Mathematical Logic 2016

Lecture 16: Normal forms and Resolution theorem proving

Instructor: Ashutosh Gupta

TIFR, India

Compile date: 2016-10-01

Where are we and where are we going?

We have seen

- Syntax, semantics, and complete proof methods for FOL

We will start looking at the architecture of a modern automated theorem prover

Modern prover architecture

A modern FOL refutation prover runs in the following two steps

- Normalize the formula in a form
- Then, runs resolution with unification

Topic 16.1

Normal forms

Replacement theorem

Theorem 16.1

Let A be a **S***-atom. Let $F(A)$,G,H are* **S***-formulas. Let m be a* **S***-model. If*

$$m \models G \Leftrightarrow H$$

then

$$m \models F(G) \Leftrightarrow F(H).$$

Proof.

Straight forward structural induction □

Exercise 16.1

Spell out the details of the proof.

Positive occurrence

Definition 16.1

*Let A be a **S**-atom and $F(A)$ is a **S**-formula.*
A is positively occurs in $F(A)$ if

- $F = A$,
- $F(A) = \neg\neg G(A)$ *and A positively occurs in* $G(A)$,
- $F = \alpha$ *and A positively occurs in* α_1 *and* α_2,
- $F = \beta$ *and A positively occurs in* β_1 *and* β_2,
- $F = \gamma/\delta$ *and A positively occurs in* $\gamma(x)/\delta(x)$,

Example 16.1

Is $R(x, y)$ positively occurs in the following formula?

1. $R(x, y)$
2. $\neg R(x, y)$
3. $\exists x.(\neg R(x, y) \Rightarrow P(x))$
4. $\exists x.(\neg R(x, y) \Rightarrow \neg R(y, x))$

Exercise 16.2

Define negative occurrence.

Implication replacement theorem

Theorem 16.2

Let A be a **S***-atom. Let $F(A)$, G,H are* **S***-formulas and A positively occurs in $F(A)$. Let m be a* **S***-model.*
If

$$m \models G \Rightarrow H$$

then

$$m \models F(G) \Rightarrow F(H).$$

Proof.

Again straight forward structural induction. □

Normalization steps

A modern FOL refutation prover first applies the following transformations

- Rename apart : rename variables for each quantifier
- Prenex : bringing quantifiers to front
- Skolemization: remove existential quantifiers (only sat preserving)
- CNF transformation: turn the internal quantifier free part of the formula into CNF
- Syntactical removal of universal quantifiers: a CNF with free variables.

Rename apart

Definition 16.2

A formula F is renamed apart if no quantifier in F use a variable that is used by another quantifier or occurs as free variable in F.

Due to the theorems like the following, we can safely assume that every quantifier has different variable. If that is not the case then we can rename quantified variables apart.

Theorem 16.3

*Let F is a **S**-formulas and y does not occur in F.*

$$\models \forall x.F \Leftrightarrow \forall y.F\{x \mapsto y\}$$

Exercise 16.3

Rename apart the following formulas

- $\neg(\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$

Prenex form

Definition 16.3

A formula F is in prenex form if all the quantifiers of the formula occur as prefix of F. The quantifier-free suffix of F is called matrix of F.

Due to the following equivalences, we can always move quantifiers in front of the formulas.

- $\neg(\exists x.F) \equiv \forall x.\neg F$
- $\neg(\forall x.F) \equiv \exists x.\neg F$
- $\forall x.F \wedge G \equiv \forall x.(F \wedge G)$
- $\exists x.F \wedge G \equiv \exists x.(F \wedge G)$
- $F \wedge \forall x.G \equiv \forall x.(F \wedge G)$
- $F \wedge \exists x.G \equiv \exists x.(F \wedge G)$
- $\forall x.F \vee G \equiv \forall x.(F \vee G)$
- $\exists x.F \vee G \equiv \exists x.(F \vee G)$
- $F \vee \forall x.G \equiv \forall x.(F \vee G)$
- $F \vee \exists x.G \equiv \exists x.(F \vee G)$
- $\forall x.F \Rightarrow G \equiv \exists x.(F \Rightarrow G)$
- $\exists x.F \Rightarrow G \equiv \forall x.(F \Rightarrow G)$
- $F \Rightarrow \forall x.G \equiv \forall x.(F \Rightarrow G)$
- $F \Rightarrow \exists x.G \equiv \exists x.(F \Rightarrow G)$

Exercise 16.4

Convert $\neg(\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ into prenex form

Skolemization

Skolemization removes existential quantifiers from prenex formulas

Theorem 16.4

*Let F be a **S**-formula, $FV(F) = \{x, y_1, \dots, y_n\}$ and $f/n \in \mathbf{F}$ does not occur in F. For every model m', there is a f-variant model m s.t.*

$$m \models \exists x.F \Rightarrow F\{x \mapsto f(y_1, \dots, y_n)\}$$

Proof.

Let $f' \triangleq f_{m'}$.

Now for each ν s.t. $\nu(y_1) = d_1, .., \nu(y_n) = d_n$.

If $m, \nu \models \exists x.F$, choose $d \in D_{m'}$ s.t. $m', \nu[x \to d] \models F$.

Otherwise, choose any d.

$f' \triangleq f'[(d_1, .., d_n) \mapsto d]$.

Let us define $m = m'[f \mapsto f']$.

Since f does not occur in F, if $m, \nu \models \exists x.F$ then $m', \nu \models \exists x.F$.

Due to construction of m, $m, \nu \models F\{x \mapsto f(y_1, \dots, y_n)\}$(why?). □

Exercise 16.5

Show there us an m s.t. $m \models F\{x \mapsto f(y_1, \dots, y_n)\} \Rightarrow \forall x.F$

Skolemization(contd.)

Theorem 16.5

Let F be a **S***-formula with $FV(F) = \{x, y_1, \dots, y_n\}$. Let $G(A)$ be a* **S***-formula in which atom A occurs positively and $G(\exists x.F(x))$ is a sentence. Let $f/n \in$* **F** *s.t. f does not occur in $F(x)$ and $G(A)$.*

$$G(\exists x.F(x)) \text{ is sat iff } G(F(f(y_1, \dots, y_n))) \text{ is sat}$$

Proof.

If $m, \nu \models F(f(y_1, \dots, y_n))$ then there is a d for every $\nu(y_1), \dots, \nu(y_n)$ such that $m, \nu[x \mapsto f_m(\nu(y_1), ..\nu(y_n))] \models F(x)$.
Therefore, $F(f(y_1, \dots, y_n)) \Rightarrow \exists x.F(x)$ is valid.
Due to implication replacement theorem, $G(F(f(y_1, \dots, y_n))) \Rightarrow G(\exists x.F(x))$ is valid.

For the other direction, we need to adjust interpretation of f in the model.
Let $m' \models G(\exists x.F(x))$. Due to the previous theorem we can obtain m s.t. $m \models \exists x.F(x) \Rightarrow F(f(y_1, \dots, y_n))$ and $m \models G(\exists x.F(x))$.
Again due to implication replacement theorem, $m \models G(F(f(y_1, .., y_n)))$. □

Skolemization of prenex formula

Since all the quantifiers occur positively in prenex form, all $\exists$s can be removed using skolem functions.

Skolemization should be done from out to inside,i.e., remove outermost $\exists$ first.

Exercise 16.6

Skolemize the following formula

$\exists x.\forall y \exists z \forall w. \neg(R(x,y) \Rightarrow R(w,z))$

FOL CNF

Consider the following skolemized prenex formula,

$$\forall x_1, \dots, x_n.\ F.$$

Since F is quantifier free, we may convert F into CNF, preferably using Tseitin encoding and obtain

$$\forall x_1, \dots, x_n.\ C_1 \wedge \dots \wedge C_k.$$

Since $\forall$ distributes over $\wedge$, we may obtain

$$(\forall x_1, \dots, x_n.\ C_1) \wedge \dots \wedge (\forall x_1, \dots, x_n.\ C_k).$$

We may rename apart variables in each of the above clauses and obtain

$$(\forall x_{11}, \dots, x_{1n}.\ C'_1) \wedge \dots \wedge (\forall x_{k1}, \dots, x_{kn}.\ C'_k)$$

We may view the above formula as conjunction of clauses

$$C'_1 \wedge \dots \wedge C'_k,$$

From here the real theorem proving begins.

without any explicit mention of quantifiers.

It assumes that all the free variables are universally quantified.

Topic 16.2

Resolution theorem proving

Resolution theorem proving

Input: a set of clauses

Base inference rules:

$$\textsc{Res}\frac{\neg F \vee C \quad G \vee D}{(C \vee D)\sigma}\sigma = mgu(F, G)$$

$$\textsc{Factoring}\frac{L_1 \vee .. \vee L_k \vee C}{(L_1 \vee C)\sigma}\sigma = mgu(L_1, .., L_k)$$

$$\textsc{Paramodulation}\frac{s \approx t \vee C \quad D[u]}{(C \vee D[t])\sigma}\sigma = mgu(s, u)$$

$$\textsc{Relexivity}\frac{t \not\approx u \vee C}{C\sigma}\sigma = mgu(t, u)$$

Issues:

- Saturation based proving
- Redundancies and deletion
- Soundness and completeness

Example: why FACTORING rule

Clauses may use same variable names. However, before applying inference rules we need to rename them apart.

Example 16.2

1. $P(x) \vee P(y)$
2. $\neg P(x) \vee \neg P(y)$ // *input*
3. $P(x) \vee \neg P(y)$ //RESOLUTION *on 1 and 2*
4. $P(x)$ //FACTORING *on 1*
5. $\neg P(x)$ //FACTORING *on 2*
6. $\bot$ //RESOLUTION *on 4 and 5*

No progress without FACTORING

Example: another resolution proof

1. $\neg R(c, y)$
2. $R(w, f(y))$ // input
3. $\bot$ //RESOLUTION on 1 and 2

Redundancies

Example 16.3

Consider the following clauses

1. $a \approx c$
2. $b \approx d$
3. $P(a, b)$
4. $\neg P(c, d)$ //*input*
5. $P(c, b)$ //Paramodulation *on 1 and 3*
6. $P(a, d)$ //Paramodulation *on 2 and 3*
7. $P(c, d)$ //Paramodulation *on 2 and 5*
8. $\perp$ //Resolution *on 4 and 7*

Redundant derivation

- Many clauses can be derived due to simple permutations
- Often derived clauses do not add new information
- We need to restrict application of the rules by imposing order

For a while we will ignore Paramodulation and Relexivity.
We will deal with them later.

Topic 16.3

Problems

FOL CNF

Exercise 16.7

Convert the following formula in FOL CNF

$$\exists z.(\exists x.Q(x,z) \vee \exists x.P(x)) \Rightarrow \neg(\neg\exists x.P(x) \wedge \forall x.\ \exists z.Q(z,x)\ )$$

Proof via resolution

Exercise 16.8

Consider the following formulas

$$\Sigma = \{ \forall x, y, z.\ (z \in x \Leftrightarrow z \in y) \Rightarrow x \approx y,$$
$$\forall x, y.\ (x \subseteq y \Leftrightarrow \forall z.\ (z \in x \Rightarrow z \in y)),$$
$$\forall x, y, z.\ (z \in x - y \Leftrightarrow (z \in x \wedge z \notin y))\}.$$

Prove the following using resolution proof system with unification

$$\Sigma \models \forall x, y.\ x \subseteq y \Rightarrow \exists z.(y - z \approx x)$$

Theorem prover

Exercise 16.9

Download EPROVER *a first order theorem prover from the following url.*

http://wwwlehre.dhbw-stuttgart.de/~sschulz/E/Usage.html

Run the prover to prove the validity of the following sentence.

$$\forall x.\exists y.\forall z.\exists w.(R(x,y) \vee \neg R(w,z))$$

Report the proof generated by the prover. Explain the rational of proof steps in your own words.

End of Lecture 16