Refutation in Predicate Logic

\[
\text{parent(pam, bob).} \quad \text{anc(X,Y) :- parent(X,Y).}
\]
\[
\text{parent(tom, bob).} \quad \text{parent(X,Y).}
\]
\[
\text{parent(tom, liz).} \quad \text{anc(X,Y) :- parent(X,Z), anc(Z,Y).}
\]

- For what values of \( Q \) is \( \text{anc(tom,Q)} \) a logical consequence of the above program?
- Negate the goal \( F \): i.e. \( \forall Q. \neg \text{anc(tom, Q)} \).
- Consider the clauses in \( P \cup \neg F \).
  - Note that a program clause written as \( p(A,B) :- q(A,C), r(B,C) \) can be rewritten as:
    \[
    \forall A, B, C. (p(A, B) \lor \neg q(A, C) \lor r(B, C))
    \]
    I.e., l.h.s. literal is \textit{positive}, while all r.h.s. literals are \textit{negative}.
  - Note also that all variables are universally quantified in a clause.
parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).

anc(X,Y) :-
    parent(X,Y).
anc(X,Y) :-
    parent(X,Z),
    anc(Z,Y).

← anc(tom, Q)
    anc(X,Y) ← parent(X,Y)
← parent(tom, Q)
    parent(tom, bob) ←
        □
        Q=bob
Refutation: An Example (contd.)

parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).

anc(X,Y) :-
    parent(X,Y).
anc(X,Y) :-
    parent(X,Z),
    anc(Z,Y).

\[ \text{anc(tom, Q)} \]
\[ \text{anc(X,Y)} \leftarrow \text{parent(X,Z)}, \text{anc(Z,Y)} \]
\[ \text{anc(tom,Z'), anc(Z', Q)} \]
\[ \text{parent(tom, bob)} \leftarrow \]
\[ \text{anc(bob, Q)} \]
\[ \text{anc(X,Y)} \leftarrow \text{parent(X,Y)} \]
\[ \text{anc(X,Y)} \leftarrow \text{parent(bob, X)} \]
\[ \text{parent(bob, ann)} \leftarrow \]
\[ \square \]

Q=ann
Unification

- Operation done to “match” the goal atom with the head of a clause in the program.
- Forms the basis for the *matching* operation we used for Prolog evaluation.
  - \( f(a, Y) \) and \( f(X, b) \) unify when \( X=a \) and \( Y=b \).
  - \( f(a, X) \) and \( f(X, b) \) do not unify.
  - \( X \) and \( f(X) \) do not unify
    (but they “match” in Prolog!)
A substitution is a mapping between variables and values (terms).

- Denoted by \( \{X_1/t_1, X_2/t_2, \ldots, X_n/t_n\} \) such that
  - \( X_i \neq t_i \), and
  - \( X_i \) and \( X_j \) are distinct variables when \( i \neq j \).
- Empty substitution is denoted by \( \epsilon \).
- A substitution is said to be a **renaming** if it is of the form
  \( \{X_1/Y_1, \ldots, X_n/Y_n\} \) and \( Y_1, \ldots, Y_n \) is a permutation of \( X_1, \ldots, X_n \).
- Example: \( \{X/Y, Y/X\} \) is a renaming substitution.
Substitutions and Terms

- Application of a substitution:
  - $X\theta = t$ if $X/t \in \theta$.
  - $X\theta = X$ if $X/t \notin \theta$ for any term $t$.

- Application of a substitution $\{X_1/t_1, \ldots, X_n/t_n\}$ to a term/formula $F$:
  - is a term/formula obtained by simultaneously replacing every free occurrence of $X_i$ in $F$ by $t_i$.
  - Denoted by $F\theta$ [and $F\theta$ is said to be an instance of $F$]

- Example:

$$p(f(X, Z), f(Y, a))\{X/g(Y), Y/Z, Z/a\} = p(f(g(Y), a), f(Z, a))$$
Composition of Substitutions

- Composition of substitutions $\theta = \{X_1/s_1, \ldots, X_m/s_m\}$ and $\sigma = \{Y_1/t_1, \ldots, Y_n/t_n\}$:
  - First form the set $\{X_1/s_1\sigma, \ldots, X_m/s_m\sigma, Y_1/t_1, \ldots, Y_n/t_n\}$
  - Remove from the set $X_i/s_i\sigma$ if $s_i\sigma = X_i$
  - Remove from the set $Y_j/t_j$ if $Y_j$ is identical to some variable $X_i$
- Example: Let $\theta = \sigma = \{X/g(Y), Y/Z, Z/a\}$. Then $\theta\sigma =$

  \[\{X/g(Y), Y/Z, Z/a\}\{X/g(Y), Y/Z, Z/a\} = \{X/g(Z), Y/a, Z/a\}\]

- More examples: Let $\theta = \{X/f(Y)\}$ and $\sigma = \{Y/a\}$
  - $\theta\sigma = \{X/f(a), Y/a\}$
  - $\theta\sigma = \{X/f(Y), Y/a\}$
- Composition is not commutative but is associative: $\theta(\sigma\gamma) = (\theta\sigma)\gamma$
- Also, $E(\theta\sigma) = (E\theta)\sigma$
A substitution $\theta$ is **idempotent** iff $\theta\theta = \theta$.

Examples:
- $\{X/g(Y), Y/Z, Z/a\}$ is not idempotent since
  \[
  \{X/g(Y), Y/Z, Z/a\}\{X/g(Y), Y/Z, Z/a\} = \{X/g(Z), Y/a, Z/a\}
  \]
- $\{X/g(Z), Y/a, Z/a\}$ is not idempotent either since
  \[
  \{X/g(Z), Y/a, Z/a\}\{X/g(Z), Y/a, Z/a\} = \{X/g(a), Y/a, Z/a\}
  \]
- $\{X/g(a), Y/a, Z/a\}$ is idempotent

For a substitution $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$,
- $\text{Dom}(\theta) = \{X_1, X_2, \ldots, X_n\}$
- $\text{Range}(\theta) = \text{set of all variables in } t_1, \ldots, t_n$

A substitution $\theta$ is idempotent iff $\text{Dom}(\theta) \cap \text{Range}(\theta) = \emptyset$
A substitution $\theta$ is a **unifier** of two terms $s$ and $t$ if $s\theta$ is identical to $t\theta$.

$\theta$ is a unifier of a set of equations $\{s_1 \equiv t_1, \ldots, s_n \equiv t_n\}$, if for all $i$, $s_i\theta = t_i\theta$.

A substitution $\theta$ is more general than $\sigma$ (written as $\theta \succeq \sigma$) if there is a substitution $\omega$ such that $\sigma = \theta \omega$.

A substitution $\theta$ is a **most general unifier (mgu)** of two terms (or a set of equations) if for every unifier $\sigma$ of the two terms (or equations) $\theta \succeq \sigma$.

**Example:** Consider two terms $f(g(X), Y, a)$ and $f(Z, W, X)$.

- $\theta_1 = \{X/a, Y/b, Z/g(a), W/b\}$ is a unifier
- $\theta_2 = \{X/a, Y/W, Z/g(a)\}$ is also a unifier
- $\theta_2$ is a most general unifier

Unifiers
Equations and Unifiers

- A set of equations $\mathcal{E}$ is in **solved form** if it is of the form
  \[ \{X_1 \doteq t_1, \ldots, X_n \doteq t_n\} \] iff no $X_i$ appears in any $t_j$.

- Given a set of equations $\mathcal{E} = \{X_1 \doteq t_1, \ldots, X_n \doteq t_n\}$ the substitution
  \[ \{X_1/t_1, \ldots X_n/t_n\} \] is an idempotent mgu of $\mathcal{E}$.

- Two sets of equations $\mathcal{E}_1$ and $\mathcal{E}_2$ are said to be **equivalent** iff they
  have the same set of unifiers.

- To find the mgu of two terms $s$ and $t$, try to find a set of equations in
  solved form that is equivalent to $\{s \doteq t\}$. 
  If there is no equivalent solved form, there is no mgu.
Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \doteq f(g(Z), Z) \} \Rightarrow
\]

Example 2: Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\{ f(X, g(X), b) \doteq f(a, g(Z), Z) \} \Rightarrow
\]

\[
\{ X \doteq a, g(X) \doteq g(Z), b \doteq Z \} \Rightarrow
\]

\[
\{ X \doteq a, g(a) \doteq g(Z), b \doteq Z \} \Rightarrow
\]

\[
\{ X \doteq a, Z \doteq a, b \doteq Z \} \Rightarrow
\]

\[
\{ X \doteq a, Z \doteq a, b \doteq a \} \Rightarrow
\]

\[
\text{fail}
\]
A Simple Unification Algorithm (via Examples)

Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \equiv f(g(Z), Z) \} \Rightarrow \{ X \equiv g(Z), g(Y) \equiv Z \}
\]
A Simple Unification Algorithm (via Examples)

Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \equiv f(g(Z), Z) \} \Rightarrow \{ X \equiv g(Z), g(Y) \equiv Z \}
\]

\[
\Rightarrow \{ X \equiv g(Z), Z \equiv g(Y) \}
\]
A Simple Unification Algorithm (via Examples)

- Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \equiv f(g(Z), Z) \} \Rightarrow \{ X \equiv g(Z), g(Y) \equiv Z \} \\
\Rightarrow \{ X \equiv g(Z), Z \equiv g(Y) \} \\
\Rightarrow \{ X \equiv g(g(Y)), Z \equiv g(Y) \}
\]
A Simple Unification Algorithm (via Examples)

- Example 1: Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

  $$
  \{f(X, g(Y)) \doteq f(g(Z), Z)\} \Rightarrow \{X \doteq g(Z), g(Y) \doteq Z\}
  \Rightarrow \{X \doteq g(Z), Z \doteq g(Y)\}
  \Rightarrow \{X \doteq g(g(Y)), Z \doteq g(Y)\}
  $$

- Example 2: Find the mgu of $f(X, g(X), b)$ and $f(a, g(Z), Z)$

  $$
  \{f(X, g(X), b) \doteq f(a, g(Z), Z)\} \Rightarrow \{X \doteq a, Z \doteq a, Z \doteq a\}
  $$
A Simple Unification Algorithm (via Examples)

- Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{f(X, g(Y)) \doteq f(g(Z), Z)\} \Rightarrow \{X \doteq g(Z), g(Y) \doteq Z\} \\
\Rightarrow \{X \doteq g(Z), Z \doteq g(Y)\} \\
\Rightarrow \{X \doteq g(g(Y)), Z \doteq g(Y)\}
\]

- Example 2: Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\{f(X, g(X), b) \doteq f(a, g(Z), Z)\} \Rightarrow \{X \doteq a, g(X) \doteq g(Z), b \doteq Z\}
\]
A Simple Unification Algorithm (via Examples)

- **Example 1:** Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \models f(g(Z), Z) \} \quad \Rightarrow \quad \{ X \models g(Z), g(Y) \models Z \}
\]

\[
\Rightarrow \quad \{ X \models g(Z), Z \models g(Y) \}
\]

\[
\Rightarrow \quad \{ X \models g(g(Y)), Z \models g(Y) \}
\]

- **Example 2:** Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\{ f(X, g(X), b) \models f(a, g(Z), Z) \} \quad \Rightarrow \quad \{ X \models a, g(X) \models g(Z), b \models Z \}
\]

\[
\Rightarrow \quad \{ X \models a, g(a) \models g(Z), b \models Z \}
\]
Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) \doteq f(g(Z), Z) \} \Rightarrow \{ X \doteq g(Z), g(Y) \doteq Z \} \\
\Rightarrow \{ X \doteq g(Z), Z \doteq g(Y) \} \\
\Rightarrow \{ X \doteq g(g(Y)), Z \doteq g(Y) \}
\]

Example 2: Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\{ f(X, g(X), b) \doteq f(a, g(Z), Z) \} \Rightarrow \{ X \doteq a, g(X) \doteq g(Z), b \doteq Z \} \\
\Rightarrow \{ X \doteq a, g(a) \doteq g(Z), b \doteq Z \} \\
\Rightarrow \{ X \doteq a, a \doteq Z, b \doteq Z \}
\]
A Simple Unification Algorithm (via Examples)

- Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\{ f(X, g(Y)) = f(g(Z), Z) \} \quad \Rightarrow \quad \{ X \equiv g(Z), g(Y) \equiv Z \} \\
\Rightarrow \quad \{ X \equiv g(Z), Z \equiv g(Y) \} \\
\Rightarrow \quad \{ X \equiv g(g(Y)), Z \equiv g(Y) \}
\]

- Example 2: Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\{ f(X, g(X), b) = f(a, g(Z), Z) \} \quad \Rightarrow \quad \{ X \equiv a, g(X) \equiv g(Z), b \equiv Z \} \\
\Rightarrow \quad \{ X \equiv a, g(a) \equiv g(Z), b \equiv Z \} \\
\Rightarrow \quad \{ X \equiv a, a \equiv Z, b \equiv Z \} \\
\Rightarrow \quad \{ X \equiv a, Z \equiv a, b \equiv Z \}
\]
A Simple Unification Algorithm (via Examples)

- **Example 1:** Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

  \[
  \{f(X, g(Y)) \equiv f(g(Z), Z)\} \Rightarrow \{X \equiv g(Z), g(Y) \equiv Z\}
  \]

  \[
  \Rightarrow \{X \equiv g(Z), Z \equiv g(Y)\}
  \]

  \[
  \Rightarrow \{X \equiv g(g(Y)), Z \equiv g(Y)\}
  \]

- **Example 2:** Find the mgu of $f(X, g(X), b)$ and $f(a, g(Z), Z)$

  \[
  \{f(X, g(X), b) \equiv f(a, g(Z), Z)\} \Rightarrow \{X \equiv a, g(X) \equiv g(Z), b \equiv Z\}
  \]

  \[
  \Rightarrow \{X \equiv a, g(a) \equiv g(Z), b \equiv Z\}
  \]

  \[
  \Rightarrow \{X \equiv a, a \equiv Z, b \equiv Z\}
  \]

  \[
  \Rightarrow \{X \equiv a, Z \equiv a, b \equiv Z\}
  \]

  \[
  \Rightarrow \{X \equiv a, Z \equiv a, b \equiv a\}
  \]
A Simple Unification Algorithm (via Examples)

- Example 1: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)

\[
\begin{align*}
\{ f(X, g(Y)) & \doteq f(g(Z), Z) \} \Rightarrow \{ X \doteq g(Z), g(Y) \doteq Z \} \\
& \Rightarrow \{ X \doteq g(Z), Z \doteq g(Y) \} \\
& \Rightarrow \{ X \doteq g(g(Y)), Z \doteq g(Y) \}
\end{align*}
\]

- Example 2: Find the mgu of \( f(X, g(X), b) \) and \( f(a, g(Z), Z) \)

\[
\begin{align*}
\{ f(X, g(X), b) & \doteq f(a, g(Z), Z) \} \Rightarrow \{ X \doteq a, g(X) \doteq g(Z), b \doteq Z \} \\
& \Rightarrow \{ X \doteq a, g(a) \doteq g(Z), b \doteq Z \} \\
& \Rightarrow \{ X \doteq a, a \doteq Z, b \doteq Z \} \\
& \Rightarrow \{ X \doteq a, Z \doteq a, b \doteq Z \} \\
& \Rightarrow \{ X \doteq a, Z \doteq a, b \doteq a \} \\
& \Rightarrow \text{fail}
\end{align*}
\]
A Simple Unification Algorithm

Given a set of equations $\mathcal{E}$:

repeat
  select $s = t \in \mathcal{E}$;
  case $s = t$ of
    1. $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$:
       replace the equation by $s_i = t_i$ for all $i$
    2. $f(s_1, \ldots, s_n) = g(t_1, \ldots, t_m)$, $f \neq g$ or $n \neq m$:
       halt with failure
    3. $X = X$: remove the equation
    4. $t = X$: where $t$ is not a variable
       replace equation by $X = t$
    5. $X = t$: where $X \neq t$ and $X$ occurs more than once in $\mathcal{E}$:
       if $X$ is a proper subterm of $t$
          then halt with failure (5a)
          else replace all other $X$ in $\mathcal{E}$ by $t$ (5b)
  until no action is possible for any equation in $\mathcal{E}$
return $\mathcal{E}$
Example 1: Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

$\{f(X, g(Y)) \doteq f(g(Z), Z)\} \Rightarrow$
Example 1: Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

\[
\{f(X, g(Y)) \doteq f(g(Z), Z)\} \Rightarrow \{X \doteq g(Z), g(Y) \doteq Z\} \quad \text{case 1}
\]
A Simple Unification Algorithm (More Examples)

- **Example 1:** Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)
  \[
  \{ f(X, g(Y)) \equiv f(g(Z), Z) \} \implies \{ X \equiv g(Z), g(Y) \equiv Z \} \quad \text{case 1}
  \]
  \[
  \implies \{ X \equiv g(Z), Z \equiv g(Y) \} \quad \text{case 4}
  \]
Example 1: Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

\[
\{f(X, g(Y)) \doteq f(g(Z), Z)\} \Rightarrow \{X \doteq g(Z), g(Y) \doteq Z\} \quad \text{case 1}
\]

\[
\Rightarrow \{X \doteq g(Z), Z \doteq g(Y)\} \quad \text{case 4}
\]

\[
\Rightarrow \{X = g(g(Y)), Z = g(Y)\} \quad \text{case 5b}
\]
Example 1: Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$

\[
\{ f(X, g(Y)) \doteq f(g(Z), Z) \} \quad \Rightarrow \quad \{ X \doteq g(Z), g(Y) \doteq Z \} \quad \text{case 1}
\]

\[
\Rightarrow \quad \{ X \doteq g(Z), Z \doteq g(Y) \} \quad \text{case 4}
\]

\[
\Rightarrow \quad \{ X \doteq g(g(Y)), Z \doteq g(Y) \} \quad \text{case 5b}
\]

Example 3: Find the mgu of $f(X, g(X))$ and $f(Z, Z)$

\[
\{ f(X, g(X)) \doteq f(Z, Z) \} \quad \Rightarrow
\]
A Simple Unification Algorithm (More Examples)

- **Example 1:** Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$
  \[
  \{f(X, g(Y)) \rightleftharpoons f(g(Z), Z)\} \Rightarrow \{X \rightleftharpoons g(Z), g(Y) \rightleftharpoons Z\}
  \Rightarrow \{X \rightleftharpoons g(Z), Z \rightleftharpoons g(Y)\}
  \Rightarrow \{X \rightleftharpoons g(g(Y)), Z \rightleftharpoons g(Y)\}
  \]
  cases: 1, 4, 5b

- **Example 3:** Find the mgu of $f(X, g(X))$ and $f(Z, Z)$
  \[
  \{f(X, g(X)) \rightleftharpoons f(Z, Z)\} \Rightarrow \{X \rightleftharpoons Z, g(X) \rightleftharpoons Z\}
  \]
  case: 1
A Simple Unification Algorithm (More Examples)

**Example 1:** Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)
\[
\{ f(X, g(Y)) \doteq f(g(Z), Z) \} \quad \Rightarrow \quad \{ X \doteq g(Z), g(Y) \doteq Z \} \\
\Rightarrow \quad \{ X \doteq g(Z), Z \doteq g(Y) \} \\
\Rightarrow \quad \{ X \doteq g(g(Y)), Z \doteq g(Y) \}
\]
case 1
case 4
case 5b

**Example 3:** Find the mgu of \( f(X, g(X)) \) and \( f(Z, Z) \)
\[
\{ f(X, g(X)) \doteq f(Z, Z) \} \quad \Rightarrow \quad \{ X \doteq Z, g(X) \doteq Z \} \\
\Rightarrow \quad \{ X \doteq Z, g(Z) \doteq Z \}
\]
case 1
case 5b
A Simple Unification Algorithm (More Examples)

- **Example 1:** Find the mgu of $f(X, g(Y))$ and $f(g(Z), Z)$
  \[
  \{f(X, g(Y)) \doteq f(g(Z), Z)\} \quad \Rightarrow \quad \{X \doteq g(Z), g(Y) \doteq Z\}
  \]
  \[
  \Rightarrow \quad \{X \doteq g(Z), Z \doteq g(Y)\}
  \]
  \[
  \Rightarrow \quad \{X \doteq g(g(Y)), Z \doteq g(Y)\}
  \]

- **Example 3:** Find the mgu of $f(X, g(X))$ and $f(Z, Z)$
  \[
  \{f(X, g(X)) \doteq f(Z, Z)\} \quad \Rightarrow \quad \{X \doteq Z, g(X) \doteq Z\}
  \]
  \[
  \Rightarrow \quad \{X \doteq Z, g(Z) \doteq Z\}
  \]
  \[
  \Rightarrow \quad \{X \doteq Z, Z \doteq g(Z)\}
  \]
A Simple Unification Algorithm (More Examples)

- **Example 1**: Find the mgu of \( f(X, g(Y)) \) and \( f(g(Z), Z) \)
  \[
  \{ f(X, g(Y)) = f(g(Z), Z) \} \quad \Rightarrow \quad \{ X = g(Z), g(Y) = Z \} \quad \text{case 1}
  \]
  \[
  \Rightarrow \quad \{ X = g(Z), Z = g(Y) \} \quad \text{case 4}
  \]
  \[
  \Rightarrow \quad \{ X = g(g(Y)), Z = g(Y) \} \quad \text{case 5b}
  \]

- **Example 3**: Find the mgu of \( f(X, g(X)) \) and \( f(Z, Z) \)
  \[
  \{ f(X, g(X)) = f(Z, Z) \} \quad \Rightarrow \quad \{ X = Z, g(X) = Z \} \quad \text{case 1}
  \]
  \[
  \Rightarrow \quad \{ X = Z, g(Z) = Z \} \quad \text{case 5b}
  \]
  \[
  \Rightarrow \quad \{ X = Z, Z = g(Z) \} \quad \text{case 4}
  \]
  \[
  \Rightarrow \quad \text{fail} \quad \text{case 5a}
  \]
Complexity of the unification algorithm

Consider
\[ \mathcal{E} = \{ g(X_1, \ldots, X_n) = g(f(X_0, X_0), f(X_1, X_1), \ldots, f(X_{n-1}, X_{n-1}) \} . \]

- By applying case 1 of the algorithm, we get
  \[ \{ X_1 = f(X_0, X_0), X_2 = f(X_1, X_1), \ldots, X_n = f(X_{n-1}, X_{n-1}) \} \]

- If terms are kept as trees, the final value for \( X_n \) is a tree of size \( O(2^n) \).
- Recall that for case 5 we need to first check if a variable appears in a term, and this could now take \( O(2^n) \) time.
- There are linear-time unification algorithms that share structures (terms as DAGs).
- \( X = t \) is the most common case for unification in Prolog. The fastest algorithms are linear in \( t \).
- Prolog cuts corners by omitting case 5a (the occur check), thereby doing \( X = t \) in constant time.
Most General Unifiers

- Note that mgu stands for a most general unifier.
- There may be more than one mgu. E.g. \( f(X) = f(Y) \) has two mgus:
  - \( \{X/Y\} \)
  - \( \{Y/X\} \)
- If \( \theta \) is an mgu of \( s \) and \( t \), and \( \omega \) is a renaming, then \( \theta \omega \) is an mgu of \( s \) and \( t \).
- If \( \theta \) and \( \sigma \) are mgus of \( s \) and \( t \), then there is a renaming \( \omega \) such that \( \theta = \sigma \omega \).
  - mgu is unique up to renaming.
SLD Resolution

\[
\text{\begin{align*}
\leftarrow A_1, \ldots, A_{i-1}, & A_i, A_{i+1}, \ldots, A_m & B_0 & \leftarrow B_1, \ldots, B_n \\
\leftarrow (A_1, \ldots, A_{i-1}, & B_1, \ldots, B_n, A_{i+1}, \ldots, A_m)\theta
\end{align*}}
\]

where:

1. \(A_j\)s are atomic formulas
2. \(B_0 \leftarrow B_1, \ldots, B_n\) is a (renamed) definite clause in the program
3. \(\theta = \text{mgu}(A_i, B_0)\)

- \(A_i\) is called the selected atom.
- Given a goal \(\leftarrow A_1, \ldots, A_n\), a function called the selection function or computation rule selects \(A_i\).
When the resolution rule is applied, from a goal $G$ and a clause $C$, we get a new goal $G'$.
We then say that $G'$ is derived directly from $G$ and $C$:

$$ G \xrightarrow{C} G' $$

An SLD Derivation is a sequence

$$ G_0 \xrightarrow{C_0} G_1 \cdots G_i \xrightarrow{C_i} G_{i+1} \cdots $$
Refutation & SLD Derivation

parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).

anc(X,Y) :-
    parent(X,Y).
anc(X,Y) :-
    parent(X,Z),
    anc(Z,Y).

← anc(tom, Q)
    → parent(tom, Q)
    □
    Q=bob

→ parent(tom, Q)

→ □

Computing with Logic
Resolution
CSE 505
### SLD Derivation (contd.)

<table>
<thead>
<tr>
<th>Parent Relations</th>
<th>Ancestral Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>parent(pam, bob)</td>
<td>anc(tom, Q)</td>
</tr>
<tr>
<td>parent(tom, bob)</td>
<td>anc(bob, Q)</td>
</tr>
<tr>
<td>parent(tom, liz)</td>
<td>parent(bob, Q)</td>
</tr>
<tr>
<td>parent(bob, ann)</td>
<td></td>
</tr>
<tr>
<td>parent(bob, pat)</td>
<td></td>
</tr>
<tr>
<td>parent(pat, jim)</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{anc}(X,Y) :\rightarrow \text{parent}(X,Y).
\]

\[
\text{anc}(X,Y) :\rightarrow \text{parent}(X, Z), \text{anc}(Z, Y).
\]

\[
\text{anc}(X,Y) :\rightarrow \text{parent}(X, Z'), \text{anc}(Z', Q).
\]

\[
\text{anc}(X,Y) :\rightarrow \text{parent}(bob, Q).
\]

\[
\text{anc}(tom, Q) :\rightarrow \text{parent}(tom, Z'), \text{anc}(Z', Q).
\]

\[
\text{anc}(tom, Q) :\rightarrow \text{parent}(bob, Q).
\]

\[
\text{anc}(tom, Q) :\rightarrow \text{parent}(bob, Q).
\]

\[
\text{anc}(tom, Q) :\rightarrow \Box.
\]

\[
\text{Q}=\text{ann}
\]
**Computed Answer Substitution**

- Let $\theta_0, \theta_1, \ldots, \theta_{n-1}$ be the sequence of mgus used in derivation

$$G_0 \xRightarrow{\theta_0} G_1 \cdots G_{n-1} \xRightarrow{\theta_{n-1}} G_n$$

Then $\theta = \theta_0 \theta_1 \cdots \theta_{n-1}$ is the *computed substitution* of the derivation.

**Example:**

<table>
<thead>
<tr>
<th>Goal</th>
<th>Clause Used</th>
<th>mgu</th>
</tr>
</thead>
<tbody>
<tr>
<td>anc(tom, Q)</td>
<td>$\text{anc}(X', Y') :- \text{parent}(X', Z'), \text{anc}(Z', Y')$</td>
<td>$\theta_0 = {X'/\text{tom}, Y'/Q}$</td>
</tr>
<tr>
<td>parent(tom, Z'), anc(Z', Q)</td>
<td>$\text{parent}(\text{tom}, \text{bob}).$</td>
<td>$\theta_1 = {Z'/\text{bob}}$</td>
</tr>
<tr>
<td>anc(bob, Q)</td>
<td>$\text{anc}(X'', Y'') :- \text{parent}(X'', Y'').$</td>
<td>$\theta_2 = {X''/\text{bob}, Y''/Q}$</td>
</tr>
<tr>
<td>parent(bob, Q)</td>
<td>$\text{parent}(\text{bob}, \text{ann}).$</td>
<td>$\theta_3 = {Q/\text{ann}}$</td>
</tr>
</tbody>
</table>

- Computed substitution for the above derivation is $\theta_0 \theta_1 \theta_2 \theta_3 = \{X'/\text{tom}, Y'/\text{ann}, Z'/\text{bob}, X''/\text{bob}, Y''/\text{ann}, Q/\text{ann}\}$
A finite derivation of the form

\[ G_0 \xrightarrow{C_0} G_1 \ldots G_{n-1} \xrightarrow{C_{n-1}} G_n \]

where \( G_n = \square \) (i.e., an empty goal) is an \textit{SLD refutation} of \( G_0 \).

The computed substitution of an SLD refutation of \( G \), restricted to variables of \( G \), is a \textit{computed answer substitution} for \( G \).

Example (contd.): The computed answer substitution for the above SLD refutation is

\[ \{ X'/\text{tom}, Y'/\text{ann}, Z'/\text{bob}, X''/\text{bob}, Y''/\text{ann}, Q/\text{ann} \} \]

restricted to \( Q \):

\[ \{ Q/\text{ann} \} \]
Failed SLD Derivation

- A derivation of a goal clause $G_0$ whose last element is not empty, and cannot be resolved with any clause of the program.

Example: consider the following program:

\[
\begin{align*}
grandfather(X,Z) & :\neg \text{father}(X,Y), \text{parent}(Y,Z). \\
\text{parent}(X,Y) & :\neg \text{father}(X,Y). \\
\text{parent}(X,Y) & :\neg \text{mother}(X,Y). \\
father(a,b). \\
mother(b,c).
\end{align*}
\]

A derivation of \text{grandfather}(a,Q) is:

\[
\begin{align*}
& \Rightarrow \text{father}(a,Y'), \text{parent}(Y',Q) \\
& \Rightarrow \text{parent}(b,Q) \\
& \Rightarrow \text{father}(b,Q)
\end{align*}
\]
A tree where every path is an SLD derivation

grandfather(X,Z) :-
    father(X,Y), parent(Y,Z).

parent(X,Y) :- father(X,Y).
parent(X,Y) :- mother(X,Y).

father(a,b).
mother(b,c).

← grandfather(a, Q)  
  ← father(a, Q'), parent(Q', Q)  
    ← parent(b, Q)  
      ← father(b, Q)  
        ← mother(b, Q)
Soundness of SLD resolution

- Let $P$ be a definite program, $\mathcal{R}$ be a computation rule, and $\theta$ be a computed answer substitution for a goal $G$. Then $\forall G \theta$ is a logical consequence of $P$.

- Proof is by induction on the number of resolution steps used in the refutation of $G$.

- Base case uses the following lemma:
  - Let $F$ be a formula and $F'$ be an instance of $F$, i.e. $F' = F\theta$ for some substitution $\theta$.
  - Then $(\forall F) \models (\forall F')$. 