Term Rewriting Systems

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

1. Motivation and basic definitions and results.
2. Equational Problems: the word problem and term rewriting
3. Termination of term rewriting systems
4. Confluence of term rewriting systems
5. Completion of term rewriting systems

Terms

and related definitions

Signature:

- set of function symbols Σ — f, g, h
- $\Sigma^{(n)}$ denotes symbols of arity n — $f, g \in \Sigma^{(2)}, h \in \Sigma^{(1)}$
- elements of $\Sigma^{(0)}$: constant symbols — $a, b \in \Sigma^{(0)}$

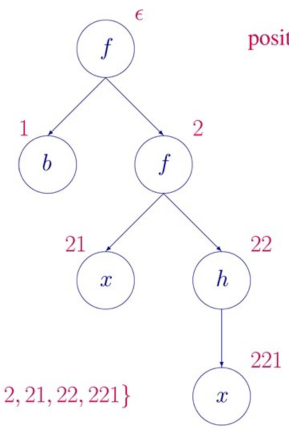
Let Σ be a signature and X be a set of variables with $\Sigma \cap X = \emptyset$.

The set $T(\Sigma, X)$ of all Σ-terms over X is inductively defined as

- $X \subseteq T(\Sigma, X)$ — x, y
- for all $n \geq 0$, all $f \in \Sigma^{(n)}$, and all $t_1, \ldots, t_n \in T(\Sigma, X)$, we have $f(t_1, \ldots, t_n) \in T(\Sigma, X)$ — $f(a, g(x, h(y)))$

Terms as trees

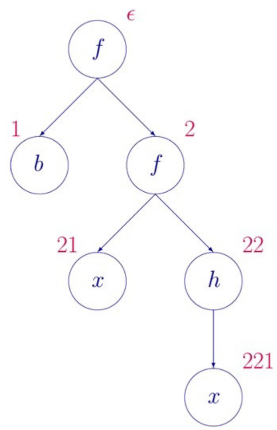
$t = f(b, (f(x, h(x)))$



positions of a term

$\mathcal{P}os(t) = \{\epsilon, 1, 2, 21, 22, 221\}$

$t = f(b, (f(x, h(x)))$



$t|_p$ subterm at position p

$t|_2 = f(x, h(x))$

$t|_{22} = h(x)$

$t[s]_p$ replace subterm at position p by s

$t[a]_2 = f(b, a)$

$t[g(a, a)]_{22} = f(b, f(x, g(a, a)))$

Substitutions

replace variables by terms

Σ a signature, V a countably infinite set of variables

A $T(\Sigma, V)$-substitution is a function

$$\sigma : V \rightarrow T(\Sigma, V)$$

such that

$$\mathcal{D}om(\sigma) := \{x \in V \mid \sigma(x) \neq x\}$$

is finite.

If $\mathcal{D}om(\sigma) = \{x_1, \ldots, x_n\}$, then we may write σ as

$$\sigma = \{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}.$$

$$\sigma := \{x \mapsto a, y \mapsto h(x)\}$$

The substitution σ can be extended to a mapping

$$\sigma : T(\Sigma, V) \rightarrow T(\Sigma, V)$$

by induction:

$$\sigma(f(s_1, \ldots, s_n)) := f(\sigma(s_1), \ldots, \sigma(s_n)).$$

$$t = f(x, f(y, x))$$

$$\sigma(t) = f(a, f(h(x), a))$$

Dresden

Identities

between terms

A Σ-identity is a pair $(s, t) \in T(\Sigma, V) \times T(\Sigma, V)$.

Identities will be written as $s \approx t$.

Semantically, an identitiy $s \approx t$ is implicitly universally quantified, i.e., stands for its universal closure $\forall \overline{x}.\ s \approx t$.

Logical consequence is then defined in the usual way.

If E is a set of identities, then we define

$$\approx_E := \{(s, t) \mid E \models \forall \overline{x}.\ s \approx t\}.$$

Word problem for E:

Given terms s, t, does $s \approx_E t$ hold or not?

Dresden

Example

a set of identities with undecidable word problem

Combinatory logic:

one of the earliest formalisms for encoding all computable functions

Signature: $\Sigma := \{\cdot, S, K\}$, where $\cdot$ is binary (infix), and S and K are constants.

Identities: describe computations

$$E := \{((S \cdot x) \cdot y) \cdot z \approx (x \cdot z) \cdot (y \cdot z),\ (K \cdot x) \cdot y \approx x\}.$$

The word problem for E is known to be undecidable.

Dresden

In the following, we consider two decidable special cases:

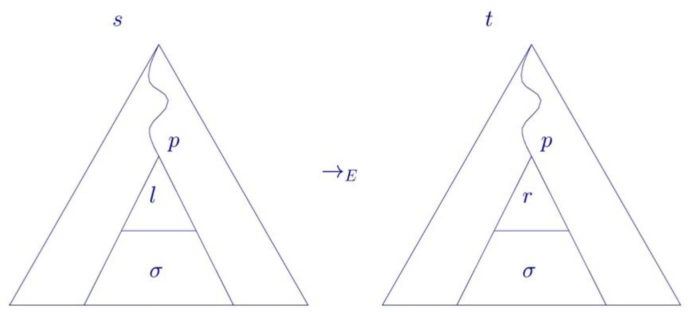
- Deciding $\approx_E$ using a finite convergent (terminating and confluent) term rewriting system.
- The word problem is decidable for finite sets of ground identities (no variables).

Dresden

Reduction relation

induced by a set of identities

$s \to_E t$ iff $\exists (l,r) \in E,\ p \in \mathcal{P}os(s),$ substitution σ.
$s|_p = \sigma(l)$ and $t = s[\sigma(r)]_p$.



Example

$s \to_E t$ iff $\exists (l,r) \in E,\ p \in \mathcal{P}os(s),$ substitution σ.
$s|_p = \sigma(l)$ and $t = s[\sigma(r)]_p$.

$G := \{f(x, f(y,z)) \approx f(f(x,y),z),\ f(e,x) \approx x,\ f(i(x),x) \approx e\}$

$f(i(e), f(e,e)) \to_G f(f(i(e),e),e) \to_G f(e,e) \to_G e$

first identity
at position $p_1 = \epsilon$
with substitution
$\sigma_1 = \{x \mapsto i(e), y \mapsto e, z \mapsto e\}$

third identity
at position $p_2 = 1$
with substitution
$\sigma_2 = \{x \mapsto e\}$

second identity
at position $p_3 = \epsilon$
with substitution
$\sigma_3 = \{x \mapsto e\}$

$\overset{*}{\to}_E$: reflexive transitive closure of $\to_E$

$\overset{*}{\leftrightarrow}_E$: reflexive transitive symmetric closure of $\to_E$

Theorem (Birkhoff)

$\overset{*}{\leftrightarrow}_E$ coincides with $\approx_E$

Consequently, to decide the word problem,
it is sufficient to decide the relation $\overset{*}{\leftrightarrow}_E$.

Word problem

Given a set of identities E and terms s, t,
does $s \overset{*}{\leftrightarrow}_E t$ hold or not?

Recall from the section on abstract reduction systems:

Theorem

If $\to_E$ is confluent and terminating, then

- every term s has a unique normal form $s{\downarrow}_E$.
- $s \overset{*}{\leftrightarrow}_E t$ iff $s{\downarrow}_E = t{\downarrow}_E$.

Consequently, to decide the word problem in this case
we must be able to compute normal forms.

This is possible if we can effectively

1. decide if a term u is already in normal form (w.r.t. $\to_E$) and
2. compute some u' such that $u \to_E u'$ if u is not in normal form.

Matching

The matching problem:

Given two terms s and l, determine if there exists a substitution σ such that

$\sigma(l) = s$,

and compute σ if it exists.

We will see later (as a consequence of results on unification) that the matching problem is decidable and that matchers can effectively be computed.

Theorem

If E is finite and $\rightarrow_E$ is convergent, then the word problem for E decidable.

Proof:

1. decide if a term u is already in normal form:

   check for all identities $(l \approx r) \in E$ and all positions $p \in \mathcal{P}os(u)$ if there is a substitution σ such that $u|_p = \sigma(l)$.

2. compute some u' such that $u \rightarrow_E u'$:

   reduce u to $u[\sigma(r)]_p$ if the above test is positive

Definition

- A rewrite rule is an identity $l \approx r$ such that
  - l is not a variable,
  - $\mathcal{V}ar(l) \supseteq \mathcal{V}ar(r)$.
- We write rewrite rules as $l \rightarrow r$ instead of $l \approx r$.
- A term rewriting system (TRS) is a set of rewrite rules.

The restrictions on rewrite rules prevent obvious cases of non-termination.

Given a TRS R, the notations $\rightarrow_R$ and $\approx_R$ are well-defined since rewrite rules are by definition identities.

If R is a finite TRS and $\rightarrow_R$ is convergent, then $\approx_R$ is decidable.

Congruence closure

decides the WP for ground identities

An identity $l \approx r$ is a ground identity if it contains no variables.

In the following, G denotes a finite set of ground identities.

Congruence closure of G computes the smallest congruence on terms that contains G:

$$\begin{aligned}
R(E) &:= \{(t,t) \mid t \in T(\Sigma, V)\},\\
S(E) &:= \{(t,s) \mid (s,t) \in E\},\\
T(E) &:= \{(s,u) \mid \exists t.\ (s,t),(t,u) \in E\},\\
C(E) &:= \{(f(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \mid f \in \Sigma^{(n)}, (s_1,t_1),\ldots,(s_n,t_n) \in E\}.\\
Cong(E) &:= E \cup R(E) \cup S(E) \cup T(E) \cup C(E)
\end{aligned}$$

E is closed under $Cong$ (i.e., $Cong(E) = E$) iff E is a congruence.

The process of closing G under $Cong$ is an iteration from below:

$$G_0 := G$$
$$G_{i+1} := Cong(G_i)$$
$$CC(G) := \bigcup_{i\geq 0} G_i$$

Lemma

$CC(G) = \approx_G$.

$CC(G)$ may be infinite: constants a and b, and a unary function f

$$CC(\{a \approx b\}) \supseteq \{f^i(a) \approx f^i(b) \mid i \geq 0\}$$

$f^2(a) \approx_G f^2(b)$ since this identity is in G_2

How can we conclude that $f^2(a) \not\approx_G f^3(b)$?

Dresden

Since G is ground, we need to consider only those terms that occur in G or in the input terms to be tested for equivalence.

$$Subterms(t) := \{t|_p \mid p \in \mathcal{P}os(t)\}$$
$$Subterms(E) := \bigcup_{(l\approx r)\in E} (Subterms(l) \cup Subterms(r))$$

Fix a finite set of ground identities G and two terms s and t.

$$S := Subterms(G) \cup Subterms(s) \cup Subterms(t)$$

Starting with G, we define the sequence

$$H_0 := G,$$
$$H_{i+1} := Cong(H_i) \cap (S \times S)$$

Dresden

Lemma

There is some m such that $H_{m+1} = H_m = \bigcup_{i\geq 0} H_i$.

Proof:

$H_0 \subseteq H_1 \subseteq H_2 \subseteq H_3 \subseteq \cdots \subseteq S \times S$
and S is finite.

The limit H_m is denoted by $CC_S(G)$

Lemma

$CC_S(G)$ can be computed in polynomial time.

Dresden

Theorem

$CC_S(G) = \approx_G \cap (S \times S)$

Proof:

($\subseteq$) Because $H_i \subseteq G_i \cap (S \times S)$, we also have

$$CC_S(G) \subseteq CC(G) \cap (S \times S) = \approx_G \cap (S \times S)$$

($\supseteq$) Assume $u, v \in S$ and $u \leftrightarrow_G^n v$.

We prove $(u, v) \in H_m = CC_S(G)$ by well-founded induction on the lexicographically ordered pair $(n, |u|)$.

If $n = 0$, then $u = v$ and hence $(u, v) \in H_1 \subseteq H_m$.

If $u \leftrightarrow_G^{n+1} v$, we distinguish two cases:

1. There is a rewrite step at the root.
2. There is no rewrite step at the root.

Dresden

1. There is a rewrite step at the root.

$$u \leftrightarrow_G^{n_1} l \leftrightarrow_G r \leftrightarrow_G^{n_2} v$$

G ground!!

for some $l \approx r \in G \cup G^{-1}$.

Since $n_1, n_2 \leq n$ and $l, r \in S$, the induction hypothesis yields

$$(u, l) \in H_m \quad \text{and} \quad (r, v) \in H_m$$

The pair (l, r) is either in $H_0 \subseteq H_m$ or, because of symmetry, in $H_1 \subseteq H_m$.

Transitivity of H_m implies $(u, v) \in H_m$.

2. There is no rewrite step at the root.

$$u = f(u_1, \ldots, u_k), \quad v = f(v_1, \ldots, v_k)$$

and $u_i \leftrightarrow_G^{n_i} v_i$ for all $1 \leq i \leq k$.

Since $n_i \leq n + 1$, $|u_i| < |u|$, and $u_i, v_i \in S$, the induction hypothesis yields

$$(u_i, v_i) \in H_m \quad \text{for all } i$$

It follows by congruence that $(u, v) \in H_{m+1} = H_m$.

Corollary

The word problem for finite sets of ground identities is decidable in polynomial time.

Proof:

We have $s \approx_G t$ iff $(s, t) \in CC_S(G)$ and $CC_S(G)$ can be computed in polynomial time.

By using good data structures, one can achieve a time complexity of $O(n \cdot logn)$.

Example

Let $\Sigma := \{f, a\}$ and $G := \{f^3(a) \approx a, f^2(a) \approx a\}$,

and $s := f(a)$, $t := a$.

Then $S := \{f^i(a) \mid 0 \leq i \leq 3\}$.

H_0 :

$a \approx a$	$a \approx f(a)$	$a \approx f^2(a)$	$a \approx f^3(a)$
$f(a) \approx a$	$f(a) \approx f(a)$	$f(a) \approx f^2(a)$	$f(a) \approx f^3(a)$
$f^2(a) \approx a$	$f^2(a) \approx f(a)$	$f^2(a) \approx f^2(a)$	$f^2(a) \approx f^3(a)$
$f^3(a) \approx a$	$f^3(a) \approx f(a)$	$f^3(a) \approx f^2(a)$	$f^3(a) \approx f^3(a)$

Example

Let $\Sigma := \{f, a\}$ and $G := \{f^3(a) \approx a, f^2(a) \approx a\}$,

and $s := f(a)$, $t := a$.

Then $S := \{f^i(a) \mid 0 \leq i \leq 3\}$.

H_1 :

$\underline{a \approx a}$	$a \approx f(a)$	$\underline{a \approx f^2(a)}$	$\underline{a \approx f^3(a)}$
$f(a) \approx a$	$\underline{f(a) \approx f(a)}$	$f(a) \approx f^2(a)$	$f(a) \approx f^3(a)$
$f^2(a) \approx a$	$f^2(a) \approx f(a)$	$\underline{f^2(a) \approx f^2(a)}$	$f^2(a) \approx f^3(a)$
$f^3(a) \approx a$	$\underline{f^3(a) \approx f(a)}$	$f^3(a) \approx f^2(a)$	$\underline{f^3(a) \approx f^3(a)}$

Example

Let $\Sigma := \{f, a\}$ and $G := \{f^3(a) \approx a, f^2(a) \approx a\}$,
and $s := f(a),\ t := a$.

Then $S := \{f^i(a) \mid 0 \leq i \leq 3\}$.

H_2 :

$$\begin{array}{llll}
a \approx a & \underline{a \approx f(a)} & a \approx f^2(a) & a \approx f^3(a) \\
f(a) \approx a & f(a) \approx f(a) & f(a) \approx f^2(a) & \underline{f(a) \approx f^3(a)} \\
f^2(a) \approx a & f^2(a) \approx f(a) & f^2(a) \approx f^2(a) & \underline{f^2(a) \approx f^3(a)} \\
f^3(a) \approx a & f^3(a) \approx f(a) & \underline{f^3(a) \approx f^2(a)} & f^3(a) \approx f^3(a)
\end{array}$$

Example

Let $\Sigma := \{f, a\}$ and $G := \{f^3(a) \approx a, f^2(a) \approx a\}$,
and $s := f(a),\ t := a$.

Then $S := \{f^i(a) \mid 0 \leq i \leq 3\}$.

H_3 :

$$\begin{array}{llll}
a \approx a & a \approx f(a) & a \approx f^2(a) & a \approx f^3(a) \\
\underline{f(a) \approx a} & f(a) \approx f(a) & \underline{f(a) \approx f^2(a)} & f(a) \approx f^3(a) \\
f^2(a) \approx a & \underline{f^2(a) \approx f(a)} & f^2(a) \approx f^2(a) & f^2(a) \approx f^3(a) \\
f^3(a) \approx a & f^3(a) \approx f(a) & f^3(a) \approx f^2(a) & f^3(a) \approx f^3(a)
\end{array}$$

Thus, $(f(a), a) \in CC_S(G)$, which shows $f(a) \approx_G a$.