3.3b Term Rewriting Systems (Part 2)

Dienstag, 10. November 2015 09:30

So we need equivalent TRSs with additional properties: termination + confluence.

Termination of a TRS $\mathcal{R}$ means that $\to_{\mathcal{R}}$ must be well founded.

Def 3.38 (Well-founded Relation)

Let $\to$ be relation on a set M. The relation $\to$ is well founded iff there is no infinite sequence of elements $t_0, t_1, \ldots \in M$ with $t_0 \to t_1 \to t_2 \to \ldots$.

Ex 3.39

- $>_{\mathbb{N}}$ (greater-relation on $\mathbb{N}$) is well founded
- $<_{\mathbb{N}}$ is not well founded ($0 < 1 < 5 < 100 < \ldots$)
- $>_{\mathbb{Z}}$ ($1 > 0 > -1 > -2 > \ldots$)
- $>_{\mathbb{Q}^+}$ (greater on positive rational numbers) ($1 > \frac{1}{2} > \frac{1}{4} > \frac{1}{8} > \ldots$)
- $\rhd$ (proper subterm relation) is well founded

$f(g(h(x)), y) \rhd g(h(x)) \rhd h(x) \rhd x$

- $\rhd$ is not well founded

- $\trianglerighteq$ is not well founded ✗
  ($t \trianglerighteq t \trianglerighteq t \trianglerighteq \dots$)

Our goal is to use TRSs, where every term s can be reduced to a normal form s_n.

Def 3.3.10 (Normal Form)

Let $\rightarrow$ be a relation on a set M. An element $q \in M$ is a normal form iff there is no $q' \in M$ with $q \rightarrow q'$.

An element q is a normal form of t iff
$t \rightarrow^* q$ and q is a normal form.

If the normal form of t is unique, then $t\downarrow$ denotes the normal form of t.

A relation $\rightarrow$ is normalizing iff every element t has (at least) one normal form.

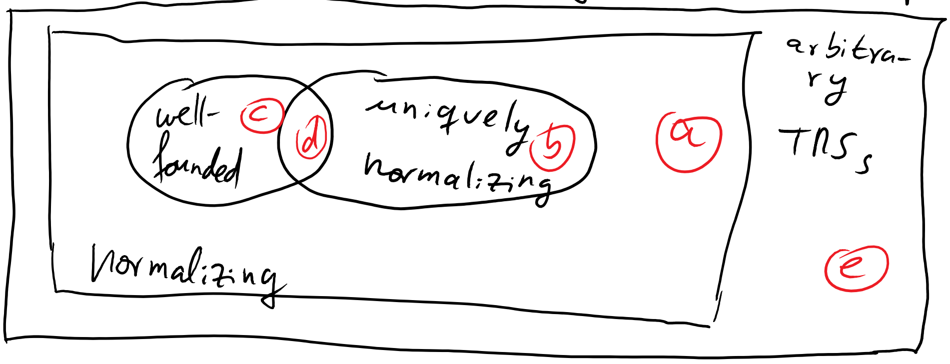
A relation $\rightarrow$ is uniquely normalizing iff every element t has exactly one normal form.

Lemma 3.3.11. (Well Foundedness $\Rightarrow$ Normalizing)

Every well founded relation is normalizing.

Proof: If $t \in M$ had no normal form, then
$t \rightarrow t_1 \rightarrow t_2 \rightarrow \dots$ which contradicts well-foundedness of "$\rightarrow$" ∎

Connection between all of these properties



Ex 3.3.13 (a) $\{b \to a, b \to f(b)\}$ is normalizing, but not terminating and not uniquely normalizing.

$$\begin{array}{ccccccccc} b & \to & f(b) & \to & f(f(b)) & \to & f^3(b) & \to & \dots \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\ a & & f(a) & & f^2(a) & & f^3(a) & & \end{array}$$

E.g.: b has the normal forms $a, f(a), \dots$

(b) $\{b \to a, b \to b\}$ is uniquely normalizing, but not terminating

(c) $\{b \to a, b \to c\}$ is terminating, but not uniquely normalizing

(d) $\{a \to b, c \to b\}$ is terminating and uniquely normalizing

(e) $\{b \to f(b)\}$ is not normalizing, as b has no normal form

Def 3.3.12. (Termination of TRSs)

A TRS $\mathcal{R}$ is terminating iff $\to_{\mathcal{R}}$ is well founded.

A TRS $\mathcal{R}$ is (uniquely) normalizing iff $\to_{\mathcal{R}}$ is (uniquely) normalizing.

For that reason, we will introduce techniques to prove termination of TRSs in Chapter 4.

Motivation: Word problem, but also many applications in program analysis + verification.

In addition to termination, we want unique normalization. Reason: otherwise the algorithm for the word problem could return "False" although $s \equiv_{\mathcal{E}} t$ holds.

More precisely: We want that $s \leftrightarrow^*_{\mathcal{R}} t$ implies that $s \to^*_{\mathcal{R}} q$ and $t \to^*_{\mathcal{R}} q$ for some term q.

This property was proven for the lambda calculus for the first time by Church + Rosser.

<u>Def 3.3.14.</u> (Church-Rosser Property, Joinability)

Let $\to$ be a relation on a set M.

Two elements $s, t \in M$ are <u>joinable</u> (denoted $s \downarrow t$) iff $s \to^* q \leftarrow^* t$ for some $q \in M$.

The relation $\to$ has the <u>Church-Rosser property</u> iff for all $s, t \in M$: if $s \leftrightarrow^* t$, then $s \downarrow t$.

A TRS $\mathcal{R}$ has the <u>Church-Rosser property</u> iff $\to_{\mathcal{R}}$ has the Church-Rosser property.

Ex 3.3.15. $\mathcal{R} = \{ b \to c \quad b \to a \}$

Ex 3.3.15. $\mathcal{R} = \{ b \to c,\ b \to a \}$

$a \xleftrightarrow{*}_{\mathcal{R}} c$, because $a \leftarrow_{\mathcal{R}} b \to_{\mathcal{R}} c$

But $a \not\downarrow_{\mathcal{R}} c$, since a and c are already in normal form.

To check whether a TRS has the CR-property, it is easier to regard a simpler property: confluence.
Here, one has to check whether every indeterminism can be resolved again.

$$\begin{tikzcd}
 & p \arrow[dl, "*"'] \arrow[dr, "*"] & \\
s \arrow[dr, "*"'] & & t \arrow[dl, "*"] \\
 & q &
\end{tikzcd}$$

if the black arrows hold,
do the green arrows hold as well?

Def 3.3.16. (Confluence)

A relation $\to$ on a set M is confluent iff for all $p, s, t \in M$:

If $p \to^* s$ and $p \to^* t$,
then there exists a $q \in M$ such that $s \to^* q$ and $t \to^* q$.

How is confluence related to the CR-property?

Thm 3.3.17. (CR property $\Leftrightarrow$ Confluence)

A relation $\to$ has the CR property iff $\to$ is confluent.

Proof: "$\Rightarrow$" $\begin{tikzcd} & p \arrow[dl, "*"'] \arrow[dr, "*"] & \\ s & & t \end{tikzcd}$ $\rightsquigarrow$ $s \leftrightarrow^* t$ $\overset{\text{CR property}}{\rightsquigarrow}$ $\begin{tikzcd} s \arrow[dr, "*"'] & & t \arrow[dl, "*"] \\ & q & \end{tikzcd}$.

"$\Leftarrow$": Let $\to$ be confluent.

Let $s \leftrightarrow^* t$, i.e., $s \leftrightarrow^n t$ for some $n \in \mathbb{N}$.

We prove that $s \downarrow t$ by induction on n.

Ind. Base: $n=0$

$s \leftrightarrow^0 t \rightsquigarrow s = t \rightsquigarrow s \downarrow t$, since $\begin{tikzcd} s \arrow[dr, "*"'] & & t \arrow[dl, "*"] \\ & s & \end{tikzcd}$

Ind. Step: $n>0$

$$\begin{tikzcd} s \arrow[r, leftrightarrow, "n-1"] \arrow[dr, "*"'] & s' \arrow[r, leftrightarrow] \arrow[d, "*"] & t \\ & q & \end{tikzcd}$$

by the ind. hyp.

Case 1: $t \to s'$

$$\begin{tikzcd} s \arrow[r, leftrightarrow, "n-1"] \arrow[dr, "*"'] & s' \arrow[d, "*"] & t \arrow[l] \\ & q & \end{tikzcd}$$

Therefore $s \downarrow t$.

Case 2: $s' \to t$

$$\begin{tikzcd} s \arrow[r, leftrightarrow, "n-1"] \arrow[dr, "*"'] & s' \arrow[r] \arrow[d, "*"] & t \arrow[ddl, "*"] \\ & q \arrow[d, "*"'] & \\ & p & \end{tikzcd}$$

because $\to$ is confluent and $\begin{tikzcd} & s' \arrow[dl, "*"'] \arrow[dr] & \\ q & & t \end{tikzcd}$

Therefore $s \downarrow t$.

$\blacksquare$

Thus: it suffices to check confluence.

Confluence is needed for programs in order to guarantee that computations have a unique result (i.e., that normal forms are unique).

Lemma 3.3.18. (Confluence means unique normal forms)

(a) If $\to$ is confluent, then every object has at most one normal form.

(b) If $\to$ is normalizing + confluent, then every object has exactly one normal form (i.e., $\to$ is uniquely normalizing).

(c) If $\to$ is uniquely normalizing, then $\to$ is confluent.

Proof: (a) Assume that t has 2 normal forms q_1 and q_2.

$$\begin{tikzcd} & t \arrow[dl, "*"'] \arrow[dr, "*"] & \\ q_1 \arrow[dr, "*"'] & & q_2 \arrow[dl, "*"] \\ & q & \end{tikzcd}$$

because of confluence

Since q_1, q_2 are normal forms, we have
$$q_1 = q = q_2.$$

(b) follows from (a) by the definition of "normalizing"

(c) Let $\to$ be uniquely normalizing.

Let
$$\begin{tikzcd} & p \arrow[dl, "*"'] \arrow[dr, "*"] & \\ s \arrow[d, "*"'] & & t \arrow[d, "*"] \\ s\downarrow & = & t\downarrow \end{tikzcd}$$
.

$\}$ $s\downarrow$ and $t\downarrow$ are the unique normal forms of s and t

So p has the normal forms $s\downarrow$ and $t\downarrow$.

By unique normalization, we have $s\downarrow = t\downarrow$.

The following theorem states that to check $s \leftrightarrow^* t$ one only has to check whether the normal forms of s and t

are equal (provided that $\to$ is normalizing + confluent).

<u>Thm 33.19</u> (Checking $\leftrightarrow^*$ by Normal Forms)

Let $\to$ be normalizing + confluent. Then: $s \leftrightarrow^* t$ iff $s{\downarrow} = t{\downarrow}$.

<u>Proof</u>: "$\Leftarrow$": $s{\downarrow} = t{\downarrow} \;\leadsto\; s \downarrow t \;\leadsto\; s \leftrightarrow^* t$

"$\Rightarrow$": $s \leftrightarrow^* t \;\leadsto\; s \downarrow t \;\leadsto$

$$\begin{tikzcd} s \arrow[dr, "*"'] & & t \arrow[dl, "*"] \\ & q \arrow[d, "*"] & \\ & q{\downarrow} & \end{tikzcd} \;\leadsto\; \underbrace{s{\downarrow} = q{\downarrow} = t{\downarrow}}$$

since $\to$ is confluent which is equivalent to the CR-property (Thm 33.17)

since $\to$ is normalizing

since $\to$ is uniquely normalizing by Lemma 33.18

☒

If $\to$ is only normalizing (but not well founded), then it could be difficult to find the normal form of a term s.

$s \to^* s{\downarrow}$, but s might also start infinite rewrite sequences

Therefore, we require that $\to$ should be well founded and confluent.

<u>Def 33.20</u> (Convergence of TRSs)

A TRS is <u>convergent</u> iff it is terminating and confluent.

<u>Ex 33.21</u> Convergent TRSs correspond to an interpreter that evaluate expressions to a result.

The plus-TRS is convergent. It can be used to evaluate expressions with "plus".

$\text{plus}(2,1) \hat{=} \text{plus}(s(s(0)), s(0))$

Evaluation will terminate with the unique result

$$3 \hat{=} s(s(s(0)))$$

Convergent TRSs can not only be used to <u>compute</u> results, but also to <u>prove</u> equations:

Algorithm WORD_PROBLEM $(\mathcal{R}, s, t)$

<u>Thm 3.3.22.</u> (Correctness of Alg. WORD_PROBLEM)

(a) The alg. WORD_PROBLEM terminates.

(b) If $\mathcal{R}$ is equivalent to $\mathcal{E}$, then the alg W_P is correct.

(c) If a set of equations $\mathcal{E}$ has an equivalent convergent TRS, then the word problem for $\mathcal{E}$ is decidable.

<u>Proof</u>: (a) Termination follows from termination of $\mathcal{R}$.

(b) $s \equiv_{\mathcal{E}} t$ iff $s \overset{*}{\longleftrightarrow}_{\mathcal{E}} t$ iff $s \overset{*}{\longleftrightarrow}_{\mathcal{R}} t$ iff $s\downarrow_{\mathcal{R}} = t\downarrow_{\mathcal{R}}$.

Thm of Birkhoff, Thm 3.1.14 — $\mathcal{E}$ and $\mathcal{R}$ are equivalent — Thm 3.3.19

(c) follows from the fact that W_P is a decision procedure. (This alg. can be executed automatically. An algorithm for matching will be presented in Section 5.) ☒

<u>Ex 3.3.23.</u> Group Example

$\mathcal{E} = \{ f(x, f(y,z)) \equiv f(f(x,y),z),\ f(x,e) \equiv x,\ f(x,i(x)) \equiv e \}$

Orienting these equations yields a TRS with 3 rules.

Using algorithm WORD_PROBLEM to check

$i(i(n)) \equiv_\mathcal{E} n$ yields "False"

Problem: The 3-rule TRS is not confluent:

$$f(x, \underline{f(y, i(y))})$$

$$f(f(x,y), i(y)) \leftarrow f(x, f(y,i(y))) \rightarrow f(x, e) \rightarrow x$$

2 different normal forms

<u>Solution</u>: Extend the TRS by additional rules in order to make it confluent: "<u>Completion</u> of TRSs".

General idea: whenever there is an indeterminism that can't be joined, add this indeterminism as an additional rule.

Here: add the rule $f(f(x,y), i(y)) \rightarrow x$

We will later introduce techniques to complete TRSs automatically (i.e., the 10-rule TRS is generated automatically from the 3 group axioms).

The 10-rule TRS is convergent $\Rightarrow$ it is a decision procedure for groups.

E.g.: check whether $i(f(i(u), f(v,u))) \equiv_\mathcal{E} f(i(u), f(i(v), u))$

So we now have a decision procedure for equations about groups.

So our goal is to have decision procedures for statements about algorithms or data structures.

Moreover, we want to synthesize these decision procedures automatically: see flowchart

Tasks

- Termination of TRSs (Sect. 4)
- Confluence — " — (Sect. 5)
- Completion — " — (Sect. 6)