2.10 Superposition Versus CDCL

We will establish a relationship between Superposition and CDCL operating on a clause set \( N \):

Superposition: Is based on an ordering \( \prec \). It computes a model assumption \( N_I \).

Either \( N_I \) is a model, \( N \) contains the empty clause, or there is an inference on the minimal false clause with respect to \( \prec \).

CDCL: Is based on a variable selection heuristic. It computes a model assumption via decision variables and propagation. Either this assumption is a model of \( N \), \( N \) contains the empty clause, or there is a backjump clause that is learned.

**Proposition 2.20** Let \((L_1 + L_2 + \ldots + L_k; N)\) be a CDCL with eager propagation state. Some of the \( L_i \) may be decision literals and the corresponding propositional variables are \( P_1, \ldots, P_k \). Furthermore, let us assume that \( L_1 + \ldots + L_{k-1} \) is a partial valuation that does not falsify any clause in \( N \) whereas \( L_1 + L_2 + \ldots + L_k \) falsifies some clause \( C \lor \overline{L_k} \in N \). Then

(a) \( L_k \) is a propagated literal.

(b) The resolvent between \( C \lor \overline{L_k} \) and the clause propagating \( L_k \) is a superposition inference and the conclusion is not redundant with respect to the ordering \( P_1 \prec P_2 \ldots \prec P_k \).

**Proof.** (a) The clause \( C \lor \overline{L_k} \) propagates \( \overline{L_k} \) with respect to \( L_1 + \ldots + L_{k-1} \), so with eager propagation, the literal \( L_k \) cannot be decision literal but was propagated by a clause \( C' \lor \overline{L_k} \in N \).

(b) Both \( C \) and \( C' \) only contain literals with variables from \( P_1, \ldots, P_{k-1} \). Since we assume duplicate literals to be removed and tautologies to be deleted, the literal \( \overline{L_k} \) is strictly maximal in \( C \lor \overline{L_k} \) and \( L_k \) is strictly maximal in \( C' \lor \overline{L_k} \). So resolving on \( L_k \) is a superposition inference with respect to the variable ordering \( P_1 \prec P_2 \ldots \prec P_k \). Now assume \( C \lor C' \) is redundant, i.e., there are clauses \( D_1, \ldots, D_n \) from \( N \) with \( D_1 \prec C \lor C' \) and \( D_1, \ldots, D_n \models C \lor C' \). Since \( C \lor C' \) is false in \( L_1 + \ldots + L_{k-1} \) there is at least one \( D_i \) that is also false in \( L_1 + \ldots + L_{k-1} \). A contradiction against the assumption that \( L_1 + \ldots + L_{k-1} \) does not falsify any clause in \( N \). \( \square \)

**Proposition 2.21** The 1UIP backjump clause is not redundant.

**Proof.** By Proposition 2.20 a one resolution step 1UIP backjump clause has this property. The argument in the proof of Proposition 2.20 can be repeated until we reach the first decision literal \( L_m \) by resolving away \( L_k, L_{k-1}, \ldots, L_{m+1} \). \( \square \)
Proposition 2.22  Let \((L_1 + L_2 + \ldots + L_k; N)\) be a CDCL with eager propagation state. We assume that all decision literals among the \(L_i\) are negative and let the corresponding propositional variables be \(P_1, \ldots, P_k\). Furthermore, let us assume that \(L_1 + \ldots + L_k\) is a partial valuation that does not falsify any clause in \(N\). Then \(N_{\preceq}^{\leq P_{k+1}} = \{P_1, \ldots, P_k\} \cap \{L_1, \ldots, L_k\}\) with ordering \(P_1 < P_2 \ldots < P_{k+1}\).

Proof. We assume that there is a variable \(P_{k+1} \in \Sigma\) for otherwise it can be added. By induction on \(k\). For the base case \(k = 1\) we distinguish two cases. If \(L_1\) is propagated then there is a clause \(L_1 \in N\). In case \(L_1\) is positive then it is also productive and \(L_1 \in N_{\preceq}^{\leq P_2}\). If it is negative then there cannot be a clause \(P_1 \in N\), so \(P_1 \not\in N_{\preceq}^{\leq P_2}\).

For the induction step assume \(N_{\preceq}^{\leq P_k} = \{P_1, \ldots, P_{k-1}\} \cap \{L_1, \ldots, L_{k-1}\}\). If \(L_k\) is propagated and positive, then there is a clause \(C \lor L_k\) where all atoms in \(C\) are from \(\{P_1, \ldots, P_{k-1}\}\) and hence \(L_k\) is strictly maximal in \(C \lor L_k\), the clause \(C\) is false in \(N_{\preceq}^{\leq P_k}\) and therefore \(L_k\) is produced, proving \(N_{\preceq}^{\leq P_{k+1}} = \{P_1, \ldots, P_k\} \cap \{L_1, \ldots, L_k\}\).

If \(L_k\) is propagated and negative, then there cannot be a clause \(C \lor P_k \in N_{\preceq}^{\leq P_{k+1}}\) with \(C\) false in \(N_{\preceq}^{\leq P_k}\), because for otherwise \(L_1 + \ldots + L_k\) falsifies a clause in \(N\). So there is no clause in \(N\) producing \(P_k\) and hence \(N_{\preceq}^{\leq P_{k+1}} = \{P_1, \ldots, P_k\} \cap \{L_1, \ldots, L_k\}\).

If \(L_k\) is a decision literal and therefore negative, there cannot be a clause \(C \lor P_k \in N_{\preceq}^{\leq P_{k+1}}\) with \(C\) false in \(N_{\preceq}^{\leq P_k}\), because we assume eager propagation and so again \(N_{\preceq}^{\leq P_{k+1}} = \{P_1, \ldots, P_k\} \cap \{L_1, \ldots, L_k\}\). \(\square\)

3 First-Order Logic

First-order logic

- formalizes fundamental mathematical concepts
- is expressive (Turing-complete)
- is not too expressive (e.g. not axiomatizable: natural numbers, uncountable sets)
- has a rich structure of decidable fragments
- has a rich model and proof theory

First-order logic is also called (first-order) predicate logic.