Equational Logic

- Consider a first order language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$.

- $\approx / 2$ binary predicate symbol written infix.

- Equation $s \approx t$.

- Equational system $\mathcal{E}$ set of universally closed equations.

- Example

\[
\mathcal{E} = \{ \left(\forall X, Y, Z\right) (X \cdot Y) \cdot Z \approx X \cdot (Y \cdot Z), \\
\left(\forall X\right) 1 \cdot X \approx X, \\
\left(\forall X\right) X \cdot 1 \approx X, \\
\left(\forall X\right) X^{-1} \cdot X \approx 1, \\
\left(\forall X\right) X \cdot X^{-1} \approx 1 \} 
\]

where $\cdot / 2, \; ^{-1}/1, \; 1 / 0 \in \mathcal{F}$. 

Axioms of Equality

\[\mathcal{E}_{\approx} = \{ (\forall X) \ X \approx X, \]
\[\ (\forall X, Y) (X \approx Y \rightarrow Y \approx X), \]
\[\ (\forall X, Y, Z) (X \approx Y \land Y \approx Z \rightarrow X \approx Z) \}\]

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

f-substitutivity

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

r-substitutivity

reflexivity

symmetry

transitivity
Equality and Logical Consequence

- ▶ $E \cup E_\approx \models (\exists X) \ X \cdot a \approx 1$?
- ▶ $E \cup E_\approx \cup \{(\forall X) \ X \cdot X \approx 1\} \models (\forall X, Y) \ X \cdot Y \approx Y \cdot X$?
- ▶ Apply resolution: $10^{21}$ resolution steps.
- ▶ Problem: $E \cup E_\approx$ causes large search space.
- ▶ Idea: Remove troublesome formulas and build them into the deductive machinery.
- ▶ Two possibilities:
  ▶ additional rule of inference: paramodulation,
  ▶ built equational theory into unification computation.
- ▶ $E \cup E_\approx$ can be written as a set of definite clauses.
- ▶ There exists a least model.
  ▶ Least congruence relation: $s \approx_E t$ iff $E \cup E_\approx \models \forall s \approx t$. 

Equational Logic (20th December 2005)
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- $L[\pi \mapsto t]$ Literal $L$ where subterm at $\pi \in \mathcal{P}(L)$ has been replaced by $t$.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:**

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

- **Notation:** $\neg s \approx t \nsim s \not\approx t$.

- **Remember:** $E \cup E_\approx \models \forall s \approx t$ iff $E \cup E_\approx \rightarrow \forall s \approx t$ is valid
  iff $\neg(E \cup E_\approx \rightarrow \forall s \approx t)$ is unsatisfiable
  iff $E \cup E_\approx \cup \{\neg \forall s \approx t\}$ is unsatisfiable
  iff $E \cup E_\approx \cup \{\exists \neg s \approx t\}$ is unsatisfiable
  iff $E \cup E_\approx \cup \{\exists s \not\approx t\}$ 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

\(\varepsilon \cup \{(\forall X) \, X \approx X, \, (\forall X) \, X \cdot X \approx 1\} \models (\forall X, Y) \, X \cdot Y \approx Y \cdot X\)

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

hypothesis

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

associativity

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

hypothesis

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

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

right unit

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

reflexivity

\(n' \quad X_5 \approx X_5\)

\(n'' \quad [\quad]\)

res \((n),(n')\)
Shorthand Notation

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

- left unit
- hypothesis
- right unit
- hypothesis
- associativity
- hypothesis
- right unit

▶ Search space: \(10^{11}\) steps (instead of \(10^{21}\)).
▶ There are still many redundant and useless steps.

▷ restricted use of equations: term rewriting system.
Term Rewriting Systems

- $s \approx t \rightsquigarrow s \to t$ and is called rewrite rule.
- Term rewriting system $\mathcal{R}$ set of rewrite rules.
- $s \lbrack \pi \rbrack$ subterm of term $s$ at position $\pi \in \mathcal{P}(s)$.
- $s \lbrack \pi \mapsto v \rbrack$ term $s$ where subterm at $\pi \in \mathcal{P}(s)$ has been replaced by $v$.
- Rewriting $s \to_{\mathcal{R}} t$ iff there are $l \to r \in \mathcal{R}$, $\pi \in \mathcal{P}(s)$ and $\theta$ such that $s \lbrack \pi \rbrack = l\theta$ and $t = s \lbrack \pi \mapsto r\theta \rbrack$.

$\mathcal{R} = \{ \text{append}([], X) \to X, \\
\text{append}([X|Y], Z) \to [X|\text{append}(Y, Z)], \\
\text{reverse}([]) \to [], \\
\text{reverse}([X|Y]) \to \text{append}(\text{reverse}(Y), [X]) \} \}

\[
\begin{align*}
\text{append}([1, 2], [3, 4]) & \to_{\mathcal{R}} [1|\text{append}([2], [3, 4])] \\
& \to_{\mathcal{R}} [1, 2|\text{append}([], [3, 4])] \\
& \to_{\mathcal{R}} [1, 2, 3, 4].
\end{align*}
\]
Term Rewriting and Equational Logic

- $\rightarrow^*_R$ denotes the reflexive and transitive closure of $\rightarrow_R$.
- $s \leftrightarrow_R t$ iff $s \leftarrow_R t$ or $s \rightarrow_R t$.
- $\leftrightarrow^*_R$ is the reflexive and transitive closure of $\leftrightarrow_R$.
- $\mathcal{E}_R = \{l \approx r \mid l \rightarrow r \in \mathcal{R}\} \cup \mathcal{E}_\approx$.
- Theorem (i) $s \rightarrow_R t$ implies $s \approx_{\mathcal{E}_R} t$.
- (ii) $s \approx_{\mathcal{E}_R} t$ iff $s \leftrightarrow^*_R t$.

Proof ⇝ Exercise

Notation We sometimes omit the subscript $R$.

Matching problem
Given terms $u$ and $l$, does there exist a substitution $\theta$ such that $u = l \theta$?
Such a $\theta$ is called matcher.
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.
- Example $[1, 2, 3, 4]$ is the normal form of $\text{append}([1, 2], [3, 4])$.
- Normal forms are not unique:

{  
  \begin{align*}
  \text{not}(\text{not}(X)) & \rightarrow X, \\
  \text{not}(\text{or}(X, Y)) & \rightarrow \text{and}(\text{not}(X), \text{not}(Y)), \\
  \text{not}(\text{and}(X, Y)) & \rightarrow \text{or}(\text{not}(X), \text{not}(Y)), \\
  \text{and}(X, \text{or}(Y, Z)) & \rightarrow \text{or}(\text{and}(X, Y), \text{and}(X, Z)), \\
  \text{and}(\text{or}(X, Y), Z) & \rightarrow \text{or}(\text{and}(Y, Z), \text{and}(Z, X))
  \end{align*}
}

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

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

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

Confluent Term Rewriting Systems

- \( s \downarrow t \iff \) there exists \( u \) such that \( s \rightarrow^* u \leftarrow^* t. \)
- \( s \uparrow t \iff \) there exists \( u \) such that \( s \leftarrow^* u \rightarrow^* t. \)
- \( \mathcal{R} \) is confluent \iff for all terms \( s \) and \( t \) we find \( s \uparrow t \implies s \downarrow t. \)
- \( \mathcal{R} \) is ground confluent \iff it is confluent for ground terms.
- \( \mathcal{R} \) is Church-Rosser \iff for all terms \( s \) and \( t \) we find \( s \leftrightarrow^* t \iff s \downarrow t. \)
- Theorem \( \mathcal{R} \) is Church-Rosser \iff \( \mathcal{R} \) is confluent.
- Proof \( \leadsto \) Exercise.
- Remember \( s \leftrightarrow_{\mathcal{R}}^* t \iff s \approx_{\mathcal{E}} t. \)
  - Rewriting has only to be applied in one direction!
Canonical Term Rewriting Systems

- \( R \) is terminating iff it has no infinite rewriting sequences.
  - The question whether \( R \) is terminating is undecidable.

- \( R \) is canonical iff \( R \) confluent and terminating.
  - If \( R \) is canonical then \( s \cong_{ER} t \) iff \( s \downarrow t \).
  - If \( R \) is canonical then \( ER \) is decidable.
Termination

- Is a given term rewriting system terminating?
- **Idea**: Find a well-founded ordering $\succ / 2$ on terms such that $s \rightarrow t$ implies $s \succ t$.
- Let $\succeq / 2$ be a partial ordering on terms.
- $s \succ t$ iff $s \succeq t$ and $s \neq t$.
- $\succ / 2$ is well-founded iff there is no infinite sequence $s_1 \succ s_2 \succ \ldots$.
- A termination ordering $\succ / 2$ is a well-founded, transitive and antisymmetric relation on the set of terms satisfying the following properties:
  - **full invariance property** if $s \succ t$ then $s\theta \succ t\theta$,
  - **replacement property** if $s \succ t$ and $\pi \in P(s)$ then $u \succ u[\pi \mapsto t]$.
- **Theorem 4.2** Let $\mathcal{R}$ be a term rewriting system and $\succ / 2$ a termination ordering. If for all rules $l \rightarrow r \in \mathcal{R}$ we find that $l \succ r$ then $\mathcal{R}$ is terminating.
  - **Proof** $\rightsquigarrow$ Exercise.
Termination Orderings: Two Examples

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

- Polynomial ordering
  assign to each $f$ a polynomial with coefficients taken from $\mathbb{N}$.
  
  $\Rightarrow f(X, Y) \leadsto 2X + Y, \ h(X, Y) \leadsto X + Y.$
  
  $s > t$ iff $s^I > t^I$, where $\cdot^I$ denotes the chosen assignment.

- 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?
- $\mathcal{R}$ is locally confluent iff for all terms $r$, $s$ and $t$ the following holds: If $r \rightarrow s$ and $r \rightarrow t$ then $s \downarrow t$.
- Theorem 4.3 Let $\mathcal{R}$ be a terminating term rewriting system. $\mathcal{R}$ is confluent iff it is locally confluent.
  
  $\blacktriangleright$ Proof $\sim$ 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 $t \leadsto$ two redeces.

  - **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 redeces overlap).
Example: Consider \( t = (g(a) \cdot f(b)) \cdot c \)

(a) \( R = \{ a \rightarrow c, \ b \rightarrow c \} \).

▷ a and b are disjoint redeces in \( t \),
▷ ok.

(b) \( R = \{ a \rightarrow c, \ g(X) \rightarrow f(X) \} \).

▷ a and \( g(a) \) are redeces in \( t \); \( a \) corresponds to the variable position in \( g(x) \),
▷ ok.

(c) \( R = \{ (X \cdot Y) \cdot Z \rightarrow X, \ g(a) \cdot f(b) \rightarrow c \} \).

▷ \( (g(a) \cdot f(b)) \cdot c \) and \( g(a) \cdot f(b) \) are overlapping redeces in \( t \).
▷ problematic!
Critical Pairs

- Suppose \( \{l_1 \rightarrow r_1, l_2 \rightarrow r_2\} \subseteq \mathcal{R} \) and \( l_2 \) is unifiable with a non-variable subterm \( u \) of \( l_1 \) using mgu \( \theta \). Then the pair

\[
\langle (l_1[u/r_2])\theta, r_1\theta \rangle
\]

is said to be critical. It is obtained by superposing \( l_1 \) and \( l_2 \).

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

- **Theorem 4.4** A term rewriting system \( \mathcal{R} \) is locally confluent
  iff for all critical pairs \( \langle s, t \rangle \) of \( \mathcal{R} \) we find \( s \downarrow t \).

  ▶ 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 $\approx_{\mathcal{R}} = \approx_{\mathcal{R}'}$.
- 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 $\triangleright / 2$.

1. If for all critical pairs $\langle s, t \rangle$ of $\mathcal{R}$ we find that $s \downarrow t$ 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 $s \triangleright t$ then add the rule $s \rightarrow t$ to $\mathcal{R}$ and goto 1.
   - If $t \triangleright s$ then add the rule $t \rightarrow s$ to $\mathcal{R}$ and goto 1.
   - If neither $s \triangleright t$ nor $t \triangleright s$ then return “fail”.

▶ The completion procedure may either succeed or fail or loop.
Completion: An Example

\[ \mathcal{R} = \{ c \to b, \ f \to b, \ f \to a, \ e \to a, \ e \to d \} \]

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

- Critical pairs: \( \langle b, a \rangle \) and \( \langle d, a \rangle \).

- New rules: \( b \to a \) and \( d \to a \).

- \( \mathcal{R}' = \{ c \to b, \ f \to b, \ f \to a, \ e \to a, \ e \to d, \ b \to a, \ d \to a \} \).

- \( \mathcal{R}' \) is canonical.

- \( s \approx_{\mathcal{R}} t \iff s \approx_{\mathcal{R}'} t \).

- All proofs for \( s \approx_{\mathcal{R}'} t \) are in valley form.
Unification Theory

- \( \mathcal{E} \)-unification problem: \( \mathcal{E} \cup \mathcal{E} \approx \models \exists s \approx t. \)
- \( \mathcal{E} \)-unifier \( \theta \) is a solution of the \( \mathcal{E} \)-unification problem iff \( s\theta \approx \mathcal{E} t\theta. \)
- \( \eta \) and \( \theta \) are \( \mathcal{E} \)-equal on set \( V \) of variables (\( \theta \approx_{\mathcal{E}} \eta[V] \)) iff \( X\eta \approx_{\mathcal{E}} X\theta \) for all \( X \in V. \)
- \( \eta \) is an \( \mathcal{E} \)-instance of \( \theta \) on set \( V \) of variables (\( \theta \leq_{\mathcal{E}} \eta[V] \)) iff there exists a substitution \( \tau \) such that \( X\eta \approx_{\mathcal{E}} X\theta\tau \) for all \( X \in V. \)
- \( \theta \leq_{\mathcal{E}} \eta[V] \) iff \( \theta \leq_{\mathcal{E}} \eta[V] \) and not \( \theta \approx_{\mathcal{E}} \eta[V]. \)
- If neither \( \theta \leq_{\mathcal{E}} \eta[V] \) nor \( \eta \leq_{\mathcal{E}} \theta[V] \) then \( \theta \) and \( \eta \) are said to be incomparable.
Example: \( \mathcal{E} \cup \mathcal{E} \approx \models (\exists X, Y) f(X, g(a, b)) \approx f(g(Y, b), X) \)

- \( \mathcal{E} = \emptyset \):
  - Decision problem is decidable.
  - Most general unifier is unique modulo variable renaming:
    \( \theta_1 = \{ X \mapsto g(a, b), Y \mapsto a \} \).
- \( \mathcal{E} = \{(\forall X, Y) f(X, Y) \approx f(Y, X)\} \)
  - \( \theta_1 \) is a solution.
  - So is \( \theta_2 = \{ Y \mapsto a \} \):

  \[
  f(X, g(a, b))\theta_2 = f(X, g(a, b)) \approx_{\mathcal{E}} f(g(a, b), X) = f(g(Y, b), X)\theta_2.
  \]
  - In general, there may be finitely many most general unifiers.
Example: $\mathcal{E} \cup \mathcal{E}_\approx \models (\exists X, Y) f(X, g(a, b)) \approx f(g(Y, b), X)$

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

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

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

\[
\begin{align*}
  f(X, g(a, b))\theta_3 &= f(f(g(a, b), g(a, b)), g(a, b)) \\
  \approx_{\mathcal{E}} &= f(g(a, b), f(g(a, b), g(a, b))) \\
  &= f(g(Y, b), X)\theta_3.
\end{align*}
\]

$\blacktriangleright \quad$ $\theta_1$ and $\theta_3$ are incomparable.

$\blacktriangleright \quad$ In general, there may be infinitely many most general unifiers.
Sets of $\mathcal{E}$-Unifiers

- Given an $\mathcal{E}$-unification problem $\mathcal{E} \cup \mathcal{E} \models \exists s \approx t$.
- $U_\mathcal{E}(s, t)$ denotes the set of all $\mathcal{E}$-unifiers of $s$ and $t$.
- Complete set $cU_\mathcal{E}(s, t)$ of $\mathcal{E}$-unifiers of $s$ and $t$:
  - $cU_\mathcal{E}(s, t) \subseteq U_\mathcal{E}(s, t)$,
  - for all $\eta \in U_\mathcal{E}(s, t)$ there exists $\theta \in cU_\mathcal{E}(s, t)$ such that $\theta \leq_\mathcal{E} \eta[V]$, where $V = \text{VAR}(s) \cup \text{VAR}(t)$.
- Minimal complete set $\mu U_\mathcal{E}(s, t)$ of $\mathcal{E}$-unifiers for $s$ and $t$:
  - complete set,
  - for all $\theta, \eta \in \mu U_\mathcal{E}(s, t)$ we find $\theta \leq_\mathcal{E} \eta[V]$ implies $\theta = \eta$.
- If $cU_\mathcal{E}(s, t)$ is finite and $\leq_\mathcal{E}$ is decidable then there exists $\mu U_\mathcal{E}(s, t)$.
- Let $\theta \equiv_\mathcal{E} \eta[V]$ iff $\theta \leq_\mathcal{E} \eta[V]$ and $\eta \leq_\mathcal{E} \theta[V]$. Then, $\mu U_\mathcal{E}(s, t)$ is unique up to $\equiv_\mathcal{E}[V]$,
Another Example

$\mathcal{F} = \{a/0, f/2\}$, $\mathcal{E} = \{f(X, f(Y, Z)) \approx f(f(X, Y), Z)\}$,

$\mathcal{E} \cup \mathcal{E}_\approx \models (\exists X) f(X, a) \approx f(a, Y)$.

$\theta = \{X \mapsto a, \ Y \mapsto a\}$ is a solution.

$\eta = \{X \mapsto f(a, Z), \ Y \mapsto f(Z, a)\}$ is another solution.

$\{\theta, \eta\}$ is a complete set of $\mathcal{E}$-unifiers.

$\theta$ and $\eta$ are incomparable under $\leq_{\mathcal{E}}$.

$\{\theta, \eta\}$ is minimal.
Unification Types

The unification type of \( \mathcal{E} \) is

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

▶ \( \mathcal{E} \)-unification procedure:

▷ input: \( s \approx t \).
▷ output: subset of \( U_\mathcal{E}(s, t) \).
▷ is complete iff for all \( s, t \) the output is a \( cU_\mathcal{E}(s, t) \).
▷ is minimal iff for all \( s, t \) the output is a \( \mu U_\mathcal{E}(s, t) \).

▶ Universal \( \mathcal{E} \)-unification procedure:

▷ input: \( \mathcal{E} \) and \( s \approx t \).
▷ output: subset of \( U_\mathcal{E}(s, t) \).
▷ is complete iff for all \( \mathcal{E} \) and \( s, t \) the output is a \( cU_\mathcal{E}(s, t) \).
▷ is minimal iff for all \( \mathcal{E} \) and \( s, t \) the output is a \( \mu U_\mathcal{E}(s, t) \).
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 free constants.
- **general** iff $s$ and $t$ may contain additional free 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>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

- **E-matching problem**: $E \cup E \approx ? \exists \theta s \approx_E t\theta$.
- **Combination problem**: Can the results and unification algorithms for $E_1$ and $E_2$ be combined to $E_1 \cup E_2$?
- **Universal E-unification problem**: 
  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 $\mathcal{E}_R \cup \mathcal{E}_\approx \models \forall \ s \approx t$.

- **Rewriting**: $s \xrightarrow{\mathcal{R}} t$ iff there are $l \rightarrow r \in \mathcal{R}$, $\pi \in \mathcal{P}(s)$ and $\theta$ such that $s[\pi] = l\theta$ and $t = s[\pi \mapsto r\theta]$.

- **Now consider** $\mathcal{E}_R \cup \mathcal{E}_\approx \models \exists \ s \approx t$.

- **Narrowing**: $s \Rightarrow_{\mathcal{R}} t$ iff there are $l \rightarrow r \in \mathcal{R}$, $\pi \in \mathcal{P}(s)$ and $\theta$ such that $s[\pi] \neq \forall$, $s[\pi] \theta = l\theta$ and $t = (s[\pi \mapsto r])\theta$.

- **Compare narrowing to rewriting and paramodulation!**

**Theorem**

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

- A complete universal $\mathcal{E}$-unification procedure for canonical theories $\mathcal{E}$ 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.

- \(X \in_k M\) iff \(x\) occurs precisely \(k\) times in \(M\).

- \(M_1 \equiv M_2\) iff for all \(X\) we find \(X \in_k M_1\) iff \(X \in_k M_2\).

- \(X \in_m M_1 \cup M_2\) iff there exist \(k, l \geq 0\) such that \(X \in_k M_1, X \in_l M_2\) and \(k + l = m\).

- \(X \in_m M_1 \setminus M_2\) iff there exist \(k, l \geq 0\) such that either \(X \in_k M_1, X \in_l M_2, k > l\) and \(m = k - l\) or \(X \in_k M_1, X \in_l M_2, k \leq l\) and \(m = 0\).

- \(X \in_m M_1 \cap M_2\) iff there exist \(k, l \geq 0\) such that \(X \in_k M_1, X \in_l M_2\) and \(m = \min\{k, l\}\).

- \(M_1 \supseteq M_2\) iff \(M_1 \cap M_2 \equiv M_1\).
Fluent Terms

- Alphabet with variables $\mathcal{V}$ and function symbols $\mathcal{F} \supseteq \{o/2, 1/0\}$.
- $\mathcal{F}^-=\mathcal{F}\setminus\{o/2, 1/0\}$
- $\mathcal{T}(\mathcal{F}^-, \mathcal{V})$: terms built over $\mathcal{V}$ and $\mathcal{F}$ without using $o/2$ and $1/0$.
- Fluents: nonvariable elements of $\mathcal{T}(\mathcal{F}^-, \mathcal{V})$.
- Fluent terms:
  - 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.

$\mathcal{E}_{AC1} = \{ (\forall X, Y, Z) \ X \circ (Y \circ Z) \approx (X \circ Y) \circ Z \ 
(\forall X, Y) \ X \circ Y \approx Y \circ X \ 
(\forall X) \ X \circ 1 \approx X \} $
Multisets vs. Fluent Terms

$\cdot^I$

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

$\cdot^{-I}$

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

- **Submultiset matching problem:**
  Does there exist a $\theta$ such that $M\theta \subseteq N$, where $N$ is ground?

- **Submultiset unification problem:**
  Does there exist a $\theta$ such that $M\theta \subseteq N\theta$?

- **Fluent matching problem:**
  Does there exist a $\theta$ such that $(s \circ X)\theta \approx_{AC1} t$, where $t$ is ground and $X$ does not occur in $s$?

- **Fluent unification problem:**
  Does there exist a $\theta$ such that $(s \circ X)\theta \approx_{AC1} t\theta$, where $X$ does not occur in $s$ or $t$?
Submultiset vs. Fluent Unification Problems

- Equivalence of matching problems:
  \[(s \circ X)\theta \approx_{AC1} t \iff (s\theta)^I \subseteq t^I \text{ and } (X\theta)^I \models t^I \setminus (s\theta)^I\]

- Equivalence of unification problems:
  \[(s \circ X)\theta \approx_{AC1} t\theta \iff (s\theta)^I \subseteq (t\theta)^I \text{ and } (X\theta)^I \models (t\theta)^I \setminus (s\theta)^I\]

- 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 $s \approx_{AC1} t$ 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, 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 $S' = (S \setminus C\theta) \cup E\theta$.
  - 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

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

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

\[
\begin{align*}
\text{action}(\text{clear}(V) \circ \text{ontable}(V) \circ \text{empty}, \text{pickup}(V), \text{holding}(V)) \\
\text{action}(\text{clear}(V) \circ \text{on}(V, W) \circ \text{empty}, \text{unstack}(V, W), \text{holding}(V) \circ \text{clear}(W)) \\
\text{action}(\text{holding}(V), \text{putdown}(V), \text{clear}(V) \circ \text{ontable}(V) \circ \text{empty}) \\
\text{action}(\text{holding}(V) \circ \text{clear}(W), \text{stack}(V, W), \text{on}(V, W) \circ \text{clear}(V) \circ \text{empty})
\end{align*}
\]

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

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

\[
\begin{align*}
\text{causes}(X, [], Y), & \leftarrow X \approx Y \circ Z \\
\text{causes}(X, [V|W], Y) & \leftarrow \text{action}(P, V, Q) \land P \circ Z \approx X \\
& \land \text{causes}(Z \circ Q, W, Y)
\end{align*}
\]

$X \approx X$

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

- The planning problem is the problem whether

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

Let

- $\mathcal{F}$ be a set of definite clauses not containing $\approx/2$ in their head plus $X \approx X$ and
- $\mathcal{E}$ be an equational system.
- Does $\mathcal{F} \cup \mathcal{E} \models G$ hold?

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

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

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

(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) \(\leftarrow \text{clear}(V_2) \circ \text{on}(V_2, W_2) \circ \text{empty} \circ Z_1 \approx \\
\text{ontable}(a) \circ \text{ontable}(b) \circ \text{on}(c, a) \circ \text{clear}(b) \circ \text{clear}(c) \circ \text{empty} \land \\
\text{causes}(Z_1 \circ \text{holding}(V_2) \circ \text{clear}(W_2), W_1, \\
\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}).\)

(4) \(\leftarrow causes(\text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b) \circ \text{clear}(a) \circ \text{holding}(c), W_1, \\
\text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}).\)

\[\vdots\]
A Solution to Sussman’s Anomaly (2)

\[\begin{align*}
(7) & \quad \leftarrow \text{causes}(\text{ontable}(a) \circ \text{ontable}(b) \circ \text{clear}(b) \circ \\
& \quad \quad \text{clear}(a) \circ \text{clear}(c) \circ \text{ontable}(c) \circ \text{empty}, \ W_4, \\
& \quad \quad \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \\
\vdots \\
(10) & \quad \leftarrow \text{causes}(\text{ontable}(a) \circ \text{clear}(c) \circ \text{ontable}(c) \circ \text{clear}(a) \circ \text{holding}(b), \ W_7, \\
& \quad \quad \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \\
\vdots \\
(13) & \quad \leftarrow \text{causes}(\text{ontable}(a) \circ \text{ontable}(c) \circ \text{clear}(a) \circ \text{on}(b, c) \circ \text{clear}(b) \circ \text{empty}, \ W_{10}, \\
& \quad \quad \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \\
\vdots \\
(16) & \quad \leftarrow \text{causes}(\text{ontable}(c) \circ \text{on}(b, c) \circ \text{clear}(b) \circ \text{holding}(a), \ W_{13}, \\
& \quad \quad \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \\
\vdots \\
(19) & \quad \leftarrow \text{causes}(\text{ontable}(c) \circ \text{on}(b, c) \circ \text{clear}(a) \circ \text{on}(a, b) \circ \text{empty}, \ W_{16}, \\
& \quad \quad \text{ontable}(c) \circ \text{on}(b, c) \circ \text{on}(a, b) \circ \text{clear}(a) \circ \text{empty}). \\
(20) & \quad \left[ \right]
\end{align*}\]
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

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

- 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 $E_{ACI1} = E_{AC1} \cup \{(\forall X) \ X \circ X \approx X\}$.
- In this case the $E$–matching problem

$$E_{ACI1} \models (\exists Z) \ s \approx t \circ Z$$

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 $F_{ACI1}$.

- 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.

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

- Indeterminate effects
- Consistency constraints
- etc; → Rational Agents
Sorts

▶ \( (\forall X, Y) \ (\text{number}(X) \land \text{number}(Y) \rightarrow \text{plus}(X, Y) = \text{plus}(Y, X)) \)

▶ \( (\forall X, Y : \text{number}) \, \text{plus}(X, Y) = \text{plus}(Y, X) \).

First order language with sorts:

▶ first order language,

▶ \( \text{sort} : \mathcal{A}_V \rightarrow 2^{\mathcal{A}_S} \),

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

▶ \( \text{Sort } s \in 2^{\mathcal{A}_S} \);

▶ \( \emptyset \in 2^{\mathcal{A}_S} \) 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 \( X : s \).
Sorts – Semantics

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

$$I : s = \{p_1, \ldots, p_n\} \mapsto s^I = D \cap p^I_1 \cap \ldots \cap p^I_n.$$  

$\triangleright$ $I : \emptyset \mapsto D$.

- State $\sigma$ is sorted iff for all $X : s$ we find $\sigma(X) \in s^I$.

- We assume that all sorts are non-empty.

\[
\begin{align*}
I, \sigma \models p(t_1, \ldots, t_n) & \quad \text{iff} \quad (\sigma(t_1), \ldots, \sigma(t_n)) \in p^I. \\
I, \sigma \models \neg F & \quad \text{iff} \quad I, \sigma \not\models F. \\
I, \sigma \models F_1 \land F_2 & \quad \text{iff} \quad I, \sigma \models F_1 \text{ and } I, \sigma \models F_2. \\
I, \sigma \models F_1 \lor F_2 & \quad \text{iff} \quad I, \sigma \models F_1 \text{ or } I, \sigma \models F_2. \\
I, \sigma \models F_1 \rightarrow F_2 & \quad \text{iff} \quad I, \sigma \not\models F_1 \text{ or } I, \sigma \models F_2. \\
I, \sigma \models F_1 \leftrightarrow F_2 & \quad \text{iff} \quad I, \sigma \models F_1 \rightarrow F_2 \text{ and } I, \sigma \models F_2 \rightarrow F_1. \\
I, \sigma \models (\exists X : s) F & \quad \text{iff} \quad \text{there exists } d \in s^I \text{ such that } I, \sigma\{X/d\} \models F. \\
I, \sigma \models (\forall X : s) F & \quad \text{iff} \quad \text{for all } d \in s^I \text{ we find } I, \sigma\{X/d\} \models F.
\end{align*}
\]
Relativization

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

$$
\begin{align*}
    r(p(t_1, \ldots, t_n)) &= p(t_1, \ldots, t_n) \\
    r(\neg F) &= \neg r(F) \\
    r(F_1 \land F_2) &= r(F_1) \land r(F_2) \\
    r(F_1 \lor F_2) &= r(F_1) \lor r(F_2) \\
    r(F_1 \rightarrow F_2) &= r(F_1) \rightarrow r(F_2) \\
    r(F_1 \leftrightarrow F_2) &= r(F_1) \leftrightarrow r(F_2) \\
    r((\forall X : s) F) &= (\forall Y)(p_1(Y) \land \ldots \land p_n(Y) \rightarrow r(F\{X/Y\})) \\
    &\text{if } sort(X) = s = \{p_1, \ldots, p_n\} \text{ and } Y \text{ is a new variable} \\
    r((\exists X : s) F) &= (\exists Y)(p_1(Y) \land \ldots \land p_n(Y) \land r(F\{X/Y\})) \\
    &\text{if } sort(X) = s = \{p_1, \ldots, p_n\} \text{ and } Y \text{ is a new variable}
\end{align*}
$$
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 of the form \( p[f(t_1, \ldots, t_n)] \) can be equivalently replaced by
  \[
  (\forall X_1 \ldots X_n) \ (p[f(t_1, \ldots, t_n)]/f(X_1, \ldots, X_n) \leftarrow X_1 \approx t_1 \land \ldots \land X_n \approx t_n).
  \]

- 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.