4.4 Termination

Termination problems:

Given a finite TRS R and a term t, are all R-reductions starting from t terminating?

Given a finite TRS R, are all R-reductions terminating?

Termination

Proposition 4.12:

Both termination problems for TRSs are undecidable in general.

Consequence:

Decidable criteria for termination are not complete.

Reduction Orderings

Goal:

Given a finite TRS R, show termination of R by looking at finitely many rules $l \to r \in R$, rather than at infinitely many possible replacement steps $s \to_R s'$.

Reduction Orderings

A binary relation $\sqsupset$ over $\mathrm{T}_\Sigma(X)$ is called compatible with Σ-operations, if $s \sqsupset s'$ implies $f(t_1, \ldots, s, \ldots, t_n) \sqsupset f(t_1, \ldots, s', \ldots, t_n)$ for all $f \in \Omega$ and $s, s', t_i \in \mathrm{T}_\Sigma(X)$.

Lemma 4.13:
The relation $\sqsupset$ is compatible with Σ-operations, if and only if $s \sqsupset s'$ implies $t[s]_p \sqsupset t[s']_p$ for all $s, s', t \in \mathrm{T}_\Sigma(X)$ and $p \in \mathrm{pos}(t)$.

Note: compatible with Σ-operations $=$ compatible with contexts.

Reduction Orderings

A binary relation $\sqsupset$ over $\mathrm{T}_\Sigma(X)$ is called stable under substitutions, if $s \sqsupset s'$ implies $s\sigma \sqsupset s'\sigma$ for all $s, s' \in \mathrm{T}_\Sigma(X)$ and substitutions σ.

Reduction Orderings

A binary relation $\sqsupset$ is called a rewrite relation, if it is compatible with Σ-operations and stable under substitutions.

Example: If R is a TRS, then $\rightarrow_R$ is a rewrite relation.

A strict partial ordering over $\mathrm{T}_\Sigma(X)$ that is a rewrite relation is called rewrite ordering.

A well-founded rewrite ordering is called reduction ordering.

Reduction Orderings

Theorem 4.14:

A TRS R terminates if and only if there exists a reduction ordering $\succ$ such that $l \succ r$ for every rule $l \to r \in R$.

Two Different Scenarios

Depending on the application, the TRS whose termination we want to show can be

(i) fixed and known in advance, or

(ii) evolving (e.g., generated by some saturation process).

Methods for case (ii) are also usable for case (i).
Many methods for case (i) are not usable for case (ii).

We will first consider case (ii);
additional techniques for case (i) will be considered later.

The Interpretation Method

Proving termination by interpretation:

Let $\mathcal{A}$ be a Σ-algebra; let $\succ$ be a well-founded strict partial ordering on its universe.

Define the ordering $\succ_{\mathcal{A}}$ over $\mathsf{T}_\Sigma(X)$ by $s \succ_{\mathcal{A}} t$ iff $\mathcal{A}(\beta)(s) \succ \mathcal{A}(\beta)(t)$ for all assignments $\beta : X \to U_{\mathcal{A}}$.

Is $\succ_{\mathcal{A}}$ a reduction ordering?

The Interpretation Method

Lemma 4.15:

$\succ_{\mathcal{A}}$ is stable under substitutions.

The Interpretation Method

A function $f : U_{\mathcal{A}}^n \to U_{\mathcal{A}}$ is called monotone (w. r. t. $\succ$), if $a \succ a'$ implies $f(b_1, \ldots, a, \ldots, b_n) \succ f(b_1, \ldots, a', \ldots, b_n)$ for all $a, a', b_i \in U_{\mathcal{A}}$.

Lemma 4.16:
If the interpretation $f_{\mathcal{A}}$ of every function symbol f is monotone w. r. t. $\succ$, then $\succ_{\mathcal{A}}$ is compatible with Σ-operations.

Theorem 4.17:
If the interpretation $f_{\mathcal{A}}$ of every function symbol f is monotone w. r. t. $\succ$, then $\succ_{\mathcal{A}}$ is a reduction ordering.

Polynomial Orderings

Polynomial orderings:

Instance of the interpretation method:

The carrier set $U_{\mathcal{A}}$ is $\mathbb{N}$ or some subset of $\mathbb{N}$.

To every function symbol f with arity n we associate a polynomial $P_f(X_1, \dots, X_n) \in \mathbb{N}[X_1, \dots, X_n]$ with coefficients in $\mathbb{N}$ and indeterminates $X_1, \dots, X_n$. Then we define $f_{\mathcal{A}}(a_1, \dots, a_n) = P_f(a_1, \dots, a_n)$ for $a_i \in U_{\mathcal{A}}$.

Polynomial Orderings

Requirement 1:

If $a_1, \dots, a_n \in U_{\mathcal{A}}$, then $f_{\mathcal{A}}(a_1, \dots, a_n) \in U_{\mathcal{A}}$. (Otherwise, $\mathcal{A}$ would not be a Σ-algebra.)

Polynomial Orderings

Requirement 2:

$f_{\mathcal{A}}$ must be monotone (w. r. t. $\succ$).

From now on:

$U_{\mathcal{A}} = \{\, n \in \mathbb{N} \mid n \geq 1 \,\}$.

If arity$(f) = 0$, then P_f is a constant ≥ 1.

If arity$(f) = n \geq 1$, then P_f is a polynomial $P(X_1, \dots, X_n)$, such that every X_i occurs in some monomial with exponent at least 1 and non-zero coefficient.

$\Rightarrow$ Requirements 1 and 2 are satisfied.

Polynomial Orderings

The mapping from function symbols to polynomials can be extended to terms: A term t containing the variables $x_1, \dots, x_n$ yields a polynomial P_t with indeterminates $X_1, \dots, X_n$ (where X_i corresponds to $\beta(x_i)$).

Example:

$$\Omega = \{b/0,\ f/1,\ g/3\}$$
$$P_b = 3, \quad P_f(X_1) = X_1^2, \quad P_g(X_1, X_2, X_3) = X_1 + X_2 X_3.$$

Let $t = g(f(b), f(x), y)$, then $P_t(X, Y) = 9 + X^2 Y$.

Polynomial Orderings

If P, Q are polynomials in $\mathbb{N}[X_1, \ldots, X_n]$, we write $P > Q$ if $P(a_1, \ldots, a_n) > Q(a_1, \ldots, a_n)$ for all $a_1, \ldots, a_n \in U_\mathcal{A}$.

Clearly, $l \succ_\mathcal{A} r$ iff $P_l > P_r$ iff $P_l - P_r > 0$.

Question: Can we check $P_l - P_r > 0$ automatically?

Polynomial Orderings

Hilbert's 10th Problem:

> Given a polynomial $P \in \mathbb{Z}[X_1, \ldots, X_n]$ with integer coefficients, is $P = 0$ for some n-tuple of natural numbers?

Theorem 4.18:
Hilbert's 10th Problem is undecidable.

Proposition 4.19:
Given a polynomial interpretation and two terms l, r, it is undecidable whether $P_l > P_r$.

Proof:
By reduction of Hilbert's 10th Problem. $\square$

Polynomial Orderings

One easy case:

If we restrict to linear polynomials, deciding whether $P_l - P_r > 0$ is trivial:

$\sum k_i a_i + k > 0$ for all $a_1, \dots, a_n \geq 1$ if and only if

$k_i \geq 0$ for all $i \in \{1, \dots, n\}$,

and $\sum k_i + k > 0$

Polynomial Orderings

Another possible solution:

Test whether $P_l(a_1, \ldots, a_n) > P_r(a_1, \ldots, a_n)$ for all $a_1, \ldots, a_n \in \{\, x \in \mathbb{R} \mid x \geq 1 \,\}$.

This is decidable (but hard). Since $U_{\mathcal{A}} \subseteq \{\, x \in \mathbb{R} \mid x \geq 1 \,\}$, it implies $P_l > P_r$.

Alternatively:

Use fast overapproximations.

Simplification Orderings

The proper subterm ordering $\rhd$ is defined by $s \rhd t$ if and only if $s|_p = t$ for some position $p \neq \varepsilon$ of s.

Simplification Orderings

A rewrite ordering $\succ$ over $\mathrm{T}_\Sigma(X)$ is called simplification ordering, if it has the subterm property: $s \rhd t$ implies $s \succ t$ for all $s, t \in \mathrm{T}_\Sigma(X)$.

Example:

Let R_{emb} be the rewrite system $R_{\text{emb}} = \{\, f(x_1, \dots, x_n) \to x_i \mid f \in \Omega,\ 1 \le i \le n = \text{arity}(f) \,\}$.

Define $\rhd_{\text{emb}} = \to^{+}_{R_{\text{emb}}}$ and $\unrhd_{\text{emb}} = \to^{*}_{R_{\text{emb}}}$ ("homeomorphic embedding relation").

$\rhd_{\text{emb}}$ is a simplification ordering.

Simplification Orderings

Lemma 4.20:

If $\succ$ is a simplification ordering, then $s \rhd_{\text{emb}} t$ implies $s \succ t$ and $s \unrhd_{\text{emb}} t$ implies $s \succeq t$.

Simplification Orderings

Goal:

Show that every simplification ordering is well-founded (and therefore a reduction ordering).

Note: This works only for finite signatures!

To fix this for infinite signatures, the definition of simplification orderings and the definition of embedding have to be modified.

Simplification Orderings

Theorem 4.21 ("Kruskal's Theorem"):

Let Σ be a finite signature, let X be a finite set of variables. Then for every infinite sequence $t_1, t_2, t_3, \ldots$ there are indices $j > i$ such that $t_j \trianglerighteq_{\text{emb}} t_i$. ($\trianglerighteq_{\text{emb}}$ is called a well-partial-ordering (wpo).)

Proof:

See Baader and Nipkow, page 113–115. $\square$

Simplification Orderings

Theorem 4.22 (Dershowitz):

If Σ is a finite signature, then every simplification ordering $\succ$ on $\mathrm{T}_\Sigma(X)$ is well-founded (and therefore a reduction ordering).

Simplification Orderings

There are reduction orderings that are not simplification orderings and terminating TRSs that are not contained in any simplification ordering.

Example:

Let $R = \{f(f(x)) \to f(g(f(x)))\}$.

R terminates and $\to_R^+$ is therefore a reduction ordering.

Assume that $\to_R$ were contained in a simplification ordering $\succ$.
Then $f(f(x)) \to_R f(g(f(x)))$ implies $f(f(x)) \succ f(g(f(x)))$,
and $f(g(f(x))) \trianglerighteq_{\text{emb}} f(f(x))$ implies $f(g(f(x))) \succeq f(f(x))$,
hence $f(f(x)) \succ f(f(x))$.

Path Orderings

Let $\Sigma = (\Omega, \Pi)$ be a finite signature, let $\succ$ be a strict partial ordering ("precedence") on Ω.

The lexicographic path ordering $\succ_{\mathrm{lpo}}$ on $\mathrm{T}_\Sigma(X)$ induced by $\succ$ is defined by: $s \succ_{\mathrm{lpo}} t$ iff

(1) $t \in \mathrm{vars}(s)$ and $t \neq s$, or

(2) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and

- (a) $s_i \succeq_{\mathrm{lpo}} t$ for some i, or
- (b) $f \succ g$ and $s \succ_{\mathrm{lpo}} t_j$ for all j, or
- (c) $f = g$, $s \succ_{\mathrm{lpo}} t_j$ for all j, and $(s_1, \ldots, s_m)\ (\succ_{\mathrm{lpo}})_{\mathrm{lex}}\ (t_1, \ldots, t_n)$.

Path Orderings

Lemma 4.23:
$s \succ_{\mathrm{lpo}} t$ implies $\mathrm{vars}(s) \supseteq \mathrm{vars}(t)$.

Theorem 4.24:
$\succ_{\mathrm{lpo}}$ is a simplification ordering on $\mathrm{T}_\Sigma(X)$.

Theorem 4.25:
If the precedence $\succ$ is total, then the lexicographic path ordering $\succ_{\mathrm{lpo}}$ is total on ground terms, i. e., for all $s, t \in \mathrm{T}_\Sigma(\emptyset)$:
$s \succ_{\mathrm{lpo}} t \lor t \succ_{\mathrm{lpo}} s \lor s = t$.

Path Orderings

Recapitulation:

Let $\Sigma = (\Omega, \Pi)$ be a finite signature, let $\succ$ be a strict partial ordering ("precedence") on Ω. The lexicographic path ordering $\succ_{\mathrm{lpo}}$ on $\mathrm{T}_\Sigma(X)$ induced by $\succ$ is defined by: $s \succ_{\mathrm{lpo}} t$ iff

(1) $t \in \mathrm{vars}(s)$ and $t \neq s$, or

(2) $s = f(s_1, \dots, s_m)$, $t = g(t_1, \dots, t_n)$, and

- (a) $s_i \succeq_{\mathrm{lpo}} t$ for some i, or
- (b) $f \succ g$ and $s \succ_{\mathrm{lpo}} t_j$ for all j, or
- (c) $f = g$, $s \succ_{\mathrm{lpo}} t_j$ for all j, and $(s_1, \dots, s_m)$ $(\succ_{\mathrm{lpo}})_{\mathrm{lex}}$ $(t_1, \dots, t_n)$.

Path Orderings

There are several possibilities to compare subterms in (2)(c):

- compare list of subterms lexicographically left-to-right ("lexicographic path ordering (lpo)", Kamin and Lévy)
- compare list of subterms lexicographically right-to-left (or according to some permutation π)
- compare multiset of subterms using the multiset extension ("multiset path ordering (mpo)", Dershowitz)
- to each function symbol f with arity$(n) \geq 1$ associate a status $\in \{ \mathit{mul} \} \cup \{ \mathit{lex}_\pi \mid \pi : \{1, \dots, n\} \to \{1, \dots, n\} \}$ and compare according to that status ("recursive path ordering (rpo) with status")

The Knuth-Bendix Ordering

Let $\Sigma = (\Omega, \Pi)$ be a finite signature, let $\succ$ be a strict partial ordering ("precedence") on Ω, let $w : \Omega \cup X \to \mathbb{R}_0^+$ be a weight function, such that the following admissibility conditions are satisfied:

$w(x) = w_0 \in \mathbb{R}^+$ for all variables $x \in X$; $w(c) \geq w_0$ for all constants $c \in \Omega$.

If $w(f) = 0$ for some $f \in \Omega$ with $\text{arity}(f) = 1$, then $f \succeq g$ for all $g \in \Omega$.

The weight function w can be extended to terms as follows:

$$w(t) = \sum_{x \in \text{vars}(t)} w(x) \cdot \#(x, t) + \sum_{f \in \Omega} w(f) \cdot \#(f, t).$$

The Knuth-Bendix Ordering

The Knuth-Bendix ordering $\succ_{\mathrm{kbo}}$ on $\mathrm{T}_\Sigma(X)$ induced by $\succ$ and w is defined by: $s \succ_{\mathrm{kbo}} t$ iff

(1) $\#(x, s) \geq \#(x, t)$ for all variables x and $w(s) > w(t)$, or

(2) $\#(x, s) \geq \#(x, t)$ for all variables x, $w(s) = w(t)$, and

  (a) $t = x$, $s = f^n(x)$ for some $n \geq 1$, or

  (b) $s = f(s_1, \ldots, s_m)$, $t = g(t_1, \ldots, t_n)$, and $f \succ g$, or

  (c) $s = f(s_1, \ldots, s_m)$, $t = f(t_1, \ldots, t_m)$, and $(s_1, \ldots, s_m)$ $(\succ_{\mathrm{kbo}})_{\mathrm{lex}}$ $(t_1, \ldots, t_m)$.

The Knuth-Bendix Ordering

Theorem 4.26:

The Knuth-Bendix ordering induced by $\succ$ and w is a simplification ordering on $\mathrm{T}_{\Sigma}(X)$.

Proof:

Baader and Nipkow, pages 125–129. $\square$

Remark

If $\Pi \neq \emptyset$, then all the term orderings described in this section can also be used to compare non-equational atoms by treating predicate symbols like function symbols.

Defining a weight $w(f) = 0$ for some unary function symbol f was in particular introduced for the application of KBO to equational systems defining groups.

4.5 Knuth-Bendix Completion

Completion:

Goal: Given a set E of equations, transform E into an equivalent convergent set R of rewrite rules.
(If R is finite: decision procedure for E.)

How to ensure termination?

Fix a reduction ordering $\succ$ and construct R in such a way that $\rightarrow_R \subseteq \succ$ (i. e., $l \succ r$ for every $l \rightarrow r \in R$).

How to ensure confluence?

Check that all critical pairs are joinable.

Knuth-Bendix Completion: Inference Rules

The completion procedure is itself presented as a set of rewrite rules working on a pair of equations E and rules R: $(E_0; R_0) \Rightarrow (E_1; R_1) \Rightarrow (E_2; R_2) \Rightarrow \ldots$

At the beginning, $E = E_0$ is the input set and $R = R_0$ is empty. At the end, E should be empty; then R is the result.

For each step $(E; R) \Rightarrow (E'; R')$, the equational theories of $E \cup R$ and $E' \cup R'$ agree: $\approx_{E \cup R} = \approx_{E' \cup R'}$.

Knuth-Bendix Completion: Inference Rules

Notations:

The formula $s \mathrel{\dot{\approx}} t$ denotes either $s \approx t$ or $t \approx s$.

$\mathrm{CP}(R)$ denotes the set of all critical pairs between rules in R.

Knuth-Bendix Completion: Inference Rules

Orient

$$(E \uplus \{s \mathrel{\dot{\approx}} t\}; R) \quad \Rightarrow_{KBC} \quad (E; R \cup \{s \to t\})$$

if $s \succ t$

Note: There are equations $s \approx t$ that cannot be oriented, i. e., neither $s \succ t$ nor $t \succ s$.

Knuth-Bendix Completion: Inference Rules

Trivial equations cannot be oriented – but we don't need them anyway:

Delete

$$(E \uplus \{s \approx s\}; R) \quad \Rightarrow_{KBC} \quad (E; R)$$

Knuth-Bendix Completion: Inference Rules

Critical pairs between rules in R are turned into additional equations:

Deduce

$$(E; R) \quad \Rightarrow_{KBC} \quad (E \cup \{s \approx t\}; R)$$

if $\langle s, t \rangle \in \mathrm{CP}(R)$

Note: If $\langle s, t \rangle \in \mathrm{CP}(R)$ then $s \;{}_R\!\!\leftarrow u \rightarrow_R t$ and hence $R \models s \approx t$.

Knuth-Bendix Completion: Inference Rules

The following inference rules are not absolutely necessary, but very useful (e. g., to get rid of joinable critical pairs and to deal with equations that cannot be oriented):

Simplify-Eq

$$(E \uplus \{s \mathrel{\dot{\approx}} t\}; R) \quad \Rightarrow_{KBC} \quad (E \cup \{u \approx t\}; R)$$

if $s \rightarrow_R u$

Knuth-Bendix Completion: Inference Rules

Simplification of the right-hand side of a rule is unproblematic.

R-Simplify-Rule

$$(E; R \uplus \{s \to t\}) \quad \Rightarrow_{KBC} \quad (E; R \cup \{s \to u\})$$

if $t \to_R u$

Simplification of the left-hand side may influence orientability and orientation. Therefore, it yields an *equation*:

L-Simplify-Rule

$$(E; R \uplus \{s \to t\}) \quad \Rightarrow_{KBC} \quad (E \cup \{u \approx t\}; R$$

if $s \to_R u$ using a rule $l \to r \in R$ such that $s \sqsupset l$ (see next slide).

Knuth-Bendix Completion: Inference Rules

For technical reasons, the lhs of $s \rightarrow t$ may only be simplified using a rule $l \rightarrow r$, if $l \rightarrow r$ *cannot* be simplified using $s \rightarrow t$, that is, if $s \sqsupset l$, where the encompassment quasi-ordering $\sqsupseteq$ is defined by

$$s \sqsupseteq l \quad \text{if} \quad s|_p = l\sigma \text{ for some } p \text{ and } \sigma$$

and $\sqsupset = \sqsupseteq \setminus \sqsubseteq$ is the strict part of $\sqsupseteq$.

Lemma 4.27:
$\sqsupset$ is a well-founded strict partial ordering.

Knuth-Bendix Completion: Inference Rules

Lemma 4.28:
If $E, R \vdash E', R'$, then $\approx_{E \cup R} = \approx_{E' \cup R'}$.

Lemma 4.29:
If $E, R \vdash E', R'$ and $\rightarrow_R \subseteq \succ$, then $\rightarrow_{R'} \subseteq \succ$.

Knuth-Bendix Completion: Correctness Proof

If we run the completion procedure on a set E of equations, different things can happen:

(1) We reach a state where no more inference rules are applicable and E is not empty.
$\Rightarrow$ Failure (try again with another ordering?)

(2) We reach a state where E is empty and all critical pairs between the rules in the current R have been checked.

(3) The procedure runs forever.

In order to treat these cases simultaneously, we need some definitions.

Knuth-Bendix Completion: Correctness Proof

A (finite or infinite sequence) $(E_0; R_0) \Rightarrow_{KBC} (E_1; R_1) \Rightarrow_{KBC} (E_2; R_2) \Rightarrow_{KBC} \ldots$ with $R_0 = \emptyset$ is called a run of the completion procedure with input E_0 and $\succ$.

For a run, $E_\infty = \bigcup_{i \geq 0} E_i$ and $R_\infty = \bigcup_{i \geq 0} R_i$.

The sets of persistent equations or rules of the run are $E_* = \bigcup_{i \geq 0} \bigcap_{j \geq i} E_j$ and $R_* = \bigcup_{i \geq 0} \bigcap_{j \geq i} R_j$.

Note: If the run is finite and ends with E_n, R_n, then $E_* = E_n$ and $R_* = R_n$.

Knuth-Bendix Completion: Correctness Proof

A run is called fair, if $CP(R_*) \subseteq E_\infty$ (i. e., if every critical pair between persisting rules is computed at some step of the derivation).

Goal:

Show: If a run is fair and E_* is empty, then R_* is convergent and equivalent to E_0.

In particular: If a run is fair and E_* is empty, then $\approx_{E_0} = \approx_{E_\infty \cup R_\infty} = \leftrightarrow^*_{E_\infty \cup R_\infty} = \downarrow_{R_*}$.