SCL: Proofs

Diagram showing the interconnection of Soundness, Non-redundancy, Termination, and completeness.
3.16.8 Definition (Sound States)

A state \((\Gamma; N; U; \beta; k; D)\) is sound if the following conditions hold:

1. \(\Gamma\) is a consistent sequence of annotated ground literals, i.e. for a ground literal \(L\) it cannot be that \(L \in \Gamma\) and \(\neg L \in \Gamma\).

2. for each decomposition \(\Gamma = \Gamma_1, L^C \cdot L^\sigma, \Gamma_2\) we have that \(C^\sigma\) is false under \(\Gamma_1\) and \(L^\sigma\) is undefined under \(\Gamma_1\), \(N \cup U \models C \lor L\),

3. for each decomposition \(\Gamma = \Gamma_1, L^k, \Gamma_2\) we have that \(L\) is undefined in \(\Gamma_1\),

4. \(N \models U\),

5. if \(D = C \cdot \sigma\) then \(C^\sigma\) is false under \(\Gamma\) and \(N \models C\). In particular, \(\text{grd} <_\beta (N) \models C^\sigma\),

6. for any \(L \in \Gamma\) we have \(\text{L} <_\beta \beta\) and there is a \(C \in N \cup U\) such that \(L \in C\).
### 3.16.9 Lemma (Soundness of the initial state)

The initial state \((\epsilon; N; \emptyset; \beta; 0; \top)\) is sound.

**Proof.**

Criteria 1–3 and 6 are trivially satisfied by \(\Gamma = \epsilon\). Furthermore, \(N \models \emptyset\), fulfilling criterion 4. Lastly, criterion 5 is trivially fulfilled for \(D = \top\).

### 3.16.10 Theorem (Soundness of SCL)

All SCL rules preserve soundness, i.e. they map a sound state onto a sound state.
3.16.9 Lemma (Soundness of the initial state)
The initial state \((\epsilon; N; \emptyset; \beta; 0; \top)\) is sound.

Proof.
Criteria 1–3 and 6 are trivially satisfied by \(\Gamma = \epsilon\). Furthermore, \(N \models \emptyset\), fulfilling criterion 4. Lastly, criterion 5 is trivially fulfilled for \(D = \top\).

3.16.10 Theorem (Soundness of SCL)
All SCL rules preserve soundness, i.e. they map a sound state onto a sound state.
Corollary (Soundness of SCL)

The rules of SCL are sound, hence SCL starting with an initial state is sound.

Proof.

Follows by induction over the size of the run. The base case is handled by Lemma 3.16.9, the induction step is contained in Theorem 3.16.10.
3.16.12 Definition (Reasonable 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.

3.16.13 Definition (Regular Runs)
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.
3.16.14 Theorem (Correct Termination)

If in a regular run no rules are applicable to a state \((\Gamma; N; U; \beta; k; D)\) then either \(D = \bot\) and \(N\) is unsatisfiable or \(D = \top\) and \(\text{grd}(N) \prec_\beta \beta\) is satisfiable and \(\Gamma \models \text{grd}(N) \prec_\beta \beta\).

Proof idea: Take a state where "no" rule is app.

- \((\Gamma; N; U; \beta; k; \top)\)
  - undec. literal \(\prec_\beta \beta\) → Decide, Propagate
    - EC for \(CE(N \cup U)\)

- no undec. literal \(\prec_\beta\)
  - \(\Gamma \models \text{grd} \prec_\beta \beta(N)\): Done
  - \(\Gamma \not\models \text{grd} \prec_\beta \beta(N)\): False clause under \(\Gamma\)
    - in \(\text{grd} \prec_\beta \beta(N)\) → Choose as culprit!
3.16.14 Theorem (Correct Termination)

If in a regular run no rules are applicable to a state \((\Gamma; N; U; \beta; k; D)\) then either \(D = \bot\) and \(N\) is unsatisfiable or \(D = \top\) and \(\text{grd}(N)^{\prec_\beta \beta}\) is satisfiable and \(\Gamma \models \text{grd}(N)^{\prec_\beta \beta}\).

\[
\begin{align*}
\bullet & \quad (\Gamma; N; U; \beta; k; C_0 \sigma) \quad \text{(ad-hoc constraint)} \\
\bullet & \quad \Gamma = \emptyset: \text{Soundness} \quad \Gamma \not\models C_0 \Rightarrow C_0 = \bot \Rightarrow C = \bot \\
& \quad \text{Soundness:} \quad N \not\models C \Rightarrow N \not\models \bot \Rightarrow N \text{ usable}.
\end{align*}
\]

\[
\begin{align*}
\bullet & \quad \Gamma = \Gamma', \bot \\
& \quad \bullet \text{L is propagated} \\
& \quad \bullet \text{comp (L) \in C_0: Resolve} \\
& \quad \bullet \text{otherwise: Skip} \\
& \quad \bullet \text{L is a decision literal: apply either Backtrack, Factorize or Skip.}
\end{align*}
\]
3.16.15 Lemma (Resolve in regular runs)

Consider the derivation of a conflict state \((\Gamma, L; N; U; \beta; k; \top) \Rightarrow \text{Conflict} (\Gamma, L; N; U; \beta; k; D)\). In a regular run, during conflict resolution \(L\) is not a decision literal and at least the literal \(L\) is resolved.

**Proof (idea).**

- Conflict, skip, Factorize, Resolve: obviously not \((D = \top)\)
- Decide: not allowed by realizability
- Backtrack: \((\Gamma, L, \Gamma'; N, U, k; \top)\)
- Backtrack \((\Gamma, L, N, U \cup \{D\} ; k, \top)\)
- Conflict to \(D\): impossible
- Conflict to any other clause in \(N \cup U\): no! regularity
3.16.15 Lemma (Resolve in regular runs)

Consider the derivation of a conflict state

\[(\Gamma, L; N; U; \beta; k; \top) \Rightarrow \text{Conflict} (\Gamma, L; N; U; \beta; k; D)\].

In a regular run, during conflict resolution \(L\) is not a decision literal and at least the literal \(L\) is resolved.

What can we apply to \((\Gamma, L; N; U; \beta; k; D)\)?

- **Backtrack**: no
  
  requires \(L\) to be a decision literal.

- **Skip**: no (if \(L\) does not occur in \(D\), apply conflict to \((\Gamma; N; U; \beta; k; \top)\))

- **Factorize**: "does not really make progress"

- **Resolve** ✓
3.16.16 Definition (State Induced Ordering)

Let \((L_1, L_2, \ldots, L_n; N; U; \beta; k; D)\) be a sound state of SCL. The trail induces a total well-founded strict order on the defined literals by

\[ L_1 \prec_\Gamma \text{comp}(L_1) \prec_\Gamma L_2 \prec_\Gamma \text{comp}(L_2) \prec_\Gamma \cdots \prec_\Gamma L_n \prec_\Gamma \text{comp}(L_n). \]

We extend \(\prec_\Gamma\) to a strict total order on all literals where all undefined literals are larger than \(\text{comp}(L_n)\). We also extend \(\prec_\Gamma\) to a strict total order on ground clauses by multiset extension and also on multisets of ground clauses and overload \(\prec_\Gamma\) for all these cases. With \(\preceq_\Gamma\) we denote the reflexive closure of \(\prec_\Gamma\).
Let \((\Gamma; N; U; \beta; k; C_0 \cdot \sigma_0)\) be the state resulting from the application of Conflict in a regular run and let \(C\) be the clause learned at the end of the conflict resolution, then \(C\) is not redundant with respect to \(N \cup U\) and \(\prec \Gamma\).

(Idea) Consider \((\Gamma'; N; U; \beta; k; C \cdot \sigma) \Rightarrow \text{Backtrack}\)

- There was a literal \(\ell\) in \(C_0\sigma\) which is not in \(C\sigma\).
- \(C_0\sigma\) is false under \(\Gamma'\) (soundness).
- Assume \(C_0\sigma\) is redundant

\[
\Gamma \not\models \text{gcd}(N \cup U) \leq r \cdot C_0 \sigma = C_0
\]

\(\Rightarrow\) There is a false clause in \(\text{gcd}(N \cup U) \leq\), we could have applied Conflict earlier.
- During a run, the ordering of literals changes
- Hence, $\prec_\Gamma$ changes as well!
- Non-redundancy property of Theorem 3.16.17 reflects state at time of creation of learned clause
- At time of creation, no need to check for redundancy
- Still, all $\prec_\Gamma$ contain the fixed clause subset ordering $\prec_{\subseteq}$
SCL: Proofs
3.16.19 Theorem (Termination)
Any regular run of $\Rightarrow_{SCL}$ terminates.

Lemma (Termination without Backtrack)
Any regular run of $\Rightarrow_{SCL}$ that does not use the Backtrack rule terminates.

$$\mathcal{M}(\Gamma, N; U; \beta; k; \top) = (1, |\{P | P \prec_B \beta\}| - |\Gamma|, 0)$$
$$\mathcal{M}(\Gamma, N; U; \beta; k; C) = (0, \#\text{possible resolutions}, |C|)$$
3.16.19 Theorem (Termination)

Any regular run of $\Rightarrow_{SCL}$ terminates.

Lemma (Termination without Backtrack)

Any regular run of $\Rightarrow_{SCL}$ that does not use the Backtrack rule terminates.

$$
\mathcal{M}(\Gamma, N; U; \beta; k; \top) = (1, \ |\{P \mid P \prec_{B} \beta\}| - |\Gamma|, 0)
$$

$$
\mathcal{M}(\Gamma, N; U; \beta; k; C) = (0, \ #\text{possible resolutions}, \ |C|)
$$
3.16.19 Theorem (Termination)
Any regular run of $\Rightarrow_{\text{SCL}}$ terminates.

Lemma (Termination without Backtrack)
Any regular run of $\Rightarrow_{\text{SCL}}$ that does not use the Backtrack rule terminates.

\[
\mathcal{M}(\Gamma, N; U; \beta; k; \top) = (1, \{P \mid P \prec_B \beta\} - |\Gamma|, 0)
\]
\[
\mathcal{M}(\Gamma, N; U; \beta; k; C) = (0, \text{#possible resolutions}, |C|)
\]

- **Decide, Propagate**
- **Resolve, Skip**
- **Factorize**
Lemma (Termination with Backtrack)

Any regular run of $\Rightarrow_{SCL}$ cannot use the Backtrack rule infinitely often.

Proof.

Firstly, for a regular run, by Theorem 3.16.17, all learned clauses are non-redundant under $\preceq_{\Gamma}$. Those clauses are also non-redundant under the fixed subset ordering $\preceq_{\subseteq}$, which is well-founded. Due to the restriction of all clauses to be smaller than $\{\beta\}$, the overall number of non-redundant ground clauses is finite. So Backtrack can only be invoked finitely many times.
Lemma (Termination with Backtrack)

Any regular run of $\Rightarrow_{\text{SCL}}$ cannot use the Backtrack rule infinitely often.

Proof.

Firstly, for a regular run, by Theorem 3.16.17, all learned clauses are non-redundant under $\prec_{\text{Γ}}$. Those clauses are also non-redundant under the fixed subset ordering $\prec_{\subseteq}$, which is well-founded. Due to the restriction of all clauses to be smaller than $\{\beta\}$, the overall number of non-redundant ground clauses is finite. So Backtrack can only be invoked finitely many times. $\square$
SCL: Proofs
If $N$ is unsatisfiable, such that some finite $N' \subseteq \text{grd}(N)$ is unsatisfiable and $\beta$ is $\prec_\beta$ larger than all literals in $N'$ then any regular run from $(\epsilon; N; \emptyset; \beta; 0; \top)$ of SCL derives $\bot$.

Proof.
By Theorem 3.16.19 and Theorem 3.16.14.
SCL: Proofs
3.16.18 Theorem (BS Non-Redundancy is NEXPTIME-Complete)

Deciding non-redundancy of a BS clause \( C \) with respect to a finite BS clause set \( N \preceq C \) is NEXPTIME-Complete.

- **containment**: solve \( N \preceq C \neq C \in \text{NEXPTIME} \)

- **hardness**: \( N = \{ C_1, \ldots, C_n \} \) finite BS-clause set, define a \( \prec_{\text{LPO}} \). Add fresh \( P \) s.t. \( P \) is \( \prec_{\text{LPO}} \)-larger than any literal in \( N \)

\[ \text{in } \{ C_1, \ldots, C_n, EP3 \} \]

\( EP3 \) is redundant \( \iff \) \( N \) unsatisfiable

(satisfiability of BS-clause is NEXPTIME-complete)
Obviously, given some unsatisfiable clause set $N$ there is no way to efficiently compute some $\beta$ such that $\text{ground}(N) \prec \beta$ is unsatisfiable. Therefore, in an implementation, the below rule Grow is needed to eventually provide a semi-decision procedure.

\[
\text{Grow} \quad (\Gamma; N; U; \beta; k; \top) \Rightarrow_{\text{SCL}} (\epsilon; N; U; \beta'; 0; \top)
\]

provided $\Gamma \models \text{grd}(N) \prec \beta$ and $\beta \prec \beta'$
3.16.21 Theorem (SCL decides the BS fragment)

SCL restricted to regular runs decides satisfiability of a BS clause set if $\beta$ is set appropriately.

Proof.

Let $B$ be the set of constants in the BS clause set $N$. Then define $\prec_\beta$ and $\beta$ such that $L \prec_\beta \beta$ for all $L \in \text{grd}^{\prec_\beta\beta}(N)$. Following the proof of Theorem 3.16.19, any SCL regular run will terminate on a BS clause set.
The End (of SCL)

orderings

compactness proof