

Term Rewriting Systems

Lecture

Tuesdays and Thursdays 16:40 – 18:10
(Rafael Peñaloza)

Tutorial

Wednesdays 14:50 – 16:20
(Marcel Lippmann)

<http://lat.inf.tu-dresden.de/teaching/ss2014/TRS/>

Literature

Franz Baader and Tobias Nipkow

Term Rewriting and *all That*

Cambridge University Press (1998)

<http://www4.informatik.tu-muenchen.de/~nipkow/TRaAT/>

Term Rewriting

Terms

Expressions built from **variables**, **constant symbols**, and **function symbols**

Example:

- **variables** x, y
- **constant symbol** 0
- **function symbols** s (unary) and $+$ (binary; infix notation)

$$0, \quad x + s(0), \quad s(s(s(0)) + 0)$$

Rewriting

Rules that express how can one term be transformed into another

Examples

Rewriting as computation mechanism

rules are applied in **one direction**
computes normal forms

- closely related to **functional programming**
- for example: **symbolic differentiation**

Rewriting as deduction mechanism

rules are applied in **both directions**
defines equivalence classes of terms

- **equational theory** in automated deduction
- for example: **group theory**

Symbolic Differentiation

Terms are **arithmetic expressions** built with the operations:

- **+** (binary function symbol)
- ***** (binary function symbol)

the indeterminates **X, Y** (constant symbols) and
the numbers **0, 1** (constant symbols).

Example: $((X+X)*Y)+1$

Additionally, the unary function symbol D_X : **partial derivative** w.r.t. X

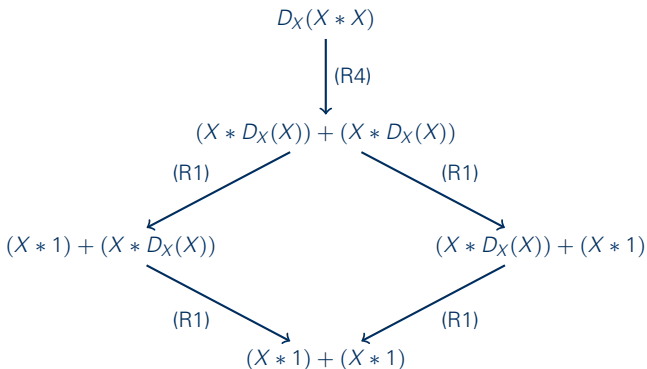
Rules for Symbolic Differentiation

$$(R1) \quad D_X(X) \longrightarrow 1$$

$$(R2) \quad D_X(Y) \longrightarrow 0$$

$$(R3) \quad D_X(u + v) \longrightarrow D_X(u) + D_X(v)$$

$$(R4) \quad D_X(u * v) \longrightarrow (u * D_X(v)) + (v * D_X(u))$$



Important Properties of Term Rewriting Systems

Termination

is it **always** true that after **finitely many** rule applications, we reach an expression that **cannot** be rewritten anymore? (**normal form**)

For the rules (R1) – (R4) this is the case (every rewriting execution terminates)

How to **prove** this?

$$D_X(u * v) \longrightarrow (u * D_X(v)) + (v * D_X(u))$$

Example

$$u + v \longrightarrow v + u$$

is **non-terminating**. Leads to an infinite sequence of applications

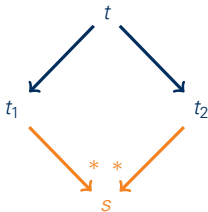
$$X + Y \longrightarrow Y + X \longrightarrow X + Y \longrightarrow Y + X \longrightarrow X + Y \longrightarrow Y + X \longrightarrow \dots$$

Important Properties of Term Rewriting Systems

Confluence

If **different rules** can be applied to a given term t , leading to different terms t_1 and t_2 , can t_1 and t_2 be **joined**?

is there a term s that can be reached from t_1 and t_2 by rule applications?

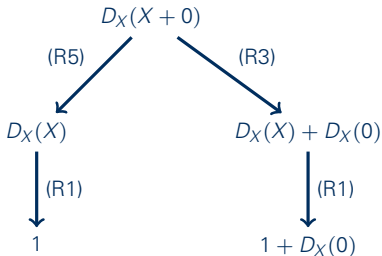


Confluence

Rules (R1) – (R4) are **confluent**

How to **prove** this?

If we add the rule: (R5) $u + 0 \rightarrow u$ to this system, we **lose confluence**



Confluence **regained** by adding the rule

$D_X(0) \rightarrow 0$

Group Theory

Consider

- \circ a binary function symbol
- i a unary function symbol
- e a constant symbol, and
- x, y, z variable symbols

The class of groups is defined by the identities

(G1)	$(x \circ y) \circ z \approx x \circ (y \circ z)$	(\circ is associative)
(G2)	$e \circ x \approx x$	(e is a left-unit)
(G3)	$i(x) \circ x \approx e$	(i is left-inverse)

Identities are rules that can be applied in both directions

Word Problem

Given identities E and terms s, t , can s be rewritten into t using E ?

Example

We can prove that i is also a **right-inverse**; i.e. e can be rewritten to $x \circ i(x)$

$$(G1) \quad (x \circ y) \circ z \approx x \circ (y \circ z)$$

$$(G2) \quad e \circ x \approx x$$

$$(G3) \quad i(x) \circ x \approx e$$

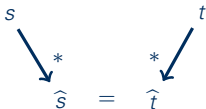
$$\begin{aligned} e &\stackrel{G3}{\approx} i(x \circ i(x)) \circ (x \circ i(x)) \\ &\stackrel{G2}{\approx} i(x \circ i(x)) \circ (x \circ (e \circ i(x))) \\ &\stackrel{G3}{\approx} i(x \circ i(x)) \circ (x \circ ((i(x) \circ x) \circ i(x))) \\ &\stackrel{G1}{\approx} i(x \circ i(x)) \circ ((x \circ (i(x) \circ x)) \circ i(x)) \\ &\stackrel{G1}{\approx} i(x \circ i(x)) \circ (((x \circ i(x)) \circ x) \circ i(x)) \\ &\stackrel{G1}{\approx} i(x \circ i(x)) \circ ((x \circ i(x)) \circ (x \circ i(x))) \\ &\stackrel{G1}{\approx} (i(x \circ i(x)) \circ (x \circ i(x))) \circ (x \circ i(x)) \\ &\stackrel{G3}{\approx} e \circ (x \circ i(x)) \\ &\stackrel{G2}{\approx} x \circ i(x) \end{aligned}$$

Word Problem

Given identities E and terms s, t , can s be rewritten into t using E ?

Idea

Solve the word problem using rewriting in one direction



Problems

- equivalent terms can have different normal forms
- normal forms may not exist

Termination and confluence ensure the existence and uniqueness of normal forms