An instance of an atom or a clause is obtained by uniformly substituting terms for variables.

A substitution is a finite set of the form \( \{ V_1/t_1, \ldots, V_n/t_n \} \), where each \( V_i \) is a distinct variable and each \( t_i \) is a term.

The application of a substitution \( \sigma = \{ V_1/t_1, \ldots, V_n/t_n \} \) to an atom or clause \( e \), written \( e\sigma \), is the instance of \( e \) with every occurrence of \( V_i \) replaced by \( t_i \).
The following are substitutions:

- $\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$
- $\sigma_2 = \{A/X, Y/b, C/Z, D/e\}$
- $\sigma_3 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$

The following shows some applications:

- $p(A, b, C, D)\sigma_1 =$
- $p(X, Y, Z, e)\sigma_1 =$
- $p(A, b, C, D)\sigma_2 =$
- $p(X, Y, Z, e)\sigma_2 =$
- $p(A, b, C, D)\sigma_3 =$
- $p(X, Y, Z, e)\sigma_3 =$
The following are substitutions:

- $\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$
- $\sigma_2 = \{A/X, Y/b, C/Z, D/e\}$
- $\sigma_3 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$

The following shows some applications:

- $p(A, b, C, D)\sigma_1 = p(A, b, C, e)$
- $p(X, Y, Z, e)\sigma_1 = p(A, b, C, e)$
- $p(A, b, C, D)\sigma_2 = p(X, b, Z, e)$
- $p(X, Y, Z, e)\sigma_2 = p(X, b, Z, e)$
- $p(A, b, C, D)\sigma_3 = p(V, b, W, e)$
- $p(X, Y, Z, e)\sigma_3 = p(V, b, W, e)$
Unifiers

- Substitution $\sigma$ is a unifier of $e_1$ and $e_2$ if $e_1 \sigma = e_2 \sigma$.
- Substitution $\sigma$ is a most general unifier (mgu) of $e_1$ and $e_2$ if
  - $\sigma$ is a unifier of $e_1$ and $e_2$; and
  - if substitution $\sigma'$ also unifies $e_1$ and $e_2$, then $e\sigma'$ is an instance of $e\sigma$ for all atoms $e$.
- If two atoms have a unifier, they have a most general unifier.
Unification Example

Which of the following are unifiers of $p(A, b, C, D)$ and $p(X, Y, Z, e)$:

- $\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$
- $\sigma_2 = \{Y/b, D/e\}$
- $\sigma_3 = \{X/A, Y/b, Z/C, D/e, W/a\}$
- $\sigma_4 = \{A/X, Y/b, C/Z, D/e\}$
- $\sigma_5 = \{X/a, Y/b, Z/c, D/e\}$
- $\sigma_6 = \{A/a, X/a, Y/b, C/c, Z/c, D/e\}$
- $\sigma_7 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$
- $\sigma_8 = \{X/A, Y/b, Z/A, C/A, D/e\}$

Which are most general unifiers?
Unification Example

\[ p(A, b, C, D) \text{ and } p(X, Y, Z, e) \text{ have as unifiers:} \]

- \( \sigma_1 = \{X/A, Y/b, Z/C, D/e\} \)
- \( \sigma_4 = \{A/X, Y/b, C/Z, D/e\} \)
- \( \sigma_7 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\} \)
- \( \sigma_6 = \{A/a, X/a, Y/b, C/c, Z/c, D/e\} \)
- \( \sigma_8 = \{X/A, Y/b, Z/A, C/A, D/e\} \)
- \( \sigma_3 = \{X/A, Y/b, Z/C, D/e, W/a\} \)

The first three are most general unifiers. The following substitutions are not unifiers:

- \( \sigma_2 = \{Y/b, D/e\} \)
- \( \sigma_5 = \{X/a, Y/b, Z/c, D/e\} \)
Proofs

- A proof is a mechanically derivable demonstration that a formula logically follows from a knowledge base.
- Given a proof procedure, \( KB \vdash g \) means \( g \) can be derived from knowledge base \( KB \).
- Recall \( KB \models g \) means \( g \) is true in all models of \( KB \).
- A proof procedure is sound if \( KB \vdash g \) implies \( KB \models g \).
- A proof procedure is complete if \( KB \models g \) implies \( KB \vdash g \).
Bottom-up proof procedure

$KB \vdash g$ if there is $g'$ added to $C$ in this procedure where $g = g'\theta$:

\[
C := \{\} ;
\]
repeat

select clause “$h \leftarrow b_1 \land \ldots \land b_m$” in $KB$ such that
there is a substitution $\theta$ such that
for all $i$, there exists $b'_i \in C$ where $b_i\theta = b'_i\theta$ and
there is no $h' \in C$ such that $h'$ is more general than $h\theta$

$C := C \cup \{h\theta\}$

until no more clauses can be selected.
Example

\[\text{live}(Y) \leftarrow \text{connected}_\to(Y, Z) \land \text{live}(Z). \quad \text{live(outside)}.\]
\[\text{connected}_\to(w_6, w_5). \quad \text{connected}_\to(w_5, \text{outside}).\]
Example

\[ \text{live}(Y) \leftarrow \text{connected \_to}(Y, Z) \land \text{live}(Z). \ \text{live(\text{outside})}. \]

\text{connected \_to}(w_6, w_5). \ \text{connected \_to}(w_5, \text{outside}).

\[ C = \{ \text{live(\text{outside})}, \]
\[ \text{connected \_to}(w_6, w_5), \]
\[ \text{connected \_to}(w_5, \text{outside}), \]
\[ \text{live}(w_5), \]
\[ \text{live}(w_6) \} \]
Soundness of bottom-up proof procedure

If $KB \vdash g$ then $KB \models g$.

- Suppose there is a $g$ such that $KB \vdash g$ and $KB \not\models g$.
- Then there must be a first atom added to $C$ that has an instance that isn’t true in every model of $KB$. Call it $h$.
  Suppose $h$ isn’t true in model $I$ of $KB$.
- There must be a clause in $KB$ of form
  \[ h' \leftarrow b_1 \land \ldots \land b_m \]
  where $h = h' \theta$. Each $b_i$ is true in $I$. $h$ is false in $I$. So this clause is false in $I$. Therefore $I$ isn’t a model of $KB$.
- Contradiction.
The $C$ generated by the bottom-up algorithm is called a fixed point.

$C$ can be infinite; we require the selection to be fair.

**Herbrand interpretation:** The domain is the set of constants. We invent one if the KB or query doesn’t contain one. Each constant denotes itself.

Let $I$ be the Herbrand interpretation in which every ground instance of every element of the fixed point is true and every other atom is false.

$I$ is a model of $KB$.

Proof: suppose $h \leftarrow b_1 \land \ldots \land b_m$ in $KB$ is false in $I$. Then $h$ is false and each $b_i$ is true in $I$. Thus $h$ can be added to $C$. Contradiction to $C$ being the fixed point.

$I$ is called a **Minimal Model**.
Completeness

If $KB \models g$ then $KB \vdash g$.

- Suppose $KB \models g$. Then $g$ is true in all models of $KB$.
- Thus $g$ is true in the minimal model.
- Thus $g$ is in the fixed point.
- Thus $g$ is generated by the bottom up algorithm.
- Thus $KB \vdash g$. 
A **generalized answer clause** is of the form

\[
\text{yes}(t_1, \ldots, t_k) \leftarrow a_1 \land a_2 \land \ldots \land a_m,
\]

where \( t_1, \ldots, t_k \) are terms and \( a_1, \ldots, a_m \) are atoms.

The **SLD resolution** of this generalized answer clause on \( a_i \) with the clause

\[
a \leftarrow b_1 \land \ldots \land b_p,
\]

where \( a_i \) and \( a \) have most general unifier \( \theta \), is

\[
\begin{align*}
\text{(yes}(t_1, \ldots, t_k) & \leftarrow \\
& a_1 \land \ldots \land a_{i-1} \land b_1 \land \ldots \land b_p \land a_{i+1} \land \ldots \land a_m) \theta.
\end{align*}
\]
To solve query \(?B\) with variables \(V_1, \ldots, V_k\):

Set \(ac\) to generalized answer clause \(yes(V_1, \ldots, V_k) \leftarrow B;\)

While \(ac\) is not an answer do

Suppose \(ac\) is \(yes(t_1, \ldots, t_k) \leftarrow a_1 \land a_2 \land \ldots \land a_m\)

Select atom \(a_i\) in the body of \(ac\);

Choose clause \(a \leftarrow b_1 \land \ldots \land b_p\) in \(KB\);

Rename all variables in \(a \leftarrow b_1 \land \ldots \land b_p\);

Let \(\theta\) be the most general unifier of \(a_i\) and \(a\).

Fail if they don’t unify;

Set \(ac\) to \((yes(t_1, \ldots, t_k) \leftarrow a_1 \land \ldots \land a_{i-1} \land b_1 \land \ldots \land b_p \land a_{i+1} \land \ldots \land a_m)\theta\)

end while.
Example

\[
\begin{align*}
live(Y) & \leftarrow \text{connected\_to}(Y, Z) \land live(Z). \quad live(\text{outside}). \\
\text{connected\_to}(w_6, w_5). & \quad \text{connected\_to}(w_5, \text{outside}). \\
?\text{live}(A).
\end{align*}
\]
Example

\[
\begin{align*}
  \text{live}(Y) & \leftarrow \text{connected\_to}(Y, Z) \land \text{live}(Z). & \text{live(\text{outside}).}
  \\
  \text{connected\_to}(w_6, w_5). & \quad \text{connected\_to}(w_5, \text{outside}).
  \\
  \text{?live}(A). & \\
  \text{yes}(A) & \leftarrow \text{live}(A).
  \\
  \text{yes}(A) & \leftarrow \text{connected\_to}(A, Z_1) \land \text{live}(Z_1).
  \\
  \text{yes}(w_6) & \leftarrow \text{live}(w_5).
  \\
  \text{yes}(w_6) & \leftarrow \text{connected\_to}(w_5, Z_2) \land \text{live}(Z_2).
  \\
  \text{yes}(w_6) & \leftarrow \text{live}(\text{outside}).
  \\
  \text{yes}(w_6) & \leftarrow .
\end{align*}
\]
Often we want to refer to individuals in terms of components.
Examples: 4:55 p.m. English sentences. A classlist.
We extend the notion of term. So that a term can be $f(t_1, \ldots, t_n)$ where $f$ is a function symbol and the $t_i$ are terms.
In an interpretation and with a variable assignment, term $f(t_1, \ldots, t_n)$ denotes an individual in the domain.
One function symbol and one constant can refer to infinitely many individuals.
A list is an ordered sequence of elements.

Let’s use the constant \texttt{nil} to denote the empty list, and the function \texttt{cons}(H, T) to denote the list with first element \textit{H} and rest-of-list \textit{T}. \textbf{These are not built-in.}

The list containing \textit{sue}, \textit{kim} and \textit{randy} is

\[
\text{cons}(\textit{sue}, \text{cons}(\textit{kim}, \text{cons}(\textit{randy}, \texttt{nil})))
\]

\textbf{append}(X, Y, Z) is true if list \textit{Z} contains the elements of \textit{X} followed by the elements of \textit{Y}

\[
\text{append}(\texttt{nil}, Z, Z).
\]

\[
\text{append}(\text{cons}(A, X), Y, \text{cons}(A, Z)) \leftarrow \text{append}(X, Y, Z).
\]