Resolution for FOL

Computability and Logic

Function Symbols

- First-Order logic has function symbols: symbols that mean functions from objects to objects.
- Functions, like predicates, have an arity: a number of ‘arguments’.
- However, whereas predicates take objects and become a claim, functions take objects and return a new object, i.e. not a claim.
- Example: father, child, +
  - String that starts with lower case, ‘+’ is an exception

Terms

- Recursive definition:
- Atomic Terms:
  - All individual constants (a,b,c,…) are atomic terms
  - All variables (x,y,z, …) are terms
- Complex Terms:
  - Where f is an n-place function symbol, and t_1,t_2, … t_n are terms, $f(t_1,t_2, \ldots t_n)$ is a term
- (nothing else is a term)

How to use Functions

- Since functions denote objects, rather than statements, functions are used like other objects.
- For example, if we are dealing with a term 1 + 1 (which is short for +(1,1)), and if we have $\forall x \forall y\ (x > y \rightarrow y < x)$, then we can instantiate the x with 1+1 to get $\forall y\ (1+1 > y \rightarrow y < 1+1)$

Unification

- Two terms t_1 and t_2 can be unified if there are substitutions that substitute terms for variables in t_1 and t_2 such that the two terms become identical.
- Examples:
  - a and x can be unified (substitute a for x)
  - y and x can be unified (substitute y for x; or x for y)
  - f(a) and x can be unified (substitute f(a) for x)
  - a and f(x) cannot be unified
  - x and f(x) cannot be unified
  - x and f(y) can be unified (f(y) for x)
  - f(a,b) and f(x,y) can be unified (use a/x and b/y)
  - f(a,a) and f(x,x) can be unified (use a/x)
  - f(a,a) and f(x,y) can be unified (use a/x, b/y)
  - f(a,x) and f(y,a) can be unified (use a/x, a/y)
  - f(a,b) and f(x,x) cannot be unified

Skolemization

- Skolemization is the process of replacing existentially quantified variables with (new!) terms:
  - If the existential is not preceded by a universal, use a (new!) constant
  - If the existential is preceded by one or more universals, use a (new!) function defined over the variables quantified by those universal quantifiers
- Examples:
  - $\exists x\ \varphi(x)$ becomes $\varphi(a)$
  - $\exists x\ \exists y\ \varphi(x,y)$ becomes $\varphi(a,b)$
  - $\forall x\ \exists y\ \varphi(x,y)$ becomes $\forall x\ \varphi(x,f(x))$
  - $\forall x\ \exists y\ \exists z\ \varphi(x,y,z)$ becomes $\forall x\ \varphi(x,f(x),g(x))$
  - $\forall x\ \forall y\ \exists z\ \varphi(x,y,z)$ becomes $\forall x\ \varphi(x,y,f(x,y))$
  - $\forall x\ \exists y\ \forall z\ \exists w\ \varphi(x,y,z,w)$ becomes $\forall x\ \forall z\ \varphi(x, f(x),z, g(x,z))$

Resolution

- Basic algorithm for resolution for FOL:
  - Determine set of statements to be tested for consistency (e.g. if argument: negate conclusion)
  - Put all statements into PNF, and body into CNF
  - Skolemize all existentials
  - Drop universals and put into clauses
  - Resolve … using unification to match atomic formulas

Example

$\forall x\,(A(x) \rightarrow B(x))$
$\forall x\,(B(x) \rightarrow C(x))$
$\therefore \forall x\,(A(x) \rightarrow C(x))$

Negate Conclusion:	$\forall x\,(A(x) \rightarrow B(x))$	$\forall x\,(B(x) \rightarrow C(x))$	$\neg\forall x\,(A(x) \rightarrow C(x))$	
PNF + CNF:	$\forall x\,(\neg A(x) \vee B(x))$	$\forall x\,(\neg B(x) \vee C(x))$	$\exists x\,(A(x) \wedge \neg C(x))$	
Skolemize:	$\forall x\,(\neg A(x) \vee B(x))$	$\forall x\,(\neg B(x) \vee C(x))$	$A(a) \wedge \neg C(a)$	
Drop $\forall$'s and clausify:	$\{\neg A(x), B(x)\}$	$\{\neg B(x), C(x)\}$	$\{A(a)\}$	$\{\neg C(a)\}$

Resolve using Unification:

$\{\neg A(x), B(x)\}$ and $\{A(a)\}$ with a/x: $\{B(a)\}$

$\{\neg B(x), C(x)\}$ and $\{\neg C(a)\}$ with a/x: $\{\neg\, B(a)\}$

$\{B(a)\}$ and $\{\neg\, B(a)\}$: $\{\ \}$

Unification in Resolution: First Comment

- So what we're really doing is not unifying terms, but unifying statements from 2 different clauses so that they can be resolved.
- To be precise: we try to find substitutions of terms for variables, such that one clause ends up containing atomic formula A, and the other ends up with $\neg$A, such that the two clauses can be resolved, and where the substitutions are applied to the atomic formulas in the resolved clause.
- E.g. (from previous slide) Applying substitution of a/x to {$\neg$A(x), B(x)} we get {$\neg$A(a), B(a)} which can be resolved with {A(a)} to {B(a)}

Unification in Resolution: Second Comment

- Remember that variables are dummies. So, feel free to replace variables with other variables so that unifications that seem impossible do become possible.
- E.g. It may seem impossible to resolve {A(x)} with {¬A(f(x)}, but you could rewrite {A(x)} as {A(y)}, and now you can use substitution f(x)/y and resolve.
- Another way of doing this is to use a separate substitution list for each clause: use f(y)/x for the first clause, and y/x for the second.
  - Doing this might be the preferred way: even if one uses all different variables in the original clause set, during resolution one may end up with different clauses with the same variables that cannot be resolved without replacing variables yet again. Using two separate substitution lists avoids this problem altogether.

Unification in Resolution: Third Comment

- Finally, it is a good idea to try and keep your substitutions as general as possible.
- E.g. {A(x), B(x)} and {¬B(y), C(y)} can be resolved using substitutions a/x and a/y, resolving to {A(a),C(a)}, but this is not the most general resolution: using x/y on the second clause, they resolve to {A(x),C(x)}. And the latter is more general, and therefore more useful: if you also had clause {¬A(b)}, you can immediately further resolve to {C(b)} in the latter case, but not in the first.

Infinite Resolutions

$\forall x\ \exists y\ R(x,y)$

$\forall x \forall y \forall z\ ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$

$\{R(x,f(x))\}$

$\{\neg R(x,y) \vee \neg R(y,z) \vee R(x,z)\}$

$f(x)/y$

$\{\neg R(f(x),z) \vee R(x,z)\}$

$f(y)/x$

$y/x,\ f(f(y))/z$

$\{R(y,f(f(y)))\}$

x/y

$f(f(x))/y$

$\{\neg R(f(f(x)),z) \vee R(x,z)\}$

$f(f(y))/x$

$f(f(f(y)))/z$

$\{R(y,f(f(f(y))))\}$

Etc.!

Note the second resolution seems blocked … so either replace variables, or use separate substitution lists (we did the latter)