3.1 Skolem Normal Form

Freitag, 28. April 2017 13:00

3. Resolution

Goal: Check $\Phi \models \varphi$ automatically

The definition of entailment is not suitable for automation, because one would have to check all (infinitely many) interpretations I and find out whether $I \models \Phi$ implies $I \models \varphi$.

Instead: develop a calculus that allows to prove $\Phi \models \varphi$ in a syntactical (automatable) way.

Calculus is <u>sound</u> if: if the calculus can deduce φ from Φ, then $\Phi \models \varphi$ really holds

Calculus is <u>complete</u> if: whenever $\Phi \models \varphi$, then the calculus can deduce φ from Φ.

logic prog. uses the <u>resolution calculus</u>, which is indeed sound + complete.

Idea: instead of entailment ($\Phi \models \varphi$), we examine an unsatisfiability problem

Lemma 3.0.1 (From Entailment to Unsatisfiability)

Let $\varphi_1, \dots, \varphi_k, \varphi \in \mathcal{F}(\Sigma, \Delta, \mathcal{V})$.

Then we have $\{\varphi_1, \dots, \varphi_k\} \models \varphi$ iff

the formula $\varphi_1 \wedge \dots \wedge \varphi_k \wedge \neg\varphi$ is unsatisfiable.

Proof: $\{\varphi_1, \dots, \varphi_k\} \models \varphi$

iff for all interpretations I with $I \models \{\varphi_1, \dots, \varphi_k\}$, we have $I \models \varphi$

iff there is no interpretation I with $I \models \{\varphi_1, \dots, \varphi_k\}$ and $I \models \neg\varphi$

iff $\varphi_1 \wedge \dots \wedge \varphi_k \wedge \neg\varphi$ is unsatisfiable. $\square$

Ex. 3.0.2

To show that in the logic prog. with the fact

mother0f(ren, sus)

the query ?- mother0f(X, sus)

holds, one has to show:

$\{\text{motherOf}(\text{ren}, \text{sus})\} \models \exists X \ \text{motherOf}(X, \text{sus})$

Instead, one can show unsatisfiability of

$\{motherOf(ren, sus), \neg \exists X\ motherOf(X, sus)\}$

In general: Unsatisfiability of logic formulas is undecidable. (Thus: Entailment is also undecidable)

This means: There is no terminating algorithm that can determine whether $\Phi \models \varphi$ holds or not.

But: Unsatisfiability (and entailment) is semi-decidable.

This means: There is an algorithm which terminates whenever $\Phi \models \varphi$ holds (and which determines that $\Phi \models \varphi$ holds). But if $\Phi \not\models \varphi$, then the algorithm might not terminate.

The resolution calculus is such a semi-decision algorithm for unsatisfiability.

Ex: empty prog.

query: $?- \exists X\ motherOf(X, sus)$

To check: $\models \exists X\ motherOf(X, sus)$

Equivalently: $\{\neg \exists X\ motherOf(X, sus)\}$ unsatisfiable?

Goal: Introduce technique to check unsatisfiability of formulas automatically.

3.1. Skolem Normal Form

Aim: Simplify any formula to the following form:

$$\forall X_1, \dots, X_n \; \psi$$

where ψ is quantifier-free and $\mathcal{V}(\psi) \subseteq \{X_1, \dots, X_n\}$.

First step: Transform any formula to <u>prenex normal form</u>.

<u>Def 3.1.1</u> (Prenex Normal Form)

A formula φ is in <u>prenex normal form</u> iff it has the form $Q_1 X_1 \; Q_2 X_2 \dots Q_n X_n \; \psi$ with $Q_1, \dots, Q_n \in \{\forall, \exists\}$ and ψ is quantifier-free.

<u>Thm 3.1.2</u> (Transformation into prenex normal form)

For every formula φ one can automatically construct an equivalent formula φ' in prenex normal form.

<u>Proof</u>:

- Replace all sub-formulas $\varphi_1 \longleftrightarrow \varphi_2$ by

$(\varphi_1 \to \varphi_2) \land (\varphi_2 \to \varphi_1)$

- Replace all sub-formulas $\varphi_1 \to \varphi_2$ by $\neg\varphi_1 \lor \varphi_2$.

Then we apply the following algorithm PRENEX:

* If φ is quantifier-free, then return φ.
* If $\varphi = \neg\varphi_1$, then compute $PRENEX(\varphi_1) = Q_1 X_1 \dots Q_n X_n\ \psi$.
  Then return
  $\overline{Q_1} X_1 \dots \overline{Q_n} X_n \neg\psi$,
  where $\overline{\forall} = \exists$, $\overline{\exists} = \forall$

  ($\neg \forall X_1\ \psi$ equiv. to $\exists X_1 \neg \psi$
  $\neg \exists X_1\ \psi$ — " — $\forall X_1 \neg\psi$)

* If $\varphi = \varphi_1 \cdot \varphi_2$ where $\cdot \in \{\land, \lor\}$, then compute
  $PRENEX(\varphi_1) = Q_1 X_1 \dots Q_n X_n\ \psi_1$
  $PRENEX(\varphi_2) = R_1 Y_1 \dots R_m Y_m\ \psi_2$
  By renaming bound variables, ensure that
  $X_1, \dots, X_n$ do not occur in
  $R_1 Y_1 \dots R_m Y_m\ \psi_2$
  and that $Y_1, \dots, Y_m$ do not occur in $Q_1 X_1 \dots Q_n X_n \psi_1$.
  Then return:
  $Q_1 X_1 \dots Q_n X_n R_1 Y_1 \dots R_m Y_m\ (\psi_1 \cdot \psi_2)$.

  ($\forall X\ \psi_1 \land \exists X\ \psi_2$
  $\Rightarrow$ rename bound variables)

* If $\varphi = Q X\ \varphi_1$ with $Q \in \{\forall, \exists\}$.
  Then compute $PRENEX(\varphi_1) = Q_1 X_1 \dots Q_n X_n\ \psi$.
  By renaming bound variables, ensure that

$X_1, \dots, X_n$ are different from X.

Then return $Q X \, Q_1 X_1 \dots Q_n X_n \; \varphi$.

[5]

Ex. 3.1.3 Transform the following formula into prenex normal form:

$$\neg \exists X \, (\mathit{married}(X,Y) \vee \underbrace{\neg \exists Y \, \mathit{motherOf}(X,Y)}))$$

$$\underbrace{\forall Y \; \neg \mathit{motherOf}(X,Y)}$$

$$\forall Z \; \neg \mathit{motherOf}(X,Z)$$

$$\underbrace{\neg \exists X \, \forall Z \, (\mathit{married}(X,Y) \vee \neg \mathit{motherOf}(X,Z))}$$

$$\forall X \, \exists Z \, \neg (\mathit{married}(X,Y) \vee \neg \mathit{motherOf}(X,Z))$$

Ex 3.1.4

Log. Prog with the fact $\mathit{motherOf}(\mathit{ren}, \mathit{sus})$.

Query $?- \mathit{motherOf}(X, \mathit{sus})$.

We have to show unsatisfiability of

$$\mathit{motherOf}(\mathit{ren}, \mathit{sus}) \wedge \neg \exists X \, \mathit{motherOf}(X, \mathit{sus}).$$

This can be transformed to prenex normal form:

$$\underbrace{\mathit{motherOf}(\mathit{ren}, \mathit{sus}) \wedge \forall X \, \neg \mathit{motherOf}(X, \mathit{sus})}$$

$$\forall X \, (\mathit{mother}(\mathit{ren}, \mathit{sus}) \wedge \neg \mathit{motherOf}(X, \mathit{sus}))$$

Def 3.1.5 (Skolem Normal Form)

A formula φ is in Skolem normal form iff it is closed and it has the form $\forall X_1, \ldots, X_n \; \psi$, where ψ is quantifier-free.

For every formula, there is an equivalent formula in prenex normal form.

This is not true for Skolem normal forms.

For example, $female(X)$ or $\exists X \; female(X)$ have no equivalent formula in Skolem normal form.

We only need that the original formula is satisfiable iff the corresponding formula in Skolem normal form is satisfiable.

In the example:

$$\exists X \; female(X) \quad \text{and} \quad female(a)$$

are satisfiability-equivalent

fresh function symbol of arity 0

$$\forall Y \exists X \; married(X,Y) \quad \text{and} \quad \forall Y \; married(f(Y), Y)$$

fresh function symbol of

arity 1

Thm 3.1.6 (Transformation into Skolem Normal Form)
For every formula φ, one can automatically construct a formula φ' in Skolem normal form such that φ is satisfiable iff φ' is satisfiable.

Proof: First, φ is transformed to prenex normal form as in Thm 3.1.2. This results in a formula φ_1.
Let $X_1, \dots, X_n$ be the free variables of φ_1.
Then transform φ_1 to $\varphi_2 = \exists X_1, \dots, X_n \; \varphi_1$.
Clearly φ_2 and φ_1 are satisfiability-equivalent:

$I \models \varphi_1$ for $I = (\mathcal{A}, \alpha, \beta)$

$\rightsquigarrow \; I[\![X_1/\beta(X_1), \dots, X_n/\beta(X_n)]\!] \models \varphi_1$

$\rightsquigarrow \; I \models \underbrace{\exists X_1, \dots, X_n \; \varphi_1}_{\varphi_2}$.

$I \models \varphi_2$ with $I = (\mathcal{A}, \alpha, \beta)$

$\rightsquigarrow \; I \models \exists X_1, \dots, X_n \; \varphi_1$

$\rightsquigarrow$ there exist $a_1, \dots, a_n \in \mathcal{A}$ with $I[\![X_1/a_1, \dots, X_n/a_n]\!] \models \varphi_1$

$\rightsquigarrow \; \varphi_1$ is satisfiable.

Now φ_2 is a closed formula in prenex normal form.

We eliminate the existential quantifiers from the outside to the inside.

If φ_2 is $\forall X_1, \dots, X_n \exists Y \psi$,

then replace it by $\forall X_1, \dots, X_n \; \psi[Y / f(X_1, \dots, X_n)]$

where f is a fresh function symbol of arity n.

One can prove that this does not change satisfiability of the formula (by substitution lemma 2.2.3).

Notation:
If $\sigma = \{X_1/t_1, \dots, X_n/t_n\}$, then we often write $\varphi[X_1/t_1, \dots, X_n/t_n]$ instead of $\sigma(\varphi)$.

$\boxed{4}$

Ex 3.1.7 $\neg \exists X (\text{married}(X,Y) \vee \neg \exists Y \text{ motherOf}(X,Y))$

$\downarrow$ Prenex normal form (Ex. 3.1.3)

$\forall X \exists Z \neg (\text{married}(X,Y) \vee \neg \text{motherOf}(X,Z))$

$\downarrow$ get rid of free var. Y

$\exists Y \forall X \exists Z \neg (\text{married}(X,Y) \vee \neg \text{motherOf}(X,Z))$

$\downarrow$ replace Y by fresh $a \in \Sigma_0$

$\forall X \exists Z \neg (\text{married}(X,a) \vee \neg \text{motherOf}(X,Z))$

$\downarrow$ replace Z by fresh $f \in \Sigma_1$

$$\forall X \quad \neg(\mathit{married}(X,a) \lor \neg \mathit{motherOf}(X, f(X)))$$