CS112 Notes: First Order Logic: Part 2

James Pustejovsky

September 7, 2004

1 Derivability

In this handout we will examine the steps involved in inference, which is usually a multi-step process. Each step leading from premises to conclusion must be *licensed* by a rule of inference in the system. Below we will review the basic rules of inference that are exploited for inference in first order logic.

We begin with *Modus ponens (MP)* and illustrate the first three rules with the propositional calculus.

(1) $\dfrac{\phi \rightarrow \psi,\ \phi}{\psi}$

Imagine that we have

$$Texan(John)$$

as a proposition in our knowledge (our database, Δ), and we also have

$$Texan(John) \rightarrow American(John)$$

Then we can derive in one step, the following proposition.

$$American(John)$$

The reverse of modus polens is called *Modus tolens (MT)*, and can be stated as follows:

(2) $\dfrac{\phi \rightarrow \psi,\ \neg\psi}{\neg\phi}$

The next rule is *And Elimination (AE)*. This states that whenever we have a conjunction of sentences, we infer each of the conjuncts.

(3) $\dfrac{\phi \wedge \psi}{\phi,\ \psi}$

The converse of this is *And Introduction (AI)*, and is stated as follows:

(4) $\dfrac{\phi,\ \psi}{\phi \wedge \psi}$

The rule of *Or Introduction (OI)* allows us to introduce a literal into a disjunction:

(4) $\dfrac{\phi_i}{\phi_1 \vee \ldots \phi_n}$

The last two rules involve the simplification of quantified expressions. *Universal instantiation (UI)* reasons from the general to the particular. Whenever we have a universally quantified sentence, we can infer a particular instance of that sentence, where the variable is globally replaced by an appropriate term.

(5) $\dfrac{\forall \nu\ \phi}{\phi_{\nu/\tau}}$ where τ is free for ν in ϕ

The rule of *existential instantiation (EI)* lets us eliminate existential quantifiers in a sentence. It is stated as follows:

(6) $\dfrac{\exists \nu\ \phi}{\phi_{\nu/\pi(\nu_1,\ldots,\nu_n)}}$ where π is function constant and ν_i are free variables in ϕ

For example, given the premise $\forall x \exists y[drive(x, y)]$, then we can create a new function constant, call it $vehicle$, and convert the expression to:

$$\forall x[drive(x, vehicle(x))]$$

For multiple variables, it works the same way. Given a premise such as

$$\forall x \forall y \exists z[buy(x, z, y)]$$

then, given a new function constant such as $gift$, the rule EI will reduce this to:

$$\forall x \forall y[buy(x, gift(x, y), y)]$$

2 Resolution

Here we present a fairly simple inference procedure called *resolution*. Expressions are input into the procedure as *clausal forms*, which contain only literals and clauses.

2.1 Conversion to Clausal Form

Below is a schematic for the procedure to convert expressions to clausal form.

```
Procedure Convert (x)
1  Begin  x  <-  Implications_out(x),
2         x  <-  Negations_in(x),
3         x  <-  Standardize_variables(x),
4         x  <-  Existentials_out(x),
5         x  <-  Universals_out(x),
6         x  <-  Disjunctions_in(x),
7         x  <-  Operators_out(x),
8         x  <-  Rename_variables(x)
   End
```

A *literal* is an atomic sentence or the negation of an atomic. A *clause* is a set of literals representing their disjunction.

To examine the steps above, consider the examples below. **First step**, implications out.

(a) $\phi \rightarrow \psi$ replace by $\neg\phi \vee \psi$

(b) $\phi \leftarrow \psi$ replace by $\phi \vee \neg\psi$

(c) $\phi \leftrightarrow \psi$ replace by $(\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$

Step two: Negations are distributed over other logical operators until each one applies to a single atomic sentence.

(a) $\neg\neg\phi$ replace by ϕ

(b) $\neg(\phi \wedge \psi)$ replace by $\neg\phi \vee \neg\psi$

(c) $\neg(\phi \vee \psi)$ replace by $(\neg\phi \wedge \neg\psi)$

(d) $\neg\forall\nu\phi$ replace by $\exists\nu\neg\phi$

(e) $\neg\exists\nu\phi$ replace by $\forall\nu\neg\phi$

In the **third step**, we rename variables so that each quantifier has a unique variable. For example we replace

$$(\forall x P(x,x)) \wedge (\exists x Q(x))$$

by the formula:

$$(\forall x P(x,x)) \wedge (\exists y Q(y))$$

Step four eliminates existential quantifiers that are not in the scope of a universal quantifier. Such variables are replaced by *Skolem constants*. For examples, in $(\exists x Q(x))$, we can simply to $Q(A)$, where A has not been used before.

If an existential is in the scope of a universal, we create a *Skolem function*, which is a new function symbol, which takes as its parameters, those variable names which the existential was in the scope of. For example, we can replace $\forall x \forall y \exists z P(x,y,z)$ with the sentence:

$$\forall x \forall y P(x,y,F(x,y))$$

The **fifth step** drops all universal quantifiers. And the **sixth step** puts expressions into *conjunctive normal form*. For example,

$$\phi \vee (\psi \wedge \chi)$$

is replaced by

$$(\phi \vee \psi) \wedge (\phi \vee \chi)$$

Finally, all variables are renamed so that no variable appears in more than one clause. Now we have clausal form expressions.

3 Unification

Unification is the process of determining whether two expressions can be made identical by appropriate substitutions for variables occurring in the expressions. Substitutions were defined in our previous handout

3.1 The Resolution Calculus Res

Resolution inference rule:

$$\frac{C \vee A \qquad \neg A \vee D}{C \vee D}$$

Terminology: $C \vee D$: *resolvent*; A: *resolved atom*

(positive) factorization:

$$\frac{C \vee A \vee A}{C \vee A}$$

These are *schematic inference rules*; for each substitution of the *schematic variables* C, D, and A, respectively, by ground clauses and ground atoms we obtain an inference rule.

As "$\vee$" is considered associative and commutative, we assume that A and $\neg A$ can occur anywhere in their respective clauses.

3.2 Example Refutation

1.	$\neg P(f(a)) \vee \neg P(f(a)) \vee Q(b)$	(given)
2.	$P(f(a)) \vee Q(b)$	(given)
3.	$\neg P(g(b,a)) \vee \neg Q(b)$	(given)
4.	$P(g(b,a))$	(given)
5.	$\neg P(f(a)) \vee Q(b) \vee Q(b)$	(Res. 2. into 1.)
6.	$\neg P(f(a)) \vee Q(b)$	(Fact. 5.)
7.	$Q(b) \vee Q(b)$	(Res. 2. into 6.)
8.	$Q(b)$	(Fact. 7.)
9.	$\neg P(g(b,a))$	(Res. 8. into 3.)
10.	$\perp$	(Res. 4. into 9.)

3.3 Resolution with Implicit Factorization RIF

$$\frac{C \vee A \vee \ldots \vee A \qquad \neg A \vee D}{C \vee D}$$

1.	$\neg P(f(a)) \vee \neg P(f(a)) \vee Q(b)$	(given)
2.	$P(f(a)) \vee Q(b)$	(given)
3.	$\neg P(g(b,a)) \vee \neg Q(b)$	(given)
4.	$P(g(b,a))$	(given)
5.	$\neg P(f(a)) \vee Q(b) \vee Q(b)$	(Res. 2. into 1.)
6.	$Q(b) \vee Q(b) \vee Q(b)$	(Res. 2. into 5.)
7.	$\neg P(g(b,a))$	(Res. 6. into 3.)
8.	$\perp$	(Res. 4. into 7.)

4 Resolution for General Clauses

First we give a definition of *most general unifier (mgu)*. A most general unifier γ of ϕ and ψ has the property that, if σ is any unifier of the two expressions, then there exists a substitution δ with the following property:

$$\phi\gamma\delta \;=\; \phi\sigma \;=\psi\sigma$$

GENERAL BINARY RESOLUTION Res:

$$\frac{C \vee A \qquad D \vee \neg B}{(C \vee D)\sigma} \quad \text{if } \sigma = \mathsf{mgu}(A, B) \qquad \text{[resolution]}$$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \qquad \text{if } \sigma = \mathsf{mgu}(A, B) \quad \text{[factorization]}$$

GENERAL RESOLUTION RIF WITH IMPLICIT FACTORIZATION:

$$\frac{C \vee A_1 \vee \ldots \vee A_n \qquad D \vee \neg B}{(C \vee D)\sigma} \quad \text{if } \sigma = \mathsf{mgu}(A_1, \ldots, A_n, B) \quad \text{[RIF]}$$

We additionally assume that the variables in one of the two premises of the resolutions rule are (bijectively) renamed such that they become different to any variable in the other premise. We do not formalize this. Which names one uses for variables is otherwise irrelevant.