}_{\text{HSCL}}$:

(Case BacktrackE) If C was added by $\Rightarrow^{\text{BacktrackE}}_{\text{HSCL}}$, then by soundness already $\text{gnd}^{\prec_B \beta}(N \cup N') \models C\sigma$. Thus, if $M \models \text{gnd}^{\prec_B \beta}(N \cup N')$ then also $M \models \text{gnd}^{\prec_B \beta}(N \cup N' \cup \{C\})$, a contradiction. Hence, it cannot be the case that C was added by $\Rightarrow^{\text{BacktrackE}}_{\text{HSCL}}$.

(Case Unstuck By the preconditions of the rule, $(\Gamma; N; U; \beta; N'; k; D)$ must be a stuck state. It remains to show that this state corresponds to M. By Lemma 9, since $D \neq (\bot)_u$, it holds that Γ forms a partial model for N under $\prec_B$ and β. Thus, all ground literals $L\sigma$ occuring in N with $L\sigma \prec_B \beta$ are defined in Γ. Hence, it is only left to prove that $\Gamma \models M$. By induction over the trail size, we show that for each literal $L\sigma \in \Gamma$ it holds that $L\sigma \in M$. For the base case of $\Gamma = \epsilon$, nothing is to do. In the induction step, consider

a trail decomposition $\Gamma' = \Gamma'', X{:}L\sigma$, where Γ' is a prefix of Γ. Then, $X{:}L\sigma$ was added to the trail by one of the following rules:
(Case Unstuck: Decide) Then, $\Gamma' = \Gamma'', \mathrm{D}{:}L\sigma$. By the definition of $C = \bigvee_{L_i \in \mathrm{decision}(\Gamma)} \mathrm{comp}(L_i)$, it holds that $C = \mathrm{comp}(L\sigma) \vee C'$, since $L\sigma$ is a decision in Γ'. Furthermore, all literals in C are already ground and $\prec_B \beta$. Hence, $\mathrm{gnd}^{\prec_B \beta}(\{C\}) = \{\mathrm{comp}(L\sigma) \vee C'\}$. Now, by assumption, $M \not\models \mathrm{gnd}^{\prec_B \beta}(\{C\})$ and, thus, $M \not\models \mathrm{comp}(L\sigma)$. Since M is a partial model that defines all ground literals $\prec_B \beta$, it must hold that $L\sigma \in M$.
(Case Unstuck: Propagate) This implies that the literal has the form $\mathrm{P}{:}L\sigma^{(C_0 \vee L)\delta \cdot \sigma}$. By the preconditions of Propagate, there is a clause $C_0 \vee C_1 \vee L \in (N \cup U)$ where $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, and $C_0\sigma$ is false under Γ''. By our induction hypothesis, we know that all literals defined in Γ'' are consistent with M. Now, assume that $\mathrm{comp}(L\sigma) \in M$. But now, $C_0\sigma$ is false under M, and $C_1\sigma$ is false as well. Hence, the overall clause $(C_0 \vee C_1 \vee L)\sigma \in (N \cup U)$ is falsified under M. If $(C_0 \vee C_1 \vee L)\sigma \in N$, this directly contradicts the assumption that M is a partial model for N. If the clause is a learned clause, i.e. $(C_0 \vee C_1 \vee L)\sigma \in U$, by soundness of the calculus ($N \models U$), this leads to the same contradiction.
(Case Unstuck: Exclude) If the literal has the form $\mathrm{E}{:}L\sigma_k^{(C_0 \vee L)\delta \cdot \sigma}$, with the same argumentation as in Propagate, there must be a clause $C_0 \vee C_1 \vee L \in N'$ where $C_1\sigma = L\sigma \vee \cdots \vee L\sigma$, and $C_0\sigma$ is false under Γ''. If we assume that $\mathrm{comp}(L\sigma) \in M$, this falsifies the clause $(C_0 \vee C_1 \vee L)\sigma$ under M. However, since this clause is a ground clause $\prec_B \beta$, this contradicts the assumption that $M \models \mathrm{gnd}^{\prec_B \beta}(N')$. □

For a final, more complex example, consider

$$N = \left\{ \begin{array}{llll} C_1 & = \neg P(x) \vee Q(x) & C_2 & = \neg Q(x) \vee R(x) \\ C_3 & = \neg R(x) \vee P(x) \vee Q(x) & C_4 & = R(a) \end{array} \right\}$$

and $\beta, \prec_B$ chosen such that exactly $\{P(a), \neg P(a), Q(a), \neg Q(a), R(a), \neg R(a)\} \prec_B \{\beta\}$. Furthermore, let σ denote the substitution $\{x \mapsto a\}$. For example, a regular HSCL could explore all partial models in the following way:

$$
\begin{array}{ll}
 & (\varepsilon; N; \emptyset; \beta; \emptyset; 0; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}; N; \emptyset; \beta; \emptyset; 0; \top) \\
\Rightarrow^{\text{Decide}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1; N; \emptyset; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1, \text{P:}P(a)^{C_3\cdot\sigma}; N; \emptyset; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{ConflictR}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1, \text{P:}P(a)^{C_3\cdot\sigma}; N; \emptyset; \beta; \emptyset; 1; (\neg P(x) \lor Q(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{ResolveR}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1, \text{P:}P(a)^{C_3\cdot\sigma}; N; \emptyset; \beta; \emptyset; 1; (\neg R(x) \lor Q(x) \lor Q(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{Factorize}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1, \text{P:}P(a)^{C_3\cdot\sigma}; N; \emptyset; \beta; \emptyset; 1; (\neg R(x) \lor Q(x)\cdot\sigma)_R) \\
\Rightarrow^{\text{Skip}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}\neg Q(a)^1; N; \emptyset; \beta; \emptyset; 1; (\neg R(x) \lor Q(x)\cdot\sigma))_R \\
\Rightarrow^{\text{BacktrackR}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}; N; U = \{C_5 = \neg R(x) \lor Q(x)\}; \beta; \emptyset; 0; \top) \\
\Rightarrow^{\text{Decide}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}P(a)^1; N; U; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{D:}P(a)^1, \text{P:}Q(a)^{C_5\cdot\sigma}; N; U; \beta; \emptyset; 1; \top) \\
\Rightarrow^{\text{Unstuck}}_{\text{HSCL}} & (\varepsilon; N; U; \beta; N' = \{\neg P(a)\}; 0; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}; N; U; \beta; N'; 0; \top) \\
\Rightarrow^{\text{Propagate}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{P:}Q(a)^{C_5\cdot\sigma}; N; U; \beta; N'; 0; \top) \\
\Rightarrow^{\text{Exclude}}_{\text{HSCL}} & (\text{P:}R(a)^{C_4\cdot\{\}}, \text{P:}Q(a)^{C_5\cdot\sigma}, \text{E:}\neg P(a)^{1:\neg P(a)\cdot\{\}}; N; U; \beta; N'; 1; \top) \\
\Rightarrow^{\text{Unstuck}}_{\text{HSCL}} & (\varepsilon; N; U; \beta; N' \cup \{\bot\}; 0; \top) \\
\Rightarrow^{\text{ConflictE}}_{\text{HSCL}} & (\varepsilon; N; U; \beta; N' \cup \{\bot\}; 0; (\bot)_{\text{E}}) \\
\Rightarrow^{\text{Finish}}_{\text{HSCL}} &
\end{array}
$$

Just before any use of Unstuck, the trail forms a partial model under $\prec_B$ and β. In this example, the partial models are $\{P(a), Q(a), R(a)\}$ and $\{\neg P(a), Q(a), R(a)\}$.

4. Discussion

We have shown that simultaneously searching for a refutation and enumerating all potential models, restricted to the current finite limit, can be effectively combined. There are several directions for future research. First a procedure that takes a stuck state, or the overall content of N' out of a some state $(\Gamma; N; U; \beta; N'; k; D)$ and checks for models for N. This problem is undecidable, in general. However, for example, the following approach will work. Take any model candidate out of N' and fit it into an effective first-order model representation formalism, e.g., see [12]. Then check whether this results in a model for the overall N. Second, any (ground) property that holds in all enumerated models is actually true, in general, and can therefore be added to the current set of clauses. Any property that holds at least in some enumerated models might be a good candidate to be tested for being true in all models. This test can be combined with the overall search for a refutation.

References

[1] A. Fiori, C. Weidenbach, SCL clause learning from simple models, in: Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings, 2019, pp. 233–249. URL: https://doi.org/10.1007/978-3-030-29436-6_14. doi:10.1007/978-3-030-29436-6_14.

[2] M. Bromberger, A. Fiori, C. Weidenbach, Deciding the Bernays-Schoenfinkel fragment over bounded difference constraints by simple clause learning over theories, in: F. Henglein, S. Shoham, Y. Vizel (Eds.), Verification, Model Checking, and Abstract Interpretation - 22nd International Conference, VMCAI 2021, Copenhagen, Denmark, January 17-19, 2021, Proceedings, volume 12597 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 511–533.

[3] H. Leidinger, C. Weidenbach, SCL(EQ): SCL for first-order logic with equality, in: J. Blanchette, L. Kovacs, D. Pattinson (Eds.), Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, Proceedings, LNCS, Springer, 2022. To Appear.

[4] A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability, volume 185 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2009. Second Edition in 2021.

[5] P. Baumgartner, Hyper tableau – the next generation, in: H. C. M. de Swart (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '98, Oisterwijk, The Netherlands, May 5-8, 1998, Proceedings, volume 1397 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 60–76.

[6] J. A. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning (in 2 volumes), Elsevier and MIT Press, 2001.

[7] P. Baumgartner, A. Fuchs, C. Tinelli, Lemma learning in the model evolution calculus, in: LPAR, volume 4246 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 572–586.

[8] R. Piskac, L. M. de Moura, N. Bjørner, Deciding effectively propositional logic using DPLL and substitution sets, Journal of Automated Reasoning 44 (2010) 401–424.

[9] M. P. Bonacina, U. Furbach, V. Sofronie-Stokkermans, On first-order model-based reasoning, in: N. Martí-Oliet, P. C. Ölveczky, C. L. Talcott (Eds.), Logic, Rewriting, and Concurrency - Essays dedicated to José Meseguer on the Occasion of His 65th Birthday, volume 9200 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 181–204.

[10] M. P. Bonacina, D. A. Plaisted, Semantically-guided goal-sensitive reasoning: Model representation, Journal of Automated Reasoning 56 (2016) 113–141.

[11] M. Bromberger, A. Fiori, C. Weidenbach, SCL with theory constraints, 2020. arXiv:2003.04627.

[12] C. G. Fermüller, R. Pichler, Model representation over finite and infinite signatures, J. Log. Comput. 17 (2007) 453–477.