First-Order Logic Theories

3.17.1 Definition (First-Order Logic Theory)

Given a first-order many-sorted signature Σ, a *theory* $\mathcal{T}$ is a set of Σ-algebras.
For some first-order formula ϕ over Σ we say that ϕ is $\mathcal{T}$*-satisfiable* if there is some $\mathcal{A} \in \mathcal{T}$ such that $\mathcal{A}(\beta) \models \phi$ for some β. We say that ϕ is $\mathcal{T}$*-valid* ($\mathcal{T}$*-unsatisfiable*) if for all $\mathcal{A} \in \mathcal{T}$ and all β it holds $\mathcal{A}(\beta) \models \phi$ ($\mathcal{A}(\beta) \not\models \phi$). In case of validity I also write $\models_{\mathcal{T}} \phi$.

Alternatively, $\mathcal{T}$ may contain a set of satisfiable axioms which then stand for all algebras satisfying the axioms.

CDCL(T)

Consider a SAT problem where the propositional variables actually stand for ground atoms over some theory, ground equations or ground atoms of LRA, i.e., LRA atoms where all variables are existentially quantified. The basic idea is to apply CDCL, Section 2.9 in order to investigate the boolean structure of the problem. If CDCL derives unsatisfiable, then the problem clearly is. If CDCL derives satisfiable, then a ground decision procedure for the theory has to check whether the actual CDCL assignment constitutes also a model in the theory.

What must hold that an LRA formula in CNF is true?

$$
\begin{aligned}
&(\ldots \vee L_{1,j_1} \vee \ldots) \wedge \\
&(\ldots \vee L_{2,j_2} \vee \ldots) \wedge \\
&\vdots \\
&(\ldots \vee L_{n,j_n} \vee \ldots)
\end{aligned}
$$

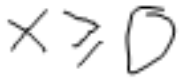
What must hold that an LRA formula in CNF is true?

$$
\begin{aligned}
&(\ldots \vee L_{1,j_1} \vee \ldots) \wedge \\
&(\ldots \vee L_{2,j_2} \vee \ldots) \wedge \\
&\vdots \\
&(\ldots \vee L_{n,j_n} \vee \ldots)
\end{aligned}
$$

There must exist an assignment β such that at least one literal per clause evaluates to true!

What must hold that an LRA formula in CNF is true?

$$(\ldots \vee L_{1,j_1} \vee \ldots) \wedge$$
$$(\ldots \vee L_{2,j_2} \vee \ldots) \wedge$$
$$\vdots$$
$$(\ldots \vee L_{n,j_n} \vee \ldots)$$



There must exist an assignment β such that at least one literal per clause evaluates to true!

Idea CDCL(T): Do simple things first!
1) Guess which literals should evaluate to true
2) Check if their conjunction has an assignment β
3) if not guess new literals.

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: $\models_{\mathcal{T}} L_1 \wedge \ldots \wedge L_n$. Let atr be a bijection from the atoms over $\Sigma_{\mathcal{T}}$ into propositional variables Σ_{PROP} auch that $\mathsf{atr}^{-1}(\mathsf{atr}(A)) = A$. Furthermore, atr distributes over the propositional operators, e.g., $\mathsf{atr}(\neg A) = \neg\, \mathsf{atr}(A)$.

Note: If $\mathsf{atr}(x \leq 5) = P$, then we want $\mathsf{atr}(x > 5) = \mathsf{atr}(\neg(x \leq 5)) = \neg P$

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: $\models_{\mathcal{T}} L_1 \wedge \ldots \wedge L_n$. Let atr be a bijection from the atoms over $\Sigma_{\mathcal{T}}$ into propositional variables Σ_{PROP} auch that $\mathsf{atr}^{-1}(\mathsf{atr}(A)) = A$. Furthermore, atr distributes over the propositional operators, e.g., $\mathsf{atr}(\neg A) = \neg\,\mathsf{atr}(A)$.

Note: If $\mathsf{atr}(x \leq 5) = P$, then we want
$\mathsf{atr}(x > 5) = \mathsf{atr}(\neg(x \leq 5)) = \neg P$

7.2.1 Lemma (Correctness of atr)

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

7.2.1 Lemma (Correctness of atr)

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

Example:

$$(\underbrace{z \geq 1}) \quad \vee \quad \underbrace{3x - y < 2})$$

$$\wedge \quad \underbrace{z < 1} \quad \wedge \quad \underbrace{3x - y \geq 2}$$

7.2.1 Lemma (Correctness of atr)

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

Example:

$$(\underbrace{z \geq 1}_{P} \quad \vee \quad \underbrace{3x - y < 2}_{Q})$$

$$\wedge \quad \underbrace{z < 1}_{\neg P} \quad \wedge \quad \underbrace{3x - y \geq 2}_{\neg Q}$$

FM (for comparison): DNF $\rightsquigarrow$ variable elimination $\rightsquigarrow$ unsat

7.2.1 Lemma (Correctness of atr)

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

Example:

$$(\underbrace{z \geq 1}_{P} \quad \vee \quad \underbrace{3x - y < 2}_{Q})$$

$$\wedge \quad \underbrace{z < 1}_{\neg P} \quad \wedge \quad \underbrace{3x - y \geq 2}_{\neg Q}$$

FM (for comparison): DNF $\rightsquigarrow$ variable elimination $\rightsquigarrow$ unsat

CDCL(T): $(P \vee Q) \wedge \neg P \wedge \neg Q$ is unsat in prop. logic

7.2.1 Lemma (Correctness of atr)

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

Example (Boolean Sat):
$[P, Q, \neg R, \ldots]$

7.2.1 Lemma (Correctness of atr)

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

Example (Boolean Sat):
$[P, Q, \neg R, \ldots]$

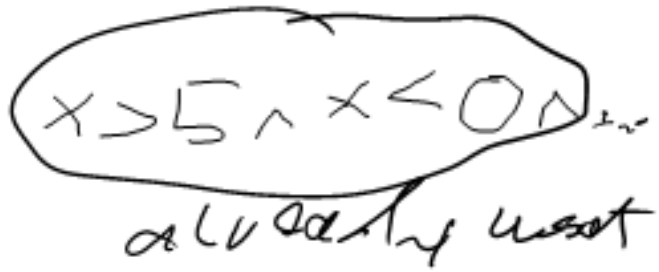
$\rightsquigarrow$ test conjunction of inequations
$\mathsf{atr}^{-1}(P) \wedge \mathsf{atr}^{-1}(Q) \wedge \mathsf{atr}^{-1}(\neg R) \wedge \ldots$

7.2.1 Lemma (Correctness of atr)

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

Example (Boolean Sat):
$[P, Q, \neg R, \ldots]$

$\rightsquigarrow$ test conjunction of inequations
$\mathrm{atr}^{-1}(P) \wedge \mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R) \wedge \ldots$



$\rightsquigarrow$ Simplex $\mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R)$ is conflict/unsat (see FailDepVar)

7.2.1 Lemma (Correctness of atr)

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

Example (Boolean Sat):
$[P, Q, \neg R, \ldots]$

$\rightsquigarrow$ test conjunction of inequations
$\mathrm{atr}^{-1}(P) \wedge \mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R) \wedge \ldots$

$\rightsquigarrow$ Simplex $\mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R)$ is conflict/unsat (see FailDepVar)

$\rightsquigarrow$ Learn $\neg Q \vee R$

7.2.1 Lemma (Correctness of atr)

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

Example (Boolean Sat):
$[P, Q, \neg R, \ldots]$

$\rightsquigarrow$ test conjunction of inequations
$\mathrm{atr}^{-1}(P) \wedge \mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R) \wedge \ldots$

$\rightsquigarrow$ Simplex $\mathrm{atr}^{-1}(Q) \wedge \mathrm{atr}^{-1}(\neg R)$ is conflict/unsat (see FailDepVar)

$\rightsquigarrow$ Learn $\neg Q \vee R$

$\rightsquigarrow$ Continue with SAT solver by analyzing the conflict and finding new partial model

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', $N = \mathsf{atr}(N')$, M a sequence of annotated propositional literals, U is a set of derived 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 $\mathsf{atr}^{-1}(M) \models_{\mathcal{T}} N'$, $\mathsf{atr}^{-1}(M)$ 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 $D \notin \{\top, \bot\}$

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

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

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

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

Forget $(M; N; U \uplus \{C\}; k; \top) \Rightarrow_{\mathsf{CDCL}} (M; N; U; k; \top)$
provided $M \not\models N$

Note that these rules are exactly the rules of CDCL from Section 2.9. 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 $(M; N; U; k; \top) \Rightarrow_{\text{CDCL(T)}} (M; N; U; -1; \top)$
provided $k \neq -1$, $M \models (N \cup U)$ and $\text{atr}^{-1}(M)$ is $\mathcal{T}$-satisfiable

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

$x \geq 0 \in M$

$x \geq 0 \models_{\mathcal{T}} x \geq c$ for $c < 0$

$\mathcal{T}$-Conflict $(M; N; U; k; \top) \Rightarrow_{\text{CDCL(T)}}$
$(\epsilon; N; U \cup \{\text{comp}(L_1) \vee \ldots \vee \text{comp}(L_n)\}; 0; \top)$
provided there are literals $L_1, \ldots, L_n$ from M with $\text{atr}^{-1}(L_1), \ldots, \text{atr}^{-1}(L_n) \models_{\mathcal{T}} \bot$

7.2.2 Definition (Reasonable CDCL(T) Strategy)

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

7.2.3 Theorem (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 $\mathrm{comp}(L_1) \vee \ldots \vee \mathrm{comp}(L_n)$ 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 $(\epsilon; N; \emptyset; 0; \top) \Rightarrow^*_{\mathrm{CDCL(T)}} (M; N; U; k; s)$ then $N' \models_{\mathcal{T}} \mathrm{atr}^{-1}(U)$.
4. If $(\epsilon; N; \emptyset; 0; \top) \Rightarrow^*_{\mathrm{CDCL(T)}} (M; N; U; k; \bot)$ 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 $\mathrm{atr}^{-1}(M) \models_{\mathcal{T}} N'$.