UPPSALA UNIVERSITET
Matematiska institutionen
Erik Palmgren

Kompletterande material
Tillämpad logik DV1
2001-09-19

1 Term rewriting systems

This handout is mainly intended as a guide and supplement to Chapter 2 of

W. Klop: Term Rewriting Systems. In: S. Abramsky *et al.* (eds.) *Handbook of Logic in Computer Science,* Vol 2. Oxford University Press 1992.

The following parts are required reading: Ch. 2.1 (pp. 11 – 19), Ch. 2.3 – 2.4 (pp. 29 – 34, 40 – 51) and Ch. 2.6 (pp. 62 – 65) and cover the following

- Birkhoff's completeness theorem for equational theories (see also section 1.1 below),
- rewrite rules
- unification of terms (see also section 1.2),
- basic definitions and results concerning well-founded relations, abstract reduction systems (see also sections 1.3 – 1.4 below),
- complete term rewriting systems, critical pairs, the Knuth-Bendix completion procedure,
- termination proofs using recursive path orderings.

1.1 Some notions from universal algebra

In *universal algebra* properties of general algebraic systems are studied. These systems include the usual, groups, semigroups, monoids, rings, but also systems with operations of arbitrary number of arguments. In algebraic specification theory these operations may describe programs or hardware components. (See Meinke and Tucker 1992, Goguen and Malcolm 1996 and Wechler 1992.)

A *signature* Σ is a set of function symbols, where each $F \in \Sigma$ takes a fixed number $n(F)$ (the *arity*) of arguments. 0-ary function symbols are considered as constant symbols. (Thus a signature is like a description of a first order language but without relation symbols.) A Σ-*algebra* $\mathcal{A}$ consists of an underlying nonempty set A, and for each function symbol $F \in \Sigma$, an operation

$$F^{\mathcal{A}} : A^{n(F)} \to A,$$

for $n(F) > 0$. If $n(F) = 0$, $F^{\mathcal{A}} \in A$.

Homomorphisms, mappings which preserves the operations of an algebra are of central importance. Let $\mathcal{A}$ and $\mathcal{B}$ be Σ-algebras. A *(Σ-algebra) homomorphism* $\varphi : \mathcal{A} \to \mathcal{B}$ is function between the underlying sets $\varphi : A \to B$ which is such that for every function symbol $F \in \Sigma$ of arity n we have for all $a_1, \ldots, a_n \in A$:

$$\varphi(F^{\mathcal{A}}(a_1, \ldots, a_n)) = F^{\mathcal{B}}(\varphi(a_1), \ldots, \varphi(a_n)).$$

If $n = 0$, this reads $\varphi(F^{\mathcal{A}}) = F^{\mathcal{B}}$.

Example 1.1 The embedding $\mathbb{Z} \hookrightarrow \mathbb{Q}$ and the quotient mapping $x \mapsto x \bmod n : \mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$ are basic examples of homomorphisms with respect to the signature $\Sigma = \{0, 1, +, \cdot\}$.

There is always a trivial homomorphism $\mathcal{A} \to \mathcal{A}$, the identity homomorphism $\mathrm{id}_{\mathcal{A}}$ defined by $\mathrm{id}_{\mathcal{A}}(x) = x$. A homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ is an *isomorphism* if there is a homomorphism $\psi : \mathcal{B} \to \mathcal{A}$ such that $\psi \circ \varphi = \mathrm{id}_{\mathcal{A}}$ and $\varphi \circ \psi = \mathrm{id}_{\mathcal{B}}$. We leave the verification of the following result to the reader:

Proposition 1.2 *A homomorphism $\varphi : \mathcal{A} \to \mathcal{B}$ is an isomorphism iff $\varphi : A \to B$ is a bijection.* $\square$

Let $\mathrm{Ter}(\Sigma)$ be the set of terms that can be formed from the function symbols in Σ and variables from a fixed set of variable symbols $\mathbb{X} = \{x_1, x_2, x_3, \ldots\}$. The set $\mathrm{Ter}(\Sigma)$ is inductively defined by the following clauses

(T1) If $x \in \mathbb{X}$, then $x \in \mathrm{Ter}(\Sigma)$.

(T2) If $F \in \Sigma$ and $n(F) = 0$, then $F \in \mathrm{Ter}(\Sigma)$.

(T3) If $F \in \Sigma$, $n = n(F) > 0$ and $t_1, \ldots, t_n \in \mathrm{Ter}(\Sigma)$, then $F(t_1, \ldots, t_n) \in \mathrm{Ter}(\Sigma)$.

Since the set $\mathrm{Ter}(\Sigma)$ is inductively defined, we may prove properties of terms by structural induction. We may also define functions on terms by structural recursion. A *substitution* is a function $\sigma : \mathbb{X} \to \mathrm{Ter}(\Sigma)$, assigning to each variable symbol a term. The effect t^σ of a substitution σ on a term t is defined recursively

$$\begin{array}{rcll} x_i^\sigma & = & \sigma(x_i) & \\ F^\sigma & = & F & \qquad (n(F) = 0) \\ F(t_1, \ldots, t_n)^\sigma & = & F(t_1^\sigma, \ldots, t_n^\sigma) & \qquad (n = n(F)) \end{array}$$

Thus we may extend σ to a function $\mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$ by $\sigma(t) = t^\sigma$. Denote by

$$\{x_{i_1} := t_1, \ldots, x_{i_k} := t_k\},$$

where $i_1 < i_2 < \cdots < i_k$, the substitution σ where $\sigma(x_{i_j}) = t_j$ for $j = 1, \ldots, k$ and $\sigma(x_i) = x_i$ for $i \notin \{i_1, i_2, \ldots, i_k\}$.

Example 1.3 Let $\Sigma = \{0, f, g\}$ where the arities are $n(0) = 0$, $n(f) = 1$ and $n(g) = 2$. Then $0, f(0), g(x_1, f(x_3))$ are examples of terms over Σ. For the substitution $\sigma = \{x_1 := g(x_1, x_3), x_2 := f(0), x_3 := x_2\}$ we have

$$g(x_1, f(x_3))^\sigma = g(x_1^\sigma, f(x_3^\sigma)) = g(g(x_1, x_3), f(x_2)). \ \Box$$

The set $\mathrm{Ter}(\Sigma)$ can be regarded as a Σ-algebra — in a kind of trivial way — by defining for each n-ary function symbol $F \in \Sigma$, a function $F^{\mathrm{Ter}(\Sigma)}$ by

$$F^{\mathrm{Ter}(\Sigma)}(t_1, \ldots, t_n) = F(t_1, \ldots, t_n).$$

We call $\mathrm{Ter}(\Sigma)$ the *term algebra of* Σ. We may also restrict ourselves to terms without variables (in case there are constant symbols) The resulting set, $\mathrm{Ter}_0(\Sigma)$, also forms a Σ-algebra.

Note that any substitution $\sigma : \mathbb{X} \to \mathrm{Ter}(\Sigma)$ extends to a Σ-algebra homomorphism $\sigma : \mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$. (Exercise: verify this.)

Let $\mathcal{A}$ be a Σ-algebra. A *variable assignment in* $\mathcal{A}$ is a function $\rho : \mathbb{X} \to A$. Given such an assignment, the value $[\![t]\!]_\rho^{\mathcal{A}}$ of a term t in $\mathcal{A}$ is determined. Define by recursion on t:

$$\begin{aligned} [\![x_i]\!]_\rho &= \rho(x_i), \\ [\![F(t_1, \ldots, t_m)]\!]_\rho &= F^{\mathcal{A}}([\![t_1]\!]_\rho, \ldots, [\![t_m]\!]_\rho). \end{aligned}$$

An equation $s = t$ *is valid in* $\mathcal{A}$ (in symbols: $\mathcal{A} \models s = t$) iff for all variable assignments ρ in $\mathcal{A}$: $[\![s]\!]_\rho^{\mathcal{A}} = [\![t]\!]_\rho^{\mathcal{A}}$.

An equational theory over Σ is given by a set E of equations $s = t$ where $s, t \in \mathrm{Ter}(\Sigma)$. The deduction rules of an equational theory essentially only tell how instances of these equations may be used to calculate inside terms. We denote by $E \vdash_{\mathrm{eq}} s = t$ that $s = t$ is derivable from E. The deduction rules are more formally

$$E \vdash_{\mathrm{eq}} s = t \qquad \text{if } s = t \in E$$

$$\frac{E \vdash_{\mathrm{eq}} s = t}{E \vdash_{\mathrm{eq}} s^\sigma = t^\sigma}\,(\mathrm{subst}) \qquad \text{for every substitution } \sigma : \mathbb{X} \to \mathrm{Ter}(\Sigma)$$

$$\frac{E \vdash_{\mathrm{eq}} s_1 = t_1 \quad \cdots \quad E \vdash_{\mathrm{eq}} s_n = t_n}{E \vdash_{\mathrm{eq}} F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)}\,(\mathrm{cong}) \qquad \text{for every } F \in \Sigma \text{ with } n = n(F)$$

$$E \vdash_{\mathrm{eq}} t = t \qquad \text{for every } t \in \mathrm{Ter}(\Sigma)$$

$$\frac{E \vdash_{\mathrm{eq}} s = t}{E \vdash_{\mathrm{eq}} t = s}\,(\mathrm{symm})$$

$$\frac{E \vdash_{\mathrm{eq}} s = v \qquad E \vdash_{\mathrm{eq}} v = t}{E \vdash_{\mathrm{eq}} s = t}\,(\mathrm{trans})$$

Example 1.4 *The equational theory of groups.* Let $\Sigma = \{1, \cdot, (\,)^{-1}\}$, where the arities are 0, 2 and 1 respectively. The equations E are

$$\begin{aligned} &1 \cdot x_1 = x_1, \qquad x_1 \cdot 1 = x_1, \\ &x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3, \\ &x_1 \cdot x_1^{-1} = 1, \qquad x_1^{-1} \cdot x_1 = 1. \end{aligned}$$

We give an example of a formal derivation of $x_2 \cdot 1^{-1} = x_2$ from the axioms of E:

$$\dfrac{\dfrac{x_2 = x_2 \qquad \dfrac{\dfrac{\dfrac{x_1 \cdot 1 = x_1}{1^{-1} \cdot 1 = 1^{-1}}\,(\text{subst})}{1^{-1} = 1^{-1} \cdot 1}\,(\text{symm}) \qquad \dfrac{x_1^{-1} \cdot x_1 = 1}{1^{-1} \cdot 1 = 1}\,(\text{subst})}{1^{-1} = 1}\,(\text{trans})}{x_2 \cdot 1^{-1} = x_2 \cdot 1}\,(\text{cong}) \qquad \dfrac{x_1 \cdot 1 = x_1}{x_2 \cdot 1 = x_2}\,(\text{subst})}{x_2 \cdot 1^{-1} = x_2}\,(\text{trans})$$

□

A Σ-algebra $\mathcal{A}$ *is a model of* E (in symbols: $\mathcal{A} \models E$) iff $\mathcal{A} \models s = t$, for each $s = t \in E$. We say that $s = t$ is a (semantic) consequence of E (in symbols: $E \models s = t$) if for every Σ-algebra $\mathcal{A}$:

$$\mathcal{A} \models E \Longrightarrow \mathcal{A} \models s = t.$$

We now prove Birkhoff's completeness theorem for equational theories. Let $=_E$ be the relation on $\mathrm{Ter}(\Sigma)$ defined by

$$s =_E t \Longleftrightarrow_{\text{def}} E \vdash_{\text{eq}} s = t.$$

This relation of *E-provable equality* is an equivalence relation and a congruence with respect to the operations $F^{\mathrm{Ter}(\Sigma)}$, according to the rules of the equational theory. We consider the set $\mathcal{T}(E) = \mathrm{Ter}(\Sigma)/=_E$ of equivalence classes $[t]$ of terms. Thus the following is a well-defined operation

$$F^{\mathcal{T}(E)}([t_1], \ldots, [t_n]) = [F(t_1, \ldots, t_n)]$$

for any $F \in \Sigma$. Thus $\mathcal{T}(E)$ is a Σ-algebra.

Theorem 1.5 (Birkhoff) *Let Σ be a signature and let E be an equational theory over Σ. Then*

$$E \vdash_{\text{eq}} s = t \Longleftrightarrow \mathcal{T}(E) \models s = t.$$

Proof. ($\Leftarrow$) Suppose $\mathcal{T}(E) \models s = t$. Then for the "identical" variable assignment $\tau(x_i) = [x_i]$ we get $[\![s]\!]_\tau^{\mathcal{T}(E)} = [\![t]\!]_\tau^{\mathcal{T}(E)}$. Hence $[s] = [t]$, so $s =_E t$ and thus $E \vdash_{\text{eq}} s = t$.

($\Rightarrow$) Note that each variable assignment $\tau : \mathbb{X} \to \mathcal{T}(E)$ gives rise to a substitution $\sigma : \mathbb{X} \to \mathrm{Ter}(\Sigma)$ where

$$\tau(x_i) = [\sigma(x_i)].$$

Thus from $E \vdash_{\text{eq}} s = t$ and the substitution rule follows $E \vdash_{\text{eq}} s^\sigma = t^\sigma$. Hence $[s^\sigma] = [t^\sigma]$. But $[\![s]\!]_\tau = [s^\sigma]$ and $[\![t]\!]_\tau = [t^\sigma]$, and hence $\mathcal{T}(E) \models s = t$, since τ was arbitrary. $\square$

Corollary 1.6 *For every equational theory E and any equation $s = t$ over Σ we have*

$$E \models s = t \Longleftrightarrow E \vdash_{\text{eq}} s = t$$

Proof. ($\Leftarrow$) This is an easy proof by induction on derivations.

($\Rightarrow$) From Theorem 1.5 ($\Rightarrow$) follows $\mathcal{T}(E) \models E$ (since $s = t \in E$ implies $E \vdash_{\text{eq}} s = t$). Suppose $E \models s = t$. Then in particular $\mathcal{T}(E) \models s = t$. By Theorem 1.5 ($\Leftarrow$) again $E \vdash_{\text{eq}} s = t$. $\square$

Remark 1.7 In view of Birkhoff's completeness theorem and the usual completeness theorem for first order logic, we have for equational theories E:

$$E \vdash_{\text{eq}} s = t \Longleftrightarrow E \vdash s = t.$$

Thus quantifiers and connectives are not necessary when proving an equation from equational axioms.

Example 1.8 *The equational theory of Abelian groups.* Let $\Sigma = \{1, \cdot, (\,)^{-1}\}$, where the arities are 0, 2 and 1 respectively. The equations E are

$$\begin{aligned}
&1 \cdot x_1 = x_1, \qquad x_1 \cdot 1 = x_1,\\
&x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3,\\
&x_1 \cdot x_2 = x_2 \cdot x_1,\\
&x_1 \cdot x_1^{-1} = 1, \qquad x_1^{-1} \cdot x_1 = 1.
\end{aligned}$$

The models of this theory are exactly the Abelian groups. Denote by $u^0 = 1$ and $u^{n+1} = u \cdot u^n$ for $n \in \mathbb{N}$. For $n > 0$, let $u^{-n} = (u^{-1})^n$. It is easy to show that for each $t \in \text{Ter}(\Sigma)$ there are sequences $n_1, \ldots, n_k \in \mathbb{Z} - \{0\}$, $1 \leq i_1 < i_2 \cdots < i_k$, where $k \geq 0$, such that

$$t =_E x_{i_1}^{n_1} \cdot x_{i_2}^{n_2} \cdot \cdots \cdot x_{i_k}^{n_k}. \tag{1}$$

(In case $k = 0$, the product is simply 1.) Thus in the model $\mathcal{T}(E)$ the equivalence classes are represented by elements of the form $x_{i_1}^{n_1} \cdot x_{i_2}^{n_2} \cdot \cdots \cdot x_{i_k}^{n_k}$. $\square$

One can in fact show that the sequences (n_j), (i_j) in (1) are unique. This can be used to decide when two terms are provably equal. A systematic method for obtaining such decidability results is provided by the theory of *term rewriting systems.*

For a signature Σ with at least one constant symbol, consider $\mathcal{T}_0(E)$ which is defined as $\mathcal{T}(E)$ but $\text{Ter}_0(\Sigma)$ is used instead of $\text{Ter}(\Sigma)$. (Exercise: What is $\mathcal{T}_0(E)$ in the case of Example 1.8? If new constants are added?)

Theorem 1.9 *Let E be an equational theory over a signature Σ, which has at least one constant symbol. Then*

(a) $\mathcal{T}_0(E) \models E$

(b) if $\mathcal{A} \models E$*, there is a unique homomorphism* $\varphi : \mathcal{T}_0(E) \to \mathcal{A}$*.*

Proof. (a): This is proved as in the direction ($\Rightarrow$) of Theorem 1.5, but using $\mathcal{T}_0(E)$ instead of $\mathcal{T}(E)$.

(b): Define $\varphi : \mathcal{T}_0(E) \to \mathcal{A}$ by $\varphi([t]) = [\![t]\!]_\tau^{\mathcal{A}}$ where τ is some fixed variable assignment (it does not matter which since t has no variables). It is well-defined because if $[s] = [t]$, then $E \vdash_{\text{eq}} s = t$. Now $\mathcal{A} \models E$, so $\mathcal{A} \models s = t$, and hence in particular $[\![s]\!]_\tau^{\mathcal{A}} = [\![t]\!]_\tau^{\mathcal{A}}$. Furthermore φ is a homomorphism, since

$$\begin{aligned}
\varphi(F^{\mathcal{T}(E)}([t_1], \ldots, [t_n])) &= \varphi([F(t_1, \ldots, t_n)]) \\
&= [\![F(t_1, \ldots, t_n)]\!]_\tau^{\mathcal{A}} \\
&= F^{\mathcal{A}}([\![t_1]\!]_\tau^{\mathcal{A}}, \ldots, [\![t_n]\!]_\tau^{\mathcal{A}}) \\
&= F^{\mathcal{A}}(\varphi([t_1]), \ldots, \varphi([t_n])).
\end{aligned}$$

Now, if ψ were another homomorphism, it is easily shown that $\psi([t]) = \varphi([t])$ by induction on t. $\square$

Because of this theorem the model $\mathcal{T}_0(E)$ is called the *initial model* of the theory E.

Remark 1.10 For algebraic specification of programs one usually consider Σ-algebras with many sorts (types). For instance, we may have a sort A for an alphabet and a sort S for a stack. The constants are $\mathsf{a}, \mathsf{b}, \mathsf{c} : \mathsf{A}$ (the letters of the alphabet), $\mathsf{nil} : \mathsf{S}$ (the empty stack), the function symbols are $\mathsf{pop} : \mathsf{S} \to \mathsf{S}$ and $\mathsf{push} : \mathsf{A} \times \mathsf{S} \to \mathsf{S}$. The equations E are

$$\begin{aligned}
\mathsf{pop}(\mathsf{nil}) &= \mathsf{nil}, \\
\mathsf{pop}(\mathsf{push}(x^{\mathsf{A}}, t^{\mathsf{S}})) &= t^{\mathsf{S}}
\end{aligned}$$

(Here $x^{\mathsf{A}}, t^{\mathsf{S}}$ indicate variables of the different sorts.) The definitions and results above easily extend to many-sorted Σ-algebras.

Exercises

1. The equational theory of semigroups is given by E_1:

$$\begin{aligned}
1 \cdot x_1 = x_1, \qquad x_1 \cdot 1 = x_1, \\
x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3,
\end{aligned}$$

   where $\Sigma = \{1, \cdot\}$. Determine the equivalence classes in $\mathcal{T}(E_1)$ analogously to Example 1.8.

2. Try to find simple representatives of equivalence classes in $\mathcal{T}_0(E)$ where E is as in Remark 1.10.

1.2 Unification of terms

Unification is an important tool in term rewriting, automatic theorem proving, and fundamental for logic programming (Prolog). Unification of terms amount to equation solving in the term algebra $\mathrm{Ter}(\Sigma)$.

Example 1.11 Let $\Sigma = \{f, g\}$ with arities 2 and 1 respectively. Find a solution in $\mathrm{Ter}(\Sigma)$ to the equation

$$f(x_1, g(f(x_2, x_1))) = f(g(x_2), x_3).$$

A solution: $x_1 := g(x_2), x_3 := g(f(x_2, g(x_2)))$.

As in ordinary equation solving we are often interested in a general solution. Over the term algebra such a solution is called a *most general unifier*. Indeed, in the example above any other solution can be gotten from the one provided, by instantiating the variables.

As explained in Section 1.1 substitutions can be regarded as Σ-algebra homomorphisms $\sigma : \mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$ determined by their values on the set $\mathbb{X}$ of variables. A substitution that is given by a permutation of the variables is called a *renaming substitution*. Two substitutions $\tau : \mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$ and $\sigma : \mathrm{Ter}(\Sigma) \to \mathrm{Ter}(\Sigma)$ may be composed $\sigma \circ \tau$ as follows

$$(\sigma \circ \tau)(t) = \sigma(\tau(t)) = (t^\tau)^\sigma.$$

We write $\tau\sigma$ for $\sigma \circ \tau$.

Example 1.12 Let $\Sigma = \{f, g\}$ with arities 2 and 1 respectively. Consider the substitutions $\sigma = \{x_2 := g(x_1), x_3 := g(x_3)\}$ and $\tau = \{x_1 := f(x_2, x_2)\}$. Then $(\tau\sigma)(x_1) = \sigma(\tau(x_1)) = \sigma(f(x_2, x_2)) = f(\sigma(x_2), \sigma(x_2)) = f(g(x_1), g(x_1))$, $(\tau\sigma)(x_2) = g(x_1)$ and $(\tau\sigma)(x_3) = g(x_3)$. Hence

$$\tau\sigma = \{x_1 := f(g(x_1), g(x_1)), x_2 := g(x_1), x_3 := g(x_3)\}.$$

On the other hand, by a similar computation,

$$\sigma\tau = \{x_1 := f(x_2, x_2), x_2 := g(f(x_2, x_2)), x_3 := g(x_3)\}. \ \square$$

Generalising this example we have for $\sigma = \{x_{i_1} := t_1, \ldots, x_{i_n} := t_n\}$ and $\tau = \{x_{i_1} := s_1, \ldots, x_{i_n} := s_n, x_{j_1} := r_1, \ldots, x_{j_m} := r_m\}$, where the indices $i_1, \ldots, i_n, j_1, \ldots, j_m$ are all distinct, that

$$\sigma\tau = \{x_{i_1} := t_1^\tau, \ldots, x_{i_n} := t_n^\tau, x_{j_1} := r_1, \ldots, x_{j_m} := r_m\}$$

We say that one substitution σ is *more general* than another substitution ρ iff $\rho = \sigma\tau$ for some substitution τ. In this case we write $\sigma \leq \rho$.

Exercise 1.13

(i) Check that the relation $\leq$ is reflexive and transtive.

(ii) Prove that if $\sigma \leq \rho$ and $\rho \leq \sigma$, then there is a renaming substitution τ such that $\rho = \sigma\tau$. $\square$

A *unifier* of a set of terms $\mathcal{T} = \{t_1, \ldots, t_n\}$ is substitution σ which makes all these terms equal, i.e. $t_1^\sigma = \cdots = t_n^\sigma$. A unifier σ of $\mathcal{T}$ is a *most general unifier (mgu),* if $\sigma \leq \rho$ for any unifier ρ of $\mathcal{T}$. By Exercise 1.13 any two mgu's σ and σ' of $\mathcal{T}$ are the same up to a renaming substitution (i.e. $\sigma = \sigma'\tau$ for some renaming substitution τ).

Note that $F(s_1, \ldots, s_n)^\sigma = F(t_1, \ldots, t_n)^\sigma$ iff $s_i^\sigma = t_i^\sigma$ for all $i = 1, \ldots, n$. Hence in order to solve one equation in the term algebra, we may have to solve a system of equations.

The unification algorithm of Martelli-Montanari. The algorithm starts with a finite set of equations $G = \{s_1 = t_1, \ldots, s_n = t_n\}$, and outputs a most general unifier σ for this set (regarded as an mgu of the set $\{F(s_1, \ldots, s_n), F(t_1, \ldots, t_n)\}$ where F is a function symbol), if there is any unifier, or reports failure otherwise. The algorithm is non-deterministic and applies certain reduction rules to the finite sets and stops at the empty set ($\emptyset$), or with a failure (denoted #). Along the way the answer substitution σ is built up. From a successful computation

$$G_1 \rightarrowtail G_2 \rightarrowtail_{\sigma_1} G_3 \rightarrowtail G_4 \rightarrowtail G_5 \rightarrowtail_{\sigma_2} G_6 \rightarrowtail \emptyset.$$

we extract $\sigma = \sigma_1\sigma_2$, the answer substitution. For a set $G = \{s_1 = t_1, \ldots, s_n = t_n\}$ we write $G^\sigma = \{s_1^\sigma = t_1^\sigma, \ldots, s_n^\sigma = t_n^\sigma\}$.

The Martelli-Montanari reduction rules are the following

1. $G \cup \{F(t_1, \ldots, t_n) = F(s_1, \ldots, s_n)\} \rightarrowtail G \cup \{t_1 = s_1, \ldots, t_n = s_n\}$ provided $F(t_1, \ldots, t_n) = F(s_1, \ldots, s_n)$ is not an element of G. ("Function decomposition")

2. $G \cup \{t = t\} \rightarrowtail G$ provided $t = t$ is not an element of G.

3. $G \cup \{t = x\} \rightarrowtail G \cup \{x = t\}$, provided t is not a variable, and that $t = x$ is not an element of G.

4. $G \cup \{x = t\} \rightarrowtail_{\{x:=t\}} G^{\{x:=t\}}$, provided x is a variable, x does not occur in t and that $x = t$ is not an element of G. ("Variable elimination")

5. $G \cup \{F(t_1, \ldots, t_n) = H(s_1, \ldots, s_m)\} \rightarrowtail \#$, if F and H are different function symbols.

6. $G \cup \{x = t\} \rightarrowtail \#$, provided $x \neq t$ and x occurs in t. ("Occur check")

Example 1.14 We compute the mgu of $f(x_1, g(f(x_2, x_1)))$ and $f(g(x_2), x_3)$ using the algorithm.

$$\begin{array}{lll}
\{f(x_1, g(f(x_2, x_1))) = f(g(x_2), x_3)\} & \rightarrowtail & \{x_1 = g(x_2), g(f(x_2, x_1)) = x_3\} \\
 & \rightarrowtail_{\{x_1:=g(x_2)\}} & \{g(f(x_2, g(x_2))) = x_3\} \\
 & \rightarrowtail & \{x_3 = g(f(x_2, g(x_2)))\} \\
 & \rightarrowtail_{\{x_3:=g(f(x_2,g(x_2)))\}} & \emptyset
\end{array}$$

The answer substitution is $\sigma = \{x_1 := g(x_2), x_3 := g(f(x_2, g(x_2)))\}$. $\square$

Example 1.15 The terms $f(g(x_1), x_1)$ and $f(x_2, g(x_2))$ are not unifiable.

$$\begin{array}{lll} \{f(g(x_1), x_1) = f(x_2, g(x_2))\} & \rightarrowtail & \{g(x_1) = x_2, x_1 = g(x_2)\} \\ & \rightarrowtail_{\{x_1 := g(x_2)\}} & \{g(g(x_2)) = x_2\} \\ & \rightarrowtail & \{x_2 = g(g(x_2))\} \\ & \rightarrowtail & \# \end{array}$$

This computation fails by occur check, since x_2 occurs in $g(g(x_2))$. $\square$

We state the following important result without proof:

Theorem 1.16 (Unification Theorem) *A set of equations $G = \{s_1 = t_1, \ldots, s_n = t_n\}$ has an mgu iff it has some unifier. Moreover, if G has an mgu, the Martelli-Montanari algorithm finds it, otherwise it stops and report failure to find a unifier.*

Exercises

1. Let $\Sigma = \{a, f, g, h, p, q\}$ where a is a constant, f, g has arity 1, h, p has arity 2 and q has arity 3. For each of the following pair of terms compute an mgu or show that no unifier exist.

   (a) $p(f(a), g(x))$, $p(y, y)$

   (b) $p(f(x), a)$, $p(y, f(w))$

   (c) $p(x, x)$, $p(y, f(y))$

   (d) $q(a, x, f(g(y)))$, $q(z, h(z, w), f(w))$.

1.3 Well-founded relations

A binary relation $(A, <)$ is *well-founded* if there is no infinite descending sequence

$$a_1 > a_2 > a_3 > \cdots$$

in A.

Example 1.17 The natural numbers $(\mathbb{N}, <)$ with the usual order is well-founded, while this is not the case for the integers $(\mathbb{Z}, <)$.

Example 1.18 Let $R \subseteq \mathbb{N} \times \mathbb{N}$ be the successor relation defined by

$$R(x, y) \iff x + 1 = y.$$

Then $(\mathbb{N}, R)$ is well-founded. Note that R is not transitive.

Example 1.19 Consider $(\mathbb{N} \times \mathbb{N}, <')$ with the lexicographic order $(a, b) <' (c, d)$ iff $a < c$ or $a = c$ and $b < d$. We have

$$(0,0) <' (0,1) <' \cdots <' (0,n) <' \cdots <' (1,0) <' (1,1) <' \cdots (2,0) <' \cdots <' (m,0).$$

This relation is well-founded. For suppose $(a_{n+1}, b_{n+1}) <' (a_n, b_n)$ for all n. Then the sequence (a_n) is eventually constant from, say N, and onwards. Hence $b_{k+1} <' b_k$ for all $k \geq N$, which is impossible. $\square$

Example 1.20 Let Σ be a signature and let $\mathbb{X}$ be a nonempty set of variables. Order the set $\mathrm{Ter}(\Sigma, \mathbb{X})$ of terms over Σ and $\mathbb{X}$ as follows

$$t \sqsubset s \iff t \neq s \text{ and } t \text{ is a subterm of } s.$$

If $t \sqsubset s$, we say that t is a *strict subterm* of s, or that it is *structurally smaller* than s. We leave as an exercise to show that $(\mathrm{Ter}(\Sigma, \mathbb{X}, \sqsubset)$ is a well-founded relation. Example: for $\Sigma = \{0, f(\cdot), g(\cdot, \cdot)\}$, $\mathbb{X} = \{x, y, z, \ldots\}$ we have

$$x \sqsubset f(x) \qquad y \not\sqsubset f(x) \qquad f(y) \sqsubset g(f(f(y)), f(0)) \qquad g(0, z) \sqsubset f(g(g(0, z), z)) \quad \square$$

This kind of order relation is useful when proving termination of functional programs.

That a relation is well-founded is the same as saying that a certain induction principle is valid, so called *Noetherian*[1] *induction,* or *well-founded induction.* Let $(A, <)$ be a binary relation. A subset $S \subseteq A$ is *progressive* iff

$$(\forall a)[(\forall b < a) b \in S \Rightarrow a \in S].$$

Thus in a progressive set, if all the elements that lie before a are in the set, then also a is in the set. A binary relation $(A, <)$ is called *inductive* iff $S = A$ whenever $S \subseteq A$ is a progressive subset. What are the progressive subsets S of $(\mathbb{N}, <)$? Clearly, there are no elements before 0, and hence trivially $0 \in S$. Now suppose that $\{0, 1, \ldots, n\} \subseteq S$. Then all elements before $n + 1$ are in S. Hence also $n + 1 \in S$. By induction $S = \mathbb{N}$. Above we just showed that $(\mathbb{N}, <)$ is inductive. In fact, we have

[1] After Emmy Noether, a pioneer in abstract algebra.

Theorem 1.21 *A binary relation is well-founded iff it is inductive.*

Proof. Suppose that $(A, <)$ is an inductive binary relation. Define the following subset of A

$$S = \{b \in A : \text{there is no infinite sequence } b > a_1 > a_2 > a_3 \cdots\}.$$

It is easily checked that S is progressive set. Hence $S = A$, so $(A, <)$ is well-founded.

Now suppose that $(A, <)$ is not inductive. Hence there is a progressive set $S \subset A$. Let $x_0 \in A \setminus S$. Since S is progressive, there must be some $x_1 < x_0$ such that $x_1 \notin S$. But then again there must be some $x_2 < x_1$ such that $x_2 \notin S$. Proceeding in this way one constructs a sequence

$$x_0 > x_1 > x_2 > \cdots$$

which shows that $(A, <)$ is not well-founded. □

Let $(A, <)$ be a binary relation. The *transitive closure* $(A, <^+)$ of $(A, <)$ is defined by $a <^+ b$ iff there is a sequence $a_1 < \cdots < a_n$, $n \geq 1$, with $a = a_1$ and $b = a_n$. Thus, for example, $(\mathbb{N}, <)$ is the transitive closure of $(\mathbb{N}, R)$ from Example 1.18. We leave the following as an easy exercise

Proposition 1.22 *Let $(A, <)$ be a binary relation. Then $(A, <)$ is well-founded iff $(A, <^+)$ is well-founded.* □

Reduction of one ordering to another. Suppose that $(A, <)$ is well-founded, $(B, <')$ a binary relation and $f : B \to A$ a function such that, for all x and y

$$x <' y \Rightarrow f(x) < f(y).$$

Then $(B, <')$ is well-founded. This fact can sometimes provide an easy proof that a relation is well-founded.

Lexicographic orderings. Let $(A, <_A)$ and $(B, <_B)$ be two binary relation. The *lexicographic combination* of these relations $(A \times B, <_{A,B})$ is defined as

$$(x, y) <_{A,B} (u, v) \Longleftrightarrow x <_A u \text{ or } x = u \text{ and } y <_B v.$$

Proposition 1.23 *Let $(A, <_A)$ and $(B, <_B)$ be well-founded binary relations. Then their lexicographic combination $(A \times B, <_{A,B})$ is well-founded.*

Proof. Analogous to Example 1.19. □

Well-quasi-orders. We introduce a notion related to that of a well-founded set. A binary relation (A, R) is a *quasi-order* if it is reflexive and transitive. A quasi-order (A, R) is a *well-quasi-order* if for every infinite sequence $a_1, a_2, a_3, \ldots$ in A there are some $m < n$ such that $R(a_m, a_n)$.

Example 1.24 $(\mathbb{N}, \leq)$ is a well-quasi-order. This is the case since every infinite sequence in $\mathbb{N}$ has a minimum.

More generally we have:

Proposition 1.25 *Let $(A, <)$ be a linear order. Define the relation*

$$x \leq y \Longleftrightarrow \neg y < x.$$

Then $(A, <)$ is wellfounded iff $(A, \leq)$ is a well-quasi-order.

Proof. Suppose that $(A, <)$ is wellfounded. Let $a_1, a_2, a_3, \ldots$ be an infinite sequence of elements of A. Then it is impossible that $a_{i+1} < a_i$ for all i. Hence $\neg(a_{i+1} < a_i)$ for some i. That is $a_i \leq a_{i+1}$.

Conversely, assume that $(A, \leq)$ is a well-quasi-order. Suppose that $a_1 > a_2 > a_3 > \cdots$ is an infinite, strictly decreasing sequence in A. Then since $\leq$ is a well-quasi-order, there are some $m < n$, such that $a_m \leq a_n$. By transtivity and linearity of $<$ it follows that $a_m = a_n$ — a contradiction. $\square$

Lemma 1.26 *In any infinite sequence $a_1, a_2, a_3, \ldots$ of natural numbers there is an infinite subsequence such that $b_1 \leq b_2 \leq b_3 \leq \cdots$.*

Proof. Let b_1 be the first minimum (say a_{i_1}) of the sequence $a_1, a_2, a_3, \ldots$. Let $b_2 = a_{i_2}$ be the first minimum of the remaining sequence $a_{i_1+1}, a_{i_1+2}, a_{i_1+3}, \ldots$. Let $b_3 = a_{i_3}$ be the first minimum of the remaining sequence $a_{i_2+1}, a_{i_2+2}, a_{i_2+3}, \ldots$ and so on. Clearly $b_1, b_2, b_3, \ldots$ forms an increasing subsequence of the given sequence. $\square$

Example 1.27 The relation $P(x, y)$: x is a substring of a permutation of y is a quasi-order on $L = \{0, 1\}^*$. For instance, we have $P(1010, 010010)$ but $\neg P(1011, 010010)$.

It is more difficult to see that P is actually a well-quasi-order. Let $s(a, x)$ denote the number of occurrences of a in x. Note that $P(x, y)$ iff $s(0, x) \leq s(0, y)$ and $s(1, x) \leq s(1, y)$. Suppose now that $u_1, u_2, u_3, \ldots$ is a given sequence strings in L. Consider the sequence $s(0, u_1), s(0, u_2), s(0, u_3), \ldots$ of natural numbers. Then by Lemma 1.26 there is a subsequence $v_1, v_2, v_3, \ldots$ of the given sequence such that

$$s(0, v_1) \leq s(0, v_2) \leq s(0, v_3) \leq \cdots$$

Since $(\mathbb{N}, \leq)$ is a well-quasi-order there is in this sequence some $m < n$ such that $s(1, v_m) \leq s(1, v_n)$. But then $P(v_m, v_n)$ which was to be proven. $\square$

This result can be generalised to arbitrary finite alphabets (See Exercises). Thus if you have an infinite row of books (which may arbitrary thick) there is always some book whose text may be obtained by cutting out letters from another book and rearranging them. Even more amazingly you do not have to rearrange the letters:

Proposition 1.28 *Let Σ be a finite alphabet. Define the relation on the set Σ^* of strings:*

$$K(u,v) \iff u \text{ is obtained by removing 0 or more symbols from } v.$$

Then K is a well-quasi-order.

Proof. A sequence $u_1, u_2, u_3, \ldots$ of strings in Δ^* is called *bad* if $\neg K(u_m, u_n)$ for all $m < n$. Suppose that K is not a well-quasi-order, i.e. that there is a bad sequence. Let v_1 be a shortest string which is the first term of a bad sequence. Then let v_2 be a shortest string such that v_1, v_2 are the first two terms of a bad sequence. More generally, let v_n be a shortest string such that $v_1, v_2, \ldots, v_n$ are the first n terms of a bad sequence. Clear each v_i is a non-empty string. Write $v_i = a_i w_i$ where $a_i \in \Delta$. Since Δ is finite, some symbol occurs as initial symbol in infinitely many of the strings v_i. Let k_1 be the least such that a_{k_1} occurs infinitely often as initial symbol. Suppose that $k_1 < k_2 < k_3 < \ldots$ are the indices of strings that begin with a_{k_1}. Then

$$a_1 w_1, a_2 w_2, \ldots, a_{k_1-1} w_{k_1-1}, w_{k_1}, w_{k_2}, w_{k_3}, \ldots$$

is a bad sequence, since all v_{k_i} begin with the same symbol a_{k_1} which is different from $a_1, \ldots, a_{k_1-1}$. But now w_{k_1} is one symbol shorter than v_{k_1}, contradicting the construction of v_{k_1}.

Hence there are no bad sequences. $\square$

Kruskal's Theorem

This theorem is very useful proving termination of term rewriting systems.

Let $\mathbb{T}$ be the set of terms formed in the following way:

(a) $m \in \mathbb{T}$ for any $m \in \mathbb{N}$,

(b) If $m \in \mathbb{N}$, and $t_1, \ldots, t_k \in \mathbb{T}$ then $m(t_1, \ldots, t_k) \in \mathbb{T}$.

Thus 3, $0(1, 0(2))$, $2(1, 3(2, 2, 1(0)))$ are some examples of such terms.

Define the following quasi-order on $\mathbb{T}$: $t \preceq s$ if s can be obtained from t by 0 or more of the following operations

(a) replace a subterm r by $m(r)$ for some $m \in \mathbb{N}$,

(b) increase the value of some number

(c) permute the arguments of some subterm $m(r_1, \ldots, r_n)$, i.e. replace $m(r_1, \ldots, r_n)$ by $m(r_{\pi(1)}, \ldots, r_{\pi(n)})$ for some permutation π

(d) replace some subterm $m(r_1, \ldots, r_n)$ by $m(r_1, \ldots, r_n, k)$ for some $k \in \mathbb{N}$. (In case $n = 0$, we replace m by $m(k)$.)

Example 1.29 We have $0(1,0(2)) \preceq 2(1,0(2)) \preceq 2(1,3(2)) \preceq 2(1,3(2,2)) \preceq 2(1,3(2,2,1)) \preceq 2(1,3(2,2,1(0)))$. But $1(1,1) \not\preceq 1(1,0)$.

Theorem 1.30 (Kruskal) *$\preceq$ is a well-quasi-order on $\mathbb{T}$.*

The proof is difficult and beyond the scope of this course. However it uses the technique of *minimal bad sequences* illustrated in Lemma 1.26 and Proposition 1.28.

Exercises

1. Show that the following program $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ terminates by using a lexicographic combination

$$\begin{aligned} f(0,y) &= y \\ f(S(x),0) &= S(x) \\ f(S(x),S(y)) &= S(f(x,f(x,y))). \end{aligned}$$

2. Prove Proposition 1.22.

3. Show that $\sqsubset$ is the transitive closure of the immediate subterm relation on $\mathrm{Ter}(\Sigma,\mathbb{X})$.

4. Extend Example 1.27 to strings over any finite alphabet. (What does Proposition 1.28 say here?)

5. Prove that the substring relation over $\{0,1\}^*$ is not a well-quasi-order.

6. Find all $t \in \mathbb{T}$ such that $t \preceq 2(1,1(2,1(0)))$.

1.4 Abstract reduction systems

An *Abstract Reduction System* (ARS) is a set A together with a binary relation $\to$. Further on we will mostly be interested in the case where A is a set of terms and $\to$ is a one-step computation, or reduction, relation. However we treat the general case first, so $(A,\to)$ could be any directed graph, finite or infinite.

An element a in A of an ARS $(A,\to)$ is said to be a *normal form*, if there is no $b \in A$ such that $a \to b$. (Intuitively a cannot be computed further, and can be considered as the *value* of a computation.)

Example 1.31 Let $A = \{0,1,2,3\}$ and $\to = \{(1,0),(1,2),(2,1),(2,3)\}$. (Draw the graph of this ARS!) It is easy to see that the elements of normal form are exactly 0 and 3.

Example 1.32 The ARS given by $A_2 = \{0,1\}$ and $\to = \{(1,0),(0,1)\}$ has no elements of normal form.

Let $(A, \to)$ be an ARS. Denote by $\twoheadrightarrow$ the reflexive and transitive closure of $\to$, that is, $a \twoheadrightarrow b$ holds iff there is a sequence $a = a_1, \ldots, a_n = b$, $n \geq 1$, such that

$$a_1 \to a_2 \to \cdots \to a_n.$$

Write $a \to^+ b$ if this holds for a sequence where $n \geq 2$. An ARS $(A, \to)$ is *weakly normalizing (WN)* if for every $a \in A$ there is some normal form $b \in B$ with $a \twoheadrightarrow b$. It is easily checked that the ARS of Example 1.31 is weakly normalizing. Note however that $1 \twoheadrightarrow 0$ and $1 \twoheadrightarrow 3$ so that 1 has two distinct normal forms.

Two elements a and b of an ARS $(A, \to)$ are said to be *convergent* (in symbols: $a \downarrow b$) if there is some c such that $a \twoheadrightarrow c$ and $b \twoheadrightarrow c$. An ARS $(A, \to)$ is *confluent* or *Church-Rosser (CR)* if $b \downarrow c$ for any $a, b, c \in A$ such that $a \twoheadrightarrow b$ and $a \twoheadrightarrow c$ The following simple result shows the importance of this property.

Proposition 1.33 *Let $(A, \to)$ be a confluent, weakly normalizing ARS. Then every element of A has a unique normal form.*

Proof. Suppose that b and c are normal forms and $a \twoheadrightarrow b$ and $a \twoheadrightarrow c$. By confluency, for some $d \in A$ with $b \twoheadrightarrow d$ and $c \twoheadrightarrow d$. Since b is normal, $b = d$ and likewise $c = d$. Hence $b = c$. $\square$

An ARS $(A, \to)$, where there are no infinite sequences $a_1 \to a_2 \to a_3 \to \cdots$ is called *strongly normalizing (SN)*, i.e. $(A, \leftarrow)$ is a wellfounded relation. Clearly, in this case any strategy of performing the reductions will lead to a normal form. The ARS of Example 1.31 is does not have this property since there is the sequence $1 \to 2 \to 1 \to 2 \to \cdots$.

Example 1.34 The ARS given by $A = \{0, 1, 2, 3\}$ and $\to = \{(1, 0), (1, 2), (2, 3)\}$ is strongly normalizing but not confluent.

The following theorem is often useful when proving confluency. An ARS $(A, \to)$ is *weakly confluent* or *Weakly Church-Rosser (WCR)* if $b \downarrow c$ for any $a, b, c \in A$ such that $a \to b$ and $a \to c$. (Note the one-step computation relations from a.)

Theorem 1.35 *(Newman's lemma) A weakly confluent, strongly normalizing ARS is confluent.*

Proof. Let $(A, \to)$ be an ARS. That it is confluent is equivalent to $P(u)$ for all u, where

$$P(u) \Leftrightarrow_{\text{def}} (\forall x, y)[u \twoheadrightarrow x \wedge u \twoheadrightarrow y \Rightarrow x \downarrow y]$$

Since the ARS is strongly normalizing, we can prove $(\forall u)\, P(u)$ by Noetherian induction. For this it suffices to show that $S = \{u \in A : P(u)\}$ is a progressive set, i.e.

$$(\forall u)[(\forall t)\,(u \to t \Rightarrow P(t)) \Rightarrow P(u)].$$

So assume that $u \in A$ is arbitrary, and as induction hypothesis $(\forall t)\,(u \to t \Rightarrow P(t))$. In case u is normal, we are done. Otherwise, suppose that $u \to b \twoheadrightarrow x$ and $u \to c \twoheadrightarrow y$.

By weak confluency there is some d such that $b \twoheadrightarrow d$ and $c \twoheadrightarrow d$. By the induction hypothesis $P(b)$, so there is a z with $x \twoheadrightarrow z$ and $d \twoheadrightarrow z$. By transitivity, $c \twoheadrightarrow z$. Using the induction hypothesis again, $P(c)$ holds, so there is some v with $z \twoheadrightarrow v$ and $y \twoheadrightarrow v$. Thus by transitivity, $x \twoheadrightarrow v$. The induction step is finished. $\square$

The following result can sometimes be used to prove that an ARS is strongly normalising.

Theorem 1.36 *Let $(A, \to)$ be an ARS such that $(A, \to^+)$ is irreflexive. Suppose that there is a well-quasi-order $(A, \preceq)$ such that for all $s \neq t \in A$:*

$$s \preceq t \Longrightarrow t \to^+ s.$$

Then $(A, \to)$ is strongly normalising.

Proof. Suppose to the contrary that the ARS is not strongly normalising. Then there is an infinite sequence in A so that

$$s_1 \to s_2 \to s_3 \to \cdots$$

Hence since $\preceq$ is a well-quasi-order, there are $m < n$ such that $s_m \preceq s_n$, and $s_m \neq s_n$. Hence $s_n \to^+ s_m$ by the assumption. By transtivitity it follows that $s_n \to^+ s_n$, contradicting the irreflexivity assumption. $\square$

References

J.A. Goguen and G. Malcolm. *Algebraic Semantics of Imperative Programming Languages.* MIT Press, 1996.

K. Meinke and J.V. Tucker. Universal Algebra. In: S. Abramsky *et al.* (eds.): *Handbook of Logic in Computer Science, Vol. 1.* Oxford University Press 1992.

W. Wechler. *Universal Algebra for Computer Scientists.* Springer 1992.