Ordering restrictions

Let $\prec$ be a total ordering on Σ.

We lift $\prec$ to a total ordering on literals by $\prec \subseteq \prec_L$ and $P \prec_L \neg P$ and $\neg P \prec_L Q$ for all $P \prec Q$.

We further lift $\prec_L$ to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.

Eventually, we overload $\prec$ with $\prec_L$ and $\prec_C$.

We define $N^{\prec C} = \{D \in N \mid D \prec C\}$.

Eventually we will restrict inferences to maximal literals with respect to $\prec$.

Abstract Redundancy

A clause C is *redundant* with respect to a clause set N if $N^{\prec C} \models C$.

Tautologies are redundant. Subsumed clauses are redundant if $\subseteq$ is strict.

Remark: Note that for finite N, $N^{\prec C} \models C$ can be decided for PROP(Σ) but is as hard as testing unsatisfiability for a clause set N.

Partial Model Construction

Given a clause set N and an ordering $\prec$ we can construct a (partial) model $N_\mathcal{I}$ for N as follows:

$N_C := \bigcup_{D \prec C} \delta_D$

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

$N_\mathcal{I} := \bigcup_{C \in N} \delta_C$

Clauses C with $\delta_C \neq \emptyset$ are called *productive*. Some properties of the partial model construction.

Proposition 2.12 *1. For every D with $(C \lor \neg P) \prec D$ we have $\delta_D \neq \{P\}$.*

2. If $\delta_C = \{P\}$ then $N_C \cup \delta_C \models C$.

3. If $N_C \models D$ then for all C' with $C \prec C'$ we have $N_{C'} \models D$ and in particular $N_\mathcal{I} \models D$.

Notation: N, $N^{\prec C}$, $N_\mathcal{I}$, N_C

Please properly distinguish:

- N is a set of clauses intepreted as the conjunction of all clauses.
- $N^{\prec C}$ is of set of clauses from N strictly smaller than C with respect to $\prec$.
- $N_\mathcal{I}$, N_C are sets of atoms, often called *Herbrand Interpretations*. $N_\mathcal{I}$ is the overall (partial) model for N, whereas N_C is generated from all clauses from N strictly smaller than C.
- Validity is defined by $N_\mathcal{I} \models P$ if $P \in N_\mathcal{I}$ and $N_\mathcal{I} \models \neg P$ if $P \notin N_\mathcal{I}$, accordingly for N_C.

Superposition

The *superposition calculus* consists of the inference rules *superposition left* and *factoring*:

Superposition Left
$(N \uplus \{C_1 \lor P, C_2 \lor \neg P\}) \quad \Rightarrow \quad (N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor C_2\})$

where P is strictly maximal in $C_1 \lor P$ and $\neg P$ is maximal in $C_2 \lor \neg P$

Factoring
$(N \uplus \{C \lor P \lor P\}) \quad \Rightarrow \quad (N \cup \{C \lor P \lor P\} \cup \{C \lor P\})$

where P is maximal in $C \lor P \lor P$

examples for specific redundancy rules are

Subsumption
$(N \uplus \{C_1, C_2\}) \quad \Rightarrow \quad (N \cup \{C_1\})$

provided $C_1 \subset C_2$

Tautology Deletion
$(N \uplus \{C \lor P \lor \neg P\}) \quad \Rightarrow \quad (N)$

Subsumption Resolution
$(N \uplus \{C_1 \lor L, C_2 \lor \bar{L}\}) \quad \Rightarrow \quad (N \cup \{C_1 \lor L, C_2\})$

where $C_1 \subseteq C_2$

Theorem 2.13 *If from a clause set N all possible superposition inferences are redundant and $\bot \notin N$ then N is satisfiable and $N_\mathcal{I} \models N$.*

Proof. The proof is by contradiction. So assume if C is any clause derived by superposition left or factoring from N that C is redundant, i.e., $N^{\prec C} \models C$. Furthermore, we assume $\bot \notin N$ but $N_{\mathcal{I}} \not\models N$. Then there is a minimal, with respect to $\prec$, clause $C_1 \lor L \in N$ such that $N_{\mathcal{I}} \not\models C_1 \lor L$ and L is a maximal literal in $C_1 \lor L$. This clause must exist because $\bot \notin N$.

(i) note that because $C_1 \lor L$ is minimal it is not redundant. For otherwise, $N^{\prec C_1 \lor L} \models C_1 \lor L$ and hence $N_{\mathcal{I}} \models C_1 \lor L$, a contradiction.

(ii) we distinguish the case wether L is a positive or negative literal. Firstly, let us assume L is positive, i.e., $L = P$ for some propositional variable P. Now if P is strictly maximal in $C_1 \lor P$ then actually $\delta_{C_1 \lor P} = \{P\}$ and hence $N_{\mathcal{I}} \models C_1 \lor P$, a contradiction. So P is not strictly maximal. But then actually $C_1 \lor P$ has the form $C_1' \lor P \lor P$ and by factoring we can derive $C_1' \lor P$ where $(C_1' \lor P) \prec C_1' \lor P \lor P$. Now $C_1' \lor P$ is not redundant (analogous to (i)), strictly smaller than $C_1 \lor L$, we have $C_1' \lor P \in N$ and $N_{\mathcal{I}} \not\models C_1' \lor P$, a contradiction against the choice of $C_1 \lor L$.

Secondly, let us assume L is negative, i.e., $L = \neg P$ for some propositional variable P. Then, since $N_{\mathcal{I}} \not\models C_1 \lor \neg P$ we know $P \in N_{\mathcal{I}}$. So there is a clause $C_2 \lor P \in N$ where $\delta_{C_2 \lor P} = \{P\}$ and P is strictly maximal in $C_2 \lor P$ and $(C_2 \lor P) \prec (C_1 \lor \neg P)$. So by superposition left we can derive $C_1 \lor C_2$ where $(C_1 \lor C_2) \prec (C_1 \lor \neg P)$. The derived clause $C_1 \lor C_2$ cannot be redundant, because for otherwise either $N^{\prec C_2 \lor P} \models C_2 \lor P$ or $N^{\prec C_1 \lor \neg P} \models C_1 \lor \neg P$. So $C_1 \lor C_2 \in N$ and $N_{\mathcal{I}} \not\models C_1 \lor C_2$, a contradiction against the choice of $C_1 \lor L$.

$\square$

So the proof actually tells us that at any point in time we need only to consider either a superposition left inference between a minimal false clause and a productive clause or a factoring inference on a minimal false clause.

A Superposition Theorem Prover STP

3 clause sets:

N(ew) containing new inferred clauses
U(sable) containing reduced new inferred clauses
clauses get into *W(orked) O(ff)* once their inferences have been computed

Strategy:

Inferences will only be computed when there are no possibilities for simplification

Rewrite Rules for STP

Tautology Deletion

$$(N \uplus \{C\}; U; WO) \quad \Rightarrow_{STP} \quad (N; U; WO)$$

if C is a tautology

Forward Subsumption

$$(N \uplus \{C\}; U; WO) \quad \Rightarrow_{STP} \quad (N; U; WO)$$

if some $D \in (U \cup WO)$ subsumes C

Backward Subsumption U

$$(N \uplus \{C\}; U \uplus \{D\}; WO) \quad \Rightarrow_{STP} \quad (N \cup \{C\}; U; WO)$$

if C strictly subsumes D ($C \subset D$)

Backward Subsumption WO

$$(N \uplus \{C\}; U; WO \uplus \{D\}) \quad \Rightarrow_{STP} \quad (N \cup \{C\}; U; WO)$$

if C strictly subsumes D ($C \subset D$)

Forward Subsumption Resolution

$$(N \uplus \{C_1 \vee L\}; U; WO) \quad \Rightarrow_{STP} \quad (N \cup \{C_1\}; U; WO)$$

if there exists $C_2 \vee \bar{L} \in (UP \cup WO)$ such that $C_2 \subseteq C_1$

Backward Subsumption Resolution U

$$(N \uplus \{C_1 \vee L\}; U \uplus \{C_2 \vee \bar{L}\}; WO) \quad \Rightarrow_{STP} \quad (N \cup \{C_1 \vee L\}; U \uplus \{C_2\}; WO)$$

if $C_1 \subseteq C_2$

Backward Subsumption Resolution WO

$$(N \uplus \{C_1 \vee L\}; U; WO \uplus \{C_2 \vee \bar{L}\}) \quad \Rightarrow_{STP} \quad (N \cup \{C_1 \vee L\}; U; WO \uplus \{C_2\})$$

if $C_1 \subseteq C_2$

Clause Processing

$$(N \uplus \{C\}; U; WO) \quad \Rightarrow_{STP} \quad (N; U \cup \{C\}; WO)$$

Inference Computation

$$(\emptyset; U \uplus \{C\}; WO) \quad \Rightarrow_{STP} \quad (N; U; WO \cup \{C\})$$

where N is the set of clauses derived by superposition inferences from C and clauses in WO.

Soundness and Completeness

Theorem 2.14

$$N \models \bot \quad \Leftrightarrow \quad (N;\emptyset;\emptyset) \quad \Rightarrow^*_{STP} \quad (N' \cup \{\bot\}; U; WO)$$

Proof in L. Bachmair, H. Ganzinger: Resolution Theorem Proving appeared in the Handbook of Automated Reasoning, 2001

Termination

Theorem 2.15 *For finite N and a strategy where the reduction rules Tautology Deletion, the two Subsumption and two Subsumption Resolution rules are always exhaustively applied before Clause Processing and Inference Computation, the rewrite relation $\Rightarrow_{STP}$ is terminating on $(N;\emptyset;\emptyset)$.*

Proof: think of it (more later on).

Fairness

Problem:

If N is inconsistent, then $(N;\emptyset;\emptyset) \Rightarrow^*_{STP} (N' \cup \{\bot\}; U; WO)$.

Does this imply that *every* derivation starting from an inconsistent set N eventually produces $\bot$?

No: a clause could be kept in U without ever being used for an inference.

We need in addition a *fairness condition:*

If an inference is possible forever (that is, none of its premises is ever deleted), then it must be computed eventually.

One possible way to guarantee fairness: Implement U as a queue (there are other techniques to guarantee fairness).

With this additional requirement, we get a stronger result: If N is inconsistent, then every *fair* derivation will eventually produce $\bot$.