Operational semantics
of formal calculi and programming languages

Rewriting theory

- **Rewriting = abstract form of program execution**
- Paradigmatic example: ***λ*-calculus**
  (functional programming language, in its essence)

1

A colony of chameleons includes 20 red, 18 blue, and 16 green individuals. Whenever two chameleons of different color meet, each changes to the third color. Some time passes during which no chameleons are born or die nor do any enter or leave the colony. Is it possible that at the end of this period, all 54 chameleons are the same color?

2

Math formalizations…

Example (Group Theory)

signature	e (constant) $^-$ (unary, postfix) $\cdot$ (binary, infix)	
equations	$e \cdot x \approx x \quad x^- \cdot x \approx e \quad (x \cdot y) \cdot z \approx x \cdot (y \cdot z)$	$\mathcal{E}$
theorems	$e^- \approx_{\mathcal{E}} e \quad (x \cdot y)^- \approx_{\mathcal{E}} y^- \cdot x^-$	
rewrite rules	$e \cdot x \to x \quad x \cdot e \to x$ $x^- \cdot x \to e \quad x \cdot x^- \to e$ $(x \cdot y) \cdot z \to x \cdot (y \cdot z) \quad x^{--} \to x$ $e^- \to e \quad (x \cdot y)^- \to y^- \cdot x^-$ $x^- \cdot (x \cdot y) \to y \quad x \cdot (x^- \cdot y) \to y$	$\mathcal{R}$

① $s \approx t$ is valid in $\mathcal{E}$ ($s \approx_{\mathcal{E}} t$) if and only if s and t have same $\mathcal{R}$-normal form

② $\mathcal{R}$ admits no infinite computations

① & ② $\implies$ $\mathcal{E}$ has decidable validity problem

3

Modelling computation

Example (Lambda Calculus)

signature	λ (binds variables) $\cdot$ (application, binary, infix)
terms	$M ::= x \mid (\lambda x.\, M) \mid (M \cdot M)$
α conversion	$\lambda x.\, x \cdot y =_\alpha \lambda z.\, z \cdot y$
β reduction	$(\lambda x.\, M) \cdot N \to_\beta M[x := N]$ replace free occurrences of x in M by N (and avoid variable capturing)
rewriting	$(\lambda x.\, x \cdot x) \cdot (\lambda x.\, x \cdot x) \to (\lambda x.\, x \cdot x) \cdot (\lambda x.\, x \cdot x)$
inventor	Alonzo Church (1932)

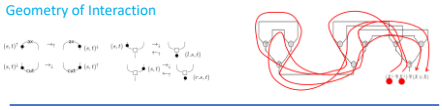
both Combinatory Logic and Lambda Calculus are Turing-complete

5

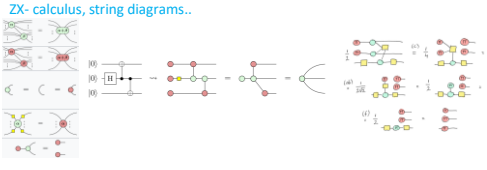
Graph Rewriting

LL proof-nets

Geometry of Interaction



ZX- calculus, string diagrams..



6

Rewriting and all that

- Rewrite Theory provides a powerful set of tools to study computational and operational properties of a system : normalization, termination , confluence, uniqueness of normal forms
- tools to study and compare strategies:
  - Is there a strategy guaranteed to lead to normal form, if any (*normalizing strat.*)?
- Abstract Rewrite Systems (ARS) capture the common substratum of rewrite theory (independently from the particular structure of terms) - can be uses in the study of any calculus or programming language.

7

Abstract Rewriting:

concrete rewrite formalisms / concrete operational semantics:

- λ-calculus
- *Quantum/ probabilistic/ non-deterministic/……* λ-calculus
- Proof-nets / graph rewriting
- Sequent calculus and cut-elimination
- string rewriting
- term rewriting

abstract rewriting

- **independent from structure** of objects that are rewritten
- **uniform** presentation of properties and proofs

8

Why a theory of rewriting matters?

- **Rewriting = abstract form of program execution**

Rewriting theory provides a *sound framework* for reasoning about

- **programs transformations**, such as compiler optimizations or parallel implementations,
- **program equivalence**.

9

Abstract Rewriting

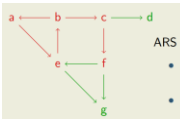
The basic language

Credits:
Figures & most material are taken from
Middeldorp's excellent course
Term Rewriting (ISR summer school, 2019)

10

ARS

Definition 1.1.1. An *abstract rewrite system* (ARS for short) is a pair $\mathcal{A} = \langle A, \to \rangle$ consisting of a set A and a binary relation $\to$ on A. Instead of $(a, b) \in \to$ we write $a \to b$ and we say that $a \to b$ is a *rewrite step*.



ARS $\mathcal{A} = \langle A, \to \rangle$

- $A = \{a, b, c, d, e, f, g\}$
- $\to = \left\{ \begin{array}{l} (a,e), (b,a), (b,c), (c,d), (c,f) \\ (e,b), (e,g), (f,e), (f,g) \end{array} \right\}$

- A (finite) *rewrite sequence* is a non-empty sequence $(a_0, \dots\ a_n)$ of elements in A such that $a_i \to a_{(i+1)}$

We write $a_0 \to^n a_n$ or simply $a_0 \to^* a_n$

- rewrite sequence
  - finite $a \to e \to b \to c \to f$
  - empty a
  - infinite $a \to e \to b \to a \to e \to b \to \cdots$

11

- $\leftarrow$ inverse of $\to$
- $\to^*$ transitive and reflexive closure of $\to$
- ${}^*\!\leftarrow$ inverse of $\to^*$

$s \leftrightarrow_{\mathcal{R}} t$ iff $s \to_{\mathcal{R}} t$ or $t \to_{\mathcal{R}} s$

$s \leftrightarrow^*_{\mathcal{R}} t$ iff $s = s_0 \leftrightarrow_{\mathcal{R}} s_1 \leftrightarrow_{\mathcal{R}} \dots \leftrightarrow_{\mathcal{R}} s_n = t$ for $n \geq 0$

- $\leftrightarrow$ symmetric closure of $\to$
- $\leftrightarrow^*$ conversion (equivalence relation generated by $\to$) **
- $\to^+$ transitive closure of $\to$
- $\to^=$ reflexive closure of $\to$

$\cdot$ is relation composition: $R \cdot S = \{ (a, c) \mid a\ R\ b \text{ and } b\ S\ c \}$

$\downarrow\ = \ \to^* \cdot {}^*\!\leftarrow$

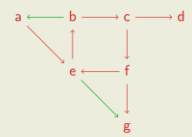
12

Terminology

- if $x \to^* y$ then x rewrites to y and y is reduct of x
- if $x \to^* z\ {}^*\!\leftarrow y$ then z is common reduct of x and y
- if $x \leftrightarrow^* y$ then x and y are convertible

$\boxed{\downarrow\ = \ \to^* \cdot {}^*\!\leftarrow}$

Example



- $a \to^* f$
- $e \downarrow f$ $f \downarrow d$ not $g \downarrow d$
- $g \leftrightarrow^* d$

13

Note: composite systems

- If $\to_1, \to_2$ are binary relations on A then $\to_1 \cdot \to_2$ denotes their composition, *i.e.* $t \to_1 \cdot \to_2 s$ iff there exists $u \in A$ such that $t \to_1 u \to_2 s$.
- We write $(A, \{\to_1, \to_2\})$ to denote the ARS $(A, \to)$ where $\to = \to_1 \cup \to_2$.

14

Note: * is a closure operator

$\twoheadrightarrow_R$ is the reflexive, transitive closure of $\to_R$:
(1) $M \to_R N \Rightarrow M \twoheadrightarrow_R N$,
(2) $M \twoheadrightarrow_R M$,
(3) $M \twoheadrightarrow_R N,\ N \twoheadrightarrow_R L \Rightarrow M \twoheadrightarrow_R L$.

The transitive-reflexive closure of a relation is a closure operator, *i.e.* satisfies

$$\to \subseteq \to^*, \qquad (\to^*)^* = \to^*, \qquad \to_1 \subseteq \to_2 \text{ implies } \to_1^* \subseteq \to_2^*$$

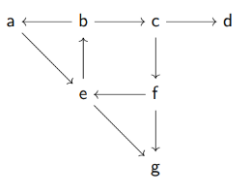
As a consequence

$$(\to_1 \cup \to_2)^* = (\to_1^* \cup \to_2^*)^*.$$

15

Normal forms: they model "results"

Definition 1.1.11. Let $\mathcal{A} = \langle A, \to \rangle$ be an ARS. An element $a \in A$ is *reducible* if there exists an element $b \in A$ with $a \to b$. A *normal form* is an element that is not reducible. The set of normal forms of $\mathcal{A}$ is denoted by $\mathsf{NF}(\mathcal{A})$ or $\mathsf{NF}(\to)$ when A can be inferred from the context. An element $a \in A$ *has a normal form* if $a \to^* b$ for some normal form b. In that case we write $a \to^! b$.



Element a has normal forms ?
How many normal forms has this ARS?

ARS $\mathcal{A} = \langle A, \to \rangle$
- d is normal form
- $\mathsf{NF}(\mathcal{A}) = \{\mathsf{d}, \mathsf{g}\}$
- $\mathsf{b} \to^! \mathsf{g}$

16

Operational properties of interest

- Termination and Confluence

Existence and uniqueness of normal forms

- How to Compute

reduction strategies with good properties:
- standardization,
- normalization

17

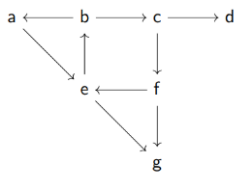
- SN strong normalization termination
  - no infinite rewrite sequences
- WN (weak) normalization
  - every element has at least one normal form
  - $\forall a\ \exists b \quad a \to^! b$
- UN unique normal forms
  - no element has more than one normal form
  - $\forall a, b, c$ if $a \to^! b$ and $a \to^! c$ then $b = c$

18

Termination

Definition 1.2.1. Let $\mathcal{A} = \langle A, \to \rangle$ be an ARS. An element $a \in A$ is called *terminating* or *strongly normalizing* (SN) if there are no infinite rewrite sequences starting at a. The ARS $\mathcal{A}$ is terminating or strongly normalizing if all its elements are terminating. An element $a \in A$ has *unique normal forms* (UN) if it does not have different normal forms ($\forall b, c \in A$ if $a \to^! b$ and $a \to^! c$ then $b = c$). The ARS $\mathcal{A}$ has unique normal forms if all its elements have unique normal forms.

An element a is *weakly normalizing* (WN) (or simply *normalizing)* if it has a normal form.



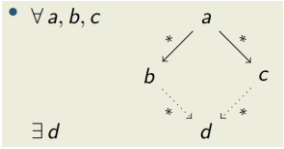
a is WN? SN?
c is WN? SN?
a or c has UN ?

The nf are convertible?

19

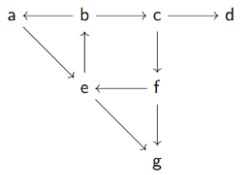
Confluence

Definition 1.2.3. Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. An element $a \in A$ is *confluent* if for all elements $b, c \in A$ with $b \,{}^*\!\leftarrow a \rightarrow^* c$ we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.



Every confluent ARS *has unique normal forms.*

20



1. a is confluent?
2. f is confluent?

3. Can you add a single arrow so that the resulting ARS has **unique normal forms without being confluent ?**

21

Given

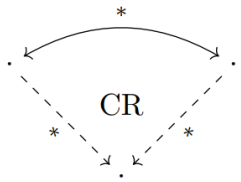
$$\mathcal{R} = \begin{cases} f(x,x) & \rightarrow & c \\ a & \rightarrow & b \\ f(x,b) & \rightarrow & d \end{cases}$$

f(a,a) has normal form?
Can you produce two different nf?

we can compute from the same term *f(a, a)* two different normal-forms *c* and *d*
different meaning for same term!

22

Same meaning for *equivalent* terms



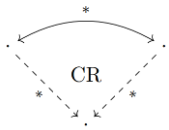
23

Confluence & CR

Definition 1.2.3. Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. An element $a \in A$ is *confluent* if for all elements $b, c \in A$ with $b \,{}^*\!\leftarrow a \rightarrow^* c$ we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.

Confluence

Church-Rosser



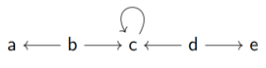
An ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ *is confluent if and only if* $\leftrightarrow^* \subseteq \downarrow$.

24

Definition 1.2.10. An ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ has *unique normal forms with respect to conversion* (UNC) if different normal forms are not convertible ($\forall\, a, b \in \mathsf{NF}(\mathcal{A})$ if $a \leftrightarrow^* b$ then $a = b$).

in an ARS with the property UNC every equivalence class of convertible elements contains at most one normal form.

Q: are UN and UNC equivalent?



25

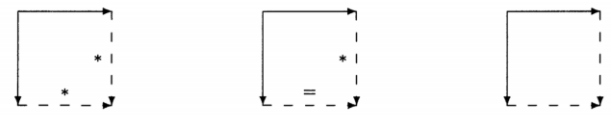
Global vs Local

26

Confluence

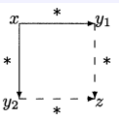
A property of term *t* is *local* if it is quantified over only *one-step reductions* from *t*; it is *global* if it is quantified over all *rewrite sequences* from *t*.

Locally confluent (WCR) **Strongly confluent** **Diamond**



confluence Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. An element $a \in A$ is *confluent* if for all elements $b, c \in A$ with $b \,{}^*\!\leftarrow a \rightarrow^* c$ we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.

Global property:

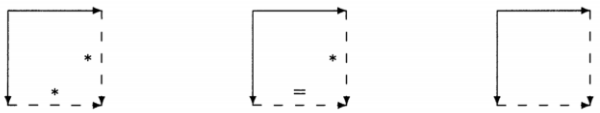


27

Confluence

A property of term *t* is *local* if it is quantified over only *one-step reductions* from *t*; it is *global* if it is quantified over all *rewrite sequences* from *t*.

Locally confluent (WCR) **Strongly confluent** **Diamond**

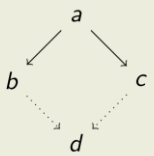


Local confluence Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. An element $a \in A$ is locally confluent for all elements $b, c \in A$ with b←a →c we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.

An ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ has the *diamond property* ($\diamond$) if $\leftarrow \cdot \rightarrow \subseteq \rightarrow \cdot \leftarrow$

28

- diamond property $\diamond$
  - $\leftarrow \cdot \rightarrow \subseteq \rightarrow \cdot \leftarrow$
  - $\forall\, a, b, c$ $\quad \exists\, d$



- *every ARS with diamond property is confluent*

Proof by tiling

29