Math 260A — Mathematical Logic — Scribe Notes
UCSD — Winter Quarter 2012
Instructor: Sam Buss

Notes by: John Dougherty
Wendesday, March 14, 2012

Some motivation

In first-order logic, using Herbrand's theorem, we've reduced the problem of arbitrary provability to the problem of refuting a set of clauses of literals. For example, refutation of $\{P(f(x), g(y)), \neg Q(x + y)\}$ shows that $\forall x \forall y (P(f(x), g(y)) \vee \neg Q(x + y))$ is invalid.

Example. Consider the predicate $\text{Knows}(x, y)$ ("x knows y") and the functions $M(x)$ (the mother of x), and $F(x)$ (the father of x), and take the set of clauses

$$\begin{aligned}\Gamma = \{&\{\text{Knows}(x, M(x))\},\\ &\{\text{Knows}(x, F(x))\},\\ &\{\neg\text{Knows}(x, y), \neg\text{Knows}(y, z), \text{Knows}(x, z)\},\\ &\{\neg\text{Knows}(x, F(M(x)))\}\}\end{aligned}$$

Note that the third clause means $\neg\text{Knows}(x, y) \vee \neg\text{Knows}(y, z) \vee \text{Knows}(x, z)$, which is equivalent to $\text{Knows}(x, y) \wedge \text{Knows}(y, z) \rightarrow \text{Knows}(x, z)$. So this set of clauses contains a contradiction, since x knows his mother, and $M(x)$ knows her father, so x must know $F(M(x))$.

We're looking for an insatisfiable/inconsistent set of ground instances of Γ — instances without variables. As before, we assume there is at least 1 constant c in our language. Then

$$\begin{aligned}&\{\{\text{Knows}(c, M(c))\},\\ &\{\text{Knows}(c, F(c))\},\\ &\{\neg\text{Knows}(c, M(c)), \neg\text{Knows}(M(c), F(M(c))), \text{Knows}(c, F(M(c)))\},\\ &\{\neg\text{Knows}(c, F(M(c)))\}\}\end{aligned}$$

is an unsatisfiable ground instance of Γ, which produces a resolution refutation.

In general, this problem is semi-satisfiable; if there is such a refutation, we're able to find it. If there is no such instance, we'll search forever. Note

that we might need more than one instance. For example, consider the set of clauses

$$\begin{aligned}\Gamma' = \{&\{\text{Knows}(x, M(x))\},\\ &\{\text{Knows}(x, F(x))\},\\ &\{\neg\text{Knows}(x, y), \neg\text{Knows}(y, z), \text{Knows}(x, z)\},\\ &\{\neg\text{Knows}(x, F(F(M(x))))\}\}\end{aligned}$$

which requires two ground instances for a resolution refutation.

Recall how resolution works: if we have two sets of clauses C and D, neither of which contain x or $\bar{x}$, then we can infer

$$\frac{C \cup \{x\} \qquad D \cup \{\bar{x}\}}{C \cup D}$$

So you might want to resolve the third and fourth clause of Γ. However, no literal in the third clause is the negation of the literal in the fourth clause, so you can't. In order to perform resolution, you have to "unify" the literals by mapping $z \mapsto F(M(x))$ to get the clause

$$\{\neg\text{Knows}(x, y), \neg\text{Knows}(y, F(M(x))), \text{Knows}(x, F(M(x)))\}.$$

No matter what you put in for x and y in this clause, you get a substitution instance of the third clause, so if Γ has an unsatisfiable ground instance that uses this clause instead of the original third clause, that ground instance is also a ground instance of the original set of clauses. So now you can resolve

$$\frac{\{\neg K(x, y), \neg K(y, F(M(x))), K(x, F(M(x)))\} \qquad \{\neg K(x, F(M(x)))\}}{\{\neg K(x, y), \neg K(y, F(M(x)))\}}$$

What could go wrong with this process?

1) What if we tried to unify $\{\text{Knows}(x, M(x))\}$ with $\{\text{Knows}(x, F(M(x))\}$? It would have to fail, since the outermost function in the second argument is M in the first and F in the second. F and M "clash", and you can't get rid of them regardless of what you substitute.

2) $\text{Knows}(x, F(x))$ can't be unified with $\text{Knows}(x, F(M(x)))$. You'd have to replace x with something such that $F(t) = F(M(t))$, which can't be done. This is an "occurs check" — you can't unify x with $M(x)$ since x occurs in $M(x)$.

Note also that we didn't use the ground clauses in performing the refutation, but we can read them off from the leaves of the proof, substituting c in for x.

Going Mathematical

The same proof strategies can work here that worked in the case of propositional resolution refutation, like prioritizing unit resolutions. We want to devise a method of automatically picking terms with an eye to unification. First, some definitions.

Definition. A *substitution* is a mapping from variables to terms. It can either be thought of as a map from the set of all variables to terms, or as a map from some subset V' of the variables that acts as the identity on $x \notin V'$.

Definition. A *ground substitution* is a substitution for variables to ground terms.

Definition. Let $t_1, \ldots, t_k$ be terms. A *unifier* for $t_1, \ldots, t_k$ is a substitution σ such that

$$t_1\sigma = t_2\sigma = \cdots = t_k\sigma$$

where we employ postfix notation: $t_1\sigma = \sigma(t_1)$.

Example. Let σ and π be the substitutions

$$\begin{aligned} &\sigma(x) = y + 0 \qquad (\text{so } \sigma(y) = y, \text{ by default}) \\ &\pi(y) = z^2 \end{aligned}$$

Then, for example,

$$\begin{aligned} (\sin(x))\sigma &= \sin(y + 0) \\ (\sin(y + 0))\pi &= \sin(z^2 + 0) \\ (\sigma\pi)(x) &+ z^2 + 0 \\ (\sigma\pi)(y) &= z^2 \\ (\sin(x \cdot 3y))(\sigma\pi) &= \sin((z^0 + 0) \cdot 3(z^2)) \end{aligned}$$

Definition. A *most general unifier* (MGU) of $t_1, \ldots, t_k$ is a unifier σ of $t_1, \ldots, t_k$ such that for all unifiers τ of $t_1, \ldots, t_k$ there is a substitution π such that $\sigma\pi = \tau$. Note that, since σ is a unifier, $\sigma\pi$ is automatically a unifier:

$$t_1\sigma = t_2\sigma = \cdots = t_k\sigma \qquad \Rightarrow \qquad t_1\sigma\pi = t_2\sigma\pi = \cdots = t_k\sigma\pi$$

Example. Find the MGU of $f(g(x), z)$ and $f(y, h(x))$.

From the first arguments, the MGU σ must map $y \mapsto g(z)$. From the second, $z \mapsto h(x)$. So define

$$\sigma = \begin{cases} y \mapsto g(x) \\ z \mapsto h(x) \end{cases}.$$

For all unifiers τ, $\tau = \sigma\pi$. In this case, $x\pi = x\tau$. Suppose that $\tau : x \mapsto c^2$. Then

$$\tau = \begin{cases} x \mapsto c^2 \\ y \mapsto g(c^2) \\ z \mapsto h(c^2) \end{cases}$$

and $\pi : x \mapsto c^2$ gives $\sigma\pi = \tau$.

Example. Find the MGU of $f(x, f(z))$ and $f(h(y, y), y)$.

The first pass gives us $x \mapsto h(y, y)$ and $y \mapsto f(z)$. Unification demands that x be further mapped to $x \mapsto h(f(z), f(z))$. So

$$\sigma = \begin{cases} x \mapsto h(f(z), f(z)) \\ y \mapsto f(z) \end{cases}.$$

An MGU isn't unique, but it *is* unique up to variable names.

Theorem. If a set of formulas has a unifier, it has a most general unifier.

Proof. (and algorithm, simultaneously).

We go in stages. For the base case, define

E_0 The things to be unified. A finite set of pairs of terms

$$\{(s_1, t_1), \ldots, (s_k, t_k)\}.$$

Our goal is to unify s_1 with t_1, s_2 with t_2, ..., s_k with t_k.

σ_0 The identity mapping.

Now define E_{i+1} and σ_{i+1} by the following procedure:

Case 1) E_i contains some clashing terms; there is a pair $(f(\cdots), g(\cdots))$ with f distinct from g. In this case, a unifier doesn't exist, so abort.

Case 2) E_i has a pair of the form $(f(t_1, \ldots, t_\ell), f(t'_1, \ldots, t'_\ell))$.

Set $E_{i+1} = (E_i \setminus \{(f(\vec{t}), f(\vec{t'})\}) \cup \{(t_i, t'_i) : i = 1, \ldots, \ell\}$.

Set $\sigma_{i+1} = \sigma_i$.

Case 3) Find a pair (x, t) or (t, x) in E_i.

(i) if $t = x$, $E_{i+1} = E_i \setminus \{(x, t)\}$, $\sigma_{i+1} = \sigma_i$.

(ii) if x occurs in t, i.e., $t = f(\cdots x \cdots)$, we have an "occurs check": there is no unifier, so abort.

(iii) if x doesn't occur in t, $\sigma_{i+1} = \sigma_i \pi_i$ where $x\pi_i = t$. $E_{i+1} = \{(s\sigma_{i+1}, t\sigma_{i+1}) : (s, t) \in E_i \setminus \{(x, t)\}\}$.

Case 4) $E_i = \emptyset$, and we're done — the MGU is σ_i.

This algorithm always halts, since at each stage we either reduce the total number of function symbols by 1 (Case 2), or we reduce the total number of variables by 1 (Case 3.iii). Any unifier of E_i is obtainable in the form $\sigma_i \pi$ for some π, and if σ unifies E_i, it unifies E_{i+1}. □