



HAL
open science

SCL(FOL) Can Simulate Non-Redundant Superposition Clause Learning

Martin Bromberger, Chaahat Jain, Christoph Weidenbach

► **To cite this version:**

Martin Bromberger, Chaahat Jain, Christoph Weidenbach. SCL(FOL) Can Simulate Non-Redundant Superposition Clause Learning. Max-Planck-Institut für Informatik, Saarbrücken, Germany. 2023. hal-04315211

HAL Id: hal-04315211

<https://inria.hal.science/hal-04315211v1>

Submitted on 30 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

SCL(FOL) Can Simulate Non-Redundant Superposition Clause Learning

Martin Bromberger¹, Chaahat Jain^{1,2}, and Christoph Weidenbach¹

¹ Max Planck Institute for Informatics, Saarbrücken, Germany
{mbromber, cjain, weidenbach}@mpi-inf.mpg.de

² Graduate School of Computer Science, Saarbrücken, Germany

Abstract. We show that SCL(FOL) can simulate the derivation of non-redundant clauses by superposition for first-order logic without equality. Superposition-based reasoning is performed with respect to a fixed reduction ordering. The completeness proof of superposition relies on the grounding of the clause set. It builds a ground partial model according to the fixed ordering, where minimal false ground instances of clauses then trigger non-redundant superposition inferences. We define a respective strategy for the SCL calculus such that clauses learned by SCL and superposition inferences coincide. From this perspective the SCL calculus can be viewed as a generalization of the superposition calculus.

Keywords: first-order reasoning · superposition · SCL · non-redundant clause learning.

1 Introduction

Superposition [1,2,15] is currently considered as the prime calculus for first-order logic reasoning where all leading first-order theorem provers implement a variant thereof [12,17,13,19]. More recently, the family of SCL calculi (Clause Learning from Simple Models, or just Simple Clause Learning) [10,4,7,14,8] was introduced. There are first experimental results [3] available, and first steps towards an overall implementation [6,5].

The main differences between superposition and SCL for first-order logic without equality are: (i) superposition assumes a fixed ordering on literals whereas the ordering in SCL is dynamic and evolves out of the satisfiability of clauses, (ii) superposition performs single superposition left and factoring inferences whereas SCL typically performs several such inferences to derive a single learned clause, (iii) the superposition model operator is not effective on the non-ground clause level whereas the SCL model assumption is effective. For first-order logic without equality superposition reduces to ordered resolution combined with the powerful superposition redundancy criterion. Our simulation result cannot be one-to-one because an SCL learned clause is typically generated by several superposition inferences and superposition factoring inferences are performed by SCL only in the context of resolution inferences. The simulation result considers the ground case, where the superposition strategy used in the completeness

proof only triggers non-redundant inferences [1]. We call this strategy SUP-MO, Def. 5. Overall first-order completeness is then obtained by a lifting argument to the non-ground clause level. We actually show that a superposition refutation of some ground clause set can be simulated by an SCL refutation on the same clause set, such that they coincide on all superposition left (ordered resolution) inferences. For the superposition calculus we refer to [1] and for SCL to [8] where all main properties of both calculi have meanwhile been verified inside the Isabelle framework [18,9,16].

For example, consider a superposition refutation of the simple ground clause set

$$N_{\text{SUP}}^0 \{ (C_1) P(a) \vee P(a), \quad (C_2) \neg P(a) \vee Q(b), \quad (C_3) \neg Q(b) \}$$

with respect to a KBO [11], where all symbols have weight one, and precedence $a \prec b \prec P \prec Q$. Superposition generates only non-redundant clauses. Then with respect to the usual superposition ordering extension to literals and clauses we get $(C_1) \prec_{\text{KBO-SUP}} (C_2) \prec_{\text{KBO-SUP}} (C_3)$ and the superposition model operator produces the Herbrand model $N_{\text{SUP},\mathcal{I}}^0 = \emptyset$. Now clause (C_1) is the minimal false clause, triggering a factoring inference resulting in $(C_4) P(a)$ and clause set $N_{\text{SUP}}^1 = N_{\text{SUP}}^0 \cup \{ (C_4) P(a) \}$. The clause $P(a)$ cannot be derived by SCL because factoring is only performed in the context of resolution inferences. Now (C_4) is the smallest clause in N_{SUP}^1 and the superposition model operator produces $N_{\text{SUP},\mathcal{I}}^1 = \{ P(a), Q(b) \}$ with minimal false clause (C_3) . A superposition left inference between (C_3) and (C_2) generates $(C_5) \neg P(a)$ and $N_{\text{SUP}}^2 = N_{\text{SUP}}^1 \cup \{ (C_5) \neg P(a) \}$. The generation of $\neg P(a)$ can now be simulated by SCL by constructing the SCL trail $[P(a)^1 Q(b)^{\{\neg P(a) \vee Q(b)\}}]$ out of $N_{\text{SUP}}^0 = N_{\text{SCL}}^0$ leading to the learned clause $(C_5) \neg P(a)$ and respective clause set $N_{\text{SCL}}^2 = N_{\text{SCL}}^0 \cup \{ (C_5) \neg P(a) \}$. Note that $P(a)$ could have also been propagated, see Sec. 2 rule Propagate, but this would eventually not lead to the learned clause $(C_5) \neg P(a)$ but \perp . Finally, the superposition model operator produces $N_{\text{SUP},\mathcal{I}}^2 = \{ P(a), Q(b) \}$ with minimal false clause (C_5) and infers \perp . The SCL simulation generates the trail $[P(a)^{\{P(a)\}}]$ and then learns \perp as well out of a conflict with (C_5) . Note that this SCL trail is based on a factoring of (C_1) to $P(a)$ that was the explicit first step of the superposition refutation. Recall that by using an exhaustive propagation strategy, SCL would start with the trail $[P(a)^{P(a)} Q(b)^{\{\neg P(a) \vee Q(b)\}}]$ and immediately derive \perp . Exhaustive propagation is not a good strategy in general, because first-order logic clauses may enable infinitely many propagations. Therefore, the *regular* strategy defined in [8] does not require exhaustive propagation, but guarantees non-redundant clause learning. The SCL-SUP strategy, Def. 8, and Def. 10, simulating superposition SUP-MO runs is also a regular strategy, Lem. 17.

The paper is now organized as follows. After repetition of the needed concepts of SCL and superposition, Sec. 2, the simulation result is contained in Sec. 3. We show that any superposition refutation of a ground clause set producing only non-redundant inferences through the SUP-MO strategy, can be simulated via the SCL-SUP strategy. Based on the 14 simulation invariants of Def. 7, we show the invariants by an inductive argument on the length of the superposition

refutation, starting from the initial state, Lem. 13, for intermediate superposition inference steps Lem. 14 until the final refutation Lem. 15, and Lem. 16. For the simulation we do not consider selection in superposition inferences in favor of a less complicated presentation. The paper ends with a discussion of the obtained results. All proofs have been moved to an appendix in favor of improved readability.

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 [8]. The complement of a literal is denoted by the function comp . The function $\text{atom}(L)$ denotes the atomic part of a literal. 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 $\text{dom}(\sigma) := \{x \mid x\sigma \neq x\}$ is finite and $\text{codom}(\sigma) := \{t \mid x\sigma = t, x \in \text{dom}(\sigma)\}$. 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 $\text{codom}(\sigma)$ 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 substitution σ that is grounding for C . The function ground 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 . A (*partial*) *Herbrand model* I for a clause set N is a set of ground atoms. A ground clause C is true in I , denoted $I \models_H C$, if there is an atom $A \in C$ such that $A \in I$, or there is a negative literal $\neg A \in C$ such that $A \notin I$, and false otherwise. A ground clause set N entails a ground clause C , denoted $N \models C$, if $M \models C$ implies $M \models \{C\}$ for all models M .

We identify sets and sequences whenever appropriate. However, the trail of an SCL run is always a sequence of ground literals.

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 $C' \in \text{ground}(C)$ it holds that C' is redundant with respect to $\text{ground}(N)$.

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. (Note that LPO does not satisfy the last condition of a \prec_B ordering although it is a well-founded, total, strict ordering.) The ordering \prec_B is lifted to literals by comparing the respective atoms and if the atoms of two literals are the same, then the negative version of the literal is larger than the positive version. It is lifted to clauses by a multiset extension.

The SCL(FOL) Calculus: The inference rules of SCL(FOL) [8] are represented by an abstract rewrite system. They operate on a problem state, a six-tuple $(\Gamma; N; U; \beta; k; D)$ 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 size of the trail; k counts the number of decisions; and D is either \top , \perp or a clause closure $C \cdot \sigma$ such that $C\sigma$ is ground and false in Γ . Literals in Γ are either annotated with a number, also called a level; i.e., they have the form L^k meaning that L is the k -th guessed decision literal, or they 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 $(\Gamma; N; U; \beta; k; D)$ if L or $\text{comp}(L)$ occurs in Γ and the first decision 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 $(\Gamma; N; U; \beta; k; D)$ if i is the maximal level of a literal in D . The level of the empty clause \perp is 0. Recall D is a non-empty closure or \top or \perp . Similarly, a trail Γ is of level i if the maximal literal in Γ is of level i .

A literal/atom L/A is *undefined* in Γ if neither L/A nor $\text{comp}(L)/\text{comp}(A)$ occur in Γ . The start state of SCL is $(\epsilon; N; \emptyset; \beta; 0; \top)$ for some initial clause set N and bound β . The below rules are exactly the rules from [8] and serve as a reference for our simulation proof in Sec. 3.

Propagate $(\Gamma; N; U; \beta; k; \top) \Rightarrow_{\text{SCL}} (\Gamma, L\sigma^{(C_0 \vee L)\delta \cdot \sigma}; N; U; \beta; k; \top)$

provided $C \vee L \in (N \cup U)$, $C = C_0 \vee C_1$, $C_1\sigma = L\sigma \vee \dots \vee L\sigma$, $C_0\sigma$ does not contain $L\sigma$, δ is the mgu of the literals in C_1 and L , $(C \vee L)\sigma$ is ground, $(C \vee L)\sigma \prec_\beta \{\beta\}$, $C_0\sigma$ is false under Γ , and $L\sigma$ is undefined in Γ

Decide $(\Gamma; N; U; \beta; k; \top) \Rightarrow_{\text{SCL}} (\Gamma, L\sigma^{k+1}; N; U; \beta; k+1; \top)$

provided $\text{atom}(L)$ occurs C for a $C \in (N \cup U)$, $L\sigma$ is a ground literal undefined in Γ , and $L\sigma \prec_\beta \beta$

Conflict $(\Gamma; N; U; \beta; k; \top) \Rightarrow_{\text{SCL}} (\Gamma; N; U; \beta; k; D \cdot \sigma)$

provided $D \in (N \cup U)$, $D\sigma$ false in Γ for a grounding substitution σ

Skip $(\Gamma, L; N; U; \beta; k; D \cdot \sigma) \Rightarrow_{\text{SCL}} (\Gamma; N; U; \beta; k-i; D \cdot \sigma)$

provided $\text{comp}(L)$ does not occur in $D\sigma$, if L is a decision literal then $i = 1$, otherwise $i = 0$

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

Resolve $(\Gamma, L\delta^{(C \vee L)\cdot\delta}; N; U; \beta; k; (D \vee L') \cdot \sigma)$
 $\Rightarrow_{\text{SCL}} (\Gamma, L\delta^{(C \vee L)\cdot\delta}; N; U; \beta; k; (D \vee C)\eta \cdot \sigma\delta)$

provided $L\delta = \text{comp}(L'\sigma)$, $\eta = \text{mgu}(L, \text{comp}(L'))$

Backtrack $(\Gamma_0, K, \Gamma_1, \text{comp}(L\sigma)^k; N; U; \beta; k; (D \vee L) \cdot \sigma)$
 $\Rightarrow_{\text{SCL}} (\Gamma_0; N; U \cup \{D \vee L\}; \beta; j; \top)$

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 , and Γ_0 is of level j

A sequence of rule applications of a particular calculus is called a *run* of the calculus. A *strategy* for a calculus restricts the set of runs we actually allow by imposing further conditions on the allowed rule applications.

Definition 2 (SCL Runs). *A sequence of SCL rule applications is called a reasonable run if the rule Decide does not enable an immediate application of rule Conflict. A sequence of SCL rule applications is called a regular run if it is a reasonable run and the rule Conflict has precedence over all other rules.*

All regular SCL runs are sound, only derive non-redundant clauses, always terminate, and SCL with a regular strategy is refutationally complete (for first-order logic without equality) [8].

The Superposition Calculus: Superposition [1,2,15] is a calculus for first-order logic reasoning that also infers/learns new clauses like SCL. In contrast to SCL, it does these inferences based on a static ordering \prec and, at the level of inference rules, independent of a partial model. A permissible ordering \prec for the superposition calculus is always a well-founded, total, strict ordering on ground literals. This ordering is then lifted to clauses and clause sets by its respective multiset extension. A problem state in the superposition calculus is just a set N of clauses. The start state the initial clause set. Due to the restriction to first-order logic without equality, the most basic version of the superposition calculus consists just of the following two rules (without selection):

Superposition Left $(N \uplus \{C_1 \vee P(t_1, \dots, t_n), C_2 \vee \neg P(s_1, \dots, s_n)\}) \Rightarrow_{\text{SUP}} (N \cup \{C_1 \vee P(t_1, \dots, t_n), C_2 \vee \neg P(s_1, \dots, s_n)\} \cup \{(C_1 \vee C_2)\sigma\})$

where (i) $P(t_1, \dots, t_n)\sigma$ is strictly maximal in $(C_1 \vee P(t_1, \dots, t_n))\sigma$

(ii) $\neg P(s_1, \dots, s_n)\sigma$ is maximal, (iii) σ is the mgu of $P(t_1, \dots, t_n)$ and $P(s_1, \dots, s_n)$

Factoring $(N \uplus \{C \vee P(t_1, \dots, t_n) \vee P(s_1, \dots, s_n)\}) \Rightarrow_{\text{SUP}} (N \cup \{C \vee P(t_1, \dots, t_n) \vee P(s_1, \dots, s_n)\} \cup \{(C \vee P(t_1, \dots, t_n))\sigma\})$

where (i) $P(t_1, \dots, t_n)\sigma$ is maximal in $(C \vee P(t_1, \dots, t_n) \vee P(s_1, \dots, s_n))\sigma$

(ii) σ is the mgu of $P(t_1, \dots, t_n)$ and $P(s_1, \dots, s_n)$

Let $\text{sfac}(C)$ represent a clause obtained by exhaustively applying superposition Factoring on C . Recall, that superposition Factoring only applies to maximal positive literals. Let $\text{sfac}(N)$ represent the clause set N after every clause has been exhaustively factorized by Superposition Factorization.

Although the superposition calculus itself is independent of a partial model and may learn non-redundant clauses, the completeness proof of superposition in [1] is based on a strategy that builds ground partial models according to the fixed ordering \prec , where minimal false ground instances of clauses then trigger non-redundant superposition inferences. Note that the completeness proof relies on a grounding of the clause set that may lead to infinitely many clauses. However, the strategy from the completeness proof can also be seen as a superposition strategy for an initial clause set, where all clauses are already ground. On ground, finite clause sets, superposition restricted to the strategy only infers non-redundant clauses, always terminates, and is complete. The partial model needed in each step of the strategy is constructed according to the following model operator:

Definition 3 (Superposition Model Operator). *Let N be a set of ground clauses. Then N_I is the Herbrand model according to the superposition model operator for clause set N and it is constructed recursively over the partial Herbrand models N_C for all $C \in N$:*

$$N_C = \bigcup_{D \prec C} \delta_D \quad N_I = \bigcup_{C \in N} \delta_C$$

$$\delta_D = \begin{cases} \{B\} & \text{if } D = D' \vee B, B \text{ strictly maximal, } N_D \not\models_H D \\ \emptyset & \text{otherwise} \end{cases}$$

We say that a clause C is productive (wrt. the model construction of a clause set N) if $\delta_C \neq \emptyset$. We say that a clause C produces an atom B (wrt. the model construction of a clause set N) if $\delta_C = \{B\}$.

After constructing the model N_I for a clause set N , the strategy selects the smallest clause in N that is false in N_I . The strategy then selects a fitting inference rule based on the reason why the clause is false in N_I . The newly inferred clause either changes the model in the next step or changes the smallest clause that is false. This is the strategy used in the superposition completeness proof [1].

Definition 4 (Minimal False Clause). *The minimal false clause $C \in N$ is the smallest clause in N according to \prec such that $N_C \cup \delta_C \not\models_H C$.*

Definition 5 (Superposition Model-Operator Strategy: SUP-MO). *The superposition model-operator strategy is defined over the minimal false clause with regards to the current clause set N . The strategy can encounter the following cases:*

- (1) *N has no minimal false clause. Then N is satisfied by N_I and we can stop the superposition run.*
- (2) *The minimal false clause in N is \perp . Then N is unsatisfiable, which means we can also stop the superposition run.*

- (3) C is the minimal false clause in N , and it has a maximal literal L that is negative. Then there must be a clause $D \in N$ with $D \prec C$, a strictly maximal literal $\text{comp}(L)$, and $\delta_D = \{\text{comp}(L)\}$. In this case, the strategy applies as its next step *Superposition Left* to C and D .
- (4) C is the minimal false clause in N , and it has a maximal literal L that is positive. Then L is not strictly maximal in C and the strategy applies *Factoring* to C .

The first two cases of the SUP-MO strategy also describe its final states according to [1]. In all other states there is always exactly one rule applicable according to the SUP-MO strategy, which also means that SUP-MO is never stuck.

Lemma 6 (SUP-MO Applicability). *Let N be a set of ground clauses. If N has a minimal false clause $C \neq \perp$, then there exists exactly one rule applicable to N according to the SUP-MO strategy.*

3 SCL Simulates Superposition

In general, it is not possible to simulate all inferences of the superposition calculus with SCL because SCL only learns/infers non-redundant clauses, whereas syntactic superposition inferences have no such guarantees. Moreover, the inferences by SCL are all based on conflicts according to a partial model driven by the satisfiability of clause instances, whereas the inferences by superposition are based on a static ordering \prec . We can mitigate these differences by restricting superposition with the SUP-MO strategy because SUP-MO has non-redundancy guarantees and it infers new clauses based on minimal false clauses with respect to a ground partial model.

Let N^0 be a set of ground clauses, totally ordered by a superposition reduction ordering \prec . Let N^i (for $i > 0$) be the result of i steps of the superposition calculus applied to N^0 according to the SUP-MO strategy, i.e., $N^0 \Rightarrow_{\text{SUP-MO}} N^1 \Rightarrow_{\text{SUP-MO}} \dots \Rightarrow_{\text{SUP-MO}} N^i$. Again, all N^i are sets of ground clauses, totally ordered by a superposition reduction ordering \prec . The SCL strategy SCL-SUP that simulates superposition restricted to SUP-MO runs is defined inductively on the clause ordering \prec . To guide and to prove the correctness of our simulation, we assign to each SCL state and every clause some additional information. For this purpose, every SCL state is annotated with a triple (i, C, γ) , where i is an integer that states that the SCL state simulates the superposition state N^i , C is the last clause that was used as a decision aid by the strategy, γ is a function such that $\gamma(C) = \text{sfac}(C)$ if $\text{sfac}(C) \in N^i$ and $\gamma(C) = C$ otherwise, the SCL state also simulates the model construction for N^i upto $N_{C'}^i \cup \delta_{C'}$, where $C' = \gamma(C)$. The annotated states are written $(\Gamma; N^0; U; \beta; k; E)_{(i, C, \gamma)}$. The overall start state is then $(\epsilon; N^0; \emptyset; \beta; 0; \top)_{(0, \perp, \gamma)}$, where we assume β large enough so $A \prec_\beta \beta$ for all $A \in \text{atom}(N^0)$, $\perp \notin N^0$, and $\gamma(C) = \text{sfac}(C)$ if $\text{sfac}(C) \in N^0$ and $\gamma(C) = C$ otherwise. We will later see that the annotated integer is not relevant for the actual choice of SCL rules by the SCL-SUP strategy but only

to prove that the strategy actually simulates superposition. Moreover, we define a new ordering \prec_γ based on our superposition ordering \prec and function γ such that $C \prec_\gamma D$ if $\gamma(C) \prec \gamma(D)$.

Definition 7 (State Simulation). *Let $(\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ be an SCL state for the input clauses N^0 . Let L be the maximal literal in D if $D \neq \perp$ and the minimal literal according to \prec otherwise. Let $N^0 \Rightarrow_{\text{SUP-MO}} N^1 \Rightarrow_{\text{SUP-MO}} \dots \Rightarrow_{\text{SUP-MO}} N^i$ be the superposition run following the SUP-MO strategy starting from the input clause set N^0 . Let $D' = \gamma(D)$. Then we say that the SCL state $(\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ simulates N^i and the model construction upto $N_{D'}^i \cup \delta_{D'}$ if*

- (i) $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$, $A \prec_\beta \beta$ for all $A \in \text{atom}(N^0)$, and $D \in \{\perp\} \cup N^0 \cup U$
- (ii) $\text{sfac}(N^0 \cup U) \subseteq \text{sfac}(N^i)$ and $\gamma(C) \in N^i$ for all $C \in N^0 \cup U$ and $\gamma(C) = \text{sfac}(C)$ or $\gamma(C) = C$.
- (iii) for all $C \in N^i$ there exists a $C' \in N^0 \cup U \cup \{E\}$ such that $\text{sfac}(C) = \text{sfac}(C')$ if the maximal literal in C is positive
- (iv) for all $C \in N^i$ there exists a $C' \in N^0 \cup U \cup \{E\}$ such that $C' \models C$ and $\gamma(C') \preceq C$
- (v) for all atoms A occurring in N^0 : $A \in N_{D'} \cup \delta_{D'}$ iff $A \in \Gamma$
- (vi) for all atoms A : $\neg A \in \Gamma$ iff $A \prec L$ and $A \notin N_{D'}$
- (vii) for every literal L in Γ , i.e., $\Gamma = \Gamma', L, \Gamma''$, and all literals L' in Γ' , $\text{atom}(L') \prec \text{atom}(L)$
- (viii) for every atom (= positive literal) B in Γ , i.e., $\Gamma = \Gamma', B, \Gamma''$, there exists $C \in N^0 \cup U$ and a $C' \in N^i$ such that $\gamma(C) = \text{sfac}(C) = \text{sfac}(C') = C'$, and C' produces B , i.e., $\delta_{C'} = \{B\}$
- (ix) for every clause $C \in N^i$ with $C \preceq \gamma(D)$ that produces an atom B , i.e., $\delta_C = \{B\}$, there exists $C' \in N^0 \cup U$ such that $C = \gamma(C')$ and $C \preceq_\gamma D$.
- (x) Γ contains only decisions if $E = \top$
- (xi) $E \notin \{\top, \perp\}$ iff $\Gamma = \Gamma' B^{\text{sfac}(D)}$, Γ' contains only decisions, there exists $E' \in N^i$ where $\gamma(E) = E = E'$ is the minimal false clause in N^i , and $\neg B \in E$
- (xii) $\Gamma \models C$ for all $C \in N^0 \cup U$ with $C \preceq_\gamma D$
- (xiii) Conflict is not applicable to $(\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$.
- (xiv) $\perp \notin N^0 \cup U$ and $E = \perp$ iff $\Gamma = \epsilon$ and $\perp \in N^i$

The above invariants can be summarized as follows: (i) All ground atoms encountered are known from the start and the trail bound β is large enough so SCL can Decide/Propagate them. (ii)-(iv) Every initial clause C or inferred clause by SUP-MO must coincide with an initial clause C' or learned clause by SCL; this means on the one hand that for every clause C learned by SCL-SUP, SUP-MO infers a clause C' that is identical up to factoring; on the other hand it means that for every clause C inferred by SUP-MO, SCL-SUP learns a clause C' that entails C (i.e. $C' \models C$) and is at most as large as C wrt. γ . (v)-(ix) The partial model constructed by SCL-SUP and SUP-MO coincide and any atom B in $N_C \cup \delta_C$ produced by clause D has a clause D' on the SCL side that could

propagate B and vice versa. (x)-(xiii) Ensure that any Conflict in SCL-SUP corresponds to a minimal false clause and that the trail is always constructed in such a way that the Resolve applications per Conflict call are limited to the maximal literal in the conflict; this property is needed or the next clause that would be learned by SCL no longer coincides with the clauses learned by SUP-MO. (xiv) Describes the final state in case the input clause set is unsatisfiable.

Now that we have defined how an SCL state must look like in order to simulate a superposition state, we define SCL-SUP, the SCL strategy that eventually simulates a SUP-MO run. First, note that not all states visited by SCL-SUP satisfy the the invariants of Def. 7. However, the invariants hold again after each so-called *atomic sequence* of SCL-SUP steps. Second, one atomic sequence of SCL-SUP steps may skip over several successive superposition states. The reason is that SCL can and must skip all steps of SUP-MO that occur because the maximal literal in a clause is not strictly maximal, i.e., superposition Factoring steps. SCL performs factoring implicitly in its Propagation rule so SCL never has to explicitly simulate case (4) of Def. 5. Third, definition of the SCL-SUP strategy is split in two parts and each part describes some atomic sequences of SCL-SUP steps.

Definition 8 (SCL Superposition Strategy: SCL-SUP Part 1). *Let $S_0 = (\Gamma; N^0; U; \beta; k; \top)_{(i,C,\gamma)}$ be an SCL state with additional annotations for the strategy. Let D be the next largest clause from C in the ordering \prec_γ with respect to the ground clause set $N^0 \cup U$. Let L be the maximal literal of D . Let $[\neg A_1, \neg A_2, \dots, \neg A_n]$ be all negative literals such that for all i we have $A_i \prec L$, all A_i undefined in Γ , A_i occurs in $N^0 \cup U$, and $A_i \prec A_{i+1}$. Let $D' = \gamma(D)$ be in N^i such that $\text{sfac}(D) = \text{sfac}(D')$. Let $j_0 + 1$ be the number of occurrences of L in D' and $j = i + j_0$. Then the SCL Superposition Strategy (SCL-SUP) performs the following steps to S_0 (possibly without any actual SCL rule applications, just changing the state annotation):*

- (1) *First decide all literals $[\neg A_1, \neg A_2, \dots, \neg A_n]$ in order, i.e., $S_0 \Rightarrow_{SCL-SUP}^* \text{Decide} S_1$, where $S_1 = (\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}; N^0; U; \beta; k + n; \top)_{(i,D,\gamma)}$.*
- (2a) *If the maximal literal L in D is positive (i.e., $L = B$), $\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n} \not\models D$, and Conflict is not applicable to $S_2 = (\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{k+n+1}; N^0; U; \beta; k + n + 1; \top)_{(j,D,\gamma')}$, then decide B , i.e., $S_1 \Rightarrow_{SCL-SUP}^{\text{Decide}} S_2$, where γ' is the same as γ except that $\gamma'(D) = \text{sfac}(D)$.*
- (2b) *If the maximal literal L in D is positive (i.e., $L = B$), $\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n} \not\models D$, and E is the smallest clause in $N^0 \cup U$ that is false in wrt. $\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{\text{sfac}(D)}$, then propagate B and apply Conflict to E , i.e., $S_1 \Rightarrow_{SCL-SUP}^{\text{Propagate}} S_2' \Rightarrow_{SCL-SUP}^{\text{Conflict}} S_2$, where $S_2 = (\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{\text{sfac}(D)}; N^0; U; \beta; k + n; E)_{(j,D,\gamma')}$ and γ' is the same as γ except that $\gamma'(D) = \text{sfac}(D)$.*
- (2c) *Otherwise, $S_2 = S_1$ and no further rules have to be applied.*

A (potentially empty) sequence of SCL rule applications according to SCL-SUP is called an atomic sequence of SCL-SUP steps if it starts from a state S_0 and ends in a state S_2 outlined in the cases (2a-c).

The first part of the strategy simulates the recursive construction of the partial model used in the SUP-MO strategy (see Def. 3). It assumes that the model is already constructed up to the current annotated clause C and extends this model for the next largest clause $D \in (N^0 \cup U)$. To this end, it uses the rule Decide in step (1) to set all atoms A to false that are still undefined but can no longer be produced by any clause greater or equal to D . Next the strategy makes a case distinction. Step (2a) handles the case where D corresponds to a clause D' in the superposition state (modulo some Factoring steps skipped by SCL) that produces atom B ; SCL-SUP then adds B to the trail with the rule Decide because producing/adding this atom does not falsify a clause. Step (2b) handles a similar case compared to step (2a); but in this case producing/adding the atom B to the trail results in a minimal false clause E ; in order to force a resolution step between clause D and E , SCL-SUP first uses Propagate to add B to the trail and then applies conflict to E . Step (2c) handles the case where D corresponds to a clause D' that will not produce an atom B even modulo some Factoring steps; in this case no further SCL rule applications are necessary as the SUP-MO model will not change. Note that the annotated function γ is needed so the SCL state knows when the superposition state would have applied Factoring to a clause C , which also means that it is now treated as its factorized version $\gamma(C) = \text{sfac}(C)$ in our inductive clause ordering.

Example 9. Let us now further demonstrate the three different cases of the first part of the SCL-SUP strategy with the help of an example. Let N^0 be our initial set of clauses:

$$N^0 = \{(C_1) P(a), (C_2) \neg P(b) \vee Q(a), (C_3) \neg P(a) \vee Q(a) \vee Q(a), \\ (C_4) P(a) \vee \neg Q(a), (C_5) \neg P(a) \vee \neg Q(a)\}$$

We compare the run of SCL-SUP for N^0 with the run of SUP-MO for N^0 to demonstrate that both runs coincide. As superposition ordering, we choose an LPO with precedence $a \prec b \prec P \prec Q$. This means that the atoms are ordered $P(a) \prec P(b) \prec Q(a) \prec Q(b)$ and the clauses in N^0 are ordered $C_1 \prec C_2 \prec C_3 \prec C_4 \prec C_5$. The initial SUP-MO state is simply the clause set N^0 and the initial SCL-SUP state is $(\epsilon, N^0, \emptyset, \beta, 0, \top)_{(0, \perp, \gamma_0)}$, where $\gamma_0(C) = C$ for all clauses C . In the first step of SCL-SUP, SCL-SUP first selects the clause C_1 as its new decision aid because it is the next largest clause in N^0 compared to \perp . Then SCL-SUP continues with step (1) of Def. 3. In this step SCL-SUP does nothing because there are no atoms smaller than $P(a)$. Next, SCL-SUP detects that the maximal literal of C_1 is positive, $\epsilon \not\models C_1$, and that the trail $[P(a)^1]$ does not result in a conflict. Therefore, SCL-SUP follows step (2a) of Def. 3 and Decides $P(a)$, which results in the state $([P(a)^1], N^0, \emptyset, \beta, 1, \top)_{(0, C_1, \gamma_0)}$. Meanwhile, SUP-MO starts with constructing a model for N^0 starting with the clause C_1 . The result is that C_1 is productive and $\delta_{C_1} = \{P(a)\}$ and $N_{C_1}^0 = \emptyset$, which coincides with our new SCL trail.

SCL-SUP considers the clause C_2 as its new decision aid and continues with step (1) of Def. 3. This time there is an atom smaller than the maximal literal of C_2 namely $P(b)$. Therefore, SCL-SUP Decides $\neg P(b)$ in step (1) of Def. 3, which results in $([P(a)^1, \neg P(b)^2], N^0, \emptyset, \beta, 2, \top)_{(0, C_2, \gamma_0)}$. Next, SCL-SUP detects that the maximal literal of C_2 is positive but that $[P(a)^1, \neg P(b)^2] \models C_2$. Therefore, SCL-SUP follows step (2c) of Def. 3 and ends this atomic sequence immediately. SUP-MO continues the model construction for N^0 with the clause C_2 . The clause C_2 is not productive because $N_{C_2}^0 \models_H C_2$, where $N_{C_2}^0 = \delta_{C_1} = \{P(a)\}$ and $\delta_{C_2} = \emptyset$, which again coincides with our new SCL trail as Herbrand models do not explicitly define atoms assigned to false.

SCL-SUP now considers the clause C_3 as its new decision aid and continues with step (1) of Def. 3. In this step SCL-SUP does nothing because all atoms smaller than $Q(a)$ are already assigned. Next, SCL-SUP detects that the maximal literal of C_3 is positive, $[P(a)^1, \neg P(b)^2] \not\models C_3$, and that the clause C_5 is false with respect to the trail $[P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}]$. Therefore, SCL-SUP follows step (2b) of Def. 3, i.e. it Propagates $P(a)$ and applies Conflict to C_5 , resulting in $([P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}], N^0, \emptyset, \beta, 2, C_3)_{(1, C_2, \gamma_1)}$, where γ_1 is identical to γ_0 except that $\gamma_1(C_3) = \text{sfac}(C_3) = \neg P(a) \vee Q(a)$. Note that SCL-SUP must change the state annotations because the maximal literal in C_3 is not strictly maximal, so SCL-SUP skips and eventually silently performs the Factorization step performed by SUP-MO. Note also that in the changed clause ordering \prec_{γ_1} the order of C_2 and C_3 changed, i.e., $C_3 \prec_{\gamma_1} C_2$, which corresponds to $\text{sfac}(C_3) \prec C_2$. Meanwhile, SUP-MO continues the model construction for N^0 with the clause C_3 . The clause C_3 is not productive because the maximal literal is not strictly maximal so $\delta_{(3)} = \emptyset$ and $N_{C_3}^0 \cup \delta_{C_3} \not\models_H C_3$ so C_3 is the minimal false clause in N^0 . SUP-MO resolves this conflict by applying Factoring to C_3 , which means SUP-MO infers the clause $C_6 = \text{sfac}(C_3) = \neg P(a) \vee Q(a)$. The new clause order in superposition state $N^1 = N^0 \cup \{C_6\}$ is $C_1 \prec C_6 \prec C_2 \prec C_3 \prec C_4 \prec C_5$, which matches the changed ordering $C_3 \prec_{\gamma_1} C_2$ because $C_6 = \gamma_1(C_3)$. Next, SUP-MO updates its model construction for N^1 . The result is that C_1 and C_6 are productive and that $N_{C_6}^1 \cup \delta_{C_6} = \{P(a), Q(a)\}$, which matches the current SCL trail. Moreover, if we continue the model construction upto C_5 then no new literals are produced and C_5 also turns into the minimal false clause for N^1 .

Definition 10 (SCL Superposition Strategy: SCL-SUP Part 2). *Let $S_0 = (\Gamma, B^{\text{sfac}(C)}; N^0; U; \beta; k; E)_{(i, C, \gamma)}$ be an SCL state with $E \notin \{\top, \perp\}$ and additional annotations for the strategy. Let $L = \neg B$ be the maximal literal of E . Let Γ contain only decision literals. Let all atoms A occurring in $N^0 \cup U$ with $A \prec B$ be defined in Γ following the order \prec , i.e., for all A occurring in $N^0 \cup U$ with $A \prec B$ there exist Γ' and Γ'' such that $\Gamma' = \Gamma, L_A, \Gamma'', L_A = A$ or $L_A = \neg A$ and all atoms $A' \in N^0 \cup U$ with $A' \prec A$ are defined in Γ' . Let E be contained in N^i . Let j_0 be the number of occurrences of L in E and $j = i + j_0$. Let $\text{sfac}(C) = C_1 \vee B$ and $E = E' \vee E''$, where E'' contains all occurrences of L in E . Then the SCL Superposition Strategy (SCL-SUP) performs the following steps to S_0 :*

- (1) First apply *Resolve* to E until all occurrences of L are resolved away, i.e., $S_0 \Rightarrow_{SCL-SUP}^* \text{Resolve} S_1$, where $S_2 = (\Gamma, B^{\text{sfac}(C)}; N^0; U; \beta; k; E_2)_{(j,C,\gamma)}$ and $E_2 = E' \vee C_1 \vee \dots \vee C_1$.
- (2a) If $E_2 = \perp$, then we apply *Skip* until the trail is empty and then stop the SCL run, i.e., $S_2 \Rightarrow_{SCL-SUP}^* \text{Skip} S_5$, where $S_5 = (\epsilon; N^0; U; \beta; 0; \perp)_{(j,\perp,\gamma)}$.
- (2b) If $E_2 \neq \perp$, then E_2 has a maximal literal L_1 . Next the strategy applies *Skip* until $\text{comp}(L_1)$ is the topmost literal on the trail, i.e., $S_2 \Rightarrow_{SCL-SUP}^* \text{Skip} S_3$, where $S_3 = (\Gamma_0, L_1^{k_1}; N^0; U; \beta; k_1; E_2)_{(j,C,\gamma)}$. (Note that this step skips at least over the literal $B^{\text{sfac}(C)}$).
- (3) Next apply *Backtrack* to S_3 , i.e., $S_3 \Rightarrow_{SCL-SUP}^* \text{Backtrack} S_4$, where $S_4 = (\Gamma_0; N^0; U \cup \{E_2\}; \beta; k_1 - 1; \top)_{(j,C,\gamma)}$.
- (4a) If L_1 is a negative literal, continue with the following rule applications. Let D be the smallest clause in $N^0 \cup U$ with maximum literal $\text{comp}(L_1) = B_1$ and $\Gamma_0 \not\models D$. Then *Propagate* B_1 from D , and apply *Conflict* to E_2 , i.e., $S_4 \Rightarrow_{SCL-SUP}^* \text{Propagate} S_4'' \Rightarrow_{SCL-SUP}^* \text{Conflict} S_5$, where $S_5 = (\Gamma_0, B_1^{\text{sfac}(D)}; N^0; U \cup \{E_2\}; \beta; k_1 - 1; E_2)_{(j,D,\gamma)}$.
- (4b) If L_1 is a positive literal (i.e., $L_1 = B$) and *Conflict* is not applicable to $S_5 = (\Gamma_0, B_1^{k_1}; N^0; U \cup \{E_2\}; \beta; k_1; \top)_{(j_2, E_2, \gamma')}$, then decide B , i.e., $S_4 \Rightarrow_{SCL-SUP}^* \text{Decide} S_5$, where $j_1 + 1$ is the number of occurrences of B_1 in E_2 , $j_2 = j + j_1$, and γ' is the same as γ except that $\gamma'(E_2) = \text{sfac}(E_2)$.
- (4c) If L_1 is a positive literal (i.e., $L_1 = B$) and E_3 is the smallest clause in $N^0 \cup U$ that is false in $S_5' = (\Gamma_0, B_1^{\text{sfac}(E_2)}; N^0; U; \beta; k_1 - 1; \top)_{(j, E_2)}$, then propagate B_1 and apply *Conflict* to E_3 , i.e., $S_4 \Rightarrow_{SCL-SUP}^* \text{Propagate} S_5' \Rightarrow_{SCL-SUP}^* \text{Conflict} S_5$, where $S_5 = (\Gamma_0, B_1^{\text{sfac}(E_2)}; N^0; U; \beta; k_1 - 1; E_3)_{(j_2, E_2, \gamma')}$, $j_1 + 1$ is the number of occurrences of B_1 in E_2 , $j_2 = j + j_1$, and γ' is the same as γ except that $\gamma'(E_2) = \text{sfac}(E_2)$.

A (potentially empty) sequence of SCL rule applications according to SCL-SUP is called an atomic sequence of SCL-SUP steps if it starts from a state S_0 and ends in a state S_5 outlined in the cases (2a) and (5a-c).

The second part of the strategy simulates the actual inferences resulting from a minimal false clause found in step (2b) of Def. 8 or found in steps (4a) and (4c) of Def. 10. These inferences always correspond to Superposition Left steps of the SUP-MO strategy that resolve minimal false clauses E' in N^i with maximal literal $\neg B$ with the clause C' in N^i that produced B . Note however that SCL-SUP may combine several Superposition Left steps of the SUP-MO strategy into one new learned clause. This is the case whenever the maximal literal $\neg B$ in the minimal false clause E' in N^i is not strictly maximal. In this case, the next minimal false clause E'' will always correspond to the last inferred clause, the maximal literal of this clause will still be $\neg B$, the clause producing B will be again C' , and therefore the next Superposition Left partner of E'' is also again C' . Moreover, all of the skipped inferences are actually redundant with respect to the final inference E_2' in this chain, which explains why SCL-SUP is still capable of simulating SUP-MO although it skips the intermediate inferences. The actual

SCL-SUP clause E_2 corresponding to final SUP-MO inference E'_2 is computed in the steps (1) and (2) of Def. 10 with greedy applications of the rules Resolve and Factorize. The following steps of Def. 10 take care of the four different cases how E'_2 changes the model and minimal false clause in N^j . The first case is that $E'_2 = \perp$ so SUP-MO has reached a final state. This case is handled by step (2a) of Def. 10 that simply empties the trail with applications of the rule Skip so the resulting SCL state has the form of a SCL-SUP final state. The second case is that the maximal literal L_1 in E'_2 is negative. In this case, the model for N^i and N^j is still the same and just the minimal false clause changes to E'_2 . This case is handled by steps (2b)-(4a) of Def. 10 that Backtrack before $\text{comp}(L_1)$ was decided, propagate it instead and apply Conflict to E_2 . In the third and fourth case the maximal literal L_1 in E'_2 is positive. In this case, the model for N^i and N^j actually changes because E'_2 is always productive. Case (2b)-(4b) of Def. 10 handles the case where producing L_1 leads to no new minimal false clause, and case (2b)-(4c) of Def. 10 handles the case where it does. Both cases work symmetrically to steps (2a) and (2b) of Def. 8.

Example 11. We continue Ex. 9 to demonstrate cases (1)→(4a) and (1)→(2a) of the second part of the SCL-SUP strategy. We left the runs in the SCL state $([P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}], N^0, \emptyset, \beta, 2, C_3)_{(1, C_2, \gamma_1)}$ that simulates the superposition state N^1 , where

$$N^1 = \{(C_1)P(a), (C_2) \neg P(b) \vee Q(a), (C_3) \neg P(a) \vee Q(a) \vee Q(a), \\ (C_4) P(a) \vee \neg Q(a), (C_5) \neg P(a) \vee \neg Q(a), (C_6) \neg P(a) \vee Q(a)\}$$

and C_5 became the minimal false clause in N^1 after C_1 and C_6 produced together the partial model $\{P(a), Q(a)\}$. SUP-MO continues from the state N^1 by applying Superposition Left to C_5 and C_6 . In the new state $N^2 = N^1 \cup \{(C_7) \neg P(a) \vee \neg P(a)\}$ the new clause order is $C_1 \prec C_7 \prec C_6 \prec C_2 \prec C_3 \prec C_4 \prec C_5$ and after constructing the model for C_1 , which produces again $P(a)$, the clause C_7 becomes again the minimal false clause. SCL-SUP follows (1) of Def. 10 and applies Resolve to C_5 and $\text{sfac}(C_3) = C_6$, resulting in the state $([P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}], N^0, \emptyset, \beta, 2, C_7)_{(2, C_2, \gamma_1)}$. Then SCL-SUP continues with steps (2b) and (3) by applying Skip twice and Backtrack once to jump to the state $(\epsilon, N^0, \{C_7\}, \beta, 0, \top)_{(2, C_2, \gamma_1)}$. Next, SCL-SUP continues with step (4a) because the maximal literal of C_7 is $\neg P(a)$ and therefore negative. This means SCL-SUP will add $P(a)$ again to the trail but this time by applying Propagate to C_1 and afterwards it applies Conflict to C_7 . The resulting state $([P(a)^{\text{sfac}(C_1)}], N^0, \{C_7\}, \beta, 0, C_7)_{(2, C_1, \gamma_1)}$ matches again the SUP-MO state N^2 .

SUP-MO continues from the state N^2 by applying Superposition Left to C_7 and C_1 , resulting in $N^3 = N^2 \cup \{(C_8) \neg P(a)\}$. Since C_8 has the same maximal literal as C_7 it becomes automatically the next minimal false clause in N^3 . As a result, SUP-MO applies Superposition Left to C_8 and C_1 , which returns $N^5 = N^3 \cup \{(C_9) \perp\}$ a final state that proves the unsatisfiability of N^0 . Meanwhile, SCL-SUP simulates both Superposition Left steps with one atomic SCL-SUP sequence. It starts with step (1) of Def. 10 and applies Resolve twice, resulting in the state $([P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}], N^0, \emptyset, \beta, 2, \perp)_{(4, C_2, \gamma_1)}$. Then it continues

with step (2a) of Def. 10 and applies Skip until the trail is empty. The resulting state $(\epsilon, N^0, \emptyset, \beta, 2, \perp)_{(4, \perp, \gamma_1)}$ is a final state and proves unsatisfiability of N^0 .

Example 12. The next example demonstrates the atomic sequence (1)→(4b) of the second part of the SCL-SUP strategy. Let N^0 be our initial set of clauses:

$N^0 = \{(C_1)P(a), (C_2)\neg P(b), (C_3)\neg P(a) \vee Q(a), (C_4)P(b) \vee \neg Q(a)\}$. As superposition ordering, we choose an LPO with precedence $a \prec b \prec P \prec Q$. This means that the atoms are ordered $P(a) \prec P(b) \prec Q(a) \prec Q(b)$ and the clauses in N^0 are ordered $C_1 \prec C_2 \prec C_3 \prec C_4$. In order to keep the example short, we skip the initial SCL-SUP steps and continue directly with the state $S = ([P(a)^1, \neg P(b)^2, Q(a)^{\text{sfac}(C_3)}], N^0, \emptyset, \beta, 2, C_4)_{(0, C_3, \gamma)}$, where $\gamma(C) = C$ for all clauses C and $\beta = Q(b)$. This state simulates the superposition state N^0 upto the model construction for C_3 , where $N_{C_3}^0 \cup \delta_{C_3} = \delta_{C_1} \cup \delta_{C_3} = \{P(a), Q(a)\}$ and C_4 is the minimal false clause. SUP-MO continues from the state N^0 by applying Superposition Left to C_4 and C_3 . In the new state $N^1 = N^0 \cup \{(C_5)\neg P(a) \vee P(b)\}$ the new clause order is $C_1 \prec C_5 \prec C_2 \prec C_3 \prec C_4$ and the partial model upto C_5 is $N_{C_5}^0 \cup \delta_{C_5} = \delta_{C_1} \cup \delta_{C_5} = \{P(a)\} \cup \{P(b)\}$, which turns C_2 into the next minimal false clause. SCL-SUP simulates the above steps by following the atomic sequence (1)→(4b) of Def. 10. The result is the state $([P(a)^1, P(b)^{\text{sfac}(C_5)}], N^0, \{C_5\}, \beta, 1, C_2)_{(1, C_5, \gamma)}$ matching again our current superposition state and model.

Without clause C_2 , SCL-SUP would apply the atomic sequence (1)→(4a) of Def. 10 to S , resulting in the state $([P(a)^1, P(b)^2], N^0 \setminus \{C_2\}, \{C_5\}, \beta, 2, \top)_{(1, C_5, \gamma)}$. This matches the state $N^1 \setminus \{C_2\}$ and its partial model upto C_5 that is still the same as for N^1 with the exception that it does not lead to a minimal false clause.

In order to actually show that every SCL-SUP run simulates a SUP-MO run, we need to prove three properties. The first property is that each state visited by an SCL-SUP run must simulate a state visited by the corresponding SUP-MO run. Note that this property does not yet say anything about the order in which SCL-SUP simulates the SUP-MO states. This property can also be seen as a soundness argument for our strategy.

Lemma 13 (Initial SCL State Simulates Initial Superposition State).

The initial SCL state $(\epsilon; N^0; \emptyset; \beta; 0; \top)_{(0, \perp, \gamma)}$ simulates the initial superposition state N^0 and the model construction upto $N_{\perp}^0 \cup \delta_{\perp}$

Lemma 14 (SCL-SUP Preserves Simulation).

Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i, C, \gamma)}$ simulate the superposition state N^i and the corresponding model construction upto $N_{C'}^i \cup \delta_{C'}$, where $C' = \gamma(C)$. Let the SCL state $S' = (\Gamma'; N^0; U'; \beta'; k'; E')_{(j, D, \gamma')}$ be the result of one atomic sequence of SCL-SUP steps. Then there exists a clause $D' \in N^j$ with $\gamma'(D) = D'$ and S' simulates the superposition state N^j and the model construction upto $N_{D'}^j \cup \delta_{D'}$.

The second property is that each atomic sequence of SCL-SUP steps always makes progress in the simulation. This means that each atomic sequence of SCL-SUP steps either advances the superposition state N^i simulated by the current

SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$, i.e., it increases the annotated i , or it still simulates the same superposition state N^i but advances the simulation of the model construction operator, i.e. it increases the annotated clause C and keeps i the same. Note that it can actually happen that an atomic sequence of SCL-SUP steps skips over several superposition states. This property can also be seen as a termination argument for our strategy because SUP-MO always terminates on ground clause sets.

Lemma 15 (SCL-SUP Advances the Simulation). *Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ simulate the superposition state N^i and the model construction upto $N_D^i \cup \delta_D$. Let the SCL state $S' = (\Gamma'; N^0; U'; \beta; k'; E')_{(j,D',\gamma')}$ be the next state reachable by one atomic sequence of SCL-SUP steps. Then either $i < j$ or $i = j$ and $\gamma' = \gamma$ and $D \prec_\gamma D'$.*

The last missing property shows that the SCL-SUP strategy can always advance the current SCL state whenever the simulated superposition state can be advanced by the SUP-MO strategy. This means SCL-SUP is never stuck when SUP-MO can still progress. These properties hold because the simulation invariants in Def. 7 either correspond to a correct final state or they satisfy the preconditions of Def. 8 or Def. 10. This property can also be seen as a partial correctness argument for our strategy.

Lemma 16 (SCL-SUP Correctness of Final States). *Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ simulate the superposition state N^i and the model construction upto $N_{\gamma(D)}^i \cup \delta_{\gamma(D)}$. Let there be no more states reachable from S following an atomic sequence of SCL-SUP steps. Then S is a final state, i.e., either (i) $E = \perp$, $D = \perp$, $\perp \in N^i$, and N^0 is unsatisfiable or (ii) $\Gamma \models N^0$.*

We can also show that any SCL-SUP run is also a regular run. Although this is not strictly necessary for the simulation proof, it is beneficial because it means that SCL-SUP inherits many properties that hold for SCL restricted to a regular strategy. For instance, that all learned clauses are non-redundant and that SCL-SUP always terminates.

Lemma 17 (SCL-SUP is a Regular SCL Strategy). *SCL-SUP is a regular SCL strategy if it is executed on a state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,C,\gamma)}$ that simulates a superposition state N^i and the corresponding model construction upto $N_{\gamma(C)}^i \cup \delta_{\gamma(C)}$.*

4 Conclusion

We have shown that the SCL(FOL) calculus [8] can simulate model driven superposition [1] refutations deriving only non-redundant clauses. The superposition calculus cannot simulate SCL refutations due to its static a priori ordering. In general, an SCL(FOL) learned clause is generated out of several resolution and factorization steps. From this perspective the SCL(FOL) calculus is more general and flexible than the superposition calculus. Furthermore, it only generates

non-redundant clauses whereas any superposition implementation generates redundant clauses due to the syntactic application of the superposition inference rules.

Selection in superposition can also be simulated, but requires an additional branch in the SCL-SUP strategy, because selection of non-maximal, negative literals by superposition requires a different trail ordering for SCL in order to simulate a respective superposition left inference.

For future work, we plan to lift our simulation result from the ground case to the non-ground case. This lifting will require the extension of the SCL calculus by an additional rule that learns clauses that are computed as intermediate steps during the conflict analysis. This rule was left out of previous versions of SCL because we would never use it in a CDCL inspired SCL-run and because it would have complicated the termination and non-redundancy proofs for SCL. (Although we are sure that the rule can be designed in such a way that all properties of the original calculus still hold.) Considering the extension to the non-ground case, this result can be used in various directions. It can be used to develop an alternative implementation of the superposition calculus. Given a fixed ordering, the trail can be developed according to the ordering, generating only non-redundant superposition inferences. On the other hand, the concept of finite saturation can be kept this way preserving a strong mechanism for detecting satisfiability. Secondly, the result means that SCL can be used to naturally combine propagation driven reasoning with fixed ordering driven reasoning. This might overcome some of the issues of the current first-order portfolio approaches implemented in the state-of-the-art provers.

References

1. Bachmair, L., Ganzinger, H.: Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation* 4(3), 217–247 (1994), revised version of Max-Planck-Institut für Informatik technical report, MPI-I-91-208, 1991
2. Bachmair, L., Ganzinger, H.: Resolution theorem proving. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, vol. I, chap. 2, pp. 19–99. Elsevier (2001)
3. Bromberger, M., Dragoste, I., Faqeh, R., Fetzer, C., González, L., Krötzsch, M., Marx, M., Murali, H.K., Weidenbach, C.: A sorted datalog hammer for supervisor verification conditions modulo simple linear arithmetic. In: Fisman, D., Rosu, G. (eds.) *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS. Springer* (2022)
4. Bromberger, M., Fiori, A., Weidenbach, C.: Deciding the bernays-schoenfinkel fragment over bounded difference constraints by simple clause learning over theories. In: Henglein, F., Shoham, S., Vizel, Y. (eds.) *Verification, Model Checking, and Abstract Interpretation - 22nd International Conference, VMCAI 2021, Copenhagen, Denmark, January 17-19, 2021, Proceedings. Lecture Notes in Computer Science*, vol. 12597, pp. 511–533. Springer (2021)

5. Bromberger, M., Gehl, T., Leutgeb, L., Weidenbach, C.: A two-watched literal scheme for first-order logic. In: Konev, B., Schon, C., Steen, A. (eds.) Proceedings of the Workshop on Practical Aspects of Automated Reasoning Co-located with the 11th International Joint Conference on Automated Reasoning (FLoC/IJCAR 2022), Haifa, Israel, August, 11 - 12, 2022. CEUR Workshop Proceedings, vol. 3201. CEUR-WS.org (2022)
6. Bromberger, M., Leutgeb, L., Weidenbach, C.: An efficient subsumption test pipeline for bs(lra) clauses. In: Blanchette, J., Kovacs, L., Pattinson, D. (eds.) Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, Proceedings. LNCS, vol. 13385, pp. 147–168. Springer (2022)
7. Bromberger, M., Schwarz, S., Weidenbach, C.: Exploring partial models with SCL. In: Konev, B., Schon, C., Steen, A. (eds.) Proceedings of the Workshop on Practical Aspects of Automated Reasoning Co-located with the 11th International Joint Conference on Automated Reasoning (FLoC/IJCAR 2022), Haifa, Israel, August, 11 - 12, 2022. CEUR Workshop Proceedings, vol. 3201 (2022)
8. Bromberger, M., Schwarz, S., Weidenbach, C.: Scl(fol) revisited (2023). <https://doi.org/10.48550/ARXIV.2302.05954>, <https://arxiv.org/abs/2302.05954>
9. Desharnais, M.: SCL: Simple clause learning https://bitbucket.org/isafol/isafol/src/master/Simple_Clause_Learning, Formal proof development
10. Fiori, A., Weidenbach, C.: Scl clause learning from simple models. In: Fontaine, P. (ed.) 27th International Conference on Automated Deduction, CADE-27. LNAI, vol. 11716. Springer (2019)
11. Knuth, D.E., Bendix, P.B.: Simple word problems in universal algebras. In: Leech, I. (ed.) Computational Problems in Abstract Algebra, pp. 263–297. Pergamon Press (1970)
12. Korovin, K.: iprover - an instantiation-based theorem prover for first-order logic (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5195, pp. 292–298. Springer (2008)
13. Kovács, L., Voronkov, A.: First-order theorem proving and vampire. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8044, pp. 1–35. Springer (2013)
14. Leidinger, H., Weidenbach, C.: SCL(EQ): SCL for first-order logic with equality. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13385, pp. 228–247. Springer (2022). https://doi.org/10.1007/978-3-031-10769-6_14
15. Nieuwenhuis, R., Rubio, A.: Paramodulation-based theorem proving. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. I, chap. 7, pp. 371–443. Elsevier (2001)
16. Schlichtkrull, A., Blanchette, J.C., Traytel, D., Waldmann, U.: Formalization of bachmair and ganzinger’s ordered resolution prover. Archive of Formal Proofs (January 2018), https://isa-afp.org/entries/Ordered_Resolution_Prover.html, Formal proof development

17. Schulz, S., Cruanes, S., Vukmirović, P.: Faster, higher, stronger: E 2.3. In: Fontaine, P. (ed.) Proc. of the 27th CADE, Natal, Brasil. pp. 495–507. No. 11716 in LNAI, Springer (2019)
18. Waldmann, U., Turret, S., Robillard, S., Blanchette, J.: A comprehensive framework for saturation theorem proving. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12166, pp. 316–334. Springer (2020). https://doi.org/10.1007/978-3-030-51074-9_18
19. Weidenbach, C., Dimova, D., Fietzke, A., Suda, M., Wischniewski, P.: Spass version 3.5. In: Schmidt, R.A. (ed.) 22nd International Conference on Automated Deduction (CADE-22). Lecture Notes in Artificial Intelligence, vol. 5663, pp. 140–145. Springer, Montreal, Canada (August 2009)

Appendix

Lemma 6 (SUP-MO Applicability)

Let N be a set of ground clauses. If N has a minimal false clause $C \neq \perp$, then there exists exactly one rule applicable to N according to the SUP-MO strategy.

Proof. For this proof, we make a case distinction:

Case 1: C is the minimal false clause in N , and it has a maximal literal L that is negative. The strategy claims that in this case there always exists a clause $D \in N$, with $D \prec C$, a strictly maximal literal $\text{comp}(L)$, and $\delta_D = \{\text{comp}(L)\}$. Note that $N_C \cup \delta_C \models L$ if $\text{comp}(L) \notin N_C \cup \delta_C$, which also would imply that $N_C \cup \delta_C \models C$. Therefore, $\text{comp}(L) \in N_C \cup \delta_C$. Moreover, we know that $\text{comp}(L) \in N_C$ because the $\delta_C = \emptyset$ for every clause C with a maximal literal that is negative. This means that, according to definition of the superposition model operator, there exists a clause $D \in N$, with $D \prec C$, a strictly maximal literal $\text{comp}(L)$, and $\delta_D = \{\text{comp}(L)\}$. These properties also satisfy the conditions of the rule Superposition Left so the strategy rightfully claims that it is applicable.

Case 2: C is the minimal false clause in N , and it has a maximal literal L that is positive. The strategy claims that in this case L is not strictly maximal in C and therefore the rule Factoring is applicable. Assume to the contrary that L is in fact strictly maximal, so $C = C' \vee L$ and $C' \prec \{L\}$. However, this also means that $N_C \not\models C'$ and, according to definition of the superposition model operator, $\delta_C = \{L\}$ because L is strictly maximal. Hence, $N_C \cup \delta_C \models C$, which is the contradiction we were aiming for. \square

Lemma 13 (Initial SCL State Simulates Initial Superposition State)

The initial SCL state $(\epsilon; N^0; \emptyset; \beta; 0; \top)_{(0, \perp, \gamma)}$ simulates the initial superposition state N^0 and the model construction upto $N_{\perp}^0 \cup \delta_{\perp}$

Proof. **(i)** – **(iv)** Trivially true because the set of learned clauses is empty, the superposition state N^0 is the same as the input clause set, β is sufficiently large by definition, and γ is the identity function.

- (v) – (xiv) Trivially true because both the trail and $N_{\perp}^0 \cup \delta_{\perp}$ are empty, no atom is smaller than the minimal atom, and no clause is less or equal than \perp .

□

Lemma 14 (SCL-SUP Preserves Simulation)

Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,C,\gamma)}$ simulate the superposition state N^i and the corresponding model construction upto $N_{C'}^i \cup \delta_{C'}$, where $C' = \gamma(C)$. Let the SCL state $S' = (\Gamma'; N^0; U'; \beta; k'; E')_{(j,D,\gamma')}$ be the result of one atomic sequence of SCL-SUP steps. Then there exists a clause $D' \in N^j$ with $\gamma'(D) = D'$ and S' simulates the superposition state N^j and the model construction upto $N_{D'}^j \cup \delta_{D'}$.

Proof. We start with the 3 cases of atomic sequences according to Def. 8. This means we can make the following additional assumptions due to Def. 8: $E = \top$; $U' = U$; D is the next largest clause from C in the ordering \prec_{γ} with respect to the ground clause set $N^0 \cup U$; L is the maximal literal of D ; $[\neg A_1, \neg A_2, \dots, \neg A_n]$ are all negative literals such that for all i we have $A_i \prec L$, all A_i undefined in Γ , A_i occurs in $N^0 \cup U$, and $A_i \prec A_{i+1}$. Due to S simulates N^i , we can assume that there exists a clause $D^* = \gamma(D)$ in N^i such that $\text{sfac}(D) = \text{sfac}(D^*)$ (by Def. 7(ii)). Moreover, $j_0 + 1$ is the number of occurrences of L in D^* .

Case (1)→(2a): We additionally know by Def. 8 that $L = B$ is a positive literal, $\Gamma'' = \Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, \Gamma'' \not\models D, \Gamma' = \Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{k+n+1}$, γ' is the same as γ except that $\gamma'(D) = \text{sfac}(D)$, $E' = \top$, $k' = k + n + 1$, $j = i + j_0$, and Conflict is not applicable to S' . Due to S simulates N^i , we also know that D is the smallest clause D_2 in $N^0 \cup U$ wrt. \prec_{γ} such that $D_2 \models D$ because otherwise $\Gamma \models D_2$ would hold (due to Def. 7(xii)), and thus $\Gamma'' \models D$, which contradicts the assumption in step (2a) that $\Gamma'' \not\models D$. This also means that D is the smallest clause D_2 in $N^0 \cup U$ wrt. \prec_{γ} such that $\text{sfac}(D_2) = \text{sfac}(D)$. Analogously, there exists no clause $D^{**} \in N^i$ with $\text{sfac}(D) = \text{sfac}(D^{**})$ and less occurrences of L than D^* because D is the smallest clause in $N^0 \cup U$ wrt. \prec_{γ} such that $D_2 \models D$ and due to Def. 7(iv) $D^* = \gamma(D) \preceq D^{**}$ must hold. Moreover, for $m = 0, \dots, j_0 - 1$ we know that the minimal false clause in N^{i+m} is $D_m^* = D_0 \vee L_1 \vee \dots \vee L_m$, where $D^* = D_0 \vee L_1 \vee \dots \vee L_{j_0}$, $L_1 = \dots = L_{j_0} = L$, and $N^{i+m+1} = N^{i+m} \cup \{D_{m+1}^*\}$. This holds by the following inductive argument for m : (a) $C' \prec D_m^* \prec D_{m-1}^* \prec \dots \prec D_0^*$ by definition of D_m^* ; (b) $\delta_{D_m^*} = \emptyset$ because L is not strictly maximal in D_m^* ; (c) $N_{D_m^*}^{i+m} = N_{C'}^i \cup \delta_{C'}$ due to (a) and (b); (d) $\Gamma'' \not\models D_m^*$ because $\Gamma'' \not\models D$, $D_m^* \models D$ as $\text{sfac}(D) = \text{sfac}(D_m^*)$, (e) $N_{D_m^*}^{i+m} \cup \delta_{D_m^*} \not\models D_m^*$ because of (b),(c),(d), S simulates $N_{C'}^i \cup \delta_{C'}$, and none of the atoms $A_1^{k+1}, \dots, A_n^{k+n}$ appear positively in D_m^* ; (f) D_0^* is the minimal false clause in N^i because of (e) and Def. 7(iv),(xii); (g) if D_m^* is the minimal false clause in N^{i+m} , then SUP-MO applies Factoring to D_m^*

- in N^{i+m} which results in $N^{i+m+1} = N^{i+m} \cup \{D_{m+1}^*\}$; (h) D_m^* is the minimal false clause in N^{i+m} because of (a),(e),(f),(g), and by induction. (j) $D' = \text{sfac}(D) \in N^j$ because of (g) and induction.
- (i) : $\text{atom}(N^{i+m+1}) = \text{atom}(N^0) = \text{atom}(N^0 \cup U)$ holds because $\text{atom}(N^i) = \text{atom}(N^0) = \text{atom}(N^0 \cup U)$ (since S simulates N^i), $\text{atom}(N^{i+m+1}) = \text{atom}(N^{i+m}) \cup \text{atom}(D_m^*)$ for $m = 0, \dots, j_0 - 1$, and $\text{atom}(D_m^*) = \text{atom}(D) \subseteq (\text{atom}(N^0 \cup U))$ (since $\text{sfac}(D) = \text{sfac}(D_m^*)$); $D \in N^0 \cup U$ due to assumptions derived from Def. 8; the rest follows directly from S simulates N^i .
- (ii) : $\text{sfac}(N^0 \cup U') \subseteq \text{sfac}(N^j)$ holds because $U' = U$, $N^i \subseteq N^j$, and $\text{sfac}(N^0 \cup U) \subseteq \text{sfac}(N^i)$ (due to S satisfies Def. 7(ii)). The rest holds for all $C^* \in (N^0 \cup U') \setminus D$ $\gamma'(C^*) \in N^j$ because $\gamma'(C^*) = \gamma(C^*)$ for $C^* \neq D$ and S satisfies Def. 7(ii). $\gamma'(D) \in N^j$ and $\gamma'(D) = \text{sfac}(D)$ for all $C \in (N^0 \cup U') \setminus D$ because $\gamma'(D) = \text{sfac}(D)$ by definition Def. 8 and $\text{sfac}(D) \in N^j$ by argument (j) of the inductive subproof. This also means that $\gamma'(D) = \text{sfac}(D)$.
- (iii) : Holds for all $C^* \in N^i$ because S satisfies Def. 7(iii) and S' Def. 7(ii). Holds for all $C^* \in N^j \setminus N^i$ because $\text{sfac}(C^*) = \text{sfac}(D^*) \in N^i$.
- (iv) : Holds for all $C^* \in N^i$ because S satisfies Def. 7(iv), S' Def. 7(ii), and $\gamma'(C^{**}) \preceq \gamma(C^{**})$ for all clauses C^{**} . Holds for all $C^* \in N^j \setminus N^i$ because $\text{sfac}(C^*) = \text{sfac}(D^*) = \text{sfac}(D)$ and $\gamma'(D) = \text{sfac}(D^*)$.
- (v),(viii),(ix) : Holds for all atoms $A \in \Gamma$ and $A \in N_{C'}^j \cup \delta_{C'} = N_{D'}^j \cup \delta_{D'}$ because S satisfies Def. 7(v),(viii),(ix), $\gamma'(C^{**}) \preceq \gamma(C^{**})$ for all clauses C^{**} , and S' satisfies Def. 7(ii), so if $\gamma(C^{**}) = \text{sfac}(C^{**})$, then $\gamma'(C^{**}) = \text{sfac}(C^{**})$. The only other positive atom in Γ' is B and it is in $N_{D'}^j \cup \delta_{D'}$ because we can deduce from the above properties proven for the D_m^* that $D_{j_0}^* \in N^j$, $D' = \gamma(D^*) = \text{sfac}(D^*) = D_{j_0}^*$, $D_{j_0}^*$ is the next larger clause to C' , $D_{j_0}^*$ is not satisfied by $N_{C'}^j \cup \delta_{C'}$, B is the strictly maximal literal in $D_{j_0}^* = \text{sfac}(D^*)$, and thus $D_{j_0}^*$ produces B . Hence, B is the only atom in $N_{D'}^j \cup \delta_{D'} \setminus (N_{C'}^j \cup \delta_{C'})$.
- (vi) : We first prove that for all $A \in \text{atom}(N^0)$ with $A \prec L$ and $\neg A \in \Gamma'$ it must hold that $A \notin N_{D'}^j \cup \delta_{D'}$. This is the case because $N_{D'}^j \cup \delta_{D'}$ contains exactly all positive atoms in Γ' (see the previous proof step for property (v)). Next we prove that for all $A \in \text{atom}(N^0)$ with $A \prec L$ and $A \notin N_{D'}^j \cup \delta_{D'}$ it must hold that $\neg A \in \Gamma'$. Let L' be the maximal literal in C (and therefore also C'). Then Γ contains $\neg A$ for all literals $A \in \text{atom}(N^0)$ with $A \prec L' \preceq L$ and $A \notin N_{C'}^j \cup \delta_{C'} = N_{D'}^j \cup \delta_{D'}$ because S satisfies Def. 7(vi). The other $\neg A$ with $A \in \text{atom}(N^0)$, $A \prec L$, and $A \notin (N_{D'}^j \cup \delta_{D'})$ are in Γ' due to the sequence of decisions $\neg A_1^{k+1}, \dots, \neg A_n^{k+n}$ since this sequence includes all undefined atoms in Γ with $A \in \text{atom}(N^0)$ and $A \prec L$.
- (vii) : All atoms A with $A \prec L'$, where L' is the maximal literal in C' are already defined in Γ (because S satisfies Def. 7(v),(vi)), all atoms in Γ are defined in ascending order of \prec (because S satisfies Def. 7(vii)), and the only atom A with $L' \preceq A$ that can be defined in Γ is the atom of L' . Hence, all atoms in Γ are ordered and contain no gaps. By Def. 8, the atoms $\neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{k+n+1}$ appended to Γ are therefore all

larger than the atoms in Γ , ordered according to \prec , and contain no gaps. Thus, the property also holds for S' .

- (x)-(xi) : trivially satisfied because Γ contains only decisions as it satisfies Def. 7(x) and SCL-SUP only adds decisions between S and S' .
- (xii) : First of all, $\Gamma' \models D$ because B is in Γ' . Let $C^* \in N^0 \cup U$ be such that $\gamma'(C^*) \prec \gamma'(D)$. Then $\gamma(C^*) \preceq \gamma(D)$ because $\gamma'(C^*) = \gamma(C^*)$ and $\gamma'(D) \preceq \gamma(D)$. Therefore, $\Gamma \models C^*$ because S satisfies Def. 7(xii), which also means that $\Gamma' \models C^*$ because Γ' is just an extension of Γ .
- (xiii) : Conflict is not applicable to S' by Def. 8.
- (xiv) : $\perp \notin N^0 \cup U$ because S satisfies Def. 7(xiv), $E' = \top \neq \perp$, and $\perp \notin N^j$ because S' satisfies Def. 7(iv).

Case (1)→(2b): We additionally know by Def. 8 that $L = B$ is a positive literal, $\Gamma'' = \Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, \Gamma'' \not\models D, \Gamma' = \Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{k+n+1}$, γ' is the same as γ except that $\gamma'(D) = \text{sfac}(D)$, E' is the smallest clause in $N^0 \cup U$ wrt. \prec_γ that is false wrt. $\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}, B^{\text{sfac}(D)}$, $k' = k + n$, and $j = i + j_0$. Moreover, we can use the same deductions as in case (1)→(2a) to show properties (a)-(j) and for $m = 0, \dots, j_0 - 1$ that the minimal false clause in N^{i+m} is $D_m^* = D_0 \vee L_1 \vee \dots \vee L_m$, where $D^* = D_0 \vee L_1 \vee \dots \vee L_{j_0}$, $L_1 = \dots = L_{j_0} = L$, and $N^{i+m+1} = N^{i+m} \cup \{D_{m+1}^*\}$.

(i)-(ix),(xii),(xiv) : Work the same as for case (1)→(2a).

(x),(xiii) : trivially satisfied because $E' \notin \{\top, \perp\}$.

(xi) : $B^{\text{sfac}(D)}$ is the topmost literal in Γ' by Def. 8; $\Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}$ contains only decisions because S satisfies Def. 7(x); there must exist a clause $E^* \in N^j$ such that $\gamma(E') = E' = E^*$ because S satisfies Def. 7(ii); $\neg B$ is the largest literal in E^* because S' satisfies Def. 7(xii) and so Γ' satisfies all clauses with maximal literal $\preceq B$; $N_{D'}^j \cup \delta_{D'} = N_{E^*}^j \cup \delta_{E^*}$ because $B \in N_{D'}^j \cup \delta_{D'}$ is the largest atom in $N_{D'}^j \cup \delta_{D'}$ and because $\neg B$ is the largest literal in E^* the model operator does not produce any other atoms between D' and E^* ; E^* is false in $N_{E^*}^j \cup \delta_{E^*}$ because E' is false in Γ' , $E^* = E'$, and S' satisfies Def. 7(v),(vi); there is no smaller clause $E_2^* \in N^j$ false in $N_{E^*}^j \cup \delta_{E^*}$ because S' satisfies Def. 7(iv), so $N^0 \cup U$ would contain a clause E_2' such that $E_2' \models E_2^*$ and $\gamma(E_2') \preceq E_2^* \prec E'$ which contradicts our assumption that E' is the smallest false clause in Γ' ; hence, E' is the minimal false clause in N^j .

(xiv) : $\perp \notin N^0 \cup U$ because S satisfies Def. 7(xiv), $E' \neq \perp$ because it has a maximal literal, and $\perp \notin N^j$ because S' satisfies Def. 7(iv).

Case (1)→(2c): We additionally know by Def. 8 that $\Gamma' = \Gamma, \neg A_1^{k+1}, \dots, \neg A_n^{k+n}$, $\Gamma' \models D$, $\gamma' = \gamma$, $D' = \gamma(D)$, $E' = \top$, $k' = k + n$, and $j = i$.

(i)-(iv),(vi),(vii),(xi),(xiv) : Work the same as for case (1)→(2a).

(v),(viii),(ix) : Holds for all atoms $A \in \Gamma$ and $A \in N_{C'}^j \cup \delta_{C'} = N_{C'}^i \cup \delta_{C'}$ because S satisfies Def. 7(v),(viii),(ix). There is no further positive atom in Γ' because SCL-SUP only adds negative literals between S and S' to the trail. The other clauses D^* between C' and D' also produce no positive atom in the superposition model construction due to the following reasons. The clause D' is not productive in the superposition model construction if its maximal literal is negative (see Def. 3). If the maximal

literal of D' is positive, then $\Gamma' \models D'$ or one of the other two atomic sequences of Def. 8 were applicable. Since the atoms $A_1^{k+1}, \dots, A_n^{k+n}$ appear if at all negatively in D and thus D' and since S simulates $N_{C'}^i \cup \delta_{C'}$ and $C' \prec D'$, it follows that $N_{D'}^i \models D'$. Therefore, D' is in case (1)→(2c) never productive. The other clauses D^* between C' and D' (i.e. with $\gamma(C) = C' \prec D^* \prec D' = \gamma(D)$) are also not productive because for every $D^* \in N^i$ there exists a $D_2 \in N^0 \cup U$ with $\gamma(D_2) \preceq \gamma(D^*)$ and $D_2 \models D^*$ (since S satisfies Def. 7(iv)). From $\gamma(D_2) \preceq \gamma(D^*)$ and $\gamma(C) = C' \prec D^* \prec D' = \gamma(D)$ follows that $\gamma(D_2) \preceq \gamma(C)$. Therefore, $N_{D^*}^i \models D^*$ because $\Gamma \models D_2$ (since S satisfies Def. 7(xii)) and S simulates $N_{C'}^i \cup \delta_{C'}$. Hence, the other clauses D^* between C' and D' are also not productive.

- (x) : Holds trivially because $E = \top$ and Γ' contains only decided literals.
- (xii) : $\Gamma' \models \Gamma$ because Γ' is an extension of Γ ; $\Gamma \models C^*$ for all $C^* \in N^0 \cup U$ with $C^* \prec_\gamma D$ because S satisfies Def. 7(xii) so $\Gamma' \models C^*$ for all $C^* \in N^0 \cup U$ with $C^* \prec C$; $\Gamma' \models D$ because either the maximal literal L of D was negative and therefore added to Γ' by step (1) of the sequence or the maximal literal L of D was positive and $\Gamma' \models D$ holds because otherwise D would satisfy the preconditions of step (2a) or (2b).
- (xiii) : Proof by contradiction. Suppose there is a clause E_2 that is not false in S , but false in S' . Then the maximal literal in E_2 is positive and one of the atoms A_i . However, this would also mean that E_2 is smaller than D and therefore $E_2 \preceq C$. This contradicts the fact that S satisfies Def. 7(xii).

We continue with the 4 cases of atomic sequences according to Def. 10. This means we can make the following additional assumptions due to Def. 10: $E \notin \{\top, \perp\}$, $L = \neg B$ is the maximal literal of E , $\Gamma = \Gamma_0 B^{\text{sfac}(C)}$, Γ_0 contains only decision literals, all literals in Γ_0 are smaller than B , $\text{sfac}(C) = C_1 \vee B$ with $C_1 \prec \{B\}$ because B was propagated from $\text{sfac}(C)$, $E = E_0 \vee E_1$, where E_1 contains all occurrences of L in E , j_0 is the number of occurrences of L in E , and $E_2 = E_0 \vee C_1 \vee \dots \vee C_{j_0}$, where $C_1 = \dots = C_{j_0}$. Due to S simulates N^i and Def. 7(xi), we can assume that $E \in N^i$, $\gamma(E) = E$, E is the minimal false clause in N^i . By Def. 7(iii),(viii), there exists a clause $C' \in N^i$ with $C' = \text{sfac}(C') = \text{sfac}(C) = \gamma(C)$, $C' = C'_1 \vee B$, and $\delta_{C'} = \{B\}$. Moreover, we can show for $m = 0, \dots, j_0 - 1$ that $N^{i+m+1} = N^{i+m} \cup E_{m+1}^*$, that E_m^* is the minimal false clause in N^{i+m} , and that E_{m+1}^* is the result of applying Superposition Left to E_m^* and C' , where $E_m^* = E_0 \vee C_1 \vee \dots \vee C_m \vee L_{m+1} \vee \dots \vee L_{j_0}$, $E = E_0 \vee L_1 \vee \dots \vee L_{j_0}$, $C_1 = \dots = C_{j_0}$, and $L_1 = \dots = L_{j_0} = L$. This holds because of induction: (a2) By induction hypothesis we can assume that E_m^* is the minimal false clause in N^i ; (b2) by Def. 5, the SUP-MO strategy reaches N^{i+m+1} by applying Superposition Left to E_m^* and C' , which adds the clause E_{m+1}^* to N^{i+m} ; (c2) $E_{m+1}^* \prec E_m^*$ (for $j_0 > m \geq 0$) and $C' \prec E_m^*$ (for $j_0 > m \geq 0$) because $C_m \prec L_m$; (d2) for $m+1 < j_0$, E_{m+1}^* is equivalent to $C_1 \vee E_0 \vee L$ modulo duplicate literals, L is negative, $C_1 \prec L$, $E_0 \prec L$, and therefore E_{m+1}^* is never productive; (e2) due to (c2) and (d2), $N_{E_{m+1}^*}^{i+m+1} \cup \delta_{E_{m+1}^*} = N_{C'}^{i+m+1} \cup \delta_{C'} = N_{C'}^i \cup \delta_{C'}$ for

$m + 1 < j_0$. (f2) due to (a2), (d2), and (e2), E_{m+1}^* is false in $N_{E_{m+1}^*}^{i+m+1} \cup \delta_{E_{m+1}^*}$ for $m + 1 < j_0$. (g2) due to (c2) and (f2), E_{m+1}^* is also the minimal false clause in N^{i+m+1} ; (h2) $E_{j_0}^* = E_2$.

Case (1)→(2a): We additionally know by Def. 10 that $\Gamma' = \epsilon$, $U' = U$, $\gamma' = \gamma$, $D = D' = \perp$, $E' = E_2 = E_0 \vee C_1 \vee \dots \vee C_1 = E_{j_0}^* = \perp$, and therefore $C_1 = \perp$ and $E_2 = \perp$.

(i) : Holds because $D = \perp$, $U' = U \cup \{E_2\}$, $N^j = N^i \cup \{E_1^*, \dots, E_{j_0}^*\}$, $\text{atom}(E_1^*) = \dots = \text{atom}(E_{j_0-1}^*) = \text{atom}(E_0^*) \cup \text{atom}(C')$, $\text{atom}(E_{j_0}^*) = \text{atom}(E_2) \subseteq \text{atom}(E_0) \cup \text{atom}(C)$, $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$ because S satisfies Def. 7(i), $\text{atom}(E_0) \cup \text{atom}(C) \cup \text{atom}(E_0^*) \cup \text{atom}(C') \subseteq \text{atom}(N^0)$ because $E, C \in N^0 \cup U$, $E^*, C' \in N^i$ and $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$.

(ii) : $\text{sfac}(N^0 \cup U) \subseteq \text{sfac}(N^i) \subseteq \text{sfac}(N^j)$ because S satisfies Def. 7(ii) and $U = U'$. The rest holds because $N^0 \cup U \subseteq N^i$, $\gamma' = \gamma$, and S satisfies Def. 7(ii).

(iii)-(iv) : Holds for all clauses in N^i and $N^0 \cup U$ because S satisfies Def. 7(iii),(iv) and $\gamma' = \gamma$. Holds for $\{E_1^*, \dots, E_{j_0}^*\} = N^j \setminus N^i$ because $E' = E_{j_0}^* = \gamma(E')$, and for $m = 1, \dots, j_0$ the maximum literal in E_m^* is L and negative, $E' \models E_m^*$, and $\gamma(E') = E' \preceq E_m^*$. This holds because the clause E_m^* has the form $E_0 \vee C_1 \vee \dots \vee C_1 \vee L \vee \dots \vee L$, $E_0 \prec L$ and $C_1 \prec L$, E' has the form $E_0 \vee C_1 \vee \dots \vee C_1$, so $E' \models E_m^*$ and $\gamma(E') = E' \preceq E_m^*$.

(v)-(ix) : Holds trivially because $\Gamma' = \epsilon$ and $D = \perp$.

(x)-(xi),(xiii) : Holds trivially because $E = \perp$ and $\Gamma' = \epsilon$.

(xii) : Holds because S satisfies Def. 7(xiv) and therefore $\perp \notin N^0 \cup U$.

(xiv) : $\perp \notin N^0 \cup U$ because S satisfies Def. 7(xiv); the rest holds because $E' = E_{j_0}^* = \perp$ and thus $\perp \in N^j$.

Case (1)→(4a): We additionally know by Def. 10 that $E' = E_2$, $U' = U \cup \{E_2\}$, $\gamma' = \gamma$, the maximal literal $L_1' = \neg B_2$ of E' is negative, Γ_1 is the prefix of Γ_0 that defines all atoms $A \prec B_2$, D is the smallest clause in $N^0 \cup U$ wrt. \prec_γ with maximum literal B_2 and $\Gamma_1 \not\models D$, $\Gamma' = \Gamma_1$, $B_2^{\text{sfac}(D)}$, k' is the size of Γ_1 , $\Gamma_1, B_2^{k'+1}$ is a prefix of Γ_0 , and because S satisfies Def. 7(viii),(ii) there exists a $D' \in N^i \subseteq N^j$ with $\gamma(D) = \text{sfac}(D) = \text{sfac}(D') = D'$, and $\delta_{D'} = \{B\}$. Moreover, $N_{C^*}^j = N_{C^*}^i$ for all $C^* \preceq D'$ because $D' \prec \gamma(E_2)$.

(i) : Holds because $U' = U \cup \{E_2\}$, $D = E_2 \in U'$, $N^j = N^i \cup \{E_1^*, \dots, E_{j_0}^*\}$, $\text{atom}(E_1^*) = \dots = \text{atom}(E_{j_0-1}^*) = \text{atom}(E_0^*) \cup \text{atom}(C')$, $\text{atom}(E_{j_0}^*) = \text{atom}(E_2) \subseteq \text{atom}(E_0) \cup \text{atom}(C)$, $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$ because S satisfies Def. 7(i), $\text{atom}(E_0) \cup \text{atom}(C) \cup \text{atom}(E_0^*) \cup \text{atom}(C') \subseteq \text{atom}(N^0)$ because $E, C \in N^0 \cup U$, $E, C' \in N^i$ and $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$.

(ii) : $\text{sfac}(N^0 \cup U) \subseteq \text{sfac}(N^i) \subseteq \text{sfac}(N^j)$ because S satisfies Def. 7(i). $\text{sfac}(E_2) \in \text{sfac}(N^j)$ because $E_{j_0}^* = E_2 \in N^j$.

(iii)-(iv) : Holds for all clauses in N^i and $N^0 \cup U$ because S satisfies Def. 7(iii),(iv) and $\gamma' = \gamma$. Holds for $\{E_1^*, \dots, E_{j_0}^*\} = N^j \setminus N^i$ and E_2 due

to (c2) and the definition of E_m^* because $E_2 = E_{j_0}^*$, the maximal literal in E_m^* is negative for $m = 1, \dots, j_0 - 1$, and $E_2 \models E_m^*$ and $E_2 \prec E_m^*$ for $m = 1, \dots, j_0$.

- (v)-(ix) : Holds because $\gamma' = \gamma$, S satisfies Def. 7(v)-(ix), Γ' is a prefix of Γ (except for the annotation of the last literal), Γ' defines by definition of the strategy all atoms $A \preceq B_2$, and any producing clause $\preceq D'$ is in N^i and any propagating clause $\preceq_\gamma D$ is in $N^0 \cup U$.
- (x) : Holds trivially because $E \notin \{\top, \perp\}$.
- (xi) : Holds because $\Gamma' = \Gamma_1, B_2^{\text{sfac}(D)}$; Γ' contains only decisions because it is a prefix of Γ_0 , which contains only decisions; N^j contains $E_{j_0}^* = E_2 = \gamma(E_2)$; and E_2 is the minimal false clause in N^j because E_2 is the smallest clause added to N^j , $E_2 \prec E'$, E' was the minimal false clause in N^i with respect to Γ , and Γ' is only a prefix of Γ .
- (xii) : Holds because S satisfies Def. 7(xii), $D \prec_\gamma C$, Γ' is a prefix of Γ , Γ' defines by definition of the strategy all atoms $A \preceq B_2$, Therefore, Γ' always satisfies D . Γ' also satisfies all clauses $C^* \in N^0 \cup U'$ with $C^* \prec D$ because all those clauses were already in $N^0 \cup U$, Γ satisfied them (since S satisfies Def. 7(xii)), and Γ' is a prefix of Γ that defines all atoms $\preceq B_2$.
- (xiii) : Conflict is not applicable because $E' \neq \top$.
- (xiv) : $\perp \notin N^0 \cup U$ because S satisfies Def. 7(xiv); the rest holds because $E_2 = E' \neq \perp$, $\perp \notin N^i$, and $E_m^* \neq \perp$ (for $j_0 \geq m \geq 0$).

Case (1)→(4b): We additionally know by Def. 10 that $E' = \top$, $U' = U \cup \{E_2\}$, the maximal literal $L'_1 = B_2$ of E_2 is positive and occurs $j_2 + 1$ times in E_2 , $D = E_2$, γ' is the same as γ except that $\gamma'(E_2) = \text{sfac}(E_2)$, Γ_1 is the prefix of Γ_0 that defines all atoms $A \prec L'_1$, $\Gamma' = \Gamma_1, B_2^{k'}$, k' is the size of Γ' , $j_1 = i + j_0$, $j = j_1 + j_2$, and Conflict is not applicable to S' . Moreover, we can use the same deductions as in case (1)→(2a) for Def. 8 to show properties (a)-(j) and for $n = 0, \dots, j_2 - 1$ that the minimal false clause in N^{j_1+n} is $D_n^* = D_0 \vee L'_1 \vee \dots \vee L'_n$, where $E_2 = D_0 \vee L'_1 \vee \dots \vee L'_{j_2}$, $L'_1 = \dots = L'_{j_2}$, and $N^{j_1+n} = N^{j_1+n} \cup \{D_{n+1}^*\}$. Therefore, $D' = \text{sfac}(E_2) = D_{j_2}^*$ and $D' \in N^j$.

- (i) : Note that $U' = U \cup \{E_2\}$, $D = E_2 \in U'$, $N^j = N^i \cup \{E_1^*, \dots, E_{j_0}^*\} \cup \{D_1^*, \dots, D_{j_2}^*\}$, $\text{atom}(E_1^*) = \dots = \text{atom}(E_{j_0-1}^*) = \text{atom}(E_0) \cup \text{atom}(C)$, $\text{atom}(E_{j_0}^*) = \text{atom}(E_2) \subseteq \text{atom}(E_0) \cup \text{atom}(C)$, $\text{atom}(D_1^*) = \dots = \text{atom}(D_{j_2}^*) = \text{atom}(E_2)$, $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$ because S satisfies Def. 7(i), $\text{atom}(E_0) \cup \text{atom}(C) \subseteq \text{atom}(N^0)$ because $E, C \in N^0 \cup U$, $E, \text{sfac}(C') \in N^i$ and $\text{atom}(N^0) = \text{atom}(N^i) = \text{atom}(N^0 \cup U)$.
- (ii) : $\text{sfac}(N^0 \cup U) \subseteq \text{sfac}(N^i) \subseteq \text{sfac}(N^j)$ because S satisfies Def. 7(i). $\text{sfac}(E_2) \in \text{sfac}(N^j)$ because $E_{j_0}^* = E_2 \in N^j$. $\gamma'(E_2) = \text{sfac}(E_2)$ and otherwise γ' is identical to γ .
- (iii)-(iv) : Note that $N^j = N^i \cup \{E_1^*, \dots, E_{j_0}^*\} \cup \{D_1^*, \dots, D_{j_2}^*\}$. The properties hold for all clauses in N^i and $N^0 \cup U$ because S satisfies Def. 7(iii),(iv) and $\gamma' = \gamma$. The properties hold for $\{E_1^*, \dots, E_{j_0}^*\}$ and E_2 due to (c2) and the definition of E_m^* because $E_2 = E_{j_0}^*$, the maximal literal in E_m^* is negative for $m = 1, \dots, j_0 - 1$, and $E_2 \models E_m^*$ and $E_2 \prec E_m^*$ for $m = 1, \dots, j_0$.

The properties hold for $\{D_1^*, \dots, D_{j_2}^*\}$ because $\gamma'(E_2) = \text{sfac}(E_2) = D_n^*$, for $n = 1, \dots, j_2$, and therefore also $E_2 \models D_n^*$ and $\gamma'(E_2) \prec D_n^*$ for $n = 1, \dots, j_2$.

- (v)-(ix) : Holds because S satisfies Def. 7(v)-(ix), Γ_1 is a prefix of Γ , Γ_1 defines by definition of the strategy all atoms $A \prec L'_1$, therefore any producing clause $\prec D' = \text{sfac}(E_2)$ is already in N^i and any propagating clause $\prec_{\gamma'} D$ in $N^0 \cup U$. The only additional atom in $\Gamma' = \Gamma_1, B_2^{k'}$ and $N_{D'}^j \cup \delta_{D'}$ is B_2 and it is produced by D' and can be propagated by D . Moreover, $D' = \text{sfac}(D') = \text{sfac}(D) = \gamma'(D)$.
- (x) : Holds because $\Gamma' = \Gamma_1, B_2^{k'}$, Γ_1 is a prefix of Γ_0 , S satisfies Def. 7(xi), and therefore Γ_0 and Γ_1 only contain decisions.
- (xi) : Holds trivially because $E = \top$.
- (xii) : Holds for $D = E_2$ because B_2 is true in Γ' . Holds for all clauses $C^* \in N^0 \cup U'$ with $C^* \prec_{\gamma'} D$ because S satisfies Def. 7(xii) and so $\Gamma \models C^*$, $\Gamma_1, \text{comp}(B_2)$ is a prefix of Γ , $\Gamma_1, \text{comp}(B_2)$ defines by definition of the strategy all atoms $A \preceq B_2$, the maximal literal in C^* is $A \preceq B_2$, so C^* is true independent of B_2 , which means $\Gamma_1 \models C^*$ and therefore $\Gamma_1, B \models C^*$.
- (xiii) : Conflict is not applicable due to the precondition of Def. 10(4b).
- (xiv) : $\perp \notin N^0 \cup U$ because S satisfies Def. 7(xiv); the rest holds because $E' \neq \perp$, $\perp \notin N^i$, and $E_m^* \neq \perp$ (for $j_0 \geq m \geq 0$).

Case (1)→(4c): We additionally know by Def. 10 that $U' = U \cup \{E_2\}$, the maximal literal $L'_1 = B_2$ of E_2 is positive and occurs $j_2 + 1$ times in E_2 , $D = E_2$, γ' is the same as γ except that $\gamma'(E_2) = \text{sfac}(E_2)$, Γ_1 is the prefix of Γ_0 that defines all atoms $A \prec L'_1$, $\Gamma' = \Gamma_1, B_2^{\text{sfac}(D)}$, k' is the size of Γ' , $j_1 = i + j_0$, $j = j_1 + j_2$, E' is the minimal clause that is false with respect to Γ' , and $\gamma'(E') = E'$ because the maximal literal in E' is negative. Moreover, we can use the same deductions as in case (1)→(2a) for Def. 8 to show properties (a)-(j) and for $n = 0, \dots, j_2 - 1$ that the minimal false clause in N^{j_1+n} is $D_n^* = D_0 \vee L'_1 \vee \dots \vee L'_n$, where $E_2 = D_0 \vee L'_1 \vee \dots \vee L'_{j_2}$, $L'_1 = \dots = L'_{j_2}$, and $N^{j_1+n} = N^{j_1+n} \cup \{D_{n+1}^*\}$. Therefore, $D' = \text{sfac}(E_2) = D_{j_2}^*$ and $D' \in N^j$.

- (i)-(ix),(xii),(xiv) : Work the same as for case (1)→(4b).
- (x) : Holds trivially because $E \neq \top$.
- (xi) : Holds because $\Gamma' = \Gamma_1, B_2^{\text{sfac}(D)}$, Γ_1 is a prefix of Γ_0 , S satisfies Def. 7(xi), and therefore Γ_0 and Γ_1 only contain decisions. $D_{j_2}^* = D' = \text{sfac}(D') = \text{sfac}(D) = \gamma'(D)$ due to properties we have proven for D_n^* ; Γ_1 contains only decisions because Γ_1 is a prefix of Γ_0 and S satisfies Def. 7(x); it must hold that $E' \in N^j$ because $\gamma'(E') = E'$ and S' satisfies Def. 7(ii); $\neg B$ is the largest literal in E' because S' satisfies Def. 7(xii) and so Γ' satisfies all clauses with maximal literal $\preceq B$; $N_{D'}^j \cup \delta_{D'} = N_{E'}^j \cup \delta_{E'}$ because $B_2 \in N_{D'}^j \cup \delta_{D'}$ is the largest atom in $N_{D'}^j \cup \delta_{D'}$ and because $\neg B$ is the largest literal in E' the model operator does not produce any other atoms between D' and E' ; E' is false in $N_{E'}^j \cup \delta_{E'}$ because E' is false in Γ' , and S' satisfies Def. 7(v),(vi); there is no smaller clause $E^{**} \in N^j$ false in $N_{E'}^j \cup \delta_{E'}$ because S' satisfies Def. 7(iv), so $N^0 \cup U'$

would contain a clause E'' such that $E'' \models E^{**}$ and $\gamma(E'') \preceq E^{**} \prec E'$ which contradicts our assumption that E' is the smallest false clause in Γ' ; hence, E' is the minimal false clause in N^j .
(xiii) : Conflict is not applicable because $E \neq \top$.

□

Lemma 15 (SCL-SUP Advances the Simulation)

Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ simulate the superposition state N^i and the model construction upto $N^i_{\gamma(D)} \cup \delta_{\gamma(D)}$. Let the SCL state $S' = (\Gamma'; N^0; U'; \beta'; k'; E')_{(j,D',\gamma')}$ be the next state reachable by one atomic sequence of SCL-SUP steps. Then either $i < j$ or $i = j$ and $\gamma' = \gamma$ and $D \prec_{\gamma} D'$.

Proof. If the atomic sequence is according to Def. 8, then D' is the next larger clause wrt. \prec_{γ} and D . If $\gamma(D') = \text{sfac}(D')$, then $i = j$, $\gamma' = \gamma$, and $D \prec_{\gamma} D'$ hold. Otherwise, SCL-SUP skips some Factoring steps, so $i < j$. If the atomic sequence is according to Def. 10, then by definition the annotated number increases by one for every resolution step in the sequence. Since there is at most one resolution step, $i < j$ always holds. □

Lemma 16 (SCL-SUP Correctness of Final States)

Let the SCL state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,D,\gamma)}$ simulate the superposition state N^i and the model construction upto $N^i_{\gamma(D)} \cup \delta_{\gamma(D)}$. Let there be no more states reachable from S following an atomic sequence of SCL-SUP steps. Then S is a *final state*, i.e., either (i) $E = \perp$, $D = \perp$, $\perp \in N^i$, and N^0 is unsatisfiable or (ii) $\Gamma \models N^0$.

Proof. Proof by case distinction:

$E = \perp$: Then by property Def. 7(xiv) $\perp \in N^i$ and $\Gamma = \epsilon$. Since superposition is sound this means that N^0 is unsatisfiable. Moreover, $D = \perp$ or due to Def. 7(xii) the empty trail would satisfy a non-empty clause, which would be a contradiction. Therefore, this SCL state is a final state.

$E = \top$ and D is the largest clause in $N^0 \cup U$ wrt. \prec_{γ} : $\Gamma \models N^0$ because of properties Def. 7(xii) and (iv). Therefore, this SCL state is a final state.

$E = \top$ and D is not the largest clause in $N^0 \cup U$ wrt. \prec_{γ} : Then an atomic sequence of SCL-SUP steps as described in Def. 8 must fulfill the following preconditions: (a) $E = \top$ and there exists a next largest clause in $N^0 \cup U$ wrt. \prec_{γ} and D , which is automatically fulfilled by the assumptions for the given case; and (b) N^i contains $D' = \gamma(D)$ and $\text{sfac}(D) = \text{sfac}(D')$, which is fulfilled because S satisfies property Def. 7(ii).

$E \notin \{\perp, \top\}$: Then by property Def. 7(xi) $\Gamma = \Gamma', B^{\text{sfac}(D)}$, Γ' contains only decisions, and there exists an $E' \in N^i$ that is the minimal false clause in N^i such that $\gamma(E) = E = E'$ and $\neg B \in E$. By properties Def. 7(v)-(vii), all atoms $A \preceq B$ are defined in Γ in ascending order \prec . By property Def. 7(viii),

there exists a clause $D' \in N^i$ such that $\text{sfac}(D') = D'$, $\gamma(D) = \text{sfac}(D) = D'$, and $\delta_{D'} = \{B\}$. This satisfies all non-trivial preconditions of Def. 10, so the atomic sequence of SCL-SUP steps described in Def. 10 is applicable.

□

Lemma 17 (SCL-SUP is a Regular SCL Strategy)

SCL-SUP is a regular SCL strategy if it is executed on a state $S = (\Gamma; N^0; U; \beta; k; E)_{(i,C,\gamma)}$ that simulates a superposition state N^i and the corresponding model construction upto $N_{\gamma(C)}^i \cup \delta_{\gamma(C)}$.

Proof. Let $S' = (\Gamma'; N^0; U'; \beta; k'; E')_{(j,D',\gamma')}$ be the next state reachable by one atomic sequence of SCL-SUP steps from S . Let L be the largest literal in D' . First note that SCL-SUP only applies the rule Conflict directly after an application of the rule Propagate (see steps Def. 8(2b), Def. 10(4a), and Def. 10(4c)). Therefore, SCL-SUP is a regular SCL strategy if it applies the rule Conflict within the atomic sequence of steps as soon as Conflict is applicable. We prove this by a case distinction:

Case 1: The atomic sequence follows Def. 8. This means Γ' is an extension of Γ , $U = U'$, and the only rules applied are Decide, Propagate, and Conflict. Furthermore, due to Lemma 14, we know that Γ' defines all atoms A with $A \prec L$ (Def. 7(v)-(vii)), Γ' defines none of the atoms A with $L \prec A$ (Def. 7(v)-(vii)), A or $\text{comp}(A)$ is the last literal in Γ' if Γ' defines A (Def. 7(v)-(vii)), and Γ satisfies all clauses with maximum literal $L' \prec L$ (Def. 7(xii)). Moreover, every clause in $N^0 \cup U$ is not false with respect to Γ' if it has a maximum literal L' such that $L \prec \text{atom}(L')$ because at least L' is not yet defined. This also means, that such a clause is not false with respect to any prefix of Γ' . Therefore, the only clauses in $N^0 \cup U$ that can be false with respect to Γ' have maximum literal L or $\text{comp}(L)$ and they are not false in a strict prefix of Γ' as L is not yet defined in a strict prefix of Γ' . Hence, Conflict is only applicable after L is defined, which is also the case when SCL-SUP checks for a false clause and applies Conflict.

Case 2: The atomic sequence follows Def. 10 and ends in case (2a). Then Conflict was never applicable between S and S' because the last argument in the SCL state was always $\neq \top$.

Case 3: The atomic sequence follows Def. 10 and ends in case (4a). Then Conflict was not applicable between steps (1) and (3) of the strategy because the last argument in the SCL state was always $\neq \top$. In part (4a) of the strategy only rules Propagate and Conflict are applied. Moreover, Γ' is a strict prefix of $\Gamma = \Gamma_0$, B (except for the annotation of the last literal on the trail), $U' = U \cup \{E'\}$, B is the top literal in C , $\text{comp}(B)$ is the top literal in E , and the top literal L_1 in E' is smaller than B (as any occurrence $\text{comp}(B)$ was resolved away). Similar to what we have shown for case 1, we know that Γ defines all atoms A with $A \prec B$ (Def. 7(v)-(vii)), Γ defines none of the atoms A with $B \prec A$ (Def. 7(v)-(vii)), B is the last literal in Γ (Def. 7(v)-(vii)), and Γ satisfies all clauses with maximum literal $L' \prec B$ (Def. 7(xii)). Moreover, every clause in $N^0 \cup U$ is not false with

respect to Γ if it has a maximum literal L' such that $B \prec \text{atom}(L')$ because at least L' is not yet defined. This also means, that such a clause is not false with respect to any prefix of Γ . Therefore, the only clauses in $N^0 \cup U$ that can be false with respect to Γ have maximum literal $L = B$ or $\text{comp}(L)$ and there are no false clauses in $N^0 \cup U$ with respect to the strict prefix Γ' of Γ as L is not yet defined in Γ' . Thus, the only clause that can be false with respect to Γ' is the newly learned clause E' and it can only be false after its maximal literal has been defined. Hence, Conflict is only applicable after L_1 is defined, which is also the case when SCL-SUP applies Conflict.

Case 4: The atomic sequence follows Def. 10 and ends in case (4b). Then Conflict was not applicable between steps (1) and (3) of the strategy because the last argument in the SCL state was always $\neq \top$. In part (4b) of the strategy only Decide is applied. Furthermore, due to Def. 7(xiii), Conflict is not applicable in state S' and therefore also not in any state $S^* = (\Gamma^*; N^0; U'; \beta; k'; \top)$ visited in part (4a) of the strategy because S^* is equivalent to S' except that Γ^* is a prefix of Γ' .

Case 5: The atomic sequence follows Def. 10 and ends in case (4c). Then Conflict was not applicable between steps (1) and (3) of the strategy because the last argument in the SCL state was always $\neq \top$. In part (4c) of the strategy only Propagate and Conflict are applied and $\Gamma' = \Gamma_1, B_2^{\text{fac}(D')}$, where Γ_1 is a prefix of Γ_0 and $\Gamma = \Gamma_0, B$ and $B_2 \prec B$. Furthermore, due to Lemma 14, we know that Γ_1 defines all atoms A with $A \prec B_2$ (Def. 7(v)-(vii)), Γ_1 defines none of the atoms A with $B_2 \preceq A$ (Def. 7(v)-(vii)), Γ satisfies all clauses in N^i with maximum literal $L' \preceq B$ (Def. 7(xii)) and therefore Γ_1 falsifies no of the clauses in N^i , and Γ' falsifies none of the clauses in $N^j \setminus N^i$ because they all contain the literal B_2 (see proof for Lemma 14) and therefore Γ_1 falsifies no of the clauses in $N^j \setminus N^i$. Therefore, Conflict is not applicable before B_2 is propagated. \square