TERM REWRITING & ... useless

classmate
Date 7.9.2019
Page 1

baader & nipkow

PREFACE

jump to ch1?

who said ↓ baader

term rewriting = logic
+ universal algebra
+ automated theorm proving
+ functional pgming.

journals. ⟶ symbolic computation
automated reasoning. } journal of R

abstract reduction systems
combining word problems

prerequisites:
discrete mathematics
(linear) algebra
theoretical computer science.

SURPRISE EXAMPLES

EQUATIONAL REASONING

pen...

classes of algebra → groups, rings.

we want to define addition of natural numbers using (constant) 0 and successor funtion s

$$x + 0 \approx x$$
$$x + s(y) \approx s(x+y)$$

identities

$$1 + 2$$
$$\approx s(0) + s(s(0))$$
$$\approx s(s(0) + s(0))$$
$$\approx s(s(s(0) + 0))$$
$$\approx s(s(s(0)))$$
$$\approx 3$$

interpreting identities as rewrite rules

term rewriting systems

- variables
- constant symbols
- function symbols

in general,
identities can be applied in both directions

rewrite rules — computation mechanism

$\rightarrow$ instead of $\approx$ — deduction mechanism

A VERY BASIC EXAMPLES THAT THE CHICKENS CALL SYMBOLIC DIFFERENTIATION

R1 $D_x(X) \rightarrow 1$

R2 $D_x(Y) \rightarrow 0$

R3 $D_x(u+v) \rightarrow D_x(u) + D_x(v)$

R4 $D_x(u * v) \rightarrow (u * D_x(v)) + (D_x(u) * v)$

$$D_x(x * x)$$

$$\downarrow R4$$

$$(x * D_x(x)) + (D_x(x) * x)$$

$$\swarrow R1 \qquad\qquad \searrow R1$$

$$(x * 1) + (D_x(x) * x) \qquad\qquad (x * D_x(x)) + (1 * x)$$

$$\searrow R1 \qquad\qquad \swarrow R1$$

$$(x * 1) + (1 * x)$$

< symolic differentiation of $D_x(x * x)$. >

TERMINATION

expression to which no more rules apply
→ normal form.

R1 – R4 } terminating (but now?)

ex (non terminating rule) →

$u + v \rightarrow v + u$ commutativity of addition

non-termination may not be because of just one rule it could also result from the interaction of several rules.

CONFLUENCE

if there are different ways of applying rules to a given term t, leading to different terms t_1 and t_2, can t_1 and t_2 be joined, i.e., can we find a common term s that can be reached from t_1 and t_2 by rule application.

$$\begin{tikzcd} & t_1 \arrow[dr] & \\ t \arrow[ur] \arrow[dr] & & s \\ & t_2 \arrow[ur] & \end{tikzcd}$$

R1 – R4 : confluent
but now?

if we add the simplification rule:

R5 $u + 0 \rightarrow u$

we lose the confluence property

$$
\begin{tikzcd}
 & D_x(x+0) \arrow[dl, "R5"'] \arrow[dr, "R3"] & \\
D_x(x) \arrow[d, "R1"] & & D_x(x) + D_x(0) \arrow[d, "R1"] \\
1 & & 1 + D_x(0)
\end{tikzcd}
$$

non confluence of R1 - R5 can be overcome by adding

R6 $D_x(0) \rightarrow 0$

can we always make a non-confluent system confluent by adding implied rules (completion of term rewriting systems).

R1-R4 → functional program

termination → D_x is a total function.

confluence → computation independent of strategy

all term rewriting systems that constitute functional programs are confluent.

GROUP THEORY

$\circ$ binary function symbol

i unary function symbol

e constant symbol

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

G2 $e \circ x \approx x$

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

e

$\approx i(x \circ i(x)) \circ (x \circ i(x))$

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

$\approx i(x \circ i(x)) \circ (x \circ ((i(x) \circ x) \circ i(x)))$

$\approx i(x \circ i(x)) \circ ((x \circ (i(x) \circ x)) \circ i(x))$

$\approx i(x \circ i(x)) \circ (((x \circ i(x)) \circ x) \circ i(x))$

$\approx (i(x \circ i(x)) \circ ((x \circ i(x)) \circ x)) \circ i(x)$

$\approx ((i(x \circ i(x)) \circ (x \circ i(x))) \circ x) \circ i(x)$

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

$\approx x \circ i(x)$

word-problems for set of identities.

we want unidirectional rewrite roles

abstract treatment of reduction
- traversal of some directed graph
- stepwise execution of some computation.
- gradual transformation of some object (i.e., term)

ARS $\overset{\text{set}}{A}, \overset{\text{reduction}}{\rightarrow}$ $\quad \rightarrow \subseteq A \times A$

$(a,b) \in \rightarrow$

$a \rightarrow b$ $\quad \{aRb\}$

Undirected search expensive

we can decide equivalence of a, b if their normal forms are identical.

- reduction terminates
- normal forms unique

$R \subseteq A \times B$ $\quad R \circ S = \{(a,c) \in A \times C \mid \exists b \in B$ where $(a,b) \in R \wedge (b,c) \in S\}$

$S \subseteq B \times C$

$\xrightarrow{0} := \{(x,x) \mid x \in A\}$ identity

$\xrightarrow{i+1} := \xrightarrow{i} \circ \rightarrow$ (i+1) fold composition

$\xrightarrow{+} := \bigcup_{i>0} \xrightarrow{i}$ transitive closure

$\xrightarrow{*} := \xrightarrow{+} \cup \xrightarrow{0}$ reflexive transitive closure

$\xrightarrow{=} := \rightarrow \cup \xrightarrow{0}$ reflexive closure

$\xrightarrow{-1} := \{(b,a) \mid a \rightarrow b\}$ inverse

$\leftarrow := \xrightarrow{-1}$ inverse

$\leftrightarrow := \rightarrow \cup \leftarrow$ symmetric closure

$\xleftrightarrow{+} := (\leftrightarrow)^+$ transitive symmetric closure.

$\xleftrightarrow{*} := (\leftrightarrow)^*$ reflexive transitive symmetric closure.

R^*, R^{-1}, ...

$a \xrightarrow{n} b$ path of length n from a to b.

$a \xrightarrow{*} b$ path from a to b.

$a \xrightarrow{+} b$ non-empty path from a to b.

P closure is the least set with property P which contains R.

1. a is reducible iff there is a b st. $a \rightarrow b$.

2. a is in normal form iff it is not reducible.

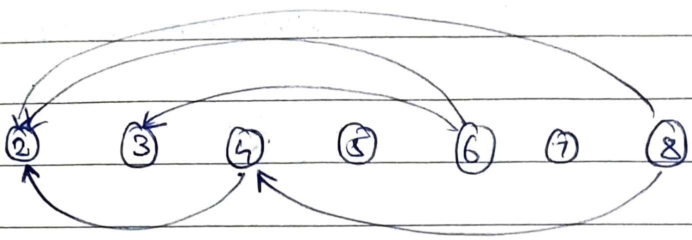
3. b is a normal form of a iff $a \xrightarrow{*} b$ and b is in normal form. if a has a uniquely defined normal form, it is denoted by $a\downarrow$.

4. b is a direct successor of a iff $a \rightarrow b$.

5. b is a successor of a iff $a \xrightarrow{+} b$.

6. a and b are joinable iff $\exists$ c st. $a \xrightarrow{*} c \xleftarrow{*} b$. $(a \downarrow b)$.

① $A := \mathbb{N} - \{0, 1\}$ $\quad \rightarrow := \{(m,n) \mid m > n \text{ and } n \mid m\}$



(a) m is in normal form iff m is prime

(b) p is a normal form of m iff p is a prime factor of m.

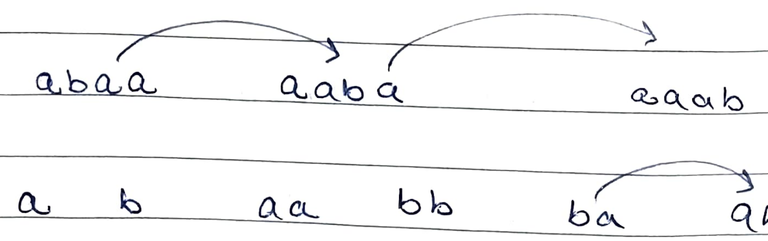
(c) $m \downarrow n$ iff m and n are not relatively prime.

(d) $\xrightarrow{+} = \rightarrow$ because $>$ and "divides" are already transitive.

(e) $\xleftrightarrow{*}$ $A \times A$

② $A := \{a, b\}^*$ (set of words over alphabet $\{a, b\}$)
$\rightarrow := \{(ubav, uabv) \mid u, v \in A\}$

(a)



(a) w is in normal form iff w is sorted, i.e. of the form a^*b^*.

(b) every w has a unique normal form $w\downarrow$, the result of sorting w.

(c) $w_1 \downarrow w_2$ iff $w_1 \xleftrightarrow{*} w_2$ iff w_1 and w_2 contain the same no. of a's and b's.

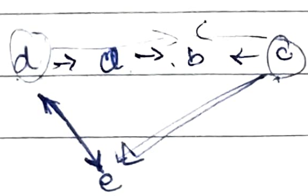
Church-rosser iff $a \xleftrightarrow{*} b \Rightarrow a \downarrow b$

confluent iff $b_1 \xleftarrow{*} a \xrightarrow{*} b_2 \Rightarrow b_1 \downarrow b_2$

terminating iff there is no infinite chain or loop.

normalizing iff every element has a normal form.

convergent iff it is both confluent & terminating



WELL FOUNDED INDUCTION

(weak induction)
$$\frac{\forall a \in A,\ (\forall b \in A\ \ a \to b \Rightarrow P(b)) \Rightarrow P(a)}{\forall a \in A\ \ P(a)}$$

$$\frac{\forall a \in A,\ (\forall b \in A\ \ a \xrightarrow{+} b \Rightarrow P(b)) \Rightarrow P(a)}{\forall a \in A\ \ P(a)}.$$

$a \in A$

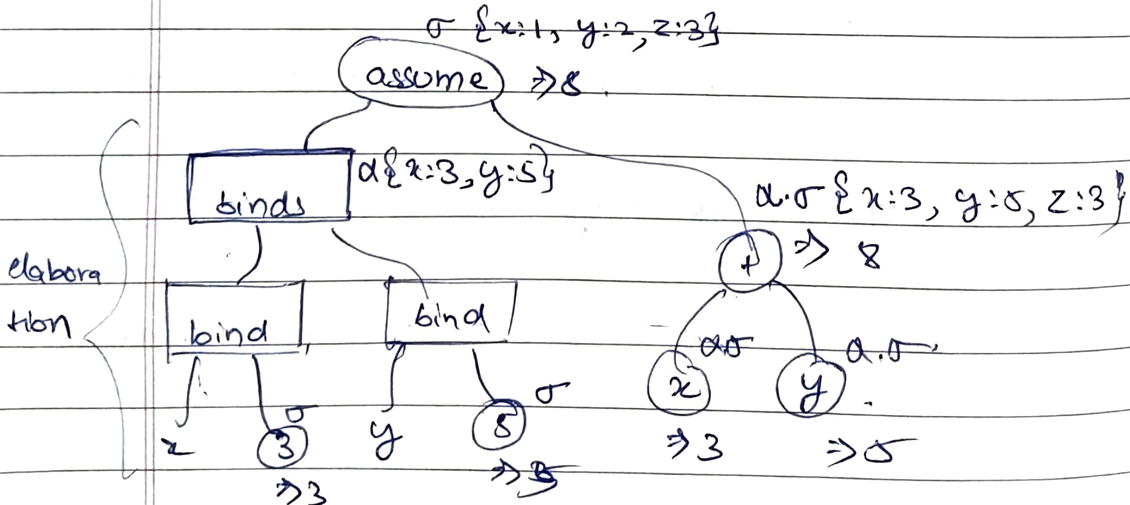
$a \xrightarrow{*} b$

$a \xrightarrow{*} c$

$\Rightarrow b \downarrow c$

$b = c.$

elaboration vs evaluation.



$x_0 \to x_1$