Corollary 4.1.6 (Convergence of E)**.** If a set of equations E is convergent then $s \approx_E t$ if and only if $s \leftrightarrow^* t$ if and only if $s \downarrow_E = t \downarrow_E$.

Corollary 4.1.7 (Decidability of $\approx_E$)**.** If a set of equations E is finite and convergent then $\approx_E$ is decidable.

The above Lemma 4.1.5 shows equivalence of the syntactically defined relations $\leftrightarrow^*_E$ and $\Rightarrow^*_E$. What is missing, in analogy to Herbrand's theorem for first-order logic without equality Theorem 3.5.5, is a semantic characterization of the relations by a particular algebra.

Definition 4.1.8 (Quotient Algebra)**.** For sets of unit equations this is a *quotient algebra*: Let X be a set of variables. For $t \in T(\Sigma, \mathcal{X})$ let $[t] = \{t' \in T(\Sigma, \mathcal{X})) \mid E \Rightarrow^*_{\mathrm{E}} t \approx t'\}$ be the *congruence class* of t. Define a Σ-algebra $\mathcal{I}_E$, called the *quotient algebra*, technically $T(\Sigma, \mathcal{X})/E$, as follows: $S^{\mathcal{I}_E} = \{[t] \mid t \in T_S(\Sigma, \mathcal{X})\}$ for all sorts S and $f^{\mathcal{I}_E}([t_1], \ldots, [t_n]) = [f(t_1, \ldots, t_n)]$ for $f : \mathrm{sort}(t_1) \times \ldots \times \mathrm{sort}(t_n) \to T \in \Omega$ for some sort T.

Lemma 4.1.9 ($\mathcal{I}_E$ is an E-algebra)**.** $\mathcal{I}_E = T(\Sigma, \mathcal{X})/E$ is an E-algebra.

Proof. Firstly, all functions $f^{\mathcal{I}_E}$ are well-defined: if $[t_i] = [t'_i]$, then $[f(t_1, \ldots, t_n)] = [f(t'_1, \ldots, t'_n)]$. This follows directly from the Congruence rule for $\Rightarrow^*$.

Secondly, let $\forall x_1 \ldots x_n (s \approx t)$ be an equation in E. Let β be an arbitrary assignment. It has to be shown that $\mathcal{I}_E(\beta)(\forall \vec{x}(s \approx t)) = 1$, or equivalently, that $\mathcal{I}_E(\gamma)(s) = \mathcal{I}_E(\gamma)(t)$ for all $\gamma = \beta[x_i \mapsto [t_i] \mid 1 \le i \le n]$ with $[t_i] \in \mathrm{sort}(x_i)^{\mathcal{I}_E}$. Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, with $t_i \in T_{\mathrm{sort}(x_i)}(\Sigma, \mathcal{X})$, then $s\sigma \in \mathcal{I}_E(\gamma)(s)$ and $t\sigma \in \mathcal{I}_E(\gamma)(t)$. By the Instance rule, $E \Rightarrow^* s\sigma \approx t\sigma$ is derivable, hence $\mathcal{I}_E(\gamma)(s) = [s\sigma] = [t\sigma] = \mathcal{I}_E(\gamma)(t)$. □

Lemma 4.1.10 ($\Rightarrow_E$ is complete)**.** Let $\mathcal{X}$ be a countably infinite set of variables; let $s, t \in T_S(\Sigma, \mathcal{X})$. If $\mathcal{I}_E \models \forall \vec{x}(s \approx t)$, then $E \Rightarrow^*_E s \approx t$ is derivable.

Proof. Assume that $\mathcal{I}_E \models \forall \vec{x}(s \approx t)$, i.e., $\mathcal{I}_E(\beta)(\forall \vec{x}(s \approx t)) = 1$. Consequently, $\mathcal{I}_E(\gamma)(s) = \mathcal{I}_E(\gamma)(t)$ for all $\gamma = \beta[\, x_i \mapsto [t_i] \mid 1 \le i \le n\,]$ with $[t_i] \in \mathrm{sort}(x_i)^{\mathcal{I}_E}$. Choose $t_i = x_i$, then $[s] = \mathcal{I}_E(\gamma)(s) = \mathcal{I}_E(\gamma)(t) = [t]$, so $E \Rightarrow^* s \approx t$ is derivable by definition of $\mathcal{I}_E$. □

Theorem 4.1.11 (Birkhoff's Theorem)**.** Let $\mathcal{X}$ be a countably infinite set of variables, let E be a set of (universally quantified) equations. Then the following properties are equivalent for all $s, t \in T_S(\Sigma, \mathcal{X})$:

1. $s \leftrightarrow^*_E t$.
2. $E \Rightarrow^*_E s \approx t$ is derivable.
3. $s \approx_E t$, i.e., $E \models \forall \vec{x}(s \approx t)$.
4. $\mathcal{I}_E \models \forall \vec{x}(s \approx t)$.

Proof. (1.)⇔(2.): Lemma 4.1.5.

(2.)⇒(3.): By induction on the size of the derivation for $E \Rightarrow^* s \approx t$.

(3.)⇒(4.): Obvious, since $\mathcal{I}_E = T(\Sigma, \mathcal{X})/E$ is an E-algebra.

(4.)⇒(2.): Lemma 4.1.10. □

Universal Algebra

$T(\Sigma, \mathcal{X})/E = T(\Sigma, \mathcal{X})/{\approx_E} = T(\Sigma, \mathcal{X})/{\leftrightarrow_E^*}$ is called the *free E-algebra* with generating set $\mathcal{X}/{\approx_E} = \{[x] \mid x \in \mathcal{X}\}$: Every mapping $\phi : \mathcal{X}/{\approx_E} \to \mathcal{B}$ for some E-algebra $\mathcal{B}$ can be extended to a homomorphism $\hat{\phi} : T(\Sigma, \mathcal{X})/E \to \mathcal{B}$.

$T(\Sigma, \emptyset)/E = T(\Sigma, \emptyset)/{\approx_E} = T(\Sigma, \emptyset)/{\leftrightarrow_E^*}$ is called the *initial E-algebra.*

$\approx_E = \{(s, t) \mid E \models s \approx t\}$ is called the *equational theory* of E.

$\approx_E^I = \{(s, t) \mid T(\Sigma, \emptyset)/E \models s \approx t\}$ is called the *inductive theory* of E.

Example 4.1.12. Let $E = \{\forall x(x + 0 \approx x),\ \forall x \forall y(x + s(y) \approx s(x + y))\}$. Then $x + y \approx_E^I y + x$, but $x + y \not\approx_E y + x$.

4.2 Critical Pairs

By Theorem 4.1.11 the semantics of E and $\leftrightarrow_E^*$ coincide. In order to decide $\leftrightarrow_E^*$ we need to turn $\to_E^*$ in a confluent and terminating relation. If $\leftrightarrow_E^*$ is terminating then confluence is equivalent to local confluence, see Newman's Lemma, Lemma 1.6.6. Local confluence is the following problem for TRS: if $t_1 \ {}_E{\leftarrow}\ t_0 \to_E t_2$, does there exist a term s so that $t_1 \to_E^* s \ {}_E^*{\leftarrow}\ t_2$? If the two rewrite steps happen in different subtrees (disjoint redexes) then a repetition of the respective other step yields the common term s. If the two rewrite steps happen below each other (overlap at or below a variable position) again a repetition of the respective other step yields the common term s. If the left-hand sides of the two rules overlap at a non-variable position there is no obvious way to generate s.

More technically two rewrite rules $l_1 \to r_1$ and $l_2 \to r_2$ overlap if there exist some non-variable subterm $l_1|_p$ such that l_2 and $l_1|_p$ have a common instance $(l_1|_p)\sigma_1 = l_2\sigma_2$. If the two rewrite rules do not have common variables, then only a single substitution is necessary, the mgu σ of $(l_1|_p)$ and l_2.

Definition 4.2.1 (Critical Pair)**.** Let $l_i \to r_i$ $(i = 1, 2)$ be two rewrite rules in a TRS R without common variables, i.e., $\mathrm{vars}(l_1) \cap \mathrm{vars}(l_2) = \emptyset$. Let $p \in \mathrm{pos}(l_1)$ be a position so 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 $r_1\sigma \downarrow_R (l_1\sigma)[r_2\sigma]_p$.

Recall that $\mathrm{vars}(l_i) \supseteq \mathrm{vars}(r_i)$ for the two rewrite rules by Definition 4.1.1. Furthermore, the definition of the rule includes overalaps of a rule with itself. Such overlaps on top-level are always joinable.

Theorem 4.2.2 ("Critical Pair Theorem")**.** A TRS R is locally confluent iff all its critical pairs are joinable.

Proof. ($\Rightarrow$) Obvious, since joinability of a critical pair is a special case of local confluence.

($\Leftarrow$) 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 $i = 1, 2$. The two rules are variable disjoint, hence $s|_{p_i} = l_i\sigma$ and $t_i = s[r_i\sigma]_{p_i}$. There are two cases to be considered:

1. Either p_1 and p_2 are in disjoint subtrees ($p_1 \mid\mid p_2$) or
2. one is a prefix of the other (w.l.o.g., $p_1 \leq p_2$).

Case 1: $p_1 \mid\mid p_2$. Then $s = s[l_1\sigma]_{p_1}[l_2\sigma]_{p_2}$, and therefore $t_1 = s[r_1\sigma]_{p_1}[l_2\sigma]_{p_2}$ and $t_2 = s[l_1\sigma]_{p_1}[r_2\sigma]_{p_2}$. Let $t_0 = s[r_1\sigma]_{p_1}[r_2\sigma]_{p_2}$. Then clearly $t_1 \to_R t_0$ using $l_2 \to r_2$ and $t_2 \to_R t_0$ using $l_1 \to r_1$.
Case 2: $p_1 \leq p_2$.
Case 2.1: $p_2 = p_1q_1q_2$, 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 $p_1q'q_2$, 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_1qq_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 σ', where $\sigma' = \sigma[x \mapsto (x\sigma)[r_2\sigma]_{q_2}]$.
Case 2.2: $p_2 = p_1p$, where p is a non-variable position of l_1. Then $s|_{p_2} = l_2\sigma$ and $s|_{p_2} = (s|_{p_1})|_p = (l_1\sigma)|_p = (l_1|_p)\sigma$, so σ is a unifier of l_2 and $l_1|_p$.Let σ' be the mgu of l_2 and $l_1|_p$, then $\sigma = \tau \circ \sigma'$ and $\langle r_1\sigma', (l_1\sigma')[r_2\sigma']_p\rangle$ is a critical pair. By assumption, it is joinable, so $r_1\sigma' \to_R^* v \leftarrow_R^* (l_1\sigma')[r_2\sigma']_p$. Consequently, $t_1 = s[r_1\sigma]_{p_1} = s[r_1\sigma'\tau]_{p_1} \to_R^* s[v\tau]_{p_1}$ and $t_2 = s[r_2\sigma]_{p_2} = s[(l_1\sigma)[r_2\sigma]_p]_{p_1} = s[(l_1\sigma'\tau)[r_2\sigma'\tau]_p]_{p_1} = s[((l_1\sigma')[r_2\sigma']_p)\tau]_{p_1} \to_R^* s[v\tau]_{p_1}$. □

Please note that critical pairs between a rule and (a renamed variant of) itself must be considered, except if the overlap is at the root, i.e., $p = \epsilon$, because this critical pair always joins.

Corollary 4.2.3. A terminating TRS R is confluent if and only if all its critical pairs are joinable.

Proof. By the Theorem 4.2.2 and because every locally confluent and terminating relation $\to$ is confluent, Newman's Lemma, Lemma 1.6.6. □

Corollary 4.2.4. For a finite terminating TRS, confluence is decidable.

Proof. For every pair of rules and every non-variable position in the first rule there is at most one critical pair $\langle u_1, u_2\rangle$. Reduce every u_i to some normal form u_i'. If $u_1' = u_2'$ for every critical pair, then R is confluent, otherwise there is some non-confluent situation $u_1' \;{}_R^*\!\leftarrow u_1 \leftarrow_R s \to_R u_2 \to_R^* u_2'$. □

4.3 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?

Proposition 4.3.1. Both termination problems for TRSs are undecidable in general.

Proof. Encode Turing machines (TM) using rewrite rules and reduce the (uniform) halting problems for TMs to the termination problems for TRSs. □

Consequence: Decidable criteria for termination are not complete.

Two Different Scenarios

Depending on the application, the TRS whose termination has to be shown can be

1. fixed and known in advance, or
2. evolving (e.g., generated by some saturation process).

Methods for case 2. are also usable for case 1.. Many methods for case 1. are not usable for case 2..

First consider case 2., additional techniques for case 1. will be considered later.

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 $s \to_R s'$.

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

Lemma 4.3.2. 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 $s, s', t \in T(\Sigma, \mathcal{X})$ and $p \in pos(t)$.

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

A binary relation $\sqsupset$ over $T(\Sigma, \mathcal{X})$ is called *stable under substitutions,* if $s \sqsupset s'$ implies $s\sigma \sqsupset s'\sigma$ for all $s, s' \in T(\Sigma, \mathcal{X})$ and substitutions σ. A binary relation $\sqsupset$ is called a *rewrite relation,* if it is compatible with Σ-operations and stable under substitutions.

Example 4.3.3. If R is a TRS, then $\to_R$ is a rewrite relation.

A strict partial ordering over $T(\Sigma, \mathcal{X})$ that is a rewrite relation is called *rewrite ordering.* A well-founded rewrite ordering is called *reduction ordering.*

Theorem 4.3.4. A TRS R terminates if and only if there exists a reduction ordering $\succ$ so that $l \succ r$ for every rule $l \to r \in R$.

Proof. ($\Rightarrow$): $s \rightarrow_R s'$ if and only if $s = t[l\sigma]_p$, $s' = t[r\sigma]_p$. If $l \succ r$, then $l\sigma \succ r\sigma$ and therefore $t[l\sigma]_p \succ t[r\sigma]_p$. This implies $\rightarrow_R \subseteq \succ$. Since $\succ$ is a well-founded ordering, $\rightarrow_R$ is terminating.

($\Leftarrow$): Define $\succ = \rightarrow_R^+$. If $\rightarrow_R$ is terminating, then $\succ$ is a reduction ordering. □

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 $T(\Sigma, \mathcal{X})$ by $s \succ_{\mathcal{A}} t$ iff $\mathcal{A}(\beta)(s) \succ \mathcal{A}(\beta)(t)$ for all assignments $\beta : \mathcal{X} \rightarrow U_{\mathcal{A}}$. Is $\succ_{\mathcal{A}}$ a reduction ordering?

Lemma 4.3.5. $\succ_{\mathcal{A}}$ is stable under substitutions.

Proof. Let $s \succ_{\mathcal{A}} s'$, that is, $\mathcal{A}(\beta)(s) \succ \mathcal{A}(\beta)(s')$ for all assignments $\beta : \mathcal{X} \rightarrow U_{\mathcal{A}}$. Let σ be a substitution. It has to be shown that $\mathcal{A}(\gamma)(s\sigma) \succ \mathcal{A}(\gamma)(s'\sigma)$ for all assignments $\gamma : \mathcal{X} \rightarrow U_{\mathcal{A}}$. Choose $\beta = \gamma \circ \sigma$, then by the substitution lemma, $\mathcal{A}(\gamma)(s\sigma) = \mathcal{A}(\beta)(s) \succ \mathcal{A}(\beta)(s') = \mathcal{A}(\gamma)(s'\sigma)$. Therefore $s\sigma \succ_{\mathcal{A}} s'\sigma$. □

A function $f : U_{\mathcal{A}}^n \rightarrow U_{\mathcal{A}}$ is called *monotone* with respect to $\succ$, 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 4.3.6. 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.

Proof. Let $s \succ s'$, that is, $\mathcal{A}(\beta)(s) \succ \mathcal{A}(\beta)(s')$ for all $\beta : \mathcal{X} \rightarrow U_{\mathcal{A}}$. Let $\beta : \mathcal{X} \rightarrow U_{\mathcal{A}}$ be an arbitrary assignment. Then

$$\begin{aligned}\mathcal{A}(\beta)(f(t_1, \ldots, s, \ldots, t_n)) &= f_{\mathcal{A}}(\mathcal{A}(\beta)(t_1), \ldots, \mathcal{A}(\beta)(s), \ldots, \mathcal{A}(\beta)(t_n)) \\ &\succ f_{\mathcal{A}}(\mathcal{A}(\beta)(t_1), \ldots, \mathcal{A}(\beta)(s'), \ldots, \mathcal{A}(\beta)(t_n)) \\ &= \mathcal{A}(\beta)(f(t_1, \ldots, s', \ldots, t_n))\end{aligned}$$

Therefore $f(t_1, \ldots, s, \ldots, t_n) \succ_{\mathcal{A}} f(t_1, \ldots, s', \ldots, t_n)$. □

Theorem 4.3.7. 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.

Proof. By the previous two lemmas, $\succ_{\mathcal{A}}$ is a rewrite relation. If there were an infinite chain $s_1 \succ_{\mathcal{A}} s_2 \succ_{\mathcal{A}} \ldots$, then it would correspond to an infinite chain $\mathcal{A}(\beta)(s_1) \succ \mathcal{A}(\beta)(s_2) \succ \ldots$ (with β chosen arbitrarily). Thus $\succ_{\mathcal{A}}$ is well-founded. Irreflexivity and transitivity are proved similarly. □

Polynomial Orderings

Polynomial orderings:

1. Instance of the interpretation method
2. The carrier set $U_{\mathcal{A}}$ is $\mathbb{N}$ or some subset of $\mathbb{N}$.
3. To every function symbol f with arity n a polynomial $P_f(X_1, \ldots, X_n) \in \mathbb{N}[X_1, \ldots, X_n]$ with coefficients in $\mathbb{N}$ is associated and indeterminates $X_1, \ldots, X_n$. Then define $f_{\mathcal{A}}(a_1, \ldots, a_n) = P_f(a_1, \ldots, a_n)$ for $a_i \in U_{\mathcal{A}}$.

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.)

Requirement 2: $f_\mathcal{A}$ must be monotone (w.r.t.$\succ$).

From now on:

1. $U_\mathcal{A} = \{n \in \mathbb{N} \mid n \geq 1\}$.
2. If $arity(f) = 0$, then P_f is a constant ≥ 1.
3. If $arity(f) = n \geq 1$, then P_f is a polynomial $P(X_1, \dots, X_n)$, so that every X_i occurs in some monomial with exponent at least 1 and non-zero coefficient. $\Rightarrow$ Requirements 1 and 2 are satisfied.

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 4.3.8. Let $\Omega = \{b/0, f/1, g/3\}, P_b = 3, P_f(X_1) = X_1^2, P_g(X_1, X_2, X_3) = X_1 + X_2 X_3$ and $t = g(f(b), f(x), y)$, then $P_t(X, Y) = 9 + X^2 Y$.

If P, Q are polynomials in $\mathbb{N}[X_1, \dots, X_n]$, $P > Q$ is written if $P(a_1, \dots, a_n) > Q(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in U_\mathcal{A}$. Clearly, $l \succ_\mathcal{A} r$ iff $P_l > P_r$ iff $P_l - P_r > 0$. The question is whether $P_l - P_r > 0$ can be checked automatically?

Hilbert's 10th Problem: Given a polynomial $P \in \mathbb{Z}[X_1, \dots, X_n]$ with integer coefficients, is $P = 0$ for some n-tuple of natural numbers?

Theorem 4.3.9. Hilbert's 10th Problem is undecidable.

Proposition 4.3.10. 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. □

One easy case: If restricted 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 $\sum k_i + k > 0$.

Another possible solution: Test whether $P_l(a_1, \dots, a_n) > P_r(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in \{x \in \mathbb{R} \mid x \geq 1\}$. This is decidable (but hard). Since $U_\mathcal{A} \subseteq \{x \in \mathbb{R} \mid x \geq 1\}$ this 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 $p \neq \epsilon$ of s.

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

Example 4.3.11. Let R_{emb} be the rewrite system $R_{emb} = \{ f(x_1, \dots, x_n) \to x_i \mid f \in \Omega, 1 \leq i \leq n = f/n\}$. Define $\rhd_{emb} = \to^+_{R_{emb}}$ and $\trianglerighteq = \to^*_{R_{emb}}$ *("homeomorphic embedding relation")* and $\rhd_{emb}$ is a simplification ordering.

Lemma 4.3.12. If $\succ$ is a simplification ordering then $s \rhd_{emb} t$ implies $s \succ t$ and $s \trianglerighteq t$ implies $s \succeq t$.

Proof. Since $\succ$ is transitive and $\succeq$ is transitive and reflexive, it suffices to show that $s \rightarrow_{R_{emb}} t$ implies $s \succ t$. By definition, $s \rightarrow_{R_{emb}} t$ if and only if $s = s[l\sigma]$ and $t = s[r\sigma]$ for some rule $l \rightarrow r \in R_{emb}$. Obviously, $l \rhd r$ for all rules in R_{emb}, hence $l \succ r$. Since $\succ$ is a rewrite relation, $s = s[l\sigma] \succ s[r\sigma] = t$. □

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.

Theorem 4.3.13 ("Kruskal's Theorem")**.** Let Σ be a finite signature, let $\mathcal{X}$ be a finite set of variables. Then for every infinite sequence $t_1, t_2, t_3, \ldots$ there are indexes $j > i$ so that $t_j \trianglerighteq_{emb} t_i$. ($\trianglerighteq_{emb}$ is called a *well-partial-ordering (wpo).*)

Proof. The proof can be found in the book of Baader and Nipkow [7] pages 113–115. □

Theorem 4.3.14 (Dershowitz)**.** If Σ is a finite signature, then every simplification ordering $\succ$ on $T(\Sigma, \mathcal{X})$ is well-founded (and therefore a reduction ordering).

Proof. Suppose that $t_1 \succ t_2 \succ t_3 \succ \ldots$ is an infinite descending chain. First assume that there is an $x \in vars(t_{i+1}) \setminus vars(t_i)$. Let $\sigma = \{x \mapsto t_i\}$, then $t_{i+1}\sigma \trianglerighteq x\sigma = t_i$ and therefore $t_i = t_i\sigma \succ t_{i+1}\sigma \succeq t_i$, contradicting reflexivity.

Consequently, $vars(t_i) \supseteq vars(t_{i+1})$ and $t_i \in T(\Sigma, \mathcal{V})$ for all i, where $\mathcal{V}$ is the finite set $vars(t_1)$. By Kruskal's Theorem, there are $i < j$ with $t_i \trianglelefteq_{emb} t_j$. Hence $t_i \preceq t_j$, contradicting $t_i \succ t_j$. □

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

Example 4.3.15.

Let $R = \{f(f(x)) \rightarrow f(g(f(x)))\}$. R terminates and $\rightarrow_R^+$ is therefore a reduction ordering. Assume that $\rightarrow_R$ was contained in a simplification ordering $\succ$. Then $f(f(x)) \rightarrow_R f(g(f(x)))$ implies $f(f(x)) \succ f(g(f(x)))$, and $f(g(f(x))) \trianglerighteq_{emb} f(f(x))$ implies $f(g(f(x))) \succeq f(f(x))$, hence $f(f(x)) \succ f(f(x))$.

4.4 Knuth-Bendix Completion (KBC)

Given a set E of equations, the goal of Knuth-Bendix completion is to transform E into an equivalent convergent set R of rewrite rules. If R is finite this yields a decision procedure for E. For ensuring termination the calculus fixes a reduction ordering $\succ$ and constructs R in such a way that $\rightarrow_R \subseteq \succ$, i.e., $l \succ r$ for every

$l \to r \in R$. For ensuring confluence the calculus checks whether all critical pairs are joinable.

The completion procedure itself is presented as a set of abstract rewrite rules working on a pair of equations E and rules R: $(E_0;R_0) \Rightarrow_{\text{KBC}} (E_1;R_1) \Rightarrow_{\text{KBC}} (E_2;R_2) \Rightarrow_{\text{KBC}} \ldots$. The initial state is $(E_0, \emptyset)$ where $E = E_0$ contains the input equations. If $\Rightarrow_{\text{KBC}}$ successfully terminates then E is empty and R is the convergent rewrite system for E_0. For each step $(E; R) \Rightarrow_{\text{KBC}} (E'; R')$ the equational theories of $E \cup R$ and $E' \cup R'$ agree: $\approx_{E\cup R} = \approx_{E'\cup R'}$. By $\text{cp}(R)$ I denote the set of critical pairs between rules in R.

Orient $(E \uplus \{s \mathrel{\dot{\approx}} t\}; R) \;\Rightarrow_{\text{KBC}}\; (E; R \cup \{s \to t\})$
if $s \succ t$

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

Deduce $(E; R) \;\Rightarrow_{\text{KBC}}\; (E \cup \{s \approx t\}; R)$
if $\langle s, t\rangle \in \text{cp}(R)$

Simplify-Eq $(E \uplus \{s \mathrel{\dot{\approx}} t\}; R) \;\Rightarrow_{\text{KBC}}\; (E \cup \{u \approx t\}; R)$
if $s \to_R u$

R-Simplify-Rule $(E; R \uplus \{s \to t\}) \;\Rightarrow_{\text{KBC}}\; (E; R \cup \{s \to u\})$
if $t \to_R u$

L-Simplify-Rule $(E; R \uplus \{s \to t\}) \;\Rightarrow_{\text{KBC}}\; (E \cup \{u \approx t\}; R)$
if $s \to_R u$ using a rule $l \to r \in R$ so that $s \sqsupset l$, see below.

Trivial equations cannot be oriented and since they are not needed they can be deleted by the Delete rule. The rule Deduce turns critical pairs between rules in R into additional equations. Note that if $\langle s, t\rangle \in \text{cp}(R)$ then $s\,{}_R{\leftarrow} u \to_R t$ and hence $R \models s \approx t$. The simplification rules are not needed but serve as reduction rules, removing redundancy from the state. Simplification of the left-hand side may influence orientability and orientation of the result. Therefore, it yields an equation. For technical reasons, the left-hand side of $s \to t$ may only be simplified using a rule $l \to r$, if $l \to r$ cannot be simplified using $s \to t$, that is, if $s \sqsupset l$, where the *encompassment quasi-ordering* $\mathrel{\underset{\sim}{\sqsupset}}$ is defined by $s \mathrel{\underset{\sim}{\sqsupset}} l$ if $s|_p = l\sigma$ for some p and σ and $\sqsupset = \mathrel{\underset{\sim}{\sqsupset}} \setminus \mathrel{\underset{\sim}{\sqsubset}}$ is the strict part of $\mathrel{\underset{\sim}{\sqsupset}}$.

Lemma 4.4.1. $\sqsupset$ is a well-founded strict partial ordering.

Lemma 4.4.2. If $(E; R) \Rightarrow_{KBC} (E'; R')$, then $\approx_{E\cup R} = \approx_{E'\cup R'}$.

Lemma 4.4.3. If $(E; R) \Rightarrow_{KBC} (E'; R')$ and $\to_R \subseteq \succ$, then $\to_{R'} \subseteq \succ$.

Proposition 4.4.4 (Knuth-Bendix Completion Correctness)**.** If the completion procedure on a set of equations E is run, different things can happen:

1. A state where no more inference rules are applicable is reached and E is not empty. ⇒ Failure (try again with another ordering?)
2. A state where E is empty is reached 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 some definitions are needed:

Definition 4.4.5 (Run)**.** 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 $R_\infty = \bigcup_{i \geq 0} R_i$.

Definition 4.4.6 (Persistent Equations)**.** The sets of *persistent equations of 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 $R_* = R_n$.

Definition 4.4.7 (Fair Run)**.** A run is called *fair* if $\mathrm{CP}(R_*) \subseteq E_\infty$ (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 $\approx_{E_0} = \approx_{E_\infty \cup R_\infty} = \leftrightarrow^*_{E_\infty \cup R_\infty} = \downarrow_{R_*}$.

From now on, $(E_0; R_0) \Rightarrow_{KBC} (E_1; R_1) \Rightarrow_{KBC} (E_2; R_2) \Rightarrow_{KBC} \ldots$ is a fair run and R_0 and E_* are empty.

A *proof* of $s \approx t$ in $E_\infty \cup R_\infty$ is a finite sequence $(s_0, \ldots, s_n)$ so that $s = s_0, t = s_n$ and for all $i \in \{1, \ldots, n\}$ it holds:

1. $s_{i-1} \leftrightarrow_{E_\infty} s_i$ or
2. $s_{i-1} \rightarrow_{R_\infty} s_i$ or
3. $s_{i-1} \;{}_{R_\infty}\!\!\leftarrow s_i$.

The pairs (s_{i-1}, s_i) are called *proof steps.* A proof is called a *rewrite proof* in R_* if there is a $k \in \{0, \ldots, n\}$ so that $s_{i-1} \rightarrow_{R_*} s_i$ for $1 \leq i \leq k$ and $s_{i-1} \;{}_{R_*}\!\!\leftarrow s_i$ for $k+1 \leq i \leq n$.

Idea (Bachmair, Derschowitz, Hsiang): Define a well-founded ordering on proofs so that for every proof that is not a rewrite proof in R_* there is an equivalent smaller proof. Consequence: For every proof there is an equivalent rewrite proof in R_*. A *cost* $c(s_{i-1}, s_i)$ is associated with every proof step as follows:

1. If $s_{i-1} \leftrightarrow_{E_\infty} s_i$ then $c(s_{i-1}, s_i) = (\{s_{i-1}, s_i\}, -, -)$ where the first component is a multiset of terms and $-$ denotes an arbitrary (irrelevant) term.
2. If $s_{i-1} \rightarrow_{R_\infty} s_i$ using $l \rightarrow r$ then $c(s_{i-1}, s_i) = (\{s_{i-1}\}, l, s_i)$.

3. If $s_{i-1} \;{}_{R_\infty}\!\!\leftarrow s_i$ using $l \to r$ then $c(s_{i-1}, s_i) = (\{s_i\}, l, s_{i-1})$.

Proof steps are compared using the lexicographical combination of the multiset extension of the reduction ordering $\succ$, the encompassment ordering $\sqsupset$ and the reduction ordering $\succ$. The cost $c(P)$ of a proof P is the multiset of the cost of its proof steps. The *proof ordering* $\succ_C$ compares the cost of proofs using the multiset extension of the proof step ordering.

Lemma 4.4.8. $\succ_C$ is well-founded ordering.

Lemma 4.4.9. Let P be a proof in $E_\infty \cup R_\infty$. If P is not a rewrite proof in R_* then there exists an equivalent proof P' in $E_\infty \cup R_\infty$ so that $P \succ_C P'$.

Proof. If P is not a rewrite proof in R_* then it contains

1. a proof step that is in E_∞ or
2. a proof step that is in $R_\infty \backslash R_*$ or
3. a subproof $s_{i-1} \;{}_{R_*}\!\!\leftarrow s_i \to s_{i+1}$ (peak).

It is shown that in all three cases the proof step or subproof can be replaced by a smaller subproof:
Case 1.: A proof step using an equation $s \dot{\approx} t$ is in E_∞. This equation must be deleted during the run.

If $s \dot{\approx} t$ is deleted using *Orient*:

$$\ldots s_{i-1} \leftrightarrow_{E_\infty} s_i \ldots \quad\Longrightarrow\quad \ldots s_{i-1} \rightarrow_{R_\infty} s_i \ldots$$

If $s \dot{\approx} t$ is deleted using *Delete*:

$$\ldots s_{i-1} \leftrightarrow_{E_\infty} s_{i-1} \ldots \quad\Longrightarrow\quad \ldots s_{i-1} \ldots$$

If $s \dot{\approx} t$ is deleted using *Simplify-Eq*:

$$\ldots s_{i-1} \leftrightarrow_{E_\infty} s_i \ldots \quad\Longrightarrow\quad \ldots s_{i-1} \rightarrow_{R_\infty} s' \leftrightarrow_{E_\infty} s_i \ldots$$

Case 2.: A proof step using a rule $s \to t$ is in $R_\infty \backslash R_*$. This rule must be deleted during the run.

If $s \to t$ is deleted using *R-Simplify-Rule*:

$$\ldots s_{i-1} \rightarrow_{R_\infty} s_i \ldots \quad\Longrightarrow\quad \ldots s_{i-1} \rightarrow_{R_\infty} s' \;{}_{R_\infty}\!\!\leftarrow s_i \ldots$$

If $s \to t$ is deleted using *L-Simplify-Rule*:

$$\ldots s_{i-1} \rightarrow_{R_\infty} s_i \ldots \quad\Longrightarrow\quad \ldots s_{i-1} \rightarrow_{R_\infty} s' \leftrightarrow_{E_\infty} s_i \ldots$$

Case 3.: A subproof has the form $s_{i-1} \;{}_{R_*}\!\!\leftarrow s_i \rightarrow_{R_*} s_{i+1}$.

If there is no overlap or a non-critical overlap:

$$\ldots s_{i-1}\ {}_{R_*}\!\!\leftarrow s_i \rightarrow_{R_*} s_{i+1} \ldots \quad \Longrightarrow \quad \ldots s_{i-1} \rightarrow^*_{R_*} s'\ {}^{*}_{R_*}\!\!\leftarrow s_{i+1} \ldots$$

If there is a critical pair that has been added using *Deduce*:

$$\ldots s_{i-1}\ {}_{R_*}\!\!\leftarrow s_i \rightarrow_{R_*} s_{i+1} \ldots \quad \Longrightarrow \quad \ldots s_{i-1} \leftrightarrow_{E_\infty} s_{i+1} \ldots$$

In all cases, checking that the replacement subproof is smaller than the replaced subproof is routine. □

Theorem 4.4.10 (KBC Soundness)**.** Let $(E_0; R_0) \Rightarrow_{KBC} (E_1; R_1) \Rightarrow_{KBC} (E_2; R_2) \Rightarrow_{KBC} \ldots$ be a fair run and let R_0 and E_* be empty. Then

1. every proof in $E_\infty \cup R_\infty$ is equivalent to a rewrite proof in R_*,

2. R_* is equivalent to E_0 and

3. R_* is convergent.

Proof. 1. By well-founded induction on $\succ_C$ using the previous lemma.

2. Clearly, $\approx_{E_\infty \cup R_\infty} = \approx_{E_0}$. Since $R_* \subseteq R_\infty$ this yields $\approx_{R_*} \subseteq \approx_{E_\infty \cup R_\infty}$. On the other hand, by 1. it holds that $\approx_{E_\infty \cup R_\infty} \subseteq \approx_{R_*}$.

3. Since $\rightarrow_{R_*} \subseteq \succ$, R_* is terminating. By 1. it holds that R_* is confluent. □

Now using the proof of Theorem 3.15.2 termination of $\Rightarrow_{KBC}$ is undecidable.

Corollary 4.4.11 (KBC Termination)**.** Termination of $\Rightarrow_{KBC}$ is undecidable for some given finite set of equations E.

Proof. Using exactly the construction of Theorem 3.15.2 it remains to be shown that all computed critical pairs can be oriented. Critical pairs corresponding to the search for a PCP solution result in equations $f_R(u(x), v(y)) \approx f_R(u'(x), v'(y))$ or $f_R(u'(x), v'(x)) \approx c$. By chosing an appropriate ordering, all these equations can be oriented. Thus $\Rightarrow_{KBC}$ does not produce any unorientable equations. The rest follows from Theorem 3.15.2. □

4.4.1 Unfailing Completion

Classical completion: Try to transform a set E of equations into an equivalent convergent TRS. Fail, if an equation can neither be oriented nor deleted.

Unfailing completion (from Bachmair, Derschowitz and Plaisted [8]): If an equation cannot be oriented, *orientable instances* can still be used for rewriting. Note: If $\succ$ is total on ground terms, then every *ground instance* of an equation is trivial or can be oriented. The goal is to derive a *ground convergent* set of equations.