Equational Logic

- Consider a first order language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$.
- We define the following precedence hierarchy:
  $\{\forall, \exists\} > \neg > \land > \lor > \{\leftarrow, \rightarrow\} > \leftrightarrow$.
- Let $\approx/2$ be a binary predicate symbol written infix.
- An equation is an atom of the form $s \approx t$.
- An equational system $\mathcal{E}$ is a finite set of universally closed equations.

$$\mathcal{E}_1 = \{ \begin{array}{ll}
(\forall X, Y, Z) (X \cdot Y) \cdot Z \approx X \cdot (Y \cdot Z), & \text{associativity} \\
(\forall X) 1 \cdot X \approx X, & \text{left unit} \\
(\forall X) X \cdot 1 \approx X, & \text{right unit} \\
(\forall X) X^{-1} \cdot X \approx 1, & \text{left inverse} \\
(\forall X) X \cdot X^{-1} \approx 1 & \text{right inverse}
\end{array} \}$$

where $\{\cdot/2, \neg^{-1}/1, 1/0\} \subseteq \mathcal{F}$.
Axioms of Equality

The equality relation enjoys some typical properties expressed by the following axioms of equality.

\[ \mathcal{E}_{\approx} = \]

\{ (\forall X) \ X \approx X, \]

\( (\forall X, Y) (X \approx Y \rightarrow Y \approx X), \)

\( (\forall X, Y, Z) (X \approx Y \land Y \approx Z \rightarrow X \approx Z) \}\]

reflexivity

symmetry

transitivity

\[ \cup \]

\{ \forall (\bigwedge_{i=1}^{n} X_i \approx Y_i \rightarrow f(X_1, \ldots, X_n) \approx f(Y_1, \ldots, Y_n)) \mid f/n \in \mathcal{F} \}\]

f-substitutivity

\[ \cup \]

\{ \forall (\bigwedge_{i=1}^{n} X_i \approx Y_i \land p(X_1, \ldots, X_n) \rightarrow p(Y_1, \ldots, Y_n)) \mid p/n \in \mathcal{R} \}\]

r-substitutivity
Equality and Logical Consequence

We are interested in computing logical consequences of $\mathcal{E} \cup \mathcal{E}\approx$

- $\mathcal{E}_1 \cup \mathcal{E}\approx \models (\exists X) X \cdot a \approx 1?$
- $\mathcal{E}_1 \cup \mathcal{E}\approx \cup \{(\forall X) X \cdot X \approx 1\} \models (\forall X, Y) X \cdot Y \approx Y \cdot X?$

One possibility is to apply resolution.

- There are $10^{21}$ resolution steps needed to solve the examples.
- $\mathcal{E} \cup \mathcal{E}\approx$ causes an extremely large search space.

Idea

Remove troublesome formulas from $\mathcal{E} \cup \mathcal{E}\approx$

and built them into the deductive machinery.

- Use additional rule of inference like paramodulation.
- Build the equational theory into the unification computation.
Finest Congruence Relation

- $\mathcal{E} \cup \mathcal{E}_\approx$ is a set of definite clauses.

- There exists a least model for $\mathcal{E} \cup \mathcal{E}_\approx$.
  
  Let $\mathcal{F} = \{a/0, b/0, g/2\}$ and consider $\mathcal{E}_2 = \{a \approx b\}$.
  
  The least model for $\mathcal{E}_2 \cup \mathcal{E}_\approx$ is
  
  \[
  \{t \approx t \mid t \in T(\mathcal{F})\} \\
  \cup \{a \approx b, b \approx a\} \\
  \cup \{g(a, a) \approx g(a, b), g(a, a) \approx g(b, a), g(a, a) \approx g(b, b), \ldots\}
  \]

- We define the least congruence relation $\approx_{\mathcal{E}} \subseteq T(\mathcal{F}, \mathcal{V})^2$ generated by $\mathcal{E}$ as $s \approx_{\mathcal{E}} t$ iff $\mathcal{E} \cup \mathcal{E}_\approx \models \forall s \approx t$.

  \[
  g(b, a) \approx_{\mathcal{E}_2} g(a, b) \\
  g(X, a) \approx_{\mathcal{E}_2} g(X, b)
  \]
Paramodulation

- $L[\pi]$ term occurring at position $\pi \in \mathcal{P}_L$ in literal $L$;
- $L[\pi \mapsto t]$ Literal $L$ where subterm at $\pi \in \mathcal{P}_L$ has been replaced by $t$.

- **Paramodulation**

\[
\frac{[L_1, \ldots, L_n]}{[L_1[\pi \mapsto r], L_2, \ldots, L_n, K_1, \ldots, K_m]} \quad \theta = \text{mgu}(L_1[\pi], l), \quad \pi \in \mathcal{P}_{L_1}
\]

- **Notation** instead of $\neg s \approx t$ we write $s \not\approx t$.

- **Remember** $E \cup E_\approx \models \forall s \approx t$ iff
  \[
  \bigwedge_{E \cup E_\approx} \rightarrow \forall s \approx t \text{ is valid}
  \]
  iff
  \[
  \neg \left(\bigwedge_{E \cup E_\approx} \rightarrow \forall s \approx t\right) \text{ is unsatisfiable}
  \]
  iff
  \[
  E \cup E_\approx \cup \{\neg \forall s \approx t\} \text{ is unsatisfiable}
  \]
  iff
  \[
  E \cup E_\approx \cup \{\exists s \not\approx t\} \text{ is unsatisfiable}.
  \]

- **Theorem 4.1** If $E \cup E_\approx \cup \{\exists s \not\approx t\}$ is unsatisfiable,
  then there is a refutation of $E \cup \{(\forall X) \ X \approx X, \ \exists s \not\approx t\}$
  with respect to paramodulation, resolution and factoring.
An Example

\[ \mathcal{E}_1 \cup \{(\forall X) \ X \approx X, \ (\forall X) \ X \cdot X \approx 1\} \models (\forall X, Y) \ X \cdot Y \approx Y \cdot X \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a \cdot b \not\approx b \cdot a)</td>
<td>initial query</td>
</tr>
<tr>
<td>2</td>
<td>(1 \cdot X_1 \approx X_1)</td>
<td>left unit</td>
</tr>
<tr>
<td>3</td>
<td>(X_2 \approx X_2)</td>
<td>reflexivity</td>
</tr>
<tr>
<td>4</td>
<td>(X_1 \approx 1 \cdot X_1)</td>
<td>pm(2,3)</td>
</tr>
<tr>
<td>5</td>
<td>(a \cdot b \not\approx (1 \cdot b) \cdot a)</td>
<td>pm(1,4)</td>
</tr>
<tr>
<td>6</td>
<td>(X_3 \cdot X_3 \approx 1)</td>
<td>hypothesis</td>
</tr>
<tr>
<td>7</td>
<td>(X_4 \approx X_4)</td>
<td>reflexivity</td>
</tr>
<tr>
<td>8</td>
<td>(1 \approx X_3 \cdot X_3)</td>
<td>pm(6,7)</td>
</tr>
<tr>
<td>9</td>
<td>(a \cdot b \not\approx ((X_3 \cdot X_3) \cdot b) \cdot a)</td>
<td>pm(5,8)</td>
</tr>
<tr>
<td></td>
<td>(a \cdot b \not\approx ((X_3 \cdot X_3) \cdot b) \cdot (a \cdot 1))</td>
<td>right unit</td>
</tr>
</tbody>
</table>

\(n\) \(a \cdot b \not\approx a \cdot b\) hypothesis

\(n'\) \(X_5 \approx X_5\) reflexivity

\(n''\) \([\ ]\) res \((n, n')\)
The Example in Shorthand Notation

\[ E_1 \cup \{ (\forall X) \; X \approx X, \; (\forall X) \; X \cdot X \approx 1 \} \models (\forall X, Y) \; X \cdot Y \approx Y \cdot X \]


1. \( a \cdot b \not\approx b \cdot a \)  
   initial query

2. \( 1 \cdot X_1 \approx X_1 \)  
   left unit

3. \( X_2 \approx X_2 \)  
   reflexivity

4. \( X_1 \approx 1 \cdot X_1 \)  
   pm(2,3)

5. \( a \cdot b \not\approx (1 \cdot b) \cdot a \)  
   pm(1,4)

6. \( X_3 \cdot X_3 \approx 1 \)  
   hypothesis

7. \( X_4 \approx X_4 \)  
   reflexivity

8. \( 1 \approx X_3 \cdot X_3 \)  
   pm(6,7)

9. \( a \cdot b \not\approx ((X_3 \cdot X_3) \cdot b) \cdot a \)  
   pm(5,8)

\[ a \cdot b \not\approx ((X_3 \cdot X_3) \cdot b) \cdot (a \cdot 1) \]

   right unit

\[ a \cdot b \not\approx (X_3 \cdot ((X_3 \cdot b) \cdot (a \cdot X_4))) \cdot X_4 \]

   hypothesis

\[ a \cdot b \not\approx (a \cdot 1) \cdot b \]

   hypothesis

\[ n \; a \cdot b \not\approx a \cdot b \]

   right unit

\[ n' \; X_5 \approx X_5 \]

   reflexivity

\[ n'' \; [ ] \]

   res \( (n, \; n') \)
The Example in Shorthand Notation Again

\[ b \cdot a \approx (1 \cdot b) \cdot a \]
\[ \approx ((X_3 \cdot X_3) \cdot b) \cdot a \]
\[ \approx ((X_3 \cdot X_3) \cdot b) \cdot (a \cdot 1) \]
\[ \approx ((X_3 \cdot X_3) \cdot b) \cdot (a \cdot (X_4 \cdot X_4)) \]
\[ \approx (X_3 \cdot ((X_3 \cdot b) \cdot (a \cdot X_4))) \cdot X_4 \]
\[ \approx (a \cdot 1) \cdot b \]
\[ \approx a \cdot b \]

- Left unit
- Hypothesis
- Right unit
- Hypothesis
- Hypothesis
- Associativity

▶ Now, the search space is \(10^{11}\) instead of \(10^{21}\) steps.

▷ Symmetry can be simulated, which leads to cycles.

▷ All \(\pi \in P_{L_1}\) are candidates.

▷ \(L_1[\pi]\) may be a variable and can be unified with any \(l\).

▶ There are still many redundant and useless steps.

▶ Next Idea use equations only from left to right: term rewriting system.
An expression of the form $s \rightarrow t$ is called rewrite rule.

A term rewriting system is a finite set of rewrite rules.

Notation term rewriting systems shall be denoted by $\mathcal{R}$.

$s[\pi]$ subterm of term $s$ at position $\pi \in P_s$.

$s[\pi \mapsto v]$ term $s$ where subterm at $\pi \in P_s$ has been replaced by $v$.

The rewrite relation $\rightarrow \mathcal{R} \subseteq T(\mathcal{F}, \mathcal{V})^2$ is defined as follows: $s \rightarrow \mathcal{R} t$ iff there are $l \rightarrow r \in \mathcal{R}$, $\pi \in P_s$ and $\theta$ such that $s[\pi] = l\theta$ and $t = s[\pi \mapsto r\theta]$.

Consider $\mathcal{R}_3 = \{ \text{append}([], X) \rightarrow X, \text{append}([X|Y], Z) \rightarrow [X|\text{append}(Y, Z)] \}$.

Then, $\text{append}([1, 2], [3, 4]) \rightarrow \mathcal{R}_3 [1|\text{append}([2], [3, 4])]$

$\rightarrow \mathcal{R}_3 [1, 2|\text{append}([], [3, 4])]$

$\rightarrow \mathcal{R}_3 [1, 2, 3, 4]$.

Matching problem

Given terms $u$ and $l$, does there exist a substitution $\theta$ such that $u = l\theta$?

If such a substitution exists, then it is called a matcher.
Term Rewriting and Equational Logic

- $\rightarrow^*_{\mathcal{R}}$ denotes the reflexive and transitive closure of $\rightarrow_{\mathcal{R}}$.

- $append([1, 2], [3, 4]) \rightarrow^*_3 [1, 2, 3, 4]$.

- $s \leftrightarrow_{\mathcal{R}} t \iff s \leftarrow_{\mathcal{R}} t \text{ or } s \rightarrow_{\mathcal{R}} t$.

- $\leftrightarrow^*_{\mathcal{R}}$ denotes the reflexive and transitive closure of $\leftrightarrow_{\mathcal{R}}$.

- Let $\mathcal{R}_4 = \{a \rightarrow b, c \rightarrow b\}$, then $a \leftrightarrow^*_{\mathcal{R}_4} c$.

- $\mathcal{E}_{\mathcal{R}} := \{l \approx r \mid l \rightarrow r \in \mathcal{R}\} \cup \mathcal{E}_\approx$.

- $\mathcal{E}_{\mathcal{R}_4} = \{a \approx b, c \approx b\} \cup \mathcal{E}_\approx$.

- Theorem 4.2  
  (i) $s \rightarrow^*_{\mathcal{R}} t$ implies $s \approx_{\mathcal{E}_{\mathcal{R}}} t$.
  (ii) $s \approx_{\mathcal{E}_{\mathcal{R}}} t$ iff $s \leftrightarrow^*_{\mathcal{R}} t$.

- Proof $\sim$ Exercise
  - $g(X, a) \rightarrow_{\mathcal{R}_4} g(X, b)$ and $g(X, a) \approx_{\mathcal{E}_{\mathcal{R}_4}} g(X, b)$.
  - $g(X, a) \approx_{\mathcal{E}_{\mathcal{R}_4}} g(X, c)$ and $g(X, a) \rightarrow_{\mathcal{R}_4} g(X, b) \leftarrow_{\mathcal{R}_4} g(X, c)$. 
Normal Form

► **s** is reducible wrt **R** iff there exists **t** such that \( s \rightarrow_R t \); otherwise it is irreducible.

► **t** is a normal form of **s** wrt **R** iff \( s \rightarrow^* R t \) and **t** irreducible.

▶ \([1, 2, 3, 4]\) is the normal form of \( \text{append}([1, 2], [3.4]) \) wrt \( R_3 \).

► Normal forms are not necessarily unique. Consider

\[
R_5 = \{ \begin{array}{ll}
\text{not(not}(X)) & \rightarrow X, \\
\text{not(or}(X, Y)) & \rightarrow \text{and(not}(X), \text{not}(Y)), \\
\text{not(and}(X, Y)) & \rightarrow \text{or(not}(X), \text{not}(Y)), \\
\text{and}(X, \text{or}(Y, Z)) & \rightarrow \text{or(and}(X, Y), \text{and}(X, Z)), \\
\text{and(or}(X, Y), Z) & \rightarrow \text{or(and}(Y, Z), \text{and}(Z, X)) \end{array} \}
\]

\( \text{and(or}(X, Y), \text{or}(U, V)) \) has the normal forms

\( \text{or(or}(\text{and}(Y, U), \text{and}(U, X)), \text{or}(\text{and}(Y, V), \text{and}(V, X))) \) and

\( \text{or(or}(\text{and}(Y, U), \text{and}(Y, V)), \text{or}(\text{and}(V, X), \text{and}(X, U))) \) wrt \( R_5 \).
Confluent Term Rewriting Systems

- $s \downarrow_\mathcal{R} t$ iff there exists $u$ such that $s \xrightarrow{\mathcal{R}}^* u \xleftarrow{\mathcal{R}}^* t$.
- $s \uparrow_\mathcal{R} t$ iff there exists $u$ such that $s \xleftarrow{\mathcal{R}}^* u \xrightarrow{\mathcal{R}}^* t$.
  - Consider $\mathcal{R}_6 = \{b \rightarrow a, b \rightarrow c\}$. Then $a \nless_{\mathcal{R}_6} c$, but $a \upeq_{\mathcal{R}_6} c$.

- $\mathcal{R}$ is confluent iff for all terms $s$ and $t$ we find $s \uparrow_\mathcal{R} t$ implies $s \downarrow_\mathcal{R} t$.
  - $\mathcal{R}_7 = \mathcal{R}_6 \cup \{a \rightarrow c\}$ is confluent.

- $\mathcal{R}$ is Church-Rosser iff for all terms $s$ and $t$ we find $s \leftrightarrow_\mathcal{R}^* t$ iff $s \downarrow_\mathcal{R} t$.

- Theorem 4.3 $\mathcal{R}$ is Church-Rosser iff $\mathcal{R}$ is confluent.
  - Proof $\Rightarrow$ Exercise.

- Remember $s \leftrightarrow_\mathcal{R}^* t$ iff $s \equiv_\mathcal{R} t$.
  - Rewriting has only to be applied in one direction!
Canonical Term Rewriting Systems

- $\mathcal{R}$ is terminating iff it has no infinite rewriting sequences.
  - The question whether $\mathcal{R}$ is terminating is undecidable.
- $\mathcal{R}$ is canonical iff $\mathcal{R}$ is confluent and terminating.
  - If $\mathcal{R}$ is canonical then $s \approx_{\mathcal{E}_\mathcal{R}} t$ iff $s \Downarrow_{\mathcal{R}} t$.
  - If $\mathcal{R}$ is canonical then $\mathcal{E}_\mathcal{R}$ is decidable.
- Given $\mathcal{E}$. If $\approx_{\mathcal{E}} = \approx_{\mathcal{E}_\mathcal{R}}$ for some canonical term rewriting system $\mathcal{R}$, then the application of paramodulation can be restricted:
  - $L_1[\pi]$ may not be a variable.
  - Symmetry can no longer be simulated.
  - Equations, i.e., rewrite rules, are only applied from left to right.
  - Further restrictions concerning $\pi \in \mathcal{P}_{L_1}$ are possible.
  - This restricted form of paramodulation is called narrowing.
Termination

► Is a given term rewriting system \( R \) terminating?

► Idea Find a well-founded ordering \( >/2 \) on terms such that \( s \to_R t \) implies \( s > t \).

▷ Let \( \geq /2 \) be a partial ordering on terms.

▷ \( s > t \) iff \( s \geq t \) and \( s \neq t \).

▷ \( >/2 \) is well-founded iff there is no infinite sequence \( s_1 > s_2 > \ldots \).

► A termination ordering \( >/2 \) is a well-founded, transitive and antisymmetric relation on the set of terms satisfying the following properties:

▷ full invariance property if \( s > t \) then \( s\theta > t\theta \),

▷ replacement property if \( s > t, \pi \in P_u \) and \( u[\pi] = s \) then \( u > u[\pi \mapsto t] \).

► Theorem 4.4 Let \( R \) be a term rewriting system and \( >/2 \) a termination ordering. If for all rules \( l \to r \in R \) we find that \( l > r \) then \( R \) is terminating.

▷ Proof \(~\) Exercise.
Termination Orderings: Two Examples

- Let $|s|$ denote the length of the term $s$, i.e., the length of the word $s$. $s > t$ iff for all grounding substitutions $\theta$ we find that $|s\theta| > |t\theta|$.
  - $f(X,Y) > g(X)$,
  - $f(X,Y)$ and $h(X,Y)$ can not be ordered.

- Polynomial ordering
  assign to each $g/n \in \mathcal{F}$ a polynomial with coefficients taken from $\mathbb{N}^+$.
  - Let $f^I : \mathbb{N}^+ \times \mathbb{N}^+ \rightarrow \mathbb{N}^+$, $(X,Y) \mapsto 2X + Y$  
    $h^I : \mathbb{N}^+ \times \mathbb{N}^+ \rightarrow \mathbb{N}^+$, $(X,Y) \mapsto X + Y$.
  - Now we define $s > t$ iff $s^I,\mathcal{E} > t^I,\mathcal{E}$.
  - Then, $f(X,Y) > h(X,Y)$.

- There are many other termination orderings!
- $>/2$ is more powerful than $>/2$ iff $s > t$ implies $s >' t$, but not vice versa.
Confluence

▶ Is a given terminating term rewriting system confluent?

▶ ∋ is locally confluent iff for all terms r, s and t the following holds:
If r → s and r → t then s ↓ t.

▶ Theorem 4.5  Let ∋ be a terminating term rewriting system.
� is confluent iff it is locally confluent.

▷ Proof ⇝ Exercise.
Local Confluence

- Is a given terminating term rewriting system locally confluent?
- A subterm $u$ of $t$ is called a **redex**
  - iff there exists $\theta$ and $l \rightarrow r \in \mathcal{R}$ such that $u = l\theta$.
- Let $l_1 \rightarrow r_1 \in \mathcal{R}$ and $l_2 \rightarrow r_2 \in \mathcal{R}$ be applicable to $t \rightsquigarrow$ two redexes.

**Case analysis**

(a) They are disjoint.
(b) one redex is a subterm of the other one and corresponds to a variable position in the left-hand-side of the other rule.
(c) one redex is a subterm of the other one but does not correspond to a variable position in the left-hand-side of the other rule (the redexes overlap).
Example

Consider the term $t = (g(a) \cdot f(b)) \cdot c$.

(a) $\mathcal{R}_8 = \{a \rightarrow c, \, b \rightarrow c\}$.
   - $a$ and $b$ are disjoint redexes in $t$,
   - $\mathcal{R}_8$ is locally confluent.

(b) $\mathcal{R}_9 = \{a \rightarrow c, \, g(X) \rightarrow f(X)\}$.
   - $a$ and $g(a)$ are redexes in $t$; $a$ corresponds to the variable position in $g(X)$,
   - $\mathcal{R}_9$ is locally confluent.

(c) $\mathcal{R}_{10} = \{(X \cdot Y) \cdot Z \rightarrow X, \, g(a) \cdot f(b) \rightarrow c\}$.
   - $(g(a) \cdot f(b)) \cdot c$ and $g(a) \cdot f(b)$ are overlapping redexes in $t$.
   - This is the problematic case!
Critical Pairs

- Suppose \( \{ l_1 \rightarrow r_1, \ l_2 \rightarrow r_2 \} \subseteq \mathcal{R}, \pi \in \mathcal{P}_{l_1}, l_1[\pi] \) is not a variable, and \( l_2 \) is unifiable with \( l_1[\pi] \) using mgu \( \theta \). Then the pair \( \langle (l_1[\pi \rightarrow r_2])\theta,\ r_1\theta \rangle \) is said to be critical. It is obtained by superimposing \( l_1 \) with \( l_2 \).

- Superimposing \((X \cdot Y) \cdot Z \rightarrow X\) with \(g(a) \cdot f(b) \rightarrow c\) yields the critical pair \( \langle c \cdot Z, g(a) \rangle \).

- Theorem 4.6  A term rewriting system \( \mathcal{R} \) is locally confluent iff for all critical pairs \( \langle s, t \rangle \) of \( \mathcal{R} \) we find \( s \downarrow_\mathcal{R} t \).

- Proof  ~ Exercise.
Completion

► Can a terminating and non-confluent $\mathcal{R}$ be turned into a confluent one?
► Two term rewriting systems $\mathcal{R}$ and $\mathcal{R}'$ are equivalent iff $\cong_{\mathcal{R}} = \cong_{\mathcal{R}'}$.
► Idea if $\langle s, t \rangle$ is a critical pair, then add either $s \rightarrow t$ or $t \rightarrow s$ to $\mathcal{R}$.

➢ This is called completion.
➢ The equational theory remains unchanged.
Completion Procedure

- Given a terminating $\mathcal{R}$ together with a termination ordering $\succ/2$.
  1. If for all critical pairs $\langle s, t \rangle$ of $\mathcal{R}$ we find that $s \downarrow_{\mathcal{R}} t$ then return “success”; $\mathcal{R}$ is canonical.
  2. If $\mathcal{R}$ has a critical pair whose elements do not rewrite to a common term, then transform the elements of the critical pair to some normal form. Let $\langle s, t \rangle$ be the normalized critical pair:
     - If $s \succ t$ then add the rule $s \rightarrow t$ to $\mathcal{R}$ and goto 1.
     - If $t \succ s$ then add the rule $t \rightarrow s$ to $\mathcal{R}$ and goto 1.
     - If neither $s \succ t$ nor $t \succ s$ then return “fail”.

- The completion procedure may either succeed or fail or loop.
- During completion the ordering $\succ/2$ may be extended to a more powerful one.
- The completion procedure may be extended to unfailing completion.
Completion: An Example

\[ \mathcal{R}_{11} = \{c \rightarrow b, \ f \rightarrow b, \ f \rightarrow a, \ e \rightarrow a, \ e \rightarrow d\} \]

\[ f > e > d > c > b > a. \]

- The critical pairs are \( \langle b, a \rangle \) and \( \langle d, a \rangle \).
- They can be oriented into the new rules \( b \rightarrow a \) and \( d \rightarrow a \).
- We obtain

\[ \mathcal{R}'_{11} = \{c \rightarrow b, \ f \rightarrow b, \ f \rightarrow a, \ e \rightarrow a, \ e \rightarrow d, \ b \rightarrow a, \ d \rightarrow a\}. \]

- \( \mathcal{R}'_{11} \) is canonical.
- \( s \approx_{\mathcal{R}} t \) iff \( s \approx_{\mathcal{R}'_{11}} t \).
- All proofs for \( s \approx_{\mathcal{R}'_{11}} t \) are in so-called valley form.
Unification Theory

- **Idea** We want to build equational axioms into the unification computation.

- **An \( \mathcal{E} \)-unification problem** consists of an equational theory \( \mathcal{E} \) and two terms \( s \) and \( t \), and is the question whether \( \mathcal{E} \cup \mathcal{E} \equiv \models \exists s \equiv t \) holds.
  
  A substitution \( \theta \) is a **solution** of the \( \mathcal{E} \)-unification problem iff \( s\theta \equiv \mathcal{E} t\theta \).
  
  In this case \( \theta \) is called **\( \mathcal{E} \)-unifier** for \( s \) and \( t \).

- If \( \mathcal{E} = \emptyset \), then \( \mathcal{E} \)-unification reduces to unification.

- Consider \( \mathcal{E}_{12} = \{ \forall f(X) \equiv X \} \).
  
  - \( \{ Y \mapsto a \} \) is an \( \mathcal{E}_{12} \)-unifier for \( g(f(a), a) \) and \( g(Y, Y) \).
  
  - The unification problem \( \{ g(f(a), a) \equiv g(Y, Y) \} \) is unsolvable.

- Substitutions \( \eta \) and \( \theta \) are **\( \mathcal{E} \)-equal** on a set \( \mathcal{V} \) of variables \( (\theta \equiv_{\mathcal{E}} \eta[\mathcal{V}]) \) iff \( X\eta \equiv_{\mathcal{E}} X\theta \) for all \( X \in \mathcal{V} \).

- \( \{ Y \mapsto a \} \) and \( \{ Y \mapsto f(a) \} \) are \( \mathcal{E}_{12} \)-equal on \( \{ X, Y \} \).
\[\mathcal{E}\text{-Instances}\]

- Substitution \(\eta\) is an \(\mathcal{E}\)-instance of \(\theta\) on a set \(\mathcal{W}\) of variables \((\theta \geq_{\mathcal{E}} \eta[\mathcal{W}]\)) iff there exists a substitution \(\tau\) such that \(X \eta \approx_{\varepsilon} X \theta \tau\) for all \(X \in \mathcal{W}\).

- \(\eta\) is a strict \(\mathcal{E}\)-instance of \(\theta\) \((\theta >_{\mathcal{E}} \eta[\mathcal{W}]\)) iff \(\theta \geq_{\mathcal{E}} \eta[\mathcal{W}]\) and not \(\theta \approx_{\varepsilon} \eta[\mathcal{W}]\).

- If neither \(\theta \geq_{\mathcal{E}} \eta[\mathcal{W}]\) nor \(\eta \geq_{\mathcal{E}} \theta[\mathcal{W}]\), then \(\theta\) and \(\eta\) are said to be incomparable on \(\mathcal{W}\).
Consider \( \mathcal{E} \cup \mathcal{E} \approx \models (\exists X, Y) \, f(X, g(a, b)) \approx f(g(Y, b), X) \)

- \( \mathcal{E} = \emptyset \)
  - Decision problem is decidable.
  - Most general unifier is unique modulo variable renaming:
    \( \theta_1 = \{ X \mapsto g(a, b), \ Y \mapsto a \} \).

- \( \mathcal{E} = \{ \forall \, f(X, Y) \approx f(Y, X) \} \)
  - \( \theta_1 \) is a solution.
  - So is \( \theta_2 = \{ Y \mapsto a \} \):
    \[
    f(X, g(a, b))\theta_2 = f(X, g(a, b)) \approx_{\mathcal{E}} f(g(a, b), X) = f(g(Y, b), X)\theta_2.
    \]
  - \( \theta_2 \geq_{\mathcal{E}} \theta_1[\{ X, Y \}] \).
  - There are at most finitely many most general unifiers.
Reconsider $\mathcal{E} \cup \mathcal{E} \approx \models (\exists X, Y) f(X, g(a, b)) \approx f(g(Y, b), X)$

$\mathcal{E} = \{\forall f(X, f(Y, Z)) \approx f(f(X, Y), Z)\}$

$\theta_1 = \{X \mapsto g(a, b), Y \mapsto a\}$ is a solution.

So is $\theta_3 = \{X \mapsto f(g(a, b), g(a, b)), Y \mapsto a\}$:

$$f(X, g(a, b))\theta_3 = f(f(g(a, b), g(a, b)), g(a, b))$$
$$\approx_{\mathcal{E}} f(g(a, b), f(g(a, b), g(a, b)))$$
$$= f(g(Y, b), X)\theta_3.$$

$\theta_1$ and $\theta_3$ are incomparable on $\{X, Y\}$.

$\theta_4 = \{X \mapsto f(g(a, b), f(g(a, b), g(a, b))), Y \mapsto a\}$

is yet another solution incomparable to $\theta_1$ and $\theta_3$ on $\{X, Y\}$.

In general, there may be infinitely many most general unifiers.

$\mathcal{E} = \{\forall f(X, f(Y, Z)) \approx f(f(X, Y), Z)), \forall f(X, Y) \approx f(Y, X) \}$

There are at most finitely many most general unifiers.
Sets of $\mathcal{E}$-Unifiers

- Given an $\mathcal{E}$-unification problem $\mathcal{E} \cup \mathcal{E} \equiv \exists s \approx t$.
- $\mathcal{U}_\mathcal{E}(s, t)$ denotes the set of all $\mathcal{E}$-unifiers of $s$ and $t$.
- Complete set $c\mathcal{U}_\mathcal{E}(s, t)$ of $\mathcal{E}$-unifiers of $s$ and $t$:
  - $c\mathcal{U}_\mathcal{E}(s, t) \subseteq \mathcal{U}_\mathcal{E}(s, t)$,
  - for all $\eta \in \mathcal{U}_\mathcal{E}(s, t)$ there exists $\theta \in c\mathcal{U}_\mathcal{E}(s, t)$ such that $\theta \geq_\mathcal{E} \eta[\mathcal{W}]$, where $\mathcal{W} = \text{var}(s) \cup \text{var}(t)$.
- Minimal complete set $\mu\mathcal{U}_\mathcal{E}(s, t)$ of $\mathcal{E}$-unifiers for $s$ and $t$:
  - complete set,
  - for all $\theta, \eta \in \mu\mathcal{U}_\mathcal{E}(s, t)$ we find $\theta \geq_\mathcal{E} \eta[\mathcal{W}]$ implies $\theta = \eta$.
- If $c\mathcal{U}_\mathcal{E}(s, t)$ is finite and $\geq_\mathcal{E}$ is decidable then there exists $\mu\mathcal{U}_\mathcal{E}(s, t)$.
- Let $\theta \equiv_\mathcal{E} \eta[\mathcal{W}]$ iff $\theta \geq_\mathcal{E} \eta[\mathcal{W}]$ and $\eta \geq_\mathcal{E} \theta[\mathcal{W}]$.
- $\mu\mathcal{U}_\mathcal{E}(s, t)$ is unique up to $\equiv_\mathcal{E} [\mathcal{W}]$, if it exists.
Another Example

- Let $\mathcal{F} = \{a/0, f/2\}$.
- Let $\mathcal{E}_{13} = \{\forall f(X, f(Y, Z)) \approx f(f(X, Y), Z)\}$.
- Consider $\mathcal{E}_{13} \cup \mathcal{E}_{\approx} \models \exists f(X, a) \approx f(a, Y)$.
- $\theta = \{X \mapsto a, Y \mapsto a\}$ is a solution.
- $\eta = \{X \mapsto f(a, Z), Y \mapsto f(Z, a)\}$ is another solution.
- $\{\theta, \eta\}$ is a complete set of $\mathcal{E}_{13}$-unifiers $\leadsto$ Exercise.
- $\theta$ and $\eta$ are incomparable under $\geq \mathcal{E}_{13}$.
- The set $\{\theta, \eta\}$ is minimal.
On the Existence of Minimal Complete Sets of $\mathcal{E}$-Unifiers

▶ **Theorem 4.7** Minimal complete sets of $\mathcal{E}$-unifiers do not always exist.

▶ **Proof** Let $\mathcal{R} = \{ f(0, X) \rightarrow X, g(f(X, Y)) \rightarrow g(Y) \}$.

▶ **Claim** There does not exist $\mu \mathcal{U}_{\mathcal{E}_\mathcal{R}}(g(X), g(0))$.

▷ $\mathcal{R}$ is canonical $\iff$ Exercise.

▷ Define $\sigma_0 = \{ X \mapsto 0 \}$,
\[
\sigma_1 = \{ X \mapsto f(X_1, 0) \} = \{ X \mapsto f(X_1, X\sigma_0) \},
\]
▷ $\vdots$
\[
\sigma_i = \{ X \mapsto f(X_i, X\sigma_{i-1}) \}.
\]

▷ Let $S = \{ \sigma_i \mid i \geq 0 \}$ and $\mathcal{W} = \{ X \}$

▷ $S$ is a $c\mathcal{U}_{\mathcal{E}_\mathcal{R}}(g(X), g(0))$ $\iff$ Exercise.

▷ With $\rho_i = \{ X_i \mapsto 0 \}$ we find $X\sigma_i\rho_i = f(0, X\sigma_{i-1}) \approx_{\mathcal{E}_\mathcal{R}} X\sigma_{i-1}$.

▷ Hence, $\sigma_i \geq_{\mathcal{E}_\mathcal{R}} \sigma_{i-1}[\mathcal{W}]$.

▷ Because $X\sigma_i = f(X_i, X\sigma_{i-1}) \not\approx_{\mathcal{E}_\mathcal{R}} X\sigma_{i-1}$ we find $\sigma_i \not\approx_{\mathcal{E}_\mathcal{R}} \sigma_{i-1}$.

▷ Thus, $\sigma_i >_{\mathcal{E}_\mathcal{R}} \sigma_{i-1}[\mathcal{W}]$. 
Proof of Theorem 4.7 Continued

- **Remember** \( \mathcal{R} = \{ f(0, X) \rightarrow X, g(f(X, Y)) \rightarrow g(Y) \} \).

- Assume \( S' \) is a \( \mu \mathcal{U}_{\mathcal{E}}(g(X), g(0)) \).

- Because \( S \) is complete we find that for all \( \rho \in S' \) there exists \( \sigma_i \in S \) such that \( \sigma_i \geq_{\mathcal{E}} \rho[\mathcal{W}] \).

- Because \( \sigma_{i+1} >_{\mathcal{E}} \sigma_i[\mathcal{W}] \) we obtain \( \sigma_{i+1} >_{\mathcal{E}} \rho[\mathcal{W}] \).

- Because \( S' \) is complete we find that there exists \( \sigma \in S' \) such that \( \sigma \geq_{\mathcal{E}} \sigma_{i+1}[\mathcal{W}] \).

- Hence, \( \sigma >_{\mathcal{E}} \rho[\mathcal{W}] \).

- Thus, \( S' \) is not minimal \( \Rightarrow \) Contradiction.
Unification Types

- The unification type of $\mathcal{E}$ is
  - unitary iff a set $\mu_{\mathcal{E}}(s, t)$ exists for all $s, t$ and has cardinality 0 or 1.
  - finitary iff a set $\mu_{\mathcal{E}}(s, t)$ exists for all $s, t$ and is finite.
  - infinitary iff a set $\mu_{\mathcal{E}}(s, t)$ exists for all $s, t,$ and there are $u$ and $v$ such that $\mu_{\mathcal{E}}(u, v)$ is infinite.
  - zero iff there are $s, t$ such that $\mu_{\mathcal{E}}(s, t)$ does not exist.
Unification procedures

► $\mathcal{E}$-unification procedure:

▶ input: $s \approx t$.
▶ output: subset of $\mathcal{U}_\mathcal{E}(s, t)$.
▶ is complete iff for all $s, t$ the output is a $c\mathcal{U}_\mathcal{E}(s, t)$.
▶ is minimal iff for all $s, t$ the output is a $\mu\mathcal{U}_\mathcal{E}(s, t)$.

► Universal $\mathcal{E}$-unification procedure:

▶ input: $\mathcal{E}$ and $s \approx t$.
▶ output: subset of $\mathcal{U}_\mathcal{E}(s, t)$.
▶ is complete iff for all $\mathcal{E}$ and $s, t$ the output is a $c\mathcal{U}_\mathcal{E}(s, t)$.
▶ is minimal iff for all $\mathcal{E}$ and $s, t$ the output is a $\mu\mathcal{U}_\mathcal{E}(s, t)$.
Typical Questions related to $\mathcal{E}$

- Is it decidable whether an $\mathcal{E}$-unification problem is solvable?
- What is the unification type of $\mathcal{E}$?
- How can we obtain an efficient $\mathcal{E}$-unification algorithm or, preferably, a minimal $\mathcal{E}$-unification procedure?
Classes of $\mathcal{E}$-Unification Problems

- The class of an $\mathcal{E}$-unification problem $\mathcal{E} \cup \mathcal{E} \approx \models \exists s \approx t$ is called
  - elementary iff $s$ and $t$ contain only symbols occurring in $\mathcal{E}$.
  - with constants iff $s$ and $t$ may contain additional so-called free constants.
  - general iff $s$ and $t$ may contain additional function symbols of arbitrary arity.
## Unification with Constants: Some Examples

<table>
<thead>
<tr>
<th>Equational System</th>
<th>Unification Type</th>
<th>Unification decidable?</th>
<th>Complexity of the decision problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{E}_A$</td>
<td>infinitary</td>
<td>yes</td>
<td>NP-hard</td>
</tr>
<tr>
<td>$\mathcal{E}_C$</td>
<td>finitary</td>
<td>yes</td>
<td>NP-complete</td>
</tr>
<tr>
<td>$\mathcal{E}_{AC}$</td>
<td>finitary</td>
<td>yes</td>
<td>NP-complete</td>
</tr>
<tr>
<td>$\mathcal{E}_{AG}$</td>
<td>unitary</td>
<td>yes</td>
<td>NP-complete polynomial</td>
</tr>
<tr>
<td>$\mathcal{E}_{AI}$</td>
<td>zero</td>
<td>yes</td>
<td>NP-hard</td>
</tr>
<tr>
<td>$\mathcal{E}_{CR1}$</td>
<td>zero</td>
<td>no</td>
<td>–</td>
</tr>
<tr>
<td>$\mathcal{E}<em>{DL}$, $\mathcal{E}</em>{DR}$</td>
<td>unitary</td>
<td>yes</td>
<td>polynomial</td>
</tr>
<tr>
<td>$\mathcal{E}_D$</td>
<td>infinitary</td>
<td>?</td>
<td>NP-hard</td>
</tr>
<tr>
<td>$\mathcal{E}_{DA}$</td>
<td>infinitary</td>
<td>no</td>
<td>–</td>
</tr>
<tr>
<td>$\mathcal{E}_{BR}$</td>
<td>unitary</td>
<td>yes</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>
Additional Remarks

- **$\mathcal{E}$-matching problem**
  \[
  \mathcal{E} \cup \mathcal{E} \models \exists \theta s \approx_{\mathcal{E}} t\theta.
  \]

- **Combination problem**
  Can the results and unification algorithms for $\mathcal{E}_1$ and $\mathcal{E}_2$ be combined to $\mathcal{E}_1 \cup \mathcal{E}_2$?

- **Universal $\mathcal{E}$-unification problem:**
  $\mathcal{E}$-unification problem, where the equational system is part of the input.
Canonical Term Rewriting Systems Revisited

Let $R$ be a canonical term rewriting system.

So far, we were able to answer questions of the form $E_R \models \forall s \approx t$.

- **Rewriting:** $s \rightarrow_R t$ iff there are $l \rightarrow r \in R$, $\pi \in P_s$ and $\theta$ such that $s[\pi] = l\theta$ and $t = s[\pi \mapsto r\theta]$.

- **Now consider** $E_R \models \exists s \approx t$.

- **Narrowing:** $s \Rightarrow_R t$ iff there are $l \rightarrow r \in R$, $\pi \in P_s$ and $\theta$ such that $s[\pi] \not\in V$, $s[\pi]\theta = l\theta$ and $t = (s[\pi \mapsto r])\theta$.

- Please compare narrowing to rewriting and paramodulation!

**Theorem 4.8**

Let $R$ be a canonical term rewriting system with $\text{var}(l) \supseteq \text{var}(r)$ for all $l \rightarrow r \in R$. Then narrowing and resolution is sound and complete.

A complete universal $E$-unification procedure for canonical theories $E$ can be built upon narrowing and resolution.
Applications

- databases
- information retrieval
- computer vision
- natural language processing
- knowledge based systems
- text manipulation systems
- planning and scheduling systems
- pattern directed programming languages
- logic programming systems
- computer algebra systems
- deduction systems
- non-classical reasoning systems
Multisets

\[ \{e_1, e_2, \ldots \}, \emptyset. \]

- \( X \in_k \mathcal{M} \) iff \( X \) occurs precisely \( k \) times in \( \mathcal{M} \).
- \( \mathcal{M}_1 \triangleq \mathcal{M}_2 \) iff for all \( X \) we find \( X \in_k \mathcal{M}_1 \) iff \( X \in_k \mathcal{M}_2 \).
- \( X \in_m \mathcal{M}_1 \cup \mathcal{M}_2 \) iff there exist \( k, l \geq 0 \) such that \( X \in_k \mathcal{M}_1, X \in_l \mathcal{M}_2 \) and \( k + l = m \).
- \( X \in_m \mathcal{M}_1 \setminus \mathcal{M}_2 \) iff there exist \( k, l \geq 0 \) such that either \( X \in_k \mathcal{M}_1, X \in_l \mathcal{M}_2, k > l \) and \( m = k - l \) or \( X \in_k \mathcal{M}_1, X \in_l \mathcal{M}_2, k \leq l \) and \( m = 0 \).
- \( X \in_m \mathcal{M}_1 \cap \mathcal{M}_2 \) iff there exist \( k, l \geq 0 \) such that \( X \in_k \mathcal{M}_1, X \in_l \mathcal{M}_2 \) and \( m = \min\{k, l\} \).
- \( \mathcal{M}_1 \subseteq \mathcal{M}_2 \) iff \( \mathcal{M}_1 \cap \mathcal{M}_2 = \mathcal{M}_1 \).
Fluent Terms

- Consider an alphabet with variables $\mathcal{V}$ and function symbols $\mathcal{F} \supseteq \{ \circ/2, 1/0 \}$.
- Now consider $\mathcal{I}(\mathcal{F} \setminus \{ \circ/2, 1/0 \}, \mathcal{V})$.
- The nonvariable elements of $\mathcal{I}(\mathcal{F} \setminus \{ \circ/2, 1/0 \}, \mathcal{V})$ are called fluents.
- The set of fluent terms can now be defined as the smallest set satisfying the following conditions:
  - Each fluent is a fluent term.
  - $1$ is a fluent term.
  - If $s$ and $t$ are fluent terms then $s \circ t$ is a fluent term as well.
- Let $\mathcal{E}_{AC1} = \{ \forall X \circ (Y \circ Z) \approx (X \circ Y) \circ Z \\
                           \forall X \circ Y \approx Y \circ X \\
                           \forall X \circ 1 \approx X \}$. 
Multisets vs. Fluent Terms

\[ t^I = \begin{cases} \\emptyset & \text{if } t = 1 \\ \{t\} & \text{if } t \text{ is a fluent} \\ u^I \cup v^I & \text{if } t = u \circ v \end{cases} \]

\[ M^{-I} = \begin{cases} 1 & \text{if } M \models \emptyset \\ s \circ N^{-I} & \text{if } M \models \{s\} \cup N \end{cases} \]
Matching and Unification Problems

➤ Submultiset matching problem
Does there exist a \( \theta \) such that \( \mathcal{M}\theta \subseteq \mathcal{N} \), where \( \mathcal{N} \) is ground?

➤ Submultiset unification problem
Does there exist a \( \theta \) such that \( \mathcal{M}\theta \subseteq \mathcal{N}\theta \)?

➤ Fluent matching problem
Does there exist a \( \theta \) such that \( (s \circ X)\theta \approx_{AC_1} t \),
where \( t \) is ground and \( X \) does not occur in \( s \)?

➤ Fluent unification problem
Does there exist a \( \theta \) such that \( (s \circ X)\theta \approx_{AC_1} t\theta \),
where \( X \) does not occur in \( s \) or \( t \)?
Submultiset vs. Fluent Unification Problems

- Equivalence of matching problems

\[(s \circ X)\theta \cong_{AC_1} t \; \text{iff} \; (s\theta)^I \subseteq t^I \; \text{and} \; (X\theta)^I \models t^I \setminus (s\theta)^I\]

- Equivalence of unification problems

\[(s \circ X)\theta \cong_{AC_1} t\theta \; \text{iff} \; (s\theta)^I \subseteq (t\theta)^I \; \text{and} \; (X\theta)^I \models (t\theta)^I \setminus (s\theta)^I\]

- Theorem 4.9 Fluent matching and fluent unification problems are
  - decidable,
  - finitary and
  - there always exists a minimal complete set of matchers and unifiers.
Fluent Matching Algorithm

Input A fluent matching problem $\exists \theta \ (s \circ X)\theta \approx_{AC1} t$?
(where $t$ is ground and $X$ does not occur in $s$).

Output A solution $\theta$ of the fluent matching problem, if it is solvable; failure, otherwise.

1. $\theta = \varepsilon$;
2. if $s \approx_{AC1} 1$ then return $\theta\{X \mapsto t\}$;
3. don’t-care non-deterministically select a fluent $f$ from $s$ and remove $f$ from $s$;
4. don’t-know non-deterministically select a fluent $g$ from $t$ such that there exists a substitution $\eta$ with $f\eta = g$;
5. if such a fluent exists then apply $\eta$ to $s$, delete $g$ from $t$ and let $\theta := \theta\eta$, otherwise stop with failure;
6. goto 2.
States, Actions and Causality

- Rational Agents, Agent Programming Languages, Cognitive Robotics.
- **Situation Calculus** (John McCarthy 1963)
- **Core Idea** A state is a snapshot of the world and can be changed by actions only.
- **Problem** Each state and each action is only partially known!
General Problems

- Frame problem
  Which fluents are unaffected by the execution of an action?

- Ramification problem
  Which fluents are really present after the execution of an action?

- Qualification problem
  Which preconditions have to be satisfied such that an action is executable?

- Prediction problem
  How long are fluents present in certain situations?

- All problems have a cognitive as well as a technical aspect.
Requirements

- (McCarthy 1963)
- General properties of causality and facts about the possibility and results of actions are given as formulas.
- It is a logical consequence of the facts of a state and the general axioms that goals can be achieved by performing certain actions.
- The formal descriptions of states should correspond as closely as possible to what people may reasonably be presumed to know about them when deciding what to do.
Conjunctive Planning Problems

- Initial state $\mathcal{I} : \{i_1, \ldots, i_m\}$ of ground fluents.
- Goal state $\mathcal{G} : \{g_1, \ldots, g_n\}$ of ground fluents.
- Finite set $\mathcal{A}$ of actions of the form
  \[
  \{c_1, \ldots, c_l\} \Rightarrow \{e_1, \ldots, e_k\},
  \]
  where $\{c_1, \ldots, c_l\}$ and $\{e_1, \ldots, e_k\}$ are multisets of fluents called conditions and effects respectively.

- Assumption
  Each variable occurring in the effects of an action occurs also in its conditions.

- A conjunctive planning problem is the question of whether there exists a sequence of actions such that its execution transforms the initial state into the goal state.
Actions and Plans

- $C \Rightarrow E$ is applicable in $S$ iff there exists $\theta$ such that $C\theta \subseteq S$.
- The application of $C \Rightarrow E$ in $S$ leads to $S' = (S \setminus C\theta) \cup E\theta$.
  - One should observe that if $S$ is ground then $S'$ is ground as well.

- A plan is a list of actions.
- A goal $G$ is satisfied iff there exists a plan $p$ which transforms $\mathcal{I}$ into $S$ and $G \subseteq S$.
- Such a plan is called solution for the planning problem.
Blocks World

- The **pickup** action:

  \[
  \text{pickup}(V) : \{ \text{clear}(V), \text{ontable}(V), \text{empty} \} \Rightarrow \{ \text{holding}(V) \} 
  \]

- The **unstack** action:

  \[
  \text{unstack}(V, W) : \{ \text{clear}(V), \text{on}(V, W), \text{empty} \} \Rightarrow \{ \text{holding}(V), \text{clear}(W) \} 
  \]

- The **putdown** action:

  \[
  \text{putdown}(V) : \{ \text{holding}(V) \} \Rightarrow \{ \text{clear}(V), \text{ontable}(V), \text{empty} \} 
  \]

- The **stack** action:

  \[
  \text{stack}(V, W) : \{ \text{holding}(V), \text{clear}(W) \} \Rightarrow \{ \text{on}(V, W), \text{clear}(V), \text{empty} \} 
  \]
Sussman’s Anomaly

\[ \mathcal{I} = \{ \text{ontable}(a), \text{ontable}(b), \text{on}(c, a), \text{clear}(b), \text{clear}(c), \text{empty}\} \]

\[ \mathcal{G} = \{ \text{ontable}(c), \text{on}(b, c), \text{on}(a, b), \text{clear}(a), \text{empty}\} \]

Solution
\[ p = [\text{unstack}(c, a), \text{putdown}(c), \text{pickup}(b), \text{stack}(b, c), \text{pickup}(a), \text{stack}(a, b)]. \]
Sussman’s Anomaly – Solution

\[
\text{unstack}(c, a) \quad \text{putdown}(c) \quad \text{pickup}(b) \quad \text{stack}(b, c)
\]
A Fluent Calculus Implementation

An action $C \Rightarrow E$ is represented by $\text{action}(C^{-I}, \text{name}, E^{-I})$:

- $\text{action}(\text{clear}(V) \circ \text{ontable}(V) \circ \text{empty}, \text{pickup}(V), \text{holding}(V))$
- $\text{action}(\text{clear}(V) \circ \text{on}(V, W) \circ \text{empty}, \text{unstack}(V, W), \text{holding}(V) \circ \text{clear}(W))$
- $\text{action}(\text{holding}(V), \text{putdown}(V), \text{clear}(V) \circ \text{ontable}(V) \circ \text{empty})$
- $\text{action}(\text{holding}(V) \circ \text{clear}(W), \text{stack}(V, W), \text{on}(V, W) \circ \text{clear}(V) \circ \text{empty})$

Let $\mathcal{F}_A$ be the set of these facts.

Causality is expressed by $\text{causes}(s, p, s')$:

- $\text{causes}(X, [], Y) \leftarrow X \approx Y \circ Z$
- $\text{causes}(X, [V|W], Y) \leftarrow \text{action}(P, V, Q)$
  $\land P \circ Z \approx X$
  $\land \text{causes}(Z \circ Q, W, Y)$

$X \approx X$

Let $\mathcal{F}_C$ be the set of these clauses.

The planning problem is the problem whether

$\mathcal{F}_A \cup \mathcal{F}_C \cup \mathcal{E}_{AC1} \cup \mathcal{E} \models (\exists P) \text{causes}(\mathcal{I}^{-I}, P, \mathcal{G}^{-I})$ holds.
SLDE-Resolution

Let

- $\mathcal{F}$ be a set of definite clauses not containing $\approx / 2$ in their head,
- $\mathcal{E}$ be an equational system and
- $G$ a goal clause.

Question Does $\mathcal{F} \cup \{X \approx X\} \cup \mathcal{E} \cup \mathcal{E} \approx \models \forall G$ hold?

Let $C$ be a new variant $H \leftarrow A_1 \land \ldots \land A_m$ of a clause in $\mathcal{F}$, $G$ the goal clause $\leftarrow B_1 \land \ldots \land B_n$ and $\text{up}_\mathcal{E}$ an $\mathcal{E}$-unification procedure. If $H$ and $B_i$, $i \in [1, n]$, are $\mathcal{E}$-unifiable with $\theta \in \text{up}_\mathcal{E}(H, B_i)$ then

$$\leftarrow (B_1 \land \ldots \land B_{i-1} \land A_1 \land \ldots \land A_m \land B_{i+1} \land \ldots \land B_n) \theta$$

is called SLDE-resolvent of $C$ and $G$.

Theorem 4.10

- SLDE-resolution is sound if $\text{up}_\mathcal{E}$ is sound.
- SLDE-resolution is complete if $\text{up}_\mathcal{E}$ is complete.
- The selection of the literal $B_i$ is don’t care non–deterministic.
A Solution to Sussman’s Anomaly (1)

(1) \[ \leftarrow \text{causes}(\text{ontable}(a) \circ \text{ontable}(b) \circ \text{on}(c, a) \circ \text{clear}(b) \circ \text{clear}(c) \circ \text{empty}, \right. \]
\[ \text{W}, \]
\[ \left. \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \]

(2) \[ \leftarrow \text{action}(P_1, V_1, Q_1) \land \]
\[ P_1 \circ Z_1 \approx \text{ontable}(a) \circ \text{ontable}(b) \circ \text{on}(c, a) \circ \text{clear}(b) \circ \text{clear}(c) \circ \text{empty} \land \]
\[ \text{causes}(Z_1 \circ Q_1, W_1, \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \]

(3) \[ \leftarrow \text{clear}(V_2) \circ \text{on}(V_2, W_2) \circ \text{empty} \circ Z_1 \approx \]
\[ \text{ontable}(a) \circ \text{ontable}(b) \circ \text{on}(c, a) \circ \text{clear}(b) \circ \text{clear}(c) \circ \text{empty} \land \]
\[ \text{causes}(Z_1 \circ \text{holding}(V_2) \circ \text{clear}(W_2), \right. \]
\[ \text{W}_1, \]
\[ \left. \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \]

(4) \[ \leftarrow \text{causes}(\text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b) \circ \text{clear}(a) \circ \text{holding}(c), \right. \]
\[ \text{W}_1, \]
\[ \left. \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \text{empty}). \]
A Solution to Sussman’s Anomaly (2)

(7) $\leftarrow$ causes($\text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b)$)$

$\text{clear}(a) \circ \text{clear}(c) \circ \text{ontable}(c) \circ \text{empty}$, $W_4$

$\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}$.

\[ \vdots \]

(10) $\leftarrow$ causes($\text{ontable}(a) \circ \text{clear}(c) \circ \text{ontable}(c) \circ \text{clear}(a) \circ \text{holding}(b)$, $W_7$

$\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}$).

\[ \vdots \]

(13) $\leftarrow$ causes($\text{ontable}(a) \circ \text{ontable}(c) \circ \text{clear}(a) \circ \text{on}(b, c) \circ \text{clear}(b) \circ \text{empty}$, $W_{10}$

$\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}$).

\[ \vdots \]

(16) $\leftarrow$ causes($\text{ontable}(c) \circ \text{on}(b, c) \circ \text{clear}(b) \circ \text{holding}(a)$, $W_{13}$

$\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}$).

\[ \vdots \]

(19) $\leftarrow$ causes($\text{ontable}(c) \circ \text{on}(b, c) \circ \text{clear}(a) \circ \text{on}(a, b) \circ \text{empty}$, $W_{16}$

$\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}$).

(20) []
Solving the Frame Problem

► In the fluent calculus the frame problem is mapped onto fluent matching and fluent unification problems.

► For example, let

\[ s = \text{ontable}(a) \circ \text{ontable}(b) \circ \text{on}(c, a) \circ \text{clear}(b) \circ \text{clear}(c) \circ \text{empty} \]

\[ t = \text{clear}(c) \circ \text{on}(c, a) \circ \text{empty}, \]

then

\[ \theta = \{ Z \mapsto \text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b) \} \]

is a most general \( \mathcal{E} \)-matcher for the \( \mathcal{E} \)-matching problem

\[ \mathcal{E}_{AC1} \cup \mathcal{E}_{\approx} \models (\exists Z) s \approx t \circ Z. \]

► Consequently, \( \text{unstack}(c, a) \) can be applied to \( s \) yielding

\[ s' = \text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b) \circ \text{clear}(a) \circ \text{holding}(c). \]
Why are Situations not Modelled by Sets?

- Let $\mathcal{E}_{ACI_1} = \mathcal{E}_{AC_1} \cup \{ (\forall X) \ X \circ X \approx X \}$.
- In this case the $\mathcal{E}$-matching problem
  \[ \mathcal{E}_{ACI_1} \cup \mathcal{E}_\approx \models (\exists Z) \ s \approx t \circ Z \]
  has an additional solution, viz.
  \[ \eta = \{ Z_1 \mapsto ontable(a) \circ ontable(b) \circ clear(b) \circ empty \} \].
- $\theta$ and $\eta$ are independent wrt $\mathcal{E}_{ACI_1}$.
- Computing the successor state in this case yields
  \[ s'' = ontable(a) \circ ontable(b) \circ clear(b) \circ clear(a) \circ holding(c) \circ empty \]
  which is not intended because the arm of the robot cannot be empty and holding an object at the same time.
Remarks

- Some people even believed that the frame problem cannot be solved within first order logic.
- Forward vs. backward planning.
- Incomplete specifications of initial situation, e.g.

\[
(\exists X, P, Y) \quad causes(ontable(b) \circ Y, \\
\quad P, \\
\quad ontable(c) \circ on(b, c) \circ on(a, b) \circ clear(a) \circ empty \circ X).
\]

- Indeterminate effects,
- Consistency constraints,
- etc;
- see lecture on Foundations of Agent Programming.
Sorts

- \((\forall X, Y) (\text{number}(X) \land \text{number}(Y) \rightarrow \text{plus}(X, Y) \approx \text{plus}(Y, X))\)

\[ (\forall X, Y : \text{number}) \text{plus}(X, Y) \approx \text{plus}(Y, X). \]

- A first order language with sorts consists of
  - a first order language \(\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})\) and
  - a function \(\text{sort} : \mathcal{V} \rightarrow 2^{\mathcal{R}_S}\),

where \(\mathcal{R}_S \subseteq \mathcal{R}\) is a finite set of unary predicate symbols called base sorts.

- Elements of \(2^{\mathcal{R}_S}\) are called sorts; \(\emptyset \in 2^{\mathcal{R}_S}\) is called top sort.

- We write \(X : s\) if \(\text{sort}(X) = s\).

- We assume that for every sort \(s\) there are countably many variables \(X : s \in \mathcal{V}\).
Sorts – Semantics

- Let $I$ be an interpretation with domain $D$,

$$I : s = \{p_1, \ldots, p_n\} \mapsto s^I = D \cap p_1^I \cap \ldots \cap p_n^I.$$ 

- $I : \emptyset \mapsto D$.

- A variable assignment $\mathcal{Z}$ is sorted iff for all $X : s \in \mathcal{V}$ we find $X^\mathcal{Z} \in s^I$.

- We assume that all sorts are non-empty.

- $F^{I,\mathcal{Z}}$ is defined as usual except for

$$[\exists X : s) F]^{I,\mathcal{Z}} = \top \text{ iff there exists } d \in s^I \text{ such that } F^{I,\{X\mapsto d\}\mathcal{Z}} = \top.$$ 

$$[\forall X : s) F]^{I,\mathcal{Z}} = \top \text{ iff for all } d \in s^I \text{ we find } F^{I,\{X\mapsto d\}\mathcal{Z}} = \top.$$
Relativization

Sorted formulas can be mapped onto unsorted ones by means of a relativization function \( r \):

\[
\begin{align*}
  r(p(t_1, \ldots, t_n)) &= p(t_1, \ldots, t_n) \\
  r(\neg F) &= \neg r(F') \\
  r(F_1 \land F_2) &= r(F_1) \land r(F_2) \\
  r(F_1 \lor F_2) &= r(F_1) \lor r(F_2) \\
  r(F_1 \rightarrow F_2) &= r(F_1) \rightarrow r(F_2) \\
  r(F_1 \leftrightarrow F_2) &= r(F_1) \leftrightarrow r(F_2) \\
  r((\forall X : s) F) &= (\forall Y) (p_1(Y) \land \ldots \land p_n(Y) \rightarrow r(F\{X/Y\})) \\
    & \text{if } \text{sort}(X) = s = \{p_1, \ldots, p_n\} \text{ and } Y \text{ is a new variable} \\
  r((\exists X : s) F) &= (\exists Y) (p_1(Y) \land \ldots \land p_n(Y) \land r(F\{X/Y\})) \\
    & \text{if } \text{sort}(X) = s = \{p_1, \ldots, p_n\} \text{ and } Y \text{ is a new variable}
\end{align*}
\]
Sorting Function and Relation Symbols

▶ Each atom of the form \( p(t_1, \ldots, t_n) \) can be equivalently replaced by

\[
(\forall X_1 \ldots X_n) (p(X_1, \ldots, X_n) \leftarrow X_1 \approx t_1 \land \ldots \land X_n \approx t_n).
\]

▶ Each atom \( A \) with \( A[\pi] = f(t_1, \ldots, t_n) \) can be equivalently replaced by

\[
(\forall X_1 \ldots X_n) A[\pi \mapsto f(X_1, \ldots, X_n)] \leftarrow X_1 \approx t_1 \land \ldots \land X_n \approx t_n.
\]

▶ Each formula \( F \) can be transformed into an equivalent formula \( F' \), in which

- all arguments of function and relation symbols different from \( \approx / 2 \) are variables and
- all equations are of the form \( d_1 \approx d_2 \) or \( f(X_1, \ldots, X_n) \approx d \), where \( X_1, \ldots, X_n \) are variables and \( d, d_1 \) and \( d_2 \) are variables or constants.

▶ Sorting the variables occurring in \( F' \) effectively sorts the function and relation symbols.
Sort Declaration

- $F'$ is usually quite lengthy and cumbersome to read.
- If $\text{sort}(X) = s$ then the sort declaration for the variable $X$ is

$$X : s.$$ 

- Let $s_i, 1 \leq i \leq n$, and $s$ be sorts, $f/n$ a function and $p/n$ a relation symbol. Then

$$f : s_1 \times \ldots \times s_n \rightarrow s$$

and

$$p : s_1 \times \ldots \times s_n$$

are sort declarations for $f/n$ and $p/n$ respectively.