CSC384: Intro to Artificial Intelligence
Knowledge Representation III

- Announcements.
  - Office hours?
- Resolution Proofs.
  - ✓Part I: Convert to clausal form
  - Part II: Dealing with variables (unification).
  - Part III: Constructing Resolution Proofs.

Unification

- Ground clauses are clauses with no variables in them. For ground clauses we can use syntactic identity to detect when we have a P and ¬P pair.
- What about variables can the clauses
  - (P(john), Q(fred), R(X))
  - (¬P(Y), R(susan), R(Y))

  Be resolved?

Unification.

- Intuitively, once reduced to clausal form, all remaining variables are universally quantified. So, implicitly (¬P(Y), R(susan), R(Y)) represents clauses like
  - (¬P(fred), R(susan), R(fred))
  - (¬P(john), R(susan), R(john))
  - …
- So there is a “specialization” of this clause that can be resolved with (P(john), Q(fred), R(X)

Unification.

- We want to be able to match conflicting literals, even when they have variables. This matching process automatically determines whether or not there is a “specialization” that matches.
- We don’t want to over specialize!

Unification.

- (¬p(X), s(X), q(fred))
- (p(Y), r(Y))
- Possible resolvants
  - (s(john), q(fred), r(john)) {Y=X, X=john}
  - (s(sally), q(fred), r(sally)) {Y=X, X=sally}
  - (s(X), q(fred), r(X)) {Y=X}
- The last resolvant is "most-general", the other two are specializations of it.
- We want to keep the most general clause so that we can use it future resolution steps.

Unification.

- unification is a mechanism for finding a "most general" matching.
- First we consider substitutions.
  - A substitution is a finite set of equations of the form

    (V = t)

    where V is a variable and t is a term not containing V. (t might contain other variables).

Substitutions.

- We can apply a substitution σ to a formula f to obtain a new formula $f\sigma$ by simultaneously replacing every variable mentioned in the left hand side of the substitution by the right hand side.

$$p(X,g(Y,Z))[X=Y,\ Y=f(a)] \rightarrow p(Y,g(f(a),Z))$$

- Note that the substitutions are not applied sequentially, i.e., the first Y is not subsequently replaced by f(a).

Substitutions.

- We can compose two substitutions. θ and σ to obtain a new substition $\theta\sigma$.

Let $\theta = \{X_1=s_1, X_2=s_2, \ldots, X_m=s_m\}$

$\sigma = \{Y_1=t_1, Y_2=t_2, \ldots, Y_k=s_k\}$

To compute $\theta\sigma$

1. $S = \{X_1=s_1\sigma, X_2=s_2\sigma, \ldots, X_m=s_m\sigma, Y_1=t_1, Y_2=t_2,\ldots, Y_k=s_k\}$

   we apply σ to each RHS of θ and then add all of the equations of σ.

Substitutions.

1. $S = \{X_1=s_1\sigma, X_2=s_2\sigma, \ldots, X_m=s_m\sigma, Y_1=t_1,$
   $Y_2=t_2, \ldots, Y_k=s_k\}$
2. Delete any identities, i.e., equations of the form V=V.
3. Delete any equation $Y_i=s_i$ where Y_i is equal to one of the X_j in θ.

The final set S is the composition $\theta\sigma$.

Composition Example.

$\theta = \{X=f(Y), Y=Z\}, \sigma = \{X=a, Y=b, Z=Y\}$

$\theta\sigma$

Substitutions.

- The empty substitution $\varepsilon = \{\}$ is also a substitution, and it acts as an identity under composition.
- More importantly substitutions when applied to formulas are associative:

$$(f\theta)\sigma = f(\theta\sigma)$$

- Composition is simply a way of converting the sequential application of a series of substitutions to a single simultaneous substitution.

Unifiers.

- A unifier of two formulas f and g is a substitution σ that makes f and g syntactically identical.
- Not all formulas can be unified—substitutions only affect variables.

$p(f(X),a) \quad p(Y,f(w))$

- This pair cannot be unified as there is no way of making a = f(w) with a substitution.

MGU.

- A substitution σ of two formulas f and g is a **Most General Unifier (MGU)** if

1. σ is a unifier.
2. For every other unifier θ of f and g there must exist a third substitution λ such that
   $\theta = \sigma\lambda$

- This says that every other unifier is “more specialized than σ. The MGU of a pair of formulas f and g is unique up to renaming.

MGU.

p(f(X),Z) p(Y,a)

1. $\sigma = \{Y = f(a), X=a, Z=a\}$ is a unifier.

   p(f(X),Z)σ =
   p(Y,a)σ =

   But it is not an MGU.

2. $\theta = \{Y=f(X), Z=a\}$ is an MGU.
   p(f(X),Z) θ =
   p(Y,a) θ =

MGU.

p(f(X),Z) p(Y,a)

3. $\sigma = \theta\lambda$, where $\lambda=\{X=a\}$

   $\sigma = \{Y = f(a), X=a, Z=a\}$
   $\lambda = \{X=a\}$
   $\theta\lambda$ =

MGU.

- The MGU is the “least specialized” way of making clauses with universal variables match.
- We can compute MGUs.
- Intuitively we line up the two formulas and find the first sub-expression where they disagree. The pair of subexpressions where they **first** disagree is called the **disagreement set**.
- The algorithm works by successively fixing disagreement sets until the two formulas become syntactically identical.

MGU.

To find the MGU of two formulas f and g.

1. $k = 0$; $\sigma_0 = \{\}$; $S_0 = \{f,g\}$
2. If S_k contains an identical pair of formulas stop, and return σ_k as the MGU of f and g.
3. Else find the disagreement set $D_k=\{e_1,e_2\}$ of S_k
4. If $e_1 = V$ a variable, and $e_2 = t$ a term not containing V (or vice-versa) then let
   $\sigma_{k+1} = \sigma_k \{V=t\}$ (Compose the additional substitution)
   $S_{k+1} = S_k\{V=t\}$ (Apply the additional substitution)
   $k = k+1$
   GOTO 2
5. Else stop, f and g cannot be unified.

MGU Example 1.

S_0 = {p(f(a), g(X)) ; p(Y,Y)}

MGU Example 2.

$S_0 = \{p(a,X,h(g(Z))) \;;\; p(Z,h(Y),h(Y))\}$

MGU Example 3.

$S_0 = \{p(X,X) \;;\; p(Y,f(Y))\}$

Non-Ground Resolution

- Resolution of non-ground clauses. From the two clauses
  (L, Q1, Q2, …, Qk)
  (¬M, R1, R2, …, Rn)

  Where there exists σ a MGU for L and M.

  We infer the new clause

  (Q1σ, …, Qkσ, R1σ, …, Rnσ)

Non-Ground Resolution E.G.

1. (p(X), q(g(X)))
2. (r(a), q(Z), ¬p(a))

   L=p(X); M=p(a)
   σ = {X=a}

3. R[1a,2c]{X=a} (q(g(a)), r(a), q(Z))

The notation is important.

- "R" means resolution step.
- "1a" means the first (a-th) literal in the first clause i.e. p(X).
- "2c" means the third (c-th) literal in the second clause, ¬p(a).
  - 1a and 2c are the "clashing" literals.
- {X=a} is the substitution applied to make the clashing literals identical.

Resolution Proof Example

"Some patients like all doctors. No patient likes any quack. Therefore no doctor is a quack."

Resolution Proof Step 1.
Pick symbols to represent these assertions.

p(X): X is a patient
d(x): X is a doctor
q(X): X is a quack
l(X,Y): X likes Y

Resolution Proof Example

Resolution Proof Step 2.
Convert each assertion to a first-order formula.

1. Some patients like all doctors.

F1.

Resolution Proof Example

2. No patient likes any quack

F2.

3. Therefore no doctor is a quack.

Query.

Resolution Proof Example

Resolution Proof Step 3.

Convert to Clausal form.

F1.

F2.

Negation of Query.

Resolution Proof Example

Resolution Proof Step 4.

Resolution Proof from the Clauses.

1. p(a)
2. (¬d(Y), l(a,Y))
3. (¬p(Z), ¬q(R), ¬l(Z,R))
4. d(b)
5. q(b)

Resolution Proof Example