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:

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

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

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

Partial Model Construction

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 $(C \vee \neg P) \prec D$ 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

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

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

Factoring

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

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

Superposition

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

$$(N \uplus \{C \vee P \vee \neg P\}) \quad \Rightarrow \quad (N)$$

Subsumption Resolution

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

where $C_1 \subseteq C_2$

Superposition

Theorem 2.13:

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

Superposition

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 ($C \subset D$)

Rewrite Rules for STP

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 $C_1 \subseteq C_2$

Rewrite Rules for *STP*

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.

Fairness

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 $\bot$.