Chapter 2

Unification
Outline

- Understanding the need for unification
- Defining alphabets, terms, and substitutions
- Introducing the Martelli-Montanari Algorithm for unification
- Proving correctness of the algorithm
The Need to Perform Unification (I)

direct(frankfurt,san_francisco).
direct(frankfurt,chicago).
direct(san_francisco,honolulu).
direct(honolulu,maui).

collection(X, Y) :- direct(X, Y).
collection(X, Y) :- direct(X, Z), collection(Z, Y).

?- collection(frankfurt, maui).

yes
The Need to Perform Unification (II)

\[ p(f(X), g(f(c), X)) \].

\[ |- p(U, g(V, f(W))). \]

\[ U = f(f(W)), \]
\[ V = f(c) \]

\[ |- p(U, g(c, f(W))). \]

no

\[ |- p(U, g(V, U)). \]

\[ U = f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(f(\ldots
\]
Ranked Alphabets and Term Universes

- Variables

- Ranked alphabet is a finite set $\Sigma$ of symbols; to every symbol a natural number $\geq 0$ (its arity or rank) is assigned ($\Sigma^{(n)}$ denotes the subset of $\Sigma$ with symbols of arity $n$)

- Parentheses, commas

- $V$ set of variables, $F$ ranked alphabet of function symbols:
  Term universe $TU_{F,V}$ (over $F$ and $V$) is smallest set $T$ of terms with
  1. $V \subseteq T$
  2. $f \in T$, if $f \in F^{(0)}$ (also called a constant)
  3. $f(t_1, \ldots, t_n) \in T$, if $f \in F^{(n)}$ with $n \geq 1$ and $t_1, \ldots, t_n \in T$
Ground Terms and Sub-Terms

- $\text{Var}(t) \iff \text{set of variables in } t$

- $t$ ground term $\iff \text{Var}(t) = \emptyset$

- $s$ sub-term of $t : \iff \text{term } s \text{ is sub-string of } t$
Substitutions (I)

\[ V \text{ set of variables, finite set } X \subseteq V, \ F \text{ ranked alphabet:} \]

**Substitution** \( \iff \) function \( \theta : X \rightarrow T U_{F,V} \) with \( x \neq \theta(x) \) for every \( x \in X \)

We use notation \( \theta = \{x_1/t_1, ..., x_n/t_n\} \), where

1. \( X = \{x_1, ..., x_n\} \)
2. \( \theta(x_i) = t_i \) for every \( x_i \in X \)
Consider a substitution $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$.

- **empty substitution** $\varepsilon :\iff n = 0$
- $\theta$ ground substitution $:\iff t_1, \ldots, t_n$ ground terms
- $\theta$ pure variable substitution $:\iff t_1, \ldots, t_n$ variables
- $\theta$ renaming $:\iff \{t_1, \ldots, t_n\} = \{x_1, \ldots, x_n\}$
- $\text{Dom}(\theta) :\iff \{x_1, \ldots, x_n\}$
- $Y \subseteq V : \theta|_Y :\iff \{y/t \mid y/t \in \theta \text{ and } y \in Y\}$
Applying Substitutions

- If $x$ is a variable and $x \in \text{Dom}(\theta)$, then $x\theta \iff \theta(x)$
- If $x$ is a variable and $x \notin \text{Dom}(\theta)$, then $x\theta \iff x$
- $f(t_1, \ldots, t_n)\theta \iff f(t_1\theta, \ldots, t_n\theta)$

- $t$ instance of $s \iff$ there is substitution $\theta$ with $s\theta = t$
- $s$ more general than $t \iff t$ instance of $s$
- $t$ variant of $s \iff$ there is renaming $\theta$ with $s\theta = t$

Lemma 2.5
$t$ variant of $s$ iff $t$ instance of $s$ and $s$ instance of $t$
Composition

Let $\theta$ and $\eta$ be substitutions.

The composition $\theta \eta$ is defined by $(\theta \eta)(x) :\Leftrightarrow (x\theta)\eta$ for each variable $x$

Lemma 2.3

Let $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$, $\eta = \{y_1/s_1, \ldots, y_m/s_m\}$.

Then $\theta \eta$ can be constructed from the sequence

$$x_1/t_1 \eta, \ldots, x_n/t_n \eta, y_1/s_1, \ldots, y_m/s_m$$

1. by removing all bindings $x_i/t_i \eta$ where $x_i = t_i \eta$,

    and all bindings $y_j/s_j$ where $y_j \in \{x_1, \ldots, x_n\}$

2. by forming a substitution from the resulting sequence

Examples:

- $\{x/y, z/x\} \cdot \{y/7, x/z\} = \{x/7, y/7\}$
- $\{y/7, x/z\} \cdot \{x/y, z/x\} = \{y/7, z/x\}$
Definition 2.6
Let \( \theta \) and \( \tau \) be substitutions.
\( \theta \) more general than \( \tau \) :\( \iff \tau = \theta \eta \) for some substitution \( \eta \)

Examples:
* \( \theta = \{x/y\} \) is more general than \( \tau = \{x/a,y/a\} \) (with \( \eta = \{y/a\} \))
* \( \theta = \{x/y\} \) is not more general than \( \tau = \{x/a\} \)
  since for every \( \eta \) with \( \tau = \theta \eta \):
  \( x/a \in \{x/y\} \eta \Rightarrow y/a \in \eta \Rightarrow y \in \text{Dom}(\theta \eta) = \text{Dom}(\tau) \)
Definition 2.9

- substitution $\theta$ is unifier of terms $s$ and $t$ : $\iff s\theta = t\theta$
- $s$ and $t$ unifiable $\iff$ a unifier of $s$ and $t$ exists
- $\theta$ most general unifier (MGU) of $s$ and $t$ : $\iff$
  - $\theta$ unifier of $s$ and $t$ that is more general than all unifiers of $s$ and $t$

Let $s_1, ..., s_n, t_1, ..., t_n$ be terms.

Let $s_i \doteq t_i$ denote the (ordered) pair $(s_i, t_i)$ and let $E = \{s_1 \doteq t_1, ..., s_n \doteq t_n\}$.

- $\theta$ is unifier of $E$ : $\iff s_i\theta = t_i\theta$ for every $i \in [1, n]$
- $\theta$ most general unifier (MGU) of $E$ : $\iff$
  - $\theta$ unifier of $E$ that is more general than all unifiers of $E$
Unifying Sets of Pairs of Terms

- Sets $E$ and $E'$ of pairs of terms equivalent:
  $\iff$ $E$ and $E'$ have the same set of unifiers

- $\{x_1 \doteq t_1, \ldots, x_n \doteq t_n\}$ solved:
  $\iff x_i, x_j$ pairwise distinct variables ($1 \leq i \neq j \leq n$) and no $x_i$ occurs in $t_j$ ($1 \leq i, j \leq n$)

Lemma 2.15
If $E = \{x_1 \doteq t_1, \ldots, x_n \doteq t_n\}$ is solved, then $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$ is an MGU of $E$.

Proof:
(i) $x_i\theta = t_i = t_i\theta$

(ii) for every unifier $\eta$ of $E$: $x_i\eta = t_i\eta = x_i\theta\eta$ for every $i \in [1, n]$ and $x\eta = x\theta\eta$ for every $x \not\in \{x_1, \ldots, x_n\}$; thus $\eta = \theta\eta$. 
Martelli-Montanari Algorithm

Let $E$ be a set of pairs of terms. As long as possible choose nondeterministically a pair of a form below and perform the associated action.

1. $f(s_1, \ldots, s_n) \equiv f(t_1, \ldots, t_n)$ replace by $s_1 \equiv t_1, \ldots, s_n \equiv t_n$
2. $f(s_1, \ldots, s_n) \equiv g(t_1, \ldots, t_m)$ where $f \neq g$ halt with failure
3. $x \equiv x$ delete the pair
4. $t \equiv x$ where $t$ is not a variable replace by $x \equiv t$
5. $x \equiv t$ where $x \not\in \text{Var}(t)$ and $x$ occurs in some other pair on all other pairs perform substitution $\{x/t\}$
6. $x \equiv t$ where $x \in \text{Var}(t)$ and $x \neq t$ halt with failure

The algorithm terminates with success when no action can be performed.

$(2) \equiv \text{“clash”}, (6) \equiv \text{“occur check”-failure}$
Martelli-Montanari (Theorem)

Theorem 2.16

If the original set $E$ has a unifier, then the algorithm successfully terminates and produces a solved set $E'$ that is equivalent to $E$; otherwise the algorithm terminates with failure.

Lemma 2.15 implies that in case of success $E'$ determines an MGU of $E$. 
Proof Steps

1. Prove that the algorithm terminates.

2. Prove that each action replaces the set of pairs by an equivalent one.

3. Prove that if the algorithm terminates successfully, then the final set of pairs is solved.

4. Prove that if the algorithm terminates with failure, then the set of pairs at the moment of failure does not have a unifier.
Relations

- $R$ relation on a set $\mathcal{A} : \iff R \subseteq \mathcal{A} \times \mathcal{A}$
- $R$ reflexive : $\iff (a, a) \in R$ for all $a \in \mathcal{A}$
- $R$ irreflexive : $\iff (a, a) \notin R$ for all $a \in \mathcal{A}$
- $R$ antisymmetric : $\iff (a, b) \in R$ and $(b, a) \in R$ implies $a = b$
- $R$ transitive : $\iff (a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$
Well-founded Orderings

- $(\mathcal{A}, \sqsubseteq)$ (reflexive) partial ordering
  $\iff \sqsubseteq$ reflexive, antisymmetric, and transitive relation on $\mathcal{A}$

- $(\mathcal{A}, \sqsupset)$ (irreflexive) partial ordering
  $\iff \sqsupset$ irreflexive and transitive relation on $\mathcal{A}$

- irreflexive partial ordering $(\mathcal{A}, \sqsupset)$ well-founded
  $\iff$ there is no infinite descending chain
  ... $\sqsupset a_2 \sqsubset a_1 \sqsubset a_0$
  of elements $a_0, a_1, a_2, ... \in \mathcal{A}$

Examples:

- $(\mathbb{N}, \leq), (\mathbb{Z}, \leq), (\mathcal{P}\{1, 2, 3\}, \subseteq)$ are partial orderings;
- $(\mathbb{N}, <), (\mathbb{Z}, <), (\mathcal{P}\{1, 2, 3\}, \subset)$ are irreflexive partial orderings;
- $(\mathbb{N}, <), (\mathcal{P}\{1, 2, 3\}, \subset)$ are well-founded, whereas $(\mathbb{Z}, <)$ is not.
Lexicographic Ordering

The lexicographic ordering $\prec_n (n \geq 1)$ is defined inductively on the set $\mathbb{N}^n$ of $n$-tuples of natural numbers:

- $(a_1) \prec_1 (b_1) :\iff a_1 < b_1$
- $(a_1, ..., a_{n+1}) \prec_{n+1} (b_1, ..., b_{n+1})$ (for $n \geq 1$)

\[ :\iff (a_1, ..., a_n) \prec_n (b_1, ..., b_n) \]

or $(a_1, ..., a_n) = (b_1, ..., b_n)$ and $a_{n+1} < b_{n+1}$

Examples:
$(3, 12, 7) \prec_3 (4, 2, 1)$ and $(8, 4, 2) \prec_3 (8, 4, 3)$.

Theorem. $(\mathbb{N}^n, \prec_n)$ is well-founded
Step 1

The MM-algorithm terminates.

Variable $x$ solved in $E$
\[ \iff x \equiv t \in E, \text{ and this is the only occurrence of } x \text{ in } E \]

$\text{uns}(E) : \iff \text{number of variables in } E \text{ that are unsolved}$

$\text{lfun}(E) : \iff \text{number of occurrences of function symbols in the first}$

$\text{card}(E) : \iff \text{number of pairs in } E$

Each successful MM-action reduces $(\text{uns}(E), \text{lfun}(E), \text{card}(E))$ wrt. $\prec_3$. 
Proof

For every \( u, l, c \in \mathbb{N} \) the reduction is as follows:

(1) \((u, l, c) \succsim_3 (u - k, l - 1, c + n - 1)\) for some \( k \in [0, ..., n] \)
(3) \((u, l, c) \succsim_3 (u - k, l, c - 1)\) for some \( k \in \{0, 1\} \)
(4) \((u, l, c) \succsim_3 (u - k_1, l - k_2, c)\) for some \( k_1 \in \{0, 1\} \) and \( k_2 \geq 1 \)
(5) \((u, l, c) \succsim_3 (u - 1, l + k, c)\) for some \( k \geq 0 \)

Termination is now a consequence of \((\mathbb{N}^3, \prec_3)\) being well-founded.
Step 2

Each action replaces the set of pairs by an equivalent one.

This is obviously true for MM-actions (1), (3), and (4).

Regarding MM-action (5), consider $E \cup \{x=\mathfrak{t}\}$ and $\{x/t\} \cup \{x=\mathfrak{t}\}$.
Then

\[ \theta \text{ is a unifier of } E \cup \{x=\mathfrak{t}\} \]
iff $(\theta \text{ is a unifier of } E) \text{ and } x\theta = t\theta$
iff $(\theta \text{ is a unifier of } E\{x/t\}) \text{ and } x\theta = t\theta$
iff $\theta \text{ is a unifier of } E\{x/t\} \cup \{x=\mathfrak{t}\}$
Step 3

If the algorithm successfully terminates, then the final set of pairs is solved.

If the algorithm successfully terminates, then MM-actions (1), (2), and (4) do not apply, so each pair in $E$ is of the form $x \doteq t$ with $x$ being a variable.

Moreover, MM-actions (3), (5), and (6) do not apply, so the variables in the first components of all pairs in $E$ are pairwise disjoint and do not occur in the second component of a pair in $E$. 
Step 4

If the algorithm terminates with failure, then the set of pairs at the moment of failure does not have a unifier.

If the failure results by MM-action (2), then some
\[ f(s_1, \ldots, s_n) ≡ g(t_1, \ldots, t_m) \]
(where \( f \neq g \)) occurs in \( E \), and for no substitution \( \theta \) we have
\[ f(s_1, \ldots, s_n)\theta = g(t_1, \ldots, t_m)\theta. \]

If the failure results by MM-action (6), then some \( x\vdash t \) (where \( x \) is a proper subterm of \( t \)) occurs in \( E \), and for no substitution \( \theta \) we have \( x\theta = t\theta \).
Unifiers may be Exponential

\[ f(x_1) \doteq f(g(x_0, x_0)) \]
\[ \theta_1 = \{ x_1 / g(x_0, x_0) \} \]

\[ f(x_1, x_2) \doteq f(g(x_0, x_0), g(x_1, x_1)) \]
\[ \theta_2 = \theta_1 \cup \{ x_2 / g(g(x_0, x_0), g(x_0, x_0)) \} \]

\[ f(x_1, x_2, x_3) \doteq f(g(x_0, x_0), g(x_1, x_1), g(x_2, x_2)) \]
\[ \theta_3 = \theta_2 \cup \{ x_3 / g(g(g(x_0, x_0), g(x_0, x_0)), g(g(x_0, x_0), g(x_0, x_0))) \} \]
\[ \vdots \]
In most PROLOG systems the occur check does not apply, for the sake of efficiency. As for the Martelli-Montanari Algorithm this amounts to drop action (6).

Then the algorithm terminates with success, e.g., for \{x \doteq f(x)\}, despite \(x\) and \(f(x)\) not being unifiable.

Also, for the sake of efficiency, action (5) is normally not implemented in PROLOG systems.

Then the algorithm may terminate with a set that only implicitly represents an \text{MGU}, e.g., \(\{x = f(y), y = g(a)\}\).
Objectives

- Understanding the need for unification
- Defining alphabets, terms, and substitutions
- Introducing the Martelli-Montanari Algorithm for unification
- Proving correctness of the algorithm