# Unifier

A *unifier* (also *substitution*, *binding list**) is a set of pairings of variables with terms:

$$\{v_1 = e_1, v_2 = e_2, v_3 = e_3, \ldots v_n = e_n\}$$

such that

- each variable is paired at most once
- a variable's pairing term may not contain the variable directly or indirectly

{x = Socrates}

* Do not confuse with bound / free variables!!!

# Most General Unifier MGU

The MGU imposes the fewest constraints, specifying the *weakest* conditions for matching

MGU is unique

order is not important

variable names are not important (alphabetic variants)

Applying the MGU to an expression yields a *most general unification instance*.

Variable substitutions are always interpreted with the unifier applied

# What is the MGU?

| | |
|---|---|
| M(Ann,x,Bob) | M(Ann,x,Bob) |
| M(Ann,x,Bob) | M(y,x,Chuck) |
| M(Ann,x,Bob) | M(y,x,Father-of(Chuck)) |
| P(w,w,Fred) | P(x,y,y) |
| Q(r,r) | Q(x,F(x)) |
| Q(r,r) | Q(x,F(y)) |
| R(G(x,Bob),y,y) | R(z,G(Fred,w),z) |

# Negation and Quantifiers

- $\neg \forall x\, P(x) \equiv \exists x\, \neg P(x)$
- $\neg \exists x\, P(x) \equiv \forall x\, \neg P(x)$
- $\forall x\, P(x) \wedge \forall y\, Q(y) \equiv \forall x\, \forall y\, [P(x) \wedge Q(y)]$
- $\forall x\, \forall y\, [P(x) \wedge Q(y)] \equiv \forall y\, \forall x\, [P(x) \wedge Q(y)]$
  (also $\vee$, also all $\exists$’s)

BUT

- $\forall x\, \exists y\, P(x,y)$ is NOT the same as
  $\exists y\, \forall x\, P(x,y)$

# Unification is easier without Quantifiers

Precludes FOPC?  No:

Eliminate existentials by *skolemizing*

Drop explicit universals (carefully) & rename

All variables are implicitly universally quantified

Need naming convention...

# Syntactic Cues/Restrictions

(without quantifiers)

P(F(x), John7, y)

Must be a predicate

Must be a function

Could be either constant or variable

Need a naming convention to distinguish constants from variables

Common:

- For constants: first letter upper case
- For variables:
  - first letter “?”
  - single characters from end of alphabet: v, w, x, y, z

# Unification Assumptions

(see text for algorithm)

- Skolemized: all variables are universally quantified
- Variables Standardized Apart: unique names
- Unifiers are *different* if they result in *different* instances and impose *different constraints* (alphabetic variant unifiers do not impose different constraints)

# Skolemization

**We have a WFF:**

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

**For any way of choosing x there is *guaranteed* to be a way of choosing y such that "R" holds between them.**

**Since y can depend on x, we can replace all occurrences of y by a new function of x, say F(x).**

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

**is equivalent to**
**("embodies the same constraints as"**
**"means the same as")**

$$\forall x \; R(x, F(x))$$

# A Skolem Function

- Introduces a new function symbol
- Directly replaces all occurrences of an existential variable
- Has as arguments all universal variables in whose scope it appears

$$\forall x\ \forall y\ \exists w\ \forall z\ Q(x, y, w, z, G(w, x))$$

**is equivalent to**

$$\forall x\ \forall y\ \forall z\ Q(x, y, P(x,y), z, G(P(x,y), x))$$

**where P is the Skolem function for w**

# Consider

**$\neg[\forall x\ \exists y\ \forall z\ P(x, y, z)]$**

**“Every boy owns a dog.”**

**where the dog need not be owned in common**

**where there is one shared dog**

# After Skolemization

- All variables are universally quantified
- Drop universal quantifier indicators “$\forall \mathbf{x}$”
- This loses scoping information
- So we must rename variables before Skolemizing
- Standardize variables apart

# Clause Form

(also clausal form)

- Set notation of CNF (conjunctive normal form - also POS)
- R & N stop with CNF – we do not
- Write axioms as a conjunction of sentences
- Each sentence is a disjunction of literals
  (recall literal: atomic WFF or negated atomic WFF)
- Braces { } denote sets; comma separates literals

# Convert FOPC to Clause Form

**1. Eliminate equivalence $\Leftrightarrow$ and implication $\Rightarrow$ symbols**

**2. Move $\neg$ inwards forming literals**

**3. Standardize variables apart - unique variable names eliminating scoping conflicts**

**4. Skolemize**

**5. Drop universal quantifiers**

**6. Distribute AND $\wedge$ over OR $\vee$**

**7. Flatten nested ANDs $\wedge$ and ORs $\vee$ yielding CNF (POS)**

**8. Write in set notation standardizing variables apart in different clauses**

# Example: $\forall x\, \exists y\, [(\exists z\, (R(x, z) \lor P(y, z))) \Rightarrow \forall z\, Q(y,z)]$

1. $\forall x\, \exists y\, [(\neg(\exists z\, (R(x, z) \lor P(y, z)))) \lor \forall z\, Q(y,z)]$

2. $\forall x\, \exists y\, [(\forall z\, (\neg R(x, z) \land \neg P(y, z))) \lor \forall z\, Q(y,z)]$

3. $\forall x1\, \exists y1\, [(\forall z1\, (\neg R(x1, z1) \land \neg P(y1, z1))) \lor \forall z2\, Q(y1, z2)]$

4. $\forall x1\, [(\forall z1\, (\neg R(x1, z1) \land \neg P(Sk1(x1), z1))) \lor$
   $\forall z2\, Q(Sk1(x1), z2)]$

5. $[(\neg R(x1, z1) \land \neg P(Sk1(x1), z1)) \lor Q(Sk1(x1), z2)]$

6. $[(\neg R(x1, z1) \lor Q(Sk1(x1), z2)) \land$
   $(\neg P(Sk1(x1), z1) \lor Q(Sk1(x1), z2))]$

7. $[(\neg R(x1, z1) \lor Q(Sk1(x1), z2)) \land$
   $(\neg P(Sk1(x1), z1) \lor Q(Sk1(x1), z2))]$

8. $\{\neg R(x2, z3), Q(Sk1(x2), z4)\}$
   $\{\neg P(Sk1(x3), z5), Q(Sk1(x3), z6)\}$