1. Introduction

Dienstag, 20. Oktober 2015 08:30

Term Rewrite Systems (Termersetzungs-Systeme)

Applications

- Specification of Programs
  (specify and execute algebraic specifications)
- program analysis + verification
- execution of programs
  (term rewriting is already a full prog. language, is the basis of functional programming)
- symbolic computation

Ex: Addition of natural numbers

Express natural numbers by terms over
0 and succ
($0 \hat{=} 0$, $succ(0) \hat{=} 1$, $succ(succ(0)) \hat{=} 2$, ...)

Addition algorithm:

$$plus(0, y) \equiv y$$

$$plus(succ(x), y) \equiv succ(plus(x, y))$$

(Functional program, looks like Haskell)

Execution of such a program:

apply equations from left to right =
term rewriting

Compute "2+1":

plus(succ(succ(0)), succ(0)) →

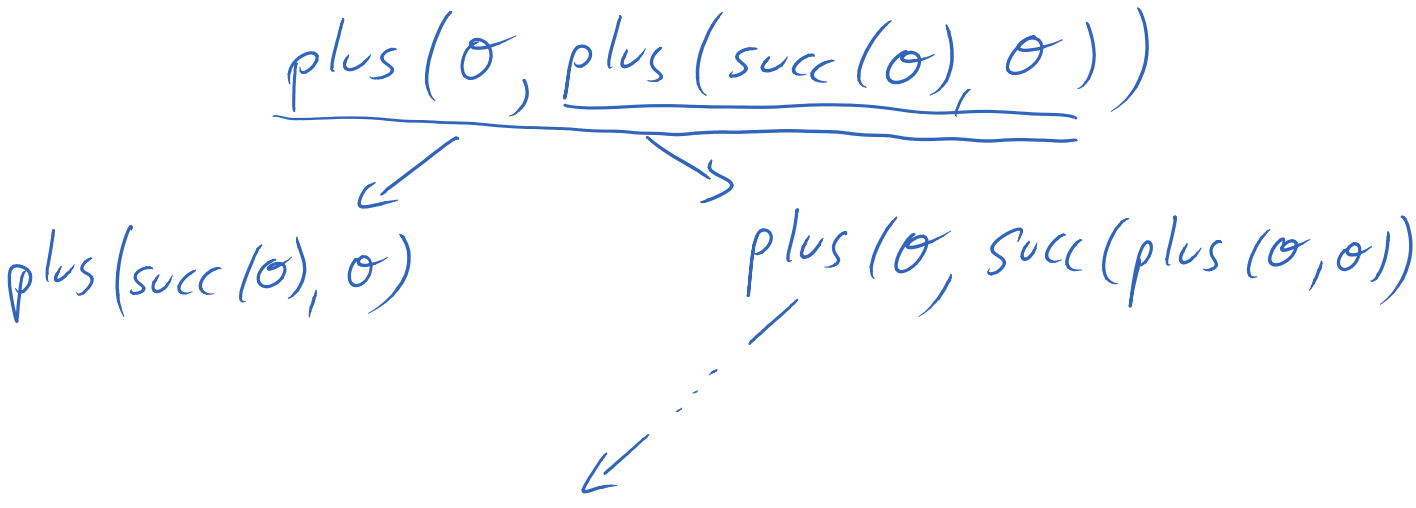
succ(plus(succ(0), succ(0))) →

succ(succ(plus(0, succ(0)))) →

succ(succ(succ(0)))

Topics of the lecture:

- Is the result of a program deterministic?
  (Confluence)



This example is confluent. How can one check confluence / determinism of programs automatically?

- Does the program always stop after finitely many steps? (Termination)

Our plus-example is terminating.
But one could add:

$$plus(x,y) \equiv plus(y,x)$$

Now the prog. is no longer terminating:

$plus(0, succ(0)) \to plus(succ(0), 0) \to$
$plus(0, succ(0)) \to \ldots$

How can one check termination automatically?

- Does a program satisfy its specification? (Correctness)

plus-program could satisfy properties like:

$$plus(succ(succ(0)), x) \equiv plus(succ(0), succ(x))$$

$$plus(plus(x,y), z) \equiv plus(x, plus(y,z))$$

Do these properties hold for the plus-program, i.e., do they follow from the 2 plus-equations?

A related question are properties of abstract data types.

Ex: Data types of groups

associativity: $f(x, f(y,z)) \equiv f(f(x,y), z)$

neutral element: $f(x, e) \equiv x$

inverse fct. i: $f(x, i(x)) \equiv e$

Does every group satisfy

$$i(i(u)) \equiv u \quad ?$$

i.e: Does this equation follow from the 3 group axioms? (Yes)

How can one find this out automatically?

- How can one make an "incomplete" program complete? (Completion)

  If a program or a data type are incompletely specified, how can one extend them in a suitable way?

Structure of the lecture:

1. Introduction
2. Preliminaries (Syntax + Semantics of terms and equations)
3. Term Rewriting and Deduction of Equations
4. Termination of Term Rewriting

5. Confluence — " ———

6. Completion — " ———

Organisation

- Lecture in English
- German Course Notes (on the web)
- English Notes from the lecture available on the web
- very old video of parts of the lecture (on the web)
- Web site: http://verify.rwth-aachen.de/tes15
- Lecture on TUE + THU

  Ex. Course on FRI (Florian Frohn)

  This week: 3 lectures (lecture instead of ex. course on FRI)
- Lecture for Master (Informatik, SSE, Math) and Bachelor (Informatik)

  ↑
  - Wahlpflicht Theorie or
  - Vorgezogene Masterprüfung

  Bachelor: register for exam at ZPA,

t. - 17. 12. 15

- Exercises
  - weekly exercise sheets
  - solve in groups of 2
  - submit at beginning of ex. course or in box (E1, 2nd floor)
  - 50% of points needed to participate in the exam
  - date of exam (written): Feb 22, 2016
    2nd exam : March 18, 2016
  - to participate in exercises:
    please register via our website
    (≠ registering for exam
    ≠ L2P)
    until Friday, Oct. 23