# 3.3 Ground Resolution

Dienstag, 9. Mai 2017 08:30

Drawbacks of Gilmore's Algorithm:

- How can one find a suitable instantiation of variables by ground terms? Sect 3.4
- How can one check satisfiability of a propositional formula efficiently? Sect 3.3

Resolution: main proof technique in logic prog.
First: ground resolution (for formulas without variables)

To check a formula $\forall X_1, \ldots, X_n \ \varphi$ in Skolem NF for unsatisfiability by resolution, one first has to transform $\varphi$ into <u>conjunctive normal form</u> (CNF).
Such formulas can then be represented as <u>clause sets</u>.

<u>Def 3.3.1</u> (CNF, Clause, Literal)

A formula $\varphi$ is in <u>conjunctive normal form</u> iff it is quantifier-free and has the form

$$(L_{1,1} \vee \ldots \vee L_{1,n_1}) \wedge \ \ldots \ \wedge (L_{m,1} \vee \ldots \vee L_{m,n_m}).$$

Here, $L_{i,j}$ are <u>literals</u>, i.e., atomic formulas or negated atomic formulas of the form $p(t_1,\ldots,t_n)$ or $\neg p(t_1,\ldots,t_n)$.
For every literal $L$, its negation $\overline{L}$ is defined as

$$\overline{L} = \begin{cases} \neg A, & \text{if } L = A \in At(\Sigma, \Delta, \mathcal{V}) \\ A & \text{if } L = \neg A,\ A \in At(\Sigma,\Delta,\mathcal{V}) \end{cases}$$

A <u>clause</u> is a set of literals and it represents the universally quantified disjunction of the literals.

A clause set represents the conjunction of its clauses.

So every formula $\varphi$ in CNF can be represented as a clause set

$$\mathcal{K}(\varphi) = \{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{m,1}, \dots, L_{m,n_m}\}\}.$$

Therefore, we also speak of satisfiability and entailment of clause sets.

The empty clause is denoted $\Box$. It is unsatisfiable by definition (empty disjunction).

**Thm 332** (Transformation to CNF)

Every quantifier-free formula $\varphi$ can be transformed into an equivalent formula $\varphi'$ in CNF automatically.

Proof: First replace all sub-formulas $\varphi_1 \leftrightarrow \varphi_2$ by
$(\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$.

Then replace all sub-formulas $\varphi_1 \rightarrow \varphi_2$ by $\neg\varphi_1 \vee \varphi_2$.

Afterwards, use the following algorithm CNF:

- Input: $\varphi$ (quantifier-free, without $\leftrightarrow$ or $\rightarrow$)
- Output: equivalent formula in CNF

- If $\varphi$ is an atomic formula, then return $\varphi$.
- If $\varphi = \varphi_1 \wedge \varphi_2$, then return $CNF(\varphi_1) \wedge CNF(\varphi_2)$.
- If $\varphi = \varphi_1 \vee \varphi_2$, then compute

$$CNF(\varphi_1) = \varphi_1' \wedge \dots \wedge \varphi_{m_1}'$$
$$CNF(\varphi_2) = \varphi_1'' \wedge \dots \wedge \varphi_{m_2}''$$

← $(\varphi_1' \wedge \dots \wedge \varphi_{m_1}') \vee (\varphi_1'' \wedge \dots \wedge \varphi_{m_2}'')$

Return $\displaystyle\bigwedge_{\substack{i \in \{1,\dots,m_1\} \\ j \in \{1,\dots,m_2\}}} \varphi_i' \vee \varphi_j''$

← use the distributivity law

- If $\varphi = \neg \varphi_1$, then compute

$$CNF(\varphi_1) = \bigwedge_{i \in \{1,\dots,m\}} \left( \bigvee_{j \in \{1,\dots,n_i\}} L_{i,j} \right)$$

De Morgan's law states that the negation of this formula is

$$\bigvee_{i \in \{1,\dots,m\}} \left( \bigwedge_{j \in \{1,\dots,n_i\}} \overline{L_{i,j}} \right)$$

Due to the distributivity law, we return

$$\bigwedge_{\substack{j_1 \in \{1,\dots,n_1\}, \\ \vdots \\ j_m \in \{1,\dots,n_m\}}} \left( \overline{L_{1,j_1}} \vee \dots \vee \overline{L_{m,j_m}} \right)$$

$\boxtimes$

Ex 333 Let $p, q, r \in \Delta_0$.
Transform the following formula into CNF:

$$\neg(\neg p \wedge (\neg q \vee r))$$

$$\downarrow \text{De Morgan}$$

$$p \vee (q \wedge \neg r)$$

$$\downarrow \text{Distributivity}$$

$$(p \vee q) \wedge (p \vee \neg r)$$

Alg. of Gilmore: To check unsatisf. of $\varphi$, consider $E(\varphi)$ and prove unsatisf. of these propositional formulas.

Therefore: Now introduce a technique to prove unsatisfiability of prop. formulas in CNF.

In other words: Prove unsatisf. of clause sets without variables.

Resolution: $(L_1 \vee L) \wedge (L_2 \vee \overline{L})$

implies $L_1 \vee L_2$

Def 334 (Propositional Resolution) — Slide 13

Let $K_1, K_2$ be two clauses without variables.
Then the clause $R$ is a resolvent of $K_1$ and $K_2$ iff
there exists a $L \in K_1$ with $\overline{L} \in K_2$
and $R = (K_1 \setminus \{L\}) \cup (K_2 \setminus \{\overline{L}\})$.

For a clause set $\mathcal{K}$ we define

$$Res(\mathcal{K}) = \mathcal{K} \cup \{R \mid R \text{ is resolvent of two clauses from } \mathcal{K}\}$$

We define

$$Res^0(\mathcal{K}) = \mathcal{K}$$

$$Res^{n+1}(\mathcal{K}) = Res(Res^n(\mathcal{K})) \quad \text{for all } n \geq 0$$

Moreover: $Res^*(\mathcal{K}) = \bigcup_{n \geq 0} Res^n(\mathcal{K})$

---

Idea: Construct $Res^*(\mathcal{K})$ until one obtains $\Box$. Since adding resolvents is equivalence-preserving, this means that $\mathcal{K}$ is unsatisfiable.

Clearly: $\Box \in Res^*(\mathcal{K})$ iff

there is a sequence of clauses $K_1, \ldots, K_m$ with $K_m = \Box$
where for all $1 \leq i \leq m$, we have

- $K_i \in \mathcal{K}$ or

- $K_i$ is a resolvent of $K_j, K_k$ for $j, k < i$.

To denote resolution proofs:

$$\begin{tikzcd}[column sep=small] K_1 \arrow[dr, no head] & & K_2 \arrow[dl, no head] \\ & R & \end{tikzcd}$$

means that $R$ is resolvent of $K_1$ and $K_2$.

Ex. 335 Let $\Delta_1 = \{p, q\}$, $\Sigma_0 = \{a, b\}$, $\Sigma_1 = \{f\}$, $\Sigma_2 = \{g\}$.
We regard the clause set $\mathcal{K} = \{K_1, K_2, K_3, K_4\}$ with

Slide 13

$$\begin{tikzcd}[column sep=tiny] \underset{K_1}{\{\neg p(f(a)), q(b)\}} \arrow[dr, no head] & & \underset{K_2}{\{p(f(a))\}} \arrow[dl, no head] & \underset{K_3}{\{p(g(b,a))\}} \arrow[dr, no head] & & \underset{K_4}{\{\neg p(g(b,a)), \neg q(b)\}} \arrow[dl, no head] \\ & \{q(b)\} \arrow[drr, no head] & & & \{\neg q(b)\} \arrow[dl, no head] & \\ & & & \Box & & \end{tikzcd}$$

The resolution calculus is sound and complete:
Completeness: If $\mathcal{K}$ is unsat., then $\Box \in Res^*(\mathcal{K})$.
Soundnesss: If $\Box \in Res^*(\mathcal{K})$, then $\mathcal{K}$ is unsat..

To prove soundness, we need the following lemma.

Lemma 336 (Propositional Resolution Lemma)
Let $\mathcal{K}$ be a set of clauses without variables.
If $K_1, K_2 \in \mathcal{K}$ and $R$ is a resolvent of $K_1$ and $K_2$, then $\mathcal{K}$ and $\mathcal{K} \cup \{R\}$ are equivalent.

Proof: "$\Leftarrow$": Every structure that satisfies $\mathcal{K} \cup \{R\}$ also satisfies $\mathcal{K}$. Thus: $\mathcal{K} \cup \{R\} \models \mathcal{K}$

"$\Rightarrow$": Let $S$ be a structure with $S \models \mathcal{K}$.

Let $L \in K_1$, $\overline{L} \in K_2$, $R = (K_1 \setminus \{L\}) \cup (K_2 \setminus \{\overline{L}\})$.

Assume $S \not\models \mathcal{K} \cup \{R\}$. Thus: $S \not\models R$

If $S \models L$, then $S \models K_2$ implies $S \models K_2 \setminus \{\overline{L}\}$

Therefore: $S \models R$ $\lightning$

If $S \not\models L$, then $S \models \overline{L}$ and $S \models K_1$ implies $S \models K_1 \setminus \{L\}$

Therefore: $S \models R$. $\lightning$

$\boxtimes$

Thm 3.3.7 (Soundness and Completeness of Prop. Resolution)

Let $\mathcal{K}$ be a set of clauses without variables.

Then $\mathcal{K}$ is unsatisfiable iff $\square \in Res^*(\mathcal{K})$.

Proof: Soundness "$\Leftarrow$":

By the resolution lemma 3.3.6., $\mathcal{K}$ and $Res(\mathcal{K})$ are equivalent. By induction on $n$, one can show that $\mathcal{K}$ and $Res^n(\mathcal{K})$ are equivalent for all $n \in \mathbb{N}$.

$\square \in Res^*(\mathcal{K})$

$\rightsquigarrow$ there exists an $n \in \mathbb{N}$ with $\square \in Res^n(\mathcal{K})$

$\rightsquigarrow$ $Res^n(\mathcal{K})$ is unsat.

$\rightsquigarrow$ $\mathcal{K}$ is unsat.

Completeness "$\Rightarrow$"

If $\mathcal{K}$ is unsatisfiable, then there is a finite subset $\mathcal{K}' \subseteq \mathcal{K}$ which is also unsat. (by the Compactness thm. of prop. logic).

Let $n$ be the number of different atomic formulas in $\mathcal{K}'$. We use induction on $n$.

Ind. Base: $n=0$

There are only two clause sets without atomic formulas:

$\mathcal{K}' = \emptyset$ ← empty conjunction: valid, i.e., true in every interpretation ↯

$\mathcal{K}' = \{\square\}$. Then $\square \in Res^{0}(\mathcal{K}') \subseteq Res^{0}(\mathcal{K})$.

Ind. Step: $n>0$

Let $A$ be an atomic formula that occurs in the unsat. clause set $\mathcal{K}'$.

Let $\mathcal{K}^+$ result from $\mathcal{K}'$ by
- removing all clauses that contain literal $A$
- drop $\neg A$ from all remaining clauses

Thus: $\mathcal{K}^+ = \{ K \setminus \{\neg A\} \mid K \in \mathcal{K}', A \notin K \}$

Similarly $\mathcal{K}^- = \{ K \setminus \{A\} \mid K \in \mathcal{K}', \neg A \notin K \}$

$\mathcal{K}^+$ is unsat.: If $S \models \mathcal{K}^+$, then extend $S$ to a structure $S'$ with $S' \models A$.
Then $S' \models \mathcal{K}'$ ↯ to the unsat. of $\mathcal{K}'$

Similarly, $\mathcal{K}^-$ is unsat.

Since $\mathcal{K}^+$ and $\mathcal{K}^-$ do not contain $A$, we can apply the ind. hypothesis, which yields:

$\square \in Res^{*}(\mathcal{K}^+)$, $\square \in Res^{*}(\mathcal{K}^-)$.

$\square \in Res^{*}(\mathcal{K}^+)$ means that there is a sequence $K_1, \dots, K_m$ with $\square = K_m$ where for all $1 \le i \le m$:
- $K_i \in \mathcal{K}^+$ or
- $K_i$ is resolvent of $K_j$ and $K_k$ for $j,k < i$

- $K_i$ is resolvent of $K_j$ and $K_k$ for $j, k < i$

If all these $K_i$ are also contained in $\mathcal{K}'$, then we have proved $\square \in Res^*(\mathcal{K}') \subseteq Res^*(\mathcal{K})$.

Otherwise: add $\neg A$ again to all clauses where it had been removed.

Then: Obtain a resolution proof for

$\{\neg A\} \in Res^*(\mathcal{K}')$

Reason:

$$\begin{tikzcd}[column sep=small] C_j \arrow[dr, dash] & & C_k \arrow[dl, dash] \\ & C_i & \end{tikzcd} \qquad \begin{tikzcd}[column sep=small] C_j \cup \{\neg A\} \arrow[dr, dash] & & C_k \cup \{\neg A\} \arrow[dl, dash] \\ & C_i \cup \{\neg A\} & \end{tikzcd}$$

Similarly: $\square \in Res^*(\mathcal{K}^-)$ implies

$\square \in Res^*(\mathcal{K}')$ or

$\{A\} \in Res^*(\mathcal{K}')$.

Thus: $\{A\}, \{\neg A\} \in Res^*(\mathcal{K}')$

implies $\square \in Res^*(\mathcal{K}')$

$$\begin{tikzcd}[column sep=small] \vdots & & \vdots \\ \{\neg A\} \arrow[dr, dash] & & \{A\} \arrow[dl, dash] \\ & \square & \end{tikzcd}$$

$\boxed{q}$

Now the alg. of Gilmore can be improved to the <u>ground resolution algorithm.</u>

Slide 14

Sound: if alg. ...

sound: if alg. returns "true", then $\{\varphi_1, \dots, \varphi_k\} \models \varphi$

complete: if $\{\varphi_1, \dots, \varphi_k\} \models \varphi$, then alg. terminates and returns "true"

if $\{\varphi_1, \dots, \varphi_k\} \not\models \varphi$, then the alg. might not terminate

$\Rightarrow$ Semi-decision procedure

Step 5: Advantage over Gilmore's alg.

$\Rightarrow$ better check for unsat. of propositional clause sets

Step 4: Still inefficient, since we don't know how to instantiate variables by ground terms in a "clever" way.

Ex 338: Show unsatisfiability of

$$\forall X, Y \ \underbrace{(\neg p(X) \vee \neg p(f(a)) \vee q(Y)) \wedge p(Y) \wedge (\neg p(g(b,X)) \vee \neg q(b))}_{\psi}$$

Corresponding clause set $\mathfrak{K}(\psi)$:

$\{\{\neg p(X), \neg p(f(a)), q(Y)\}, \{p(Y)\}, \{\neg p(g(b,X)), \neg q(b)\}\}$

$K_1: [X/f(a), Y/b]$ $\quad K_2: [Y/f(a)]$ $\quad K_3: [Y/g(b,a)]$ $\quad K_4: [X/a]$

$\{\neg p(f(a)), q(b)\}$ $\quad \{p(f(a))\}$ $\quad \{p(g(b,a))\}$ $\quad \{\neg p(g(b,a)), \neg q(b)\}$

$\{q(b)\}$ $\quad \{\neg q(b)\}$

$\Box$

- Instantiations can unify several literals of the same clause.
- We can use several different instantiations of the same

clause.

How can one find such suitable instantiations automatically?