## Memory Matters: SPASS-SATT

<table>
<thead>
<tr>
<th>Category</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forget-Start</td>
<td>800</td>
<td>108800</td>
</tr>
<tr>
<td>Restarts</td>
<td>412</td>
<td>369</td>
</tr>
<tr>
<td>Conflicts</td>
<td>153640</td>
<td>133403</td>
</tr>
<tr>
<td>Decisions</td>
<td>184034</td>
<td>159005</td>
</tr>
<tr>
<td>Propagations</td>
<td>17770298</td>
<td>15544812</td>
</tr>
<tr>
<td>Time</td>
<td>11</td>
<td>23</td>
</tr>
<tr>
<td>Memory</td>
<td>16</td>
<td>41</td>
</tr>
</tbody>
</table>
Propositional Logic Calculi

1. Tableau: classics, natural from the semantics
2. Resolution: classics, first-order, prepares for CDCL
3. CDCL: current prime calculus for propositional logic
4. Superposition: first-order, prepares for first-order
Propositional Superposition

Propositional Superposition refines the propositional resolution calculus by

(i) ordering and selection restrictions on inferences,
(ii) an abstract redundancy notion,
(iii) the notion of a partial model, based on the ordering for inference guidance
(iv) a *saturation* concept.

Important: No implicit Condensation of literals!
2.7.1 Definition (Clause Ordering)

Let $\prec$ be a total strict ordering on $\Sigma$.

Then $\prec$ can be lifted to a total ordering on literals by $\prec \subseteq \prec_L$ and $P \prec_L \neg P$ and $\neg P \prec_L Q$, $\neg P \prec_L \neg Q$ for all $P \prec Q$.

The ordering $\prec_L$ can be lifted to a total ordering on clauses $\prec_C$ by considering the multiset extension of $\prec_L$ for clauses.
2.7.2 Proposition (Properties of the Clause Ordering)

(i) The orderings on literals and clauses are total and well-founded.

(ii) Let $C$ and $D$ be clauses with $P = \text{atom}(\text{max}(C))$, $Q = \text{atom}(\text{max}(D))$, where \(\text{max}(C)\) denotes the maximal literal in $C$.

   (i) If $Q \prec L P$ then $D \prec_C C$.

   (ii) If $P = Q$, $P$ occurs negatively in $C$ but only positively in $D$, then $D \prec_C C$.

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

For a clause set $N$, I define $N \prec_C = \{ D \in N \mid D \prec_C C \}$. 
Definition (Abstract Redundancy)

A clause $C$ is *redundant* with respect to a clause set $N$ if $N^{<C} \models C$. 
2.7.4 Definition (Selection Function)

The selection function sel maps clauses to one of its negative literals or \( \bot \).

If \( \text{sel}(C) = \neg P \) then \( \neg P \) is called *selected* in \( C \).

If \( \text{sel}(C) = \bot \) then no literal in \( C \) is *selected*. 
2.7.5 Definition (Partial Model Construction)

Given a clause set $N$ and an ordering $\prec$ we can construct a (partial) Herbrand model $N_I$ for $N$ inductively as follows:

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

$$\delta_D := \begin{cases} 
\{P\} & \text{if } D = D' \lor P, P \text{ strictly maximal, no literal selected in } D \text{ and } N_D \notmodels D \\
\emptyset & \text{otherwise}
\end{cases}$$

$$N_I := \bigcup_{C \in N} \delta_C$$

Clauses $C$ with $\delta_C \neq \emptyset$ are called *productive*. 
2.7.6 Proposition (Model Construction Properties)

Some properties of the partial model construction.

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

(ii) If $\delta_C = \{P\}$ then $N_C \cup \delta_C \models C$.

(iii) If $N_C \models D$ and $D \prec C$ then for all $C'$ with $C \prec C'$ we have $N_{C'} \models D$ and in particular $N_\emptyset \models D$.

(iv) There is no clause $C$ with $P \lor P \prec C$ such that $\delta_C = \{P\}$.
Superposition Inference Rules

**Superposition Left** \( (N \uplus \{ C_1 \lor P, C_2 \lor \neg P \}) \Rightarrow \text{SUP} (N \uplus \{ C_1 \lor P, C_2 \lor \neg P \} \cup \{ C_1 \lor C_2 \}) \)

where (i) \( P \) is strictly maximal in \( C_1 \lor P \) (ii) no literal in \( C_1 \lor P \) is selected (iii) \( \neg P \) is maximal and no literal selected in \( C_2 \lor \neg P \), or \( \neg P \) is selected in \( C_2 \lor \neg P \)

**Factoring** \( (N \uplus \{ C \lor P \lor P \}) \Rightarrow \text{SUP} (N \uplus \{ C \lor P \lor P \} \cup \{ C \lor P \}) \)

where (i) \( P \) is maximal in \( C \lor P \lor P \) (ii) no literal is selected in \( C \lor P \lor P \)
2.7.7 Definition (Saturation)

A set $N$ of clauses is called \textit{saturated up to redundancy}, if any inference from non-redundant clauses in $N$ yields a redundant clause with respect to $N$ or is already contained in $N$. 