7.2 CDCL(T)

Consider a SAT problem where the propositional variables actually stand for ground atoms over some theory $\mathcal{T}$, or a Nelson-Oppen combination of theories, e.g., ground equations or ground atoms of LRA, i.e., LRA atoms where all variables are existentially quantified. The basic idea of all procedures in this section is to apply CDCL, Section **??**, in order to investigate the boolean structure of the problem. If CDCL derives unsatisfiability, then the problem clearly is. If CDCL derives satisfiability, then a ground decision procedure for $\mathcal{T}$ has to check whether the actual CDCL assignment constitutes also a model in $\mathcal{T}$.

For example, let $\mathcal{T}$ be the purely equational ground theory over free symbols (EUF) where we consider Congruence Closure (Section 6.1) as a decision procedure. Now consider a formula

$$f(a) \approx b \land b \approx c \land (f(a) \not\approx c \lor a \not\approx c)$$

and its boolean abstraction (clauses)

$$P_1 \land P_2 \land (P_3 \lor P_4).$$

A CDCL algorithm might find the propositional model $M_1 = P_1P_2P_3$. Obviously, the respective literals $f(a) \approx b$, $b \approx c$, $f(a) \not\approx c$ are contradictory in EUF. So M_1 does not correspond to a $\mathcal{T}$-model. The congruence closure algorithm can easily justify this contradiction with respect to the literals P_1, P_2, P_3, and hence the CDCL algorithm can learn the clause $\neg P_1 \lor \neg P_2 \lor \neg P_3$. Adding this clause to the above clauses

$$P_1 \land P_2 \land (P_3 \lor P_4) \land (\neg P_1 \lor \neg P_2 \lor \neg P_3)$$

the CDCL algorithm finds the next model $M_2 = P_1P_2\neg P_3P_4$ corresponding to the literals $f(a) \approx b$, $b \approx c$, $f(a) \approx c$, and $a \not\approx c$ which are satisfiable in EUF. So, an overall model is found.

Let N be a finite set of clauses over some theory $\mathcal{T}$ over signature $\Sigma_\mathcal{T}$ such that there exists a decision procedure for satisfiability of a conjunction of literals: $\models_\mathcal{T} L_1 \land \ldots \land L_n$. Let atr be a bijection from the atoms over $\Sigma_\mathcal{T}$ into propositional variables Σ_{PROP} auch that $\mathrm{atr}^{-1}(\mathrm{atr}(A)) = A$. Furthermore, atr distributes over the propositional operators, e.g., $\mathrm{atr}(\neg A) = \neg\,\mathrm{atr}(A)$.

Lemma 7.2.1 (Correctness of atr)**.** Let N be a set of clauses over some theory $\mathcal{T}$. If $\mathrm{atr}(N) \models \bot$ then $N \models_\mathcal{T} \bot$.

A CDCL(T) problem state is a five-tuple $(M; N; U; k; C)$ where N is the propositional abstraction of some clause set N', $N = \text{atr}(N')$, M a sequence of annotated propositional literals, U is a set of detived propositional clauses, $k \in \mathbb{N} \cup \{-1\}$, and C is a propositional clause or $\top$ or $\bot$. In particular, the following states can be distinguished:

$(\epsilon; N; \emptyset; 0; \top)$	is the start state for some clause set N
$(M; N; U; -1; \top)$	is a final state, where $\text{atr}^{-1}(M) \models_{\mathcal{T}} N'$, $\text{atr}^{-1}(M)$ satisfiable
$(M; N; U; k; \bot)$	is a final state, where N' has no model
$(M; N; U; k; \top)$	is a model search state if $k \neq 0$
$(M; N; U; k; D)$	is a backtracking state if $D \notin \{\top, \bot\}$

Literals in $L \in M$ are either annotated with a number, a level, i.e., they have the form L^k meaning that L is the $k-th$ guessed decision literal, or they are annotated with a clause that forced the literal to become true. A literal L is of *level* k with respect to a problem state $(M; N; U; j; C)$ if L or $\text{comp}(L)$ occurs in M and L itself or the first decision literal left from L ($\text{comp}(L)$) in M is annotated with k. If there is no such decision literal then $k = 0$. A clause D is of *level* k with respect to a problem state $(M; N; U; j; C)$ if k is the maximal level of a literal in D. Recall C is a non-empty clause or $\top$ or $\bot$. The rules are

Propagate $(M; N; U; k; \top) \ \Rightarrow_{\text{CDCL}} \ (ML^{C \vee L}; N; U; k; \top)$
provided $C \vee L \in (N \cup U)$, $M \models \neg C$, and L is undefined in M

Decide $(M; N; U; k; \top) \ \Rightarrow_{\text{CDCL}} \ (ML^{k+1}; N; U; k+1; \top)$
provided L is undefined in M

Conflict $(M; N; U; k; \top) \ \Rightarrow_{\text{CDCL}} \ (M; N; U; k; D)$
provided $D \in (N \cup U)$ and $M \models \neg D$

Skip $(ML^{C \vee L}; N; U; k; D) \ \Rightarrow_{\text{CDCL}} \ (M; N; U; k; D)$
provided $D \notin \{\top, \bot\}$ and $\text{comp}(L)$ does not occur in D

Resolve $(ML^{C \vee L}; N; U; k; D \vee \text{comp}(L)) \ \Rightarrow_{\text{CDCL}} \ (M; N; U; k; D \vee C)$
provided D is of level k

Backtrack $(M_1 K^{i+1} M_2; N; U; k; D \vee L) \ \Rightarrow_{\text{CDCL}} \ (M_1 L^{D \vee L}; N; U \cup \{D \vee L\}; i; \top)$
provided L is of level k and D is of level i.

Restart $(M; N; U; k; \top) \ \Rightarrow_{\text{CDCL}} \ (\epsilon; N; U; 0; \top)$
provided $M \not\models N$

Forget $(M; N; U \uplus \{C\}; k; \top) \ \Rightarrow_{\text{CDCL}} \ (M; N; U; k; \top)$

provided $M \not\models N$

Note that these rules are exactly the rules of CDCL from Section **??**. The only difference that any normal form $(M; N; U; k; \top)$ was a final state in CDCL, but not in CDCL(T) because $k \neq -1$. On the other hand, if CDCL derives the empty clause, i.e., $\bot$, then this is also a final state for CDCL(T), see Lemma 7.2.1. The $\mathcal{T}$ rules are missing that in particular check whether the propoistional model is in fact also a theory model.

$\mathcal{T}$-Success $(M; N; U; k; \top) \;\Rightarrow_{\mathrm{CDCL(T)}}\; (M; N; U; -1; \top)$

provided $k \neq -1$, $M \models (N \cup U)$ and $\mathrm{atr}^{-1}(M)$ is $\mathcal{T}$-satisfiable

$\mathcal{T}$-Propagate $(M; N; U; k; \top) \;\Rightarrow_{\mathrm{CDCL(T)}}\; (ML^{C\vee L}; N; U; k; \top)$

provided , $\mathrm{atr}^{-1}(M)$ is $\mathcal{T}$-satisfiable, L is undefined in M but $\mathrm{atom}(L)$ occurs in $N \cup U$, and there are literals $L_1, \ldots, L_n$ from M with $\mathrm{atr}^{-1}(L_1), \ldots, \mathrm{atr}^{-1}(L_n) \models_{\mathcal{T}} \mathrm{atr}^{-1}(L)$ and $C = \mathrm{comp}(L_1) \vee \ldots \vee \mathrm{comp}(L_n)$

$\mathcal{T}$-Conflict $(M; N; U; k; \top) \;\Rightarrow_{\mathrm{CDCL(T)}}\; (\epsilon; N; U \cup \{\mathrm{comp}(L_1) \vee \ldots \vee \mathrm{comp}(L_n)\}; 0; \top)$

provided there are literals $L_1, \ldots, L_n$ from M with $\mathrm{atr}^{-1}(L_1), \ldots, \mathrm{atr}^{-1}(L_n) \models_{\mathcal{T}} \bot$

Note that the clause $L_1 \wedge \ldots \wedge L_n \rightarrow L$ used to justify $\mathcal{T}$-Propagate as well as the $\mathcal{T}$-Conflict clause $\neg L_1 \vee \ldots \vee \neg L_n$ are tautologies in $\mathcal{T}$. For rule $\mathcal{T}$-Conflict the literal L_i of maximal level could be a decision literal, hence a restart is a safe way that CDCL(T) does not get stuck.

The rule $\mathcal{T}$-Propagate is not needed for soundness nor for completeness. Just for "efficiency". But in contrast to CDCL, where boolean propagation can be very efficiently computed, for some arbitrary theory $\mathcal{T}$ this might not be the case. So there is a trade off between at any time checking M with respect to the theory and thus avoiding $\mathcal{T}$ conflicts, called *eager* theory consideration, and computing with respect to the boolean structure and taking into account eventual extra $\mathcal{T}$ conflicts, called *lazy* theory consideration. Similarly, it is not obvious whether the applicability of $\mathcal{T}$-Conflict should be checked eagerly, because this might be expensive.

So the minimal requirement for $\mathcal{T}$ is a decision procedure that checks for a conjunction of literals whether it is satisfiable or not and in case it is not ideally provides a minimal unsatisfiable subset.

Definition 7.2.2 (Reasonable CDCL(T) Strategy)**.** A CDCL(T) strategy is *reasonable* if the rules Conflict and Propagate are always preferred over all other rules.

Theorem 7.2.3 (CDCL(T) Properties)**.** Consider a clause set $N = \mathrm{atr}(N')$ for a clause set N' over some theory $\mathcal{T}$ and a reasonable run of CDCL(T) with start state $(\epsilon; N; \emptyset; 0; \top)$. Then

1. The clause $\text{comp}(L_1) \lor \ldots \lor \text{comp}(L_n)$ learned by $\mathcal{T}$-Conflict is not contained in $N \cup U$.

2. Any CDCL(T) run where the rules Restart and Forget are only applied finitely often terminates.

3. If $(\epsilon; N; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL(T)}} (M; N; U; k; s)$ then $N' \models_{\mathcal{T}} \text{atr}^{-1}(U)$.

4. If $(\epsilon; N; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL(T)}} (M; N; U; k; \bot)$ then N' is unsatisfiable.

5. If N is satisfiable, then any CDCL(T) run where the rules Restart and Forget are only applied finitely often eventually produces a success state $(M; N; U; -1; \top)$ with $\text{atr}^{-1}(M) \models_{\mathcal{T}} N'$.

Proof. 1. By contradiction. If the clause $\neg L_1 \lor \ldots \lor \neg L_n$ is already $N \cup U$ then after deciding/propagating $n-1$ of its literals either Propagate or eventually Conflict is applied to the clause by a reasonable strategy. This contradicts the application of $\mathcal{T}$-Conflict.

2. The proof of Lemma **??** carries over to CDCL(T), i.e., also clauses learned by rule Backtrack are not contained in $N \cup U$, even if $\mathcal{T}$-Propagate is applied. All learned clauses are restricted to atoms from N, so there are only finitely many such clauses that can be learned by Backtrack or $\mathcal{T}$-Conflict. Any run restricted to the rules Decide, Propagate, Skip, Resolve, $\mathcal{T}$-Propagate, and $\mathcal{T}$-Success terminates. Therefore, if Restart and Forget are only applied finitely often, CDCL(T) eventually terminates.

3. From CDCL any clause learned by Backtrack is entailed by N. For any clause C learned by $\mathcal{T}$-Conflict, $\text{atr}^{-1}(C)$ is a tautology in $\mathcal{T}$, and any clause $\text{atr}^{-1}(L_1 \land \ldots \land L_n \to L)$ used in $\mathcal{T}$-Propagate is a $\mathcal{T}$ tautology as well, hence it is also a $\mathcal{T}$ consequences of N'.

4. For all learned clauses C by 3. it holds $N' \models_{\mathcal{T}} \text{atr}^{-1}(C)$, hence $N' \models_{\mathcal{T}} \text{atr}^{-1}(\bot)$ and therefore N' is unsatisfiable.

5. Because of 2. and 4. it suffices to show that CDCL(T) can't get stuck. The CDCL rules terminate either in a state $(M; N; U; k; \top)$, $k \neq -1$, where neither Decide, Propagate nor Conflict is applicable, or in a state $(M; N; U; k; \bot)$, see Lemma **??**. The latter state implies unsatisfiability of N' and in the former state either $\mathcal{T}$-Success or $\mathcal{T}$-Conflict is applicable. Note that $\mathcal{T}$-Propagate can be simulated by the rules Decide and Propagate. $\square$