i. {P(x), P(A)} P(A) {A/x} ~~{x/A}~~ ti/vi

ii. {P[f(x),y,g(y)], P[f(x),z,g(x)]} P[f(x), x, g(x)] {x/y, x/z}

iii. {P[f(x,g(A,y)), g(A,y)], P[f(x,z),z]} P[f(x,g(A,y)), g(A,y)]

The disagreement set of E is the set D obtained by comparing each symbol of all expressions in E from left to right and extracting from E the subexpression whose first symbols do not agree.

E = { P(f(x), g(y), a), P(f(x), z, a), P (f(x), b, h(u))} --- D={}

Unification Algorithm

1. Set k = 0 , mgu_k = {}

2. If the set $Emgu_k$ is a singleton then stop; mgu_k is an mgu of E. Otherwise, find the disagreement set D_k of $Emgu_k$

3. If there is a var v and term t in D_k such that v does not occur in t, put mgu_{k+1} = mgu_k{t/v}, set k = k+1, and return to step 2. Otherwise, stop, E is not unifiable.

E = {P(x,z,y), P(w,u,w), P(A,u,u)}

mgu ??

Resolution

An important *rule of inference* that can be applied to certain class of wffs, called clauses

A clause is defined as a wff consisting of a disjunction of literals.

The resolution process, when it is applicable, is applied to a pair of parent clauses to produce a derived clause.

Any Predicate calculus wff can be converted to a set of clauses.

Clause form

- A clause is a disjunction ("**or**") of zero or more literals, some or all of which may be negated
- Example:
  sinks(X) $\vee$ dissolves(X, water) $\vee$ $\neg$denser(X, water)
- Notice that clauses use only "or" and "not"—they do not use "and," "implies," or either of the quantifiers "for all" or "there exists"
- The impressive part is that *any* predicate calculus expression can be put into clause form
  - Existential quantifiers $\exists$ are the trickiest ones

The magic of resolution

- Here's how resolution works:
  - transform each of your facts into a particular form, called a clause
  - apply a *single rule,* the resolution principle, to a pair of clauses
    - Clauses are closed with respect to resolution--that is, when you resolve two clauses, you get a new clause
  - add the new clause to your fact base
- So the number of facts grows linearly
  - You still have to choose a pair of facts to resolve
  - You never have to choose a rule, because there's only one

The resolution principle

- Here it is:
  - From $X \lor$ someLiterals
    and $\neg X \lor$ someOtherLiterals
    ----
    conclude: someLiterals $\lor$ someOtherLiterals
- That's all there is to it!
- Example:

  broke(Bob) $\lor$ well-fed(Bob)
  ¬broke(Bob) $\lor$ ¬hungry(Bob)
  ----
  well-fed(Bob) $\lor$ ¬hungry(Bob)

Contradiction

- A special case occurs when the result of a resolution (the resolvent) is empty, or “NIL”
- Example:

```
  hungry(Bob)
¬hungry(Bob)
---------------
NIL
```

- In this case, the fact base is inconsistent
- This will turn out to be a very useful observation in doing resolution theorem proving

Converting sentences to CNF

$$(\forall x)(P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(F(x,y))) \wedge \neg(\forall y)(Q(x,y) \rightarrow P(y))))$$

1. Eliminate all $\leftrightarrow$ connectives

   $(P \leftrightarrow Q) \Rightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$

2. Eliminate all $\rightarrow$ connectives

   $(P \rightarrow Q) \Rightarrow (\neg P \vee Q)$

3. Reduce the scope of each negation symbol to a single predicate

   $\neg\neg P \Rightarrow P$

   $\neg(P \vee Q) \Rightarrow \neg P \wedge \neg Q$

   $\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q$

   $\neg(\forall x)P \Rightarrow (\exists x)\neg P$

   $\neg(\exists x)P \Rightarrow (\forall x)\neg P$

4. Standardize variables: rename all variables so that each

Converting sentences to clausal form: Skolem constants and functions

5. Eliminate existential quantification by introducing Skolem constants/functions
   - $(\exists x)P(x) \Rightarrow P(C)$
     - **C is a Skolem constant** (a brand-new constant symbol that is not used in any other sentence)
   - $(\forall x)(\exists y)P(x,y)$ becomes $(\forall x)P(x, F(x))$
     - since $\exists$ is within the scope of a universally quantified variable, use a **Skolem function F** to construct a new value that **depends on** the universally quantified variable
   - F must be a brand-new function name not occurring in any other sentence in the KB.
   - E.g., $(\forall x)(\exists y)loves(x,y)$ becomes $(\forall x)loves(x,F(x))$
     - In this case, F(x) specifies the person that x loves
   - E.g., $\forall x1\ \forall x2\ \forall x3\ \exists y\ P(\ldots\ y\ \ldots)$ becomes

skolemization

- $(\forall y)[(\exists x)P(x,y)]$
- x as fn of y   g(y) --- skolem fn
- $(\forall y)[P(g(y),y)]$

- $(\exists x)P(x)$ $P(A)$ A – skolem constant

Converting sentences to clausal form

6. Remove universal quantifiers by: (1) moving them all to the left end; (2) making the scope of each the entire sentence; and (3) dropping the "prefix" part

(Prenex form :Prefix Matrix) Ex: $(\forall x)P(x) \Rightarrow P(x)$

7. Put matrix into conjunctive normal form (conjunction of disjunctions) using distributive and associative laws repeatedly

   x1 or (x2 and x3) by (x1 or x2) and (x1 or x3)

   $(P \wedge Q) \vee R \Rightarrow (P \vee R) \wedge (Q \vee R)$??? chk

   $(P \vee Q) \vee R \Rightarrow (P \vee Q \vee R)$

8. Split conjuncts into separate clauses

(x1 and x2)   {x1, x2}

9. Standardize variables so that each clause contains only variable names that do not occur in any other clause

7. Convert to conjunction of disjunctions

$(\neg P(x) \vee \neg P(y) \vee P(F(x,y))) \wedge (\neg P(x) \vee Q(x,G(x))) \wedge$
$(\neg P(x) \vee \neg P(G(x)))$

8. Create separate clauses

$\neg P(x) \vee \neg P(y) \vee P(F(x,y))$
$\neg P(x) \vee Q(x,G(x))$
$\neg P(x) \vee \neg P(G(x))$

9. Standardize variables

$\neg P(x) \vee \neg P(y) \vee P(F(x,y))$
$\neg P(z) \vee Q(z,G(z))$
$\neg P(w) \vee \neg P(G(w))$

Note: Now that quantifiers are gone, we do need the upper/lower-case distinction