Automated Theorem Proving

Lecture 9: Rewrite Systems

Prof. Dr. Jasmin Blanchette

based on slides by Dr. Uwe Waldmann

Winter Term 2024/25

Part 4: First-Order Logic with Equality

Equality is the most important relation in mathematics and functional programming.

In principle, problems in first-order logic with equality can be handled by any prover for first-order logic without equality, as follows.

4.1 Handling Equality Naively

Proposition 4.1.1:

Let F be a closed first-order formula with equality. Let $\sim \notin \Pi$ be a new predicate symbol. The set $Eq(\Sigma)$ contains the formulas

$$\begin{gathered}
\forall x\,(x \sim x)\\
\forall x, y\,(x \sim y \rightarrow y \sim x)\\
\forall x, y, z\,(x \sim y \wedge y \sim z \rightarrow x \sim z)\\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_n \sim y_n \rightarrow f(x_1, \dots, x_n) \sim f(y_1, \dots, y_n))\\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_m \sim y_m \wedge P(x_1, \dots, x_m) \rightarrow P(y_1, \dots, y_m))
\end{gathered}$$

for every $f/n \in \Omega$ and $P/m \in \Pi$. Let $\tilde{F}$ be the formula that one obtains from F if every occurrence of $\approx$ is replaced by $\sim$. Then F is satisfiable if and only if $Eq(\Sigma) \cup \{\tilde{F}\}$ is satisfiable.

Handling Equality Naively

An analogous proposition holds for *sets* of closed first-order formulas with equality.

By giving the equality axioms explicitly, first-order problems with equality can in principle be solved by a standard resolution or tableaux prover.

But this is unfortunately not efficient
(mainly due to the transitivity and congruence axioms).

Handling Equality Naively

Equality is theoretically difficult:
First-order functional programming is Turing-complete.

But resolution theorem provers cannot even solve equational problems that are intuitively easy.

Consequence: To handle equality efficiently, knowledge must be integrated into the theorem prover.

Roadmap

How to proceed:

- This part: Equations (unit clauses with equality).

  Term rewrite systems.

  Knuth–Bendix completion.

- Next part: Equational clauses.

  Combining resolution and Knuth–Bendix completion.

  $\rightarrow$ Superposition.

4.2 Rewrite Systems

Let E be a set of (implicitly universally quantified) equations.

The rewrite relation $\rightarrow_E \subseteq \mathrm{T}_\Sigma(X) \times \mathrm{T}_\Sigma(X)$ is defined by

$$
\begin{array}{lll}
s \rightarrow_E t & \text{if and only if} & \text{there exist } (l \approx r) \in E,\ p \in \mathrm{pos}(s), \\
& & \text{and } \sigma : X \rightarrow \mathrm{T}_\Sigma(X), \\
& & \text{such that } s|_p = l\sigma \text{ and } t = s[r\sigma]_p.
\end{array}
$$

An instance of the lhs (left-hand side) of an equation is called a redex (reducible expression).

Contracting a redex means replacing it with the corresponding instance of the rhs (right-hand side) of the rule.

Rewrite Systems

An equation $l \approx r$ is also called a rewrite rule if l is not a variable and $\mathrm{var}(l) \supseteq \mathrm{var}(r)$.

Notation: $l \to r$.

A set of rewrite rules is called a term rewrite system (TRS).

Rewrite Systems

We say that a set of equations E or a TRS R is terminating
if the rewrite relation $\rightarrow_E$ or $\rightarrow_R$ has this property.

(Analogously for other properties of abstract reduction systems.)

Note: If E is terminating, then it is a TRS.

E-Algebras

Let E be a set of universally quantified equations.
A model of E is also called an E-algebra.

If $E \models \forall\vec{x}\,(s \approx t)$, i.e., $\forall\vec{x}\,(s \approx t)$ is valid in all E-algebras,
we write this also as $s \approx_E t$.

Goal:
Use the rewrite relation $\rightarrow_E$ to express the semantic consequence relation syntactically:

$s \approx_E t$ if and only if $s \leftrightarrow_E^* t$.

E-Algebras

Let E be a set of equations over $\mathrm{T}_\Sigma(X)$. The following inference system allows us to derive consequences of E:

E-Algebras

$E \vdash t \approx t$ (Reflexivity)

for every $t \in \mathrm{T}_\Sigma(X)$

$$\frac{E \vdash t \approx t'}{E \vdash t' \approx t}$$ (Symmetry)

$$\frac{E \vdash t \approx t' \qquad E \vdash t' \approx t''}{E \vdash t \approx t''}$$ (Transitivity)

$$\frac{E \vdash t_1 \approx t_1' \quad \ldots \quad E \vdash t_n \approx t_n'}{E \vdash f(t_1, \ldots, t_n) \approx f(t_1', \ldots, t_n')}$$ (Congruence)

$E \vdash t\sigma \approx t'\sigma$ (Instance)

if $(t \approx t') \in E$ and $\sigma : X \to \mathrm{T}_\Sigma(X)$

E-Algebras

Lemma 4.2.1:

The following properties are equivalent:

(i) $s \leftrightarrow_E^* t$

(ii) $E \vdash s \approx t$ is derivable.

E-Algebras

Constructing a quotient algebra:

Let X be a set of variables.

For $t \in \mathrm{T}_\Sigma(X)$ let $[t] = \{t' \in \mathrm{T}_\Sigma(X) \mid E \vdash t \approx t'\}$ be the congruence class of t.

Define a Σ-algebra $\mathrm{T}_\Sigma(X)/E$ (abbreviated by $\mathcal{T}$) as follows:

$U_\mathcal{T} = \{[t] \mid t \in \mathrm{T}_\Sigma(X)\}$.

$f_\mathcal{T}([t_1], \dots, [t_n]) = [f(t_1, \dots, t_n)]$ for $f/n \in \Omega$.

E-Algebras

Lemma 4.2.2:

$f_{\mathcal{T}}$ is well-defined:

If $[t_i] = [t_i']$, then $[f(t_1, \dots, t_n)] = [f(t_1', \dots, t_n')]$.

Lemma 4.2.3:

$\mathcal{T} = \mathrm{T}_\Sigma(X)/E$ is an E-algebra.

Lemma 4.2.4:

Let X be a countably infinite set of variables; let $s, t \in \mathrm{T}_\Sigma(Y)$.

If $\mathrm{T}_\Sigma(X)/E \models \forall\vec{x}\,(s \approx t)$, then $E \vdash s \approx t$ is derivable.

E-Algebras

Theorem 4.2.5 (“Birkhoff’s Theorem”):
Let X be a countably infinite set of variables, let E be a set of (universally quantified) equations. Then the following properties are equivalent for all $s, t \in \mathrm{T}_\Sigma(X)$:

(i) $s \leftrightarrow_E^* t$.

(ii) $E \vdash s \approx t$ is derivable.

(iii) $s \approx_E t$, i.e., $E \models \forall \vec{x} (s \approx t)$.

(iv) $\mathrm{T}_\Sigma(X)/E \models \forall \vec{x} (s \approx t)$.

4.3 Confluence

Let $(A, \rightarrow)$ be an abstract reduction system.

b and $c \in A$ are joinable if there is an a such that $b \rightarrow^* a \leftarrow^* c$.
Notation: $b \downarrow c$.

The relation $\rightarrow$ is called

Church–Rosser if $b \leftrightarrow^* c$ implies $b \downarrow c$;

confluent if $b \leftarrow^* a \rightarrow^* c$ implies $b \downarrow c$;

locally confluent if $b \leftarrow a \rightarrow c$ implies $b \downarrow c$;

convergent if it is confluent and terminating.

Confluence

Theorem 4.3.1:

The following properties are equivalent:

(i) $\rightarrow$ has the Church–Rosser property.

(ii) $\rightarrow$ is confluent.

Confluence

Lemma 4.3.2:
If → is confluent, then every element has at most one normal form.

Corollary 4.3.3:
If → is normalizing and confluent, then every element b has a unique normal form.

Proposition 4.3.4:
If → is normalizing and confluent, then $b \leftrightarrow^* c$ if and only if $b{\downarrow} = c{\downarrow}$.

Confluence and Local Confluence

Theorem 4.3.5 (“Newman’s Lemma”):

If a terminating relation $\rightarrow$ is locally confluent, then it is confluent.

Rewrite Relations

Corollary 4.3.6:
If E is convergent (i.e., terminating and confluent),
then $s \approx_E t$ if and only if $s \leftrightarrow_E^* t$ if and only if $s{\downarrow}_E = t{\downarrow}_E$.

Corollary 4.3.7:
If E is finite and convergent, then $\approx_E$ is decidable.

Reminder:
If E is terminating, then it is confluent if and only if it is locally confluent.

Rewrite Relations

Problems:

Show local confluence of E.

Show termination of E.

Transform E into an equivalent set of equations that is locally confluent and terminating.

4.4 Critical Pairs

Showing local confluence (sketch):

Problem: If $t_1 \leftarrow_E t_0 \rightarrow_E t_2$, does there exist a term s such that $t_1 \rightarrow_E^* s \leftarrow_E^* t_2$?

If the two rewrite steps happen in different subtrees (disjoint redexes): yes.

If the two rewrite steps happen below each other (overlap at or below a variable position): yes.

If the left-hand sides of the two rules overlap at a nonvariable position: needs further investigation.

Critical Pairs

Showing local confluence (sketch):

Question:
Are there rewrite rules $l_1 \to r_1$ and $l_2 \to r_2$ such that some subterm $l_1|_p$ and l_2 have a common instance $(l_1|_p)\sigma_1 = l_2\sigma_2$?

Observation:
If we assume without loss of generality that the two rewrite rules do not have common variables, then only a single substitution is necessary: $(l_1|_p)\sigma = l_2\sigma$.

Further observation:
The mgu of $l_1|_p$ and l_2 subsumes all unifiers σ of $l_1|_p$ and l_2.

Critical Pairs

Let $l_i \to r_i$ ($i \in \{1, 2\}$) be two rewrite rules in a TRS R whose variables have been renamed such that $\mathrm{var}(l_1) \cap \mathrm{var}(l_2) = \emptyset$. (Recall that $\mathrm{var}(l_i) \supseteq \mathrm{var}(r_i)$.)

Let $p \in \mathrm{pos}(l_1)$ be a position such that $l_1|_p$ is not a variable and σ is an mgu of $l_1|_p$ and l_2.

Then $r_1\sigma \leftarrow l_1\sigma \to (l_1\sigma)[r_2\sigma]_p$.

$\langle r_1\sigma, (l_1\sigma)[r_2\sigma]_p\rangle$ is called a critical pair of R.

The critical pair is joinable (or: converges) if $r_1\sigma \downarrow_R (l_1\sigma)[r_2\sigma]_p$.

Critical Pairs

Theorem 4.4.1 (“Critical Pair Theorem”):
A TRS R is locally confluent if and only if all its critical pairs are joinable.

Proof:
“only if”: Obvious, since joinability of a critical pair is a special case of local confluence.

Critical Pairs

"if": Suppose s rewrites to t_1 and t_2 using rewrite rules $l_i \to r_i \in R$ at positions $p_i \in \mathrm{pos}(s)$, where $i \in \{1, 2\}$.
Without loss of generality, we can assume that the two rules are variable disjoint, hence $s|_{p_i} = l_i\theta$ and $t_i = s[r_i\theta]_{p_i}$.

We distinguish between two cases: Either p_1 and p_2 are in disjoint subtrees ($p_1 \parallel p_2$) or one is a prefix of the other (without loss of generality, $p_1 \leq p_2$).

Critical Pairs

Case 1: $p_1 \parallel p_2$.

Then $s = s[l_1\theta]_{p_1}[l_2\theta]_{p_2}$,
and therefore $t_1 = s[r_1\theta]_{p_1}[l_2\theta]_{p_2}$ and $t_2 = s[l_1\theta]_{p_1}[r_2\theta]_{p_2}$.

Let $t_0 = s[r_1\theta]_{p_1}[r_2\theta]_{p_2}$.
Then clearly $t_1 \rightarrow_R t_0$ using $l_2 \rightarrow r_2$ and $t_2 \rightarrow_R t_0$ using $l_1 \rightarrow r_1$.

Critical Pairs

Case 2: $p_1 \leq p_2$.

Case 2.1: $p_2 = p_1\, q_1\, q_2$, where $l_1|_{q_1}$ is some variable x.

In other words, the second rewrite step takes place at or below a variable in the first rule. Suppose that x occurs m times in l_1 and n times in r_1 (where $m \geq 1$ and $n \geq 0$).

Then $t_1 \to_R^* t_0$ by applying $l_2 \to r_2$ at all positions $p_1\, q'\, q_2$, where q' is a position of x in r_1.

Conversely, $t_2 \to_R^* t_0$ by applying $l_2 \to r_2$ at all positions $p_1\, q\, q_2$, where q is a position of x in l_1 different from q_1, and by applying $l_1 \to r_1$ at p_1 with the substitution θ', where $\theta' = \theta[x \mapsto (x\theta)[r_2\theta]_{q_2}]$.

Critical Pairs

Case 2.2: $p_2 = p_1\, p$, where p is a nonvariable position of l_1.

Then $s|_{p_2} = l_2\theta$ and $s|_{p_2} = (s|_{p_1})|_p = (l_1\theta)|_p = (l_1|_p)\theta$,
so θ is a unifier of l_2 and $l_1|_p$.

Let σ be the mgu of l_2 and $l_1|_p$,
then $\theta = \tau \circ \sigma$ and $\langle r_1\sigma, (l_1\sigma)[r_2\sigma]_p\rangle$ is a critical pair.

By assumption, it is joinable, so $r_1\sigma \rightarrow_R^* v \leftarrow_R^* (l_1\sigma)[r_2\sigma]_p$.

Consequently, $t_1 = s[r_1\theta]_{p_1} = s[r_1\sigma\tau]_{p_1} \rightarrow_R^* s[v\tau]_{p_1}$ and $t_2 = s[r_2\theta]_{p_2} = s[(l_1\theta)[r_2\theta]_p]_{p_1} = s[(l_1\sigma\tau)[r_2\sigma\tau]_p]_{p_1} = s[((l_1\sigma)[r_2\sigma]_p)\tau]_{p_1} \rightarrow_R^* s[v\tau]_{p_1}$.

This completes the proof of the Critical Pair Theorem. □

Critical Pairs

Note: Critical pairs between a rule and (a renamed variant of) itself must be considered—except if the overlap is at the root (i.e., $p = \varepsilon$).

Critical Pairs

Corollary 4.4.2:
A terminating TRS R is confluent if and only if all its critical pairs are joinable.

Corollary 4.4.3:
For a finite terminating TRS, confluence is decidable.

Critical Pairs: Example

We compute the critical pairs for the following rewrite system and determine whether they are joinable:

$$f(g(f(x))) \to x \quad (1) \qquad f(g(x)) \to g(f(x)) \quad (2)$$

- Between (1) at position 11 and a renamed copy of (1):
  $\sigma = \{x \mapsto g(f(x'))\}$,
  $g(f(x')) \leftarrow f(g(f(g(f(x'))))) \to f(g(x'))$,
  critical pair: $\langle g(f(x')), f(g(x'))\rangle$, joinable at $f(g(x'))$.

- Between (1) at position ε and a renamed copy of (2):
  $\sigma = \{x' \mapsto f(x)\}$,
  $x \leftarrow f(g(f(x))) \to g(f(f(x)))$,
  critical pair: $\langle x, g(f(f(x)))\rangle$, not joinable.

Critical Pairs: Example

- Between (1) at position 11 and a renamed copy of (2):
  $\sigma = \{x \mapsto g(x')\}$,
  $f(g(g(f(x')))) \leftarrow f(g(f(g(x')))) \rightarrow g(x')$,
  critical pair: $\langle f(g(g(f(x')))), g(x') \rangle$, joinable at $g(x')$.