Preuves et programmes : Outils classiques

Claudia Faggian	CNRS (IRIF)
	faggian@irif.fr
	https://www.irif.fr/~faggian/

This part focus on **Operational Semantics**
of formal calculi (and programming languages)

1

Topics

- Tools to study the operational properties of a system:
  - ➢ Rewrite Theory (rewriting=abstract form of program execution)
- Induction and Co-induction proof principles.
- Linear Logic and Proof-Nets.

- Bridging between lambda-calculus and functional programming.
  - ➢ Call-by-Value and Call-by Name, weak and lazy calculi.
  - ➢ Big-Step and Small-Step operational semantics.
  - ➢ Observational equivalence

- Reasoning on programs equivalence:
  - ➢ Bisimulation and coinductive methods.

- Beyond pure functional:
  - ➢ Probabilistic programming and Bayesian Inference:
    Probabilistic lambda calculi, Bayesian Networks & proof-nets

2

Resources

- **Reference Books**:
  - ➢ *R. AMADIO : Operational methods in semantics*
    (available on HAL https://hal.archives-ouvertes.fr/cel-01422101v1).
  - ➢ *D. SANGIORGI: Introduction to Bisimulation and Coinduction*
    (Cambridge University Press, 2011)

- **Lecture Notes** (by Middeldorp, Laurent, Ong)

Please send me an email
(with LMFI in the subject)
to have the **lecture notes**
on Rewriting Theory

3

Rewriting theory

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

4

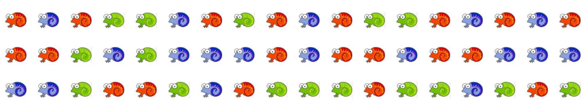
Operational semantics
of formal calculi and programming languages

Rewriting theory

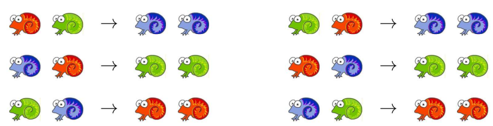
- **Rewriting = abstract form of program execution**

- Paradigmatic example: ***λ*-calculus**
  (functional programming language, in its essence)

5



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?



6

Example (Group Theory)

signature: e (constant) $^-$ (unary, postfix) $\cdot$ (binary, infix)

equations: $e \cdot x \approx x$ $x^- \cdot x \approx e$ $(x \cdot y) \cdot z \approx x \cdot (y \cdot z)$ $\mathcal{E}$

theorems: $e^- \approx_{\mathcal{E}} e$ $(x \cdot y)^- \approx_{\mathcal{E}} y^- \cdot x^-$

rewrite rules: $\mathcal{R}$

$$\begin{array}{rcl@{\qquad}rcl} e \cdot x & \to & x & x \cdot e & \to & x \\ x^- \cdot x & \to & e & x \cdot x^- & \to & e \\ (x \cdot y) \cdot z & \to & x \cdot (y \cdot z) & x^{--} & \to & x \\ e^- & \to & e & (x \cdot y)^- & \to & y^- \cdot x^- \\ x^- \cdot (x \cdot y) & \to & y & x \cdot (x^- \cdot y) & \to & y \end{array}$$

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 $\Longrightarrow$ $\mathcal{E}$ has decidable validity problem

7

Example (Combinatory Logic)

signature: S K I (constants) $\cdot$ (application, binary, infix)

terms: S $((K \cdot I) \cdot I) \cdot S$ $(x \cdot z) \cdot (y \cdot z)$

rewrite rules:

$$I \cdot x \to x$$
$$(K \cdot x) \cdot y \to x$$
$$((S \cdot x) \cdot y) \cdot z \to (x \cdot z) \cdot (y \cdot z)$$

rewriting: $((S \cdot K) \cdot K) \cdot x \to (K \cdot x) \cdot (K \cdot x) \to x$

inventor: Moses Schönfinkel (1924)

8

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

9

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)

10

Rewriting

- 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.

11

Abstract Rewriting: motivations

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

12

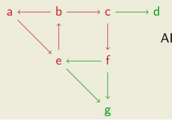
Abstract Rewriting

Basic language

13

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

- A (finite) *rewrite sequence* is a non-empty sequence $(a_0, \ldots a_n)$ 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 → e → b → c → f
  - empty a
  - infinite a → e → b → a → e → b → ⋯

14

- $\leftarrow$ 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 $s = s_0 \leftrightarrow_{\mathcal{R}} s_1 \leftrightarrow_{\mathcal{R}} \ldots \leftrightarrow_{\mathcal{R}} s_n = t$ 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: $R \cdot S = \{ (a, c) \mid a \; R \; b \text{ and } b \; S \; c \}$

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

15

Composition

- We denote $\to^*$ (resp. $\to^=$) the transitive-reflexive (resp. reflexive) closure of $\to$;
- 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 $\to = \to_1 \cup \to_2$.

16

Closure

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

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

As a consequence

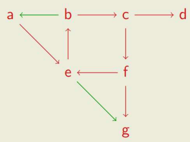
$(\to_1 \cup \to_2)^* = (\to_1^* \cup \to_2^*)^*.$

17

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

Example

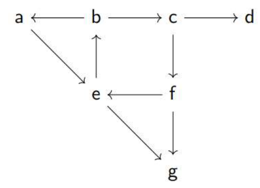


- a $\to^*$ f
- e $\downarrow$ f f $\downarrow$ d not g $\downarrow$ d
- g $\leftrightarrow^*$ d

18

Normal forms 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
- $\mathsf{NF}(\mathcal{A}) = \{$ d, g $\}$
- b $\to^!$ g

19

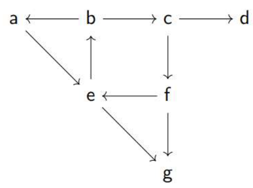
- SN strong normalization termination
  - no infinite rewrite sequences
- WN (weak) normalization
  - every element has at least one normal form
  - $\forall\, a\ \exists\, b \quad a \to^! b$
- UN unique normal forms
  - no element has more than one normal form
  - $\forall\, a, b, c \quad$ if $a \to^!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

20

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.



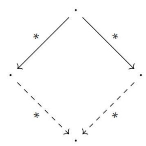
a is WN? SN?
c is WN? SN?
a or c has UN ?

The nf are convertible?

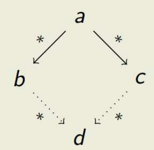
21

Confluence

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



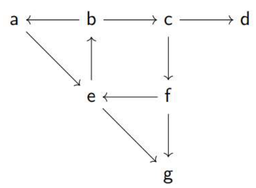
- $\forall\, a, b, c$



$\exists\, d$

Every confluent ARS *has unique normal forms.*

22



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

Bonus Point

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

23

Given

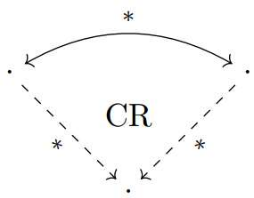
$$\mathcal{R} = \begin{cases} f(x,x) & \to & c \\ a & \to & b \\ f(x,b) & \to & d \end{cases}$$

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 equivalent terms
(different meaning for same term!)

24

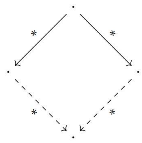
Same meaning for *equivalent* terms

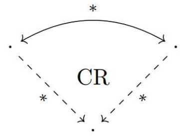


25

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.





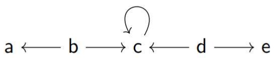
An ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ is confluent if and only if $\leftrightarrow^* \subseteq \downarrow$.

26

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?



27

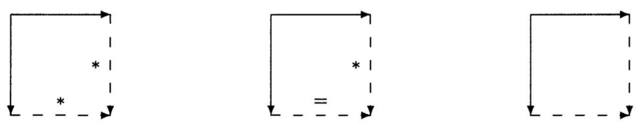
Global vs Local

28

Confluence

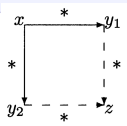
A property of term *t* is *local* if it is quantified over only *one-step reductions* from *t*; it is *global* if it is quantified over all *rewrite sequences* from *t*.

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:

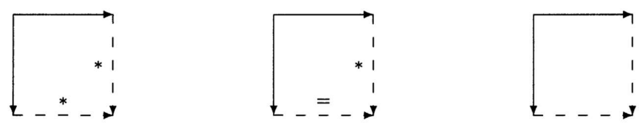


29

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**

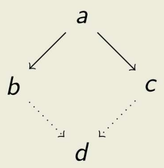


Local confluence Let $\mathcal{A} = \langle A, \rightarrow \rangle$ be an ARS. An element $a \in A$ is locally confluent for all elements $b, c \in A$ with b←a→c we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.

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

30

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



- *every ARS with diamond property is confluent*

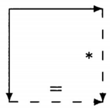
31

An ARS $\mathcal{A} = \langle A, \to \rangle$ is *strongly confluent* (SCR) if $\leftarrow \cdot \to \subseteq \to^{=} \cdot {}^{*}\!\leftarrow$, see Figure

a Show that every strongly confluent ARS is confluent.

b Does the converse also hold?

c Show that an ARS $\mathcal{A} = \langle A, \to \rangle$ is confluent if and only if ${}^{*}\!\leftarrow \cdot \to \;\subseteq\; \to^{*} \cdot \leftarrow^{*}$



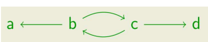
32

Which is true?

1. SN => WN
2. WN => SN

3. Confluence => UN
4. UN => Confluence

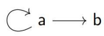
5. Confluence => Local confluence
6. Local confluence => Confluence



7. WN & UN => Confluence
8. WN & Local Conf. => Confluence
9. SN & Local Conf. => Confluence

33

WN vs SN



(ii) WN $\not\Rightarrow$ SN

$$\mathcal{R} = \begin{cases} f(a) & \to & c \\ f(x) & \to & f(a) \end{cases}$$

The system is weakly normalising but not strongly normalising:

Can you find an infinite reduction sequence?

$$f(b) \to f(a) \to c$$

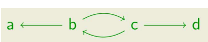
$$f(b) \to f(a) \to f(a) \ldots$$

34

1. SN => WN
2. WN => SN

3. Confluence => UN
4. UN => Confluence

5. Confluence => Local confluence
6. Local confluence => Confluence



7. WN & UN => Confluence
8. WN & Local Conf. => Confluence
9. SN & Local Conf. => Confluence

Newman's Lemma

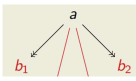
35

Lemma

WN & UN $\Longrightarrow$ CR

Proof

- WN $\Longrightarrow$ $\exists\, n_1, n_2$: $b_1 \to^{!} n_1$ and $b_2 \to^{!} n_2$
- UN $\Longrightarrow$ $n_1 = n_2$ $\Longrightarrow$ $b_1 \downarrow b_2$



36

Newman Lemma

Newman's Lemma. *Every terminating and locally confluent* ARS *is confluent.*

By well-founded induction

37

Memo: Well-founded Induction

Définition :[Relation bien fondée] Une relation d'ordre $>\subseteq E\times E$ est *bien fondée* si il n'existe pas de suite infinie d'éléments de E décroissante par rapport à $>$.

Theorem :[Principe d'induction bien fondée] Soient donnés un ensemble E quelconque, un ordre strict $<$ sur E (dont $\mathscr{M}$ est son ensemble d'éléments minimaux), et une propriété P sur E.

Si

1. pour tout élément **minimal** $m\in\mathscr{M}$ on a **P(m)**
2. le fait que **P(k)** soit vérifiée pour **tout** élément $k<x$ implique **P(x)**

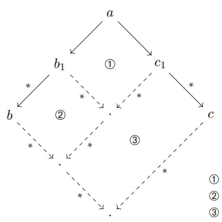
alors
pour tout $x\in E$ on a $P(x)$

The proof technique of well-founded induction states that a property $\mathcal{P}$ of elements of a terminating ARS $\mathcal{A}=\langle A,\to\rangle$ holds for all elements in A if the following condition is satisfied: An element $a\in A$ has the property $\mathcal{P}$ if all elements b with $a\to b$ have the property $\mathcal{P}$. In particular every normal form has to satisfy the property $\mathcal{P}$.

38

Newman Lemma

Newman's Lemma. *Every terminating and locally confluent* ARS *is confluent.*



① WCR
② induction hypothesis ($a\to b_1 \implies b_1$ is CR)
③ induction hypothesis ($a\to c_1 \implies c_1$ is CR)

39

Newman Lemma

Bonus Exercise

Newman's Lemma. *Every terminating and locally confluent* ARS *is confluent.*

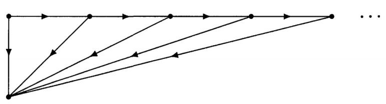
A second Proof. Let $\mathcal{A}=\langle A,\to\rangle$ *terminating and locally confluent*
It suffices to show that every element has unique normal forms

- suppose $B=\{\,a\in A\mid \neg\mathrm{UN}(a)\,\}\neq\varnothing$
- let $b\in B$ be minimal element (with respect to $\to$)
- $b\to^! n_1$ and $b\to^! n_2$ with $n_1\neq n_2$

➢ **Conclude** by showing that it is impossible (**absurd**)

40

Recap Flash Ex



➢ **EX** Say which properties hold

1. Confluent
2. Locally confluent
3. Normalizing (weakly normalizing, WN)
4. Terminating (strongly normalizing, SN)

41

Recap basics

- An *abstract rewriting system (ARS)* is a pair $(\mathcal{A},\to)$ consisting of a set $\mathcal{A}$ and a binary relation $\to$ on $\mathcal{A}$ whose pairs are written $t\to s$ and called *steps*.
- We denote $\to^*$ (resp. $\to^=$) the transitive-reflexive (resp. reflexive) closure of $\to$. We write $t\leftarrow u$ if $u\to t$.
- If $\to_1,\to_2$ are binary relations on $\mathcal{A}$ then $\to_1\cdot\to_2$ denotes their composition, *i.e.* $t\to_1\cdot\to_2 s$ if there exists $u\in\mathcal{A}$ such that $t\to_1 u\to_2 s$.
- We write $(\mathcal{A},\{\to_1,\to_2\})$ to denote the *compound system* $(\mathcal{A},\to)$ where $\to\;=\;\to_1\cup\to_2$.

- A $\to$-*sequence* (or **reduction sequence**) from t is a (possibly infinite) sequence $t,\ t_1,\ t_2,\ldots$ such that $t_i\to t_{i+1}$.
  $t\to^* s$ indicates that there is a finite sequence from t to s.
  A $\to$-sequence from t is *maximal* if it is either infinite or ends in a $\to$-*nf*.

42

The heart of confluence is a diamond

Prop. DIAMOND implies CONFLUENCE

*Can rarely be used **directly**:*
Most relations of interest do not satisfy it

Lemma **(Characterize Confluence).** $\to$ *is confluent* if and only if *there exists a relation* $\Leftrightarrow$ *such that*

a. $\Leftrightarrow^* = \to^*$,
b. $\Leftrightarrow$ *is diamond.*

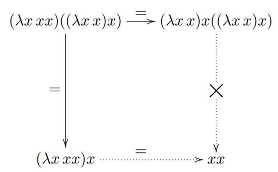
43

You have already seen an example: in the class by Joly

Definition *The* development relation *is the least reflexive relation* $\triangleright$ *on* Λ *such that:*

- $t \triangleright t' \implies \lambda x\, t \triangleright \lambda x\, t'$
- $t \triangleright t',\ u \triangleright u' \implies tu \triangleright t'u'$
- $t \triangleright t',\ u \triangleright u' \implies (\lambda x\, t)u \triangleright t'[x:=u']$.

Lemma 1 $\to\ \subseteq\ \triangleright\ \subseteq\ \twoheadrightarrow$.



44

Closure

$\twoheadrightarrow_R$ 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

$\to \subseteq \to^*$, $(\to^*)^* = \to^*$, $\to_1 \subseteq \to_2$ implies $\to_1^* \subseteq \to_2^*$

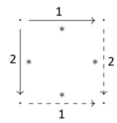
As a consequence

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

45

Commutation

Commutation. Two relations $\to_1$ and $\to_2$ on A *commute* if $\leftarrow_1{}^* \cdot \to_2{}^* \subseteq \to_2{}^* \cdot \leftarrow_1{}^*$.



Confluence. A relation $\to$ on A is confluent if it commutes with itself.

46

Proving confluence modularly

Lemma (Hindley-Rosen)

*If two relations $\to_1$ and $\to_2$ are **confluent** and **commute with each other**, then*

$\to_1 \cup \to_2$ is confluent.

47

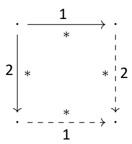
An effective usable technique

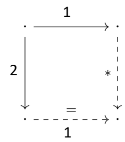
Lemma (Hindley-Rosen)

*If two relations $\to_1$ and $\to_2$ are **confluent** and **commute with each other**, then*

$\to_1 \cup \to_2$ is confluent.

Global condition
(all sequences)





Local condition
(one-step test)

Lemma (Hindley's local test)

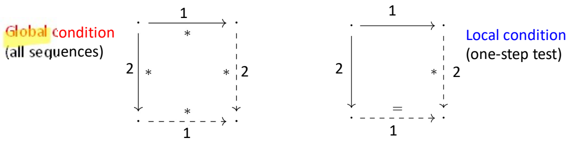
Strong commutation $\leftarrow_1 \cdot \to_2\ \subseteq\ \to_2{}^* \cdot \leftarrow_1{}^=$ *implies commutation.*

48

an effective usable technique

Lemma (Hindley-Rosen)

If two relations $\rightarrow_1$ and $\rightarrow_2$ are ***confluent*** *and* ***commute with each other****, then $\rightarrow_1 \cup \rightarrow_2$ is confluent.*



$\leftarrow_1 \cdot \rightarrow_2 \subseteq \rightarrow_2^* \cdot \leftarrow_1^=$ **(Strong Commutation)**

▸ **Lemma** (Local test). *Strong commutation implies commutation.*

49

Strategies and subreductions

50

Normalization

▸ **Def.** *$(\mathcal{A},\rightarrow)$ is strongly (weakly, uniformly) normalizing if each $t \in \mathcal{A}$ is, where the three normalization notions are as follows.*

- *t is* strongly $\rightarrow$-normalizing*: every maximal $\rightarrow$-sequence from t ends in a normal form.*
- *t is* weakly $\rightarrow$-normalizing*: there exist a $\rightarrow$-sequence from t which ends in a normal form.*
- *t is* uniformly $\rightarrow$-normalizing*: t weakly $\rightarrow$-normalizing implies t strongly $\rightarrow$-normalizing.*

If terms are not strongly normalizing, how do we compute a normal form, or even test if any exists? This is the problem tackled by *normalization*. By repeatedly performing *only specific steps* $\underset{e}{\rightarrow}$, we are guaranteed that a normal form will eventually be computed, if any exists.

51

Normalizing strategis

▸ **Def.** *$(\mathcal{A},\rightarrow)$ is strongly (weakly, uniformly) normalizing if each $t \in \mathcal{A}$ is, where the three normalization notions are as follows.*

- *t is* strongly $\rightarrow$-normalizing*: every maximal $\rightarrow$-sequence from t ends in a normal form.*
- *t is* weakly $\rightarrow$-normalizing*: there exist a $\rightarrow$-sequence from t which ends in a normal form.*
- *t is* uniformly $\rightarrow$-normalizing*: t weakly $\rightarrow$-normalizing implies t strongly $\rightarrow$-normalizing.*

▸ **Def.**

- *$\underset{e}{\rightarrow}$ is a* ***strategy for*** *$\rightarrow$ if $\underset{e}{\rightarrow} \subseteq \rightarrow$, and it has the same normal forms as $\rightarrow$.*
- *It is a* ***normalizing strategy*** *for $\rightarrow$ if whenever $t \in \mathcal{A}$ has $\rightarrow$-normal form, then* every *maximal $\underset{e}{\rightarrow}$-sequence from t ends in normal form.*

52

Completeness

Completeness. The restriction to a *subreduction* is a way to control the non-determinism which arises from different possible choices of reduction.

In general, we are interested in subreductions which are complete w.r.t. certain subset of interests (*i.e.*: values, normal forms, head normal forms).

Given $\mathcal{B} \subseteq \mathcal{A}$, we say that $\underset{e}{\rightarrow} \subseteq \rightarrow$ is $\mathcal{B}$-**complete** if whenever $t \rightarrow^* u$ with $u \in \mathcal{B}$, then $t \underset{e}{\rightarrow}^* u'$, with $u' \in B$.

53

Factorization
(*aka weak Standardization*)

another commutation!

54

Operational properties of interest

- Termination and Confluence

Existence and uniqueness of normal forms

- How to Compute

reduction strategies with good properties:

- standardization,
- normalization

Factorization
(aka Semi-Standardization, Postponement, or often simply Standardization)

- most basic property about *how to compute*

$t \to_\beta^* u \quad \Rightarrow \quad t \underset{h}{\to}^* \cdot \underset{\neg h}{\to}^* u$ *head factorization*

A **key building-block** in proofs of more sophisticated *how-to-compute* properties:

- allows immediate proofs of **normalization**
  (a reduction strategy reaches a normal form, whenever one exists)
- simplest way to prove **standardization**, by using Mitschke's argument
  (left-to-right standardization = iterate head factorization)

Factorization
(aka Semi-Standardization, Postponement, or often simply Standardization)

Melliès 97:
the **meaning of *factorization*** is that the *essential* part of a computation can always be separated from its junk.

Assume computations consists of

- steps $\underset{e}{\to}$ which are in some sense *essential*, and
- steps $\underset{i}{\to}$ which are not.

Factorization says that every rewrite sequence can be reorganized/factorized as a sequence of essential steps followed by inessential ones.

$t \to^* u \quad \Rightarrow \quad t \underset{e}{\to}^* \cdot \underset{i}{\to}^* u$ *e-factorization*

Factorization. Let $\mathcal{A} = (A, \{\underset{e}{\to}, \underset{i}{\to}\})$ be an ARS.

- The relation $\to = \underset{e}{\to} \cup \underset{i}{\to}$ satisfies **e-factorization**, written $\mathtt{Fact}(\underset{e}{\to}, \underset{i}{\to})$, if

$$\mathtt{Fact}(\underset{e}{\to}, \underset{i}{\to}) : \quad (\underset{e}{\to} \cup \underset{i}{\to})^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^* \qquad \textbf{(Factorization)}$$

- The relation $\underset{i}{\to}$ **postpones** after $\underset{e}{\to}$, written $\mathtt{PP}(\underset{e}{\to}, \underset{i}{\to})$, if

$$\mathtt{PP}(\underset{e}{\to}, \underset{i}{\to}) : \quad \underset{i}{\to}^* \cdot \underset{e}{\to}^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*. \qquad \textbf{(Postponement)}$$

▶ **Lemma.** *For any two relations $\underset{e}{\to}, \underset{i}{\to}$ the following are equivalent:*

1. $\underset{i}{\to}^* \cdot \underset{e}{\to} \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*$
2. $\underset{i}{\to} \cdot \underset{e}{\to}^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*)$
3. Postponement: $\underset{i}{\to}^* \cdot \underset{e}{\to}^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*$
4. Factorization: $(\underset{e}{\to} \cup \underset{i}{\to})^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*$

Local test ?

We say that $\underset{i}{\to}$ **strongly postpones** after $\underset{e}{\to}$, if

$$\mathtt{SP}(\underset{e}{\to}, \underset{i}{\to}) : \quad \underset{i}{\to} \cdot \underset{e}{\to} \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^= \qquad \textbf{(Strong Postponement)}$$

▶ **Lemma** (Local test for postponement [26]). *Strong postponement implies postponement:*

$$\mathtt{SP}(\underset{e}{\to}, \underset{i}{\to}) \textit{ implies } \mathtt{PP}(\underset{e}{\to}, \underset{i}{\to}), \textit{ and so } \mathtt{Fact}(\underset{e}{\to}, \underset{i}{\to}).$$

Does SP hold for λ-calculus?

▶ **Ex** (λ-calculus and strong postponement). *β reduction is decomposed in head reduction $\underset{h}{\to}_\beta$ and its dual $\underset{\neg h}{\to}_\beta$*

$$\to_\beta = \underset{h}{\to}_\beta \cup \underset{\neg h}{\to}_\beta$$

Consider:

$$(\lambda x.xxx)(Iz) \underset{\neg h}{\to}_\beta (\lambda x.xxx)z \underset{h}{\to}_\beta zzz.$$

Does SP hold for λ-calculus?

▶ **Ex** (λ-calculus and strong postponement). *β reduction is decomposed in head reduction $\underset{h}{\to}_\beta$ and its dual $\underset{\neg h}{\to}_\beta$*

$$\to_\beta = \underset{h}{\to}_\beta \cup \underset{\neg h}{\to}_\beta$$

Consider:

$$(\lambda x.xxx)(Iz) \underset{\neg h}{\to}_\beta (\lambda x.xxx)z \underset{h}{\to}_\beta zzz.$$

$$(\lambda x.xxx)(Iz) \underset{h}{\to}_\beta (Iz)(Iz)(Iz) \underset{h}{\to}_\beta z(Iz)(Iz) \underset{\neg h}{\to}_\beta zz(Iz) \underset{\neg h}{\to}_\beta zzz$$

61

The heart of confluence is a diamond

TRICK

Prop. DIAMOND implies CONFLUENCE

Can rarely be used ***directly****:*
Most relations of interest do not satisfy it

Lemma **(Characterize Confluence).** *$\to$ is confluent* if and only if *there exists a relation $\multimap$ such that*

a. $\multimap^* = \to^*$,
b. $\multimap$ *is diamond.*

62

▶ **Property 2** (Criterion). *Given* $\to = \underset{e}{\to} \cup \underset{i}{\to}$, e-*factorization holds*

$$\to^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*$$

iff exists $\underset{i}{\multimap}$

- $\underset{i}{\multimap}^* = \underset{i}{\to}^*$ *(same closure)*
- $\underset{i}{\multimap} \cdot \underset{e}{\to} \subseteq \underset{e}{\to}^* \cdot \underset{i}{\multimap}^=$ (strong postponement)

Hence: $\underset{i}{\to}^* \cdot \underset{e}{\to}^* \subseteq \underset{e}{\to}^* \cdot \underset{i}{\to}^*.$ **(Postponement)**

63

Examples of uses for factorization

64

Call-by-Name and Call-by-Value λ-calculus

65

Call-by-Name and Call-by-Value λ–calculus

Terms and *values* are generated by the following grammars

$$V \;::=\; x \mid \lambda x.M \qquad (\textit{Values}, \mathcal{V})$$
$$M \;::=\; x \mid c \mid \lambda x.M \mid MM \qquad (\textit{Terms})$$

where x ranges over a countable set of *variables*, and c over a disjoint (possibly empty) set $\mathcal{O}$ of constants.

- If the set of constants is empty, the calulus is *pure*, and the set of terms is denoted Λ.
- Otherwise, the calculus is called *applied*, and the set of terms is often indicated as $\Lambda_{\mathcal{O}}$.

Terms are identified up to renaming of bound variables, where λx is the only binder constructor. $P\{Q/x\}$ is the capture-avoiding substitution of Q for the free occurrences of x in P.

66

Contexts (with one hole $(\!|\ |\!)$) are generated as follows. $\mathbf{C}(\!|M|\!)$ stands for the term obtained from $\mathbf{C}$ by replacing the hole with the term M (possibly capturing free variables of M).

$\mathbf{C} ::= (\!|\ |\!) \,||\, M\mathbf{C} \,|\, \mathbf{C}M \,|\, \lambda x.\mathbf{C}$ (*Contexts*)

67

Call-by-Name and Call-by-Value λ–calculus

Contexts (with one hole $(\!|\ |\!)$) are generated as follows. $\mathbf{C}(\!|M|\!)$ stands for the term obtained from $\mathbf{C}$ by replacing the hole with the term M (possibly capturing free variables of M).

$\mathbf{C} ::= (\!|\ |\!) \,||\, M\mathbf{C} \,|\, \mathbf{C}M \,|\, \lambda x.\mathbf{C}$ (*Contexts*)

- A **rule** ρ is a binary relation on $\Lambda_{\mathcal{O}}$, which we also denote $\mapsto_\rho$, writing $R \mapsto_\rho R'$. R is called a *ρ-redex*.
  The best known rule is β:

  $(\lambda x.M)N \mapsto_\beta M\{N/x\}$

- A **reduction step** $\to_\rho$ is the closure under context $\mathbf{C}$ of ρ.
  Explicitly, $T \to T'$ holds if $T = \mathbf{C}(\!|R|\!)$, $T' = \mathbf{C}(\!|R'|\!)$, and $R \mapsto_\rho R'$.

68

Call-by-Name and Call-by-Value λ–calculus

- *The λ-calculus can be seen both as an equational theory on terms and as an abstract model of computation.*
- *With the functional paradigm point of view, the meaning of any λ-term is the value it evaluates to.*

69

Call-by-Name and Call-by-Value λ–calculus

CbN and CbV Calculi.

- The (pure) **Call-by-Name** calculus $\Lambda^{\mathsf{cbn}} = (\Lambda, \to_\beta)$ is the set of terms equipped with the contextual closure of the β-rule.

  $(\lambda x.M)N \mapsto_\beta M\{N/x\}$

- The (pure) **Call-by-Value** calculus $\Lambda^{\mathsf{cbv}} = (\Lambda, \to_{\beta_v})$ is the same set equipped with the contextual closure of the β_v-rule.

  $(\lambda x.M)V \mapsto_{\beta_v} M\{V/x\}$ where $V \in \mathcal{V}$

70

CbN: Head Reduction

Head reduction in CbN

Head reduction is the closure of β under head context

$\lambda x_1 \ldots x_n.(\!|\ |\!)M_1 \ldots M_k$

Head normal forms (*hnf*), whose set is denoted by $\mathcal{H}$, are its normal forms.

- Given a rule ρ, we write $\underset{\mathsf{h}}{\to}_\rho$ for its closure under head context.
- A step $\to_\rho$ is non-head, written $\underset{\neg\mathsf{h}}{\to}_\rho$ if it is not head.

What about?

$\mathbf{H} ::= (\!|\ |\!) \,|\, \lambda x.\mathbf{H} \,|\, \mathbf{H}M$

71

CbN Head Factorization

Head Factorization

Head factorization allows for a characterization of the terms which have head normal form, that is M has *hnf* if and only if $\underset{\mathsf{h}}{\to}$-reduction from M terminates.

▶ **Theorem 2** (Head Factorization).

- Head Factorization: $\to_\beta{}^* \subseteq \underset{\mathsf{h}}{\to}_\beta{}^* \cdot \underset{\neg\mathsf{h}}{\to}_\beta{}^*$.
- Head Normalization: M *has* hnf *if and only if* $M \underset{\mathsf{h}}{\to}_\beta{}^* S$ (*for some* $S \in \mathcal{H}$) .

72

- According to the function paradigm of computation the goal of every computation is to determine its value
- Since functions are seen as values, it is natural to consider *weak evaluation*. In practical implementations, weak evaluation is more realistic than the full beta reduction

73

CbV: Weak Reduction

Weak reductions in CbV

The result of interest are **values** (*i.e.* functions).
In languages, in general the reduction is *weak*, that is, it does not reduce in the body of a function.

There are three main weak schemes: left, right and in arbitrary order.
Left contexts **L**, *right* contexts **R**, and (arbitrary order) *weak* contexts **W** are defined by

$\mathbf{L} ::= (\!|\,|\!) \mid \mathbf{L}M \mid V\mathbf{L}$
$\mathbf{R} ::= (\!|\,|\!) \mid M\mathbf{R} \mid \mathbf{R}V$
$\mathbf{W} ::= (\!|\,|\!) \mid \mathbf{W}M \mid M\mathbf{W}$

Given a rule $\mapsto$ on Λ, *weak reduction* $\underset{w}{\to}$ is the closure of $\mapsto$ under context **W**.
A step $T \to S$ is non-weak, written $T \underset{\neg w}{\to} S$ if it is not weak. Similarly for left ($\underset{l}{\to}$ and $\underset{\neg l}{\to}$), and right ($\underset{r}{\to}$ and $\underset{\neg r}{\to}$).

▶ **Fact 3** (Weak normal forms). *Given M a closed term, M is $\underset{s}{\to}$-normal* iff *M is a value.*

74

CbV: *Left* contexts **L**, *right* contexts **R**, and (arbitrary order) *weak* contexts **W** are defined by

$\mathbf{L} ::= (\!|\,|\!) \mid \mathbf{L}M \mid V\mathbf{L}$
$\mathbf{R} ::= (\!|\,|\!) \mid M\mathbf{R} \mid \mathbf{R}V$
$\mathbf{W} ::= (\!|\,|\!) \mid \mathbf{W}M \mid M\mathbf{W}$

The closure under **L** (resp. **W**,**R**) context is noted $\underset{l}{\to}$ (resp $\underset{w}{\to}$, $\underset{r}{\to}$)

▶ **Fact 3** (Weak normal forms). *Given M a closed term, M is $\underset{s}{\to}$-normal* iff *M is a value.*

Question: which of the above reductions are deterministic?

▶ **Fact 6** (?). *Let M be a closed term.*
1. *$M \underset{l}{\to}^* V$ iff $M \underset{r}{\to}^* V$. True?*
2. *$M \underset{w}{\to}^V$ iff $M \underset{l}{\to}^*$. True?*
3. *Assume you proved $M \underset{l}{\to}^k V$ (runtime is k). Does the sequence of $\underset{r}{\to}$-steps also terminates? Can we say how long does it take?*
4. *With the same assumption as above, what about $\underset{w}{\to}$?*

75

CbV Weak Factorization

Weak Factorization.

Let $\mathsf{s} \in \{\mathsf{w}, \mathsf{l}, \mathsf{r}\}$
- *weak factorization of* $\to_{\beta_v}$: $\quad \to_{\beta_v}{}^* \subseteq \underset{s}{\to}_{\beta_v}{}^* \cdot \underset{\neg s}{\to}_{\beta_v}{}^*$.
- *Convergence:* $\quad T \to_{\beta_v} W (W \in \mathcal{V})$ if and only if $T \underset{s}{\to}_{\beta_v}{}^* V \; (V \in \mathcal{V})$

▶ **Corollary 4.** *Given M a closed term, M has a β_v-reduction to a value, if and only if the $\underset{s}{\to}_{\beta_v}$-reduction from M terminates.*

76

Basic properties of the contextual closure

77

Basic properties of contextual closure

If a step $T \to_\gamma T'$ is obtained by closure under *non-empty context* of a rule $\mapsto_\gamma$, then T and T' have the same shape, i.e. *both terms are an application (resp. an abstraction, a variable).*

▶ **Fact 5** (Shape preservation).
- *Assume $T = \mathbf{C}(\!|R|\!) \to \mathbf{C}(\!|R'|\!) = T'$ and that the context* **C** *is* non-empty. *Then T and T' have the same shape.*
- *Hence, for any internal step $M \underset{\neg s}{\to} M'$ ($\mathsf{s} \in \{\mathsf{h}, \mathsf{w}, \mathsf{l}, \mathsf{r}, ...\}$) M and M have the same shape.*

The following is an easy to verify consequence.

▶ **Lemma 6** (Redexes preservation).
1. *CbN: Assume $T \underset{\neg h}{\to}_\beta S$. T is a β-redex* iff *so is S.*
2. *CbV. Assume $T \underset{\neg w}{\to}_{\beta_v} S$. T is a β_v-redex* iff *so is S.*

78

Internal steps preserve head and weak normal nf

Fixed a set of redexes $\mathcal{R}$, M is w-normal (resp. h-normal) if there is no redex $R \in \mathcal{R}$ such that $M = \mathbf{W}\langle R \rangle$ (resp. $M = \mathbf{H}\langle R \rangle$)

- **Lemma 7** (Surface normal forms). 1. *CbN. Let $\mathcal{R}$ be the set of β-redexes. Assume $M \rightarrow_{\neg h \beta} M'$. M is h-normal $\Leftrightarrow$ M' is h-normal.*

2. *CbV. Let $\mathcal{R}$ be the set of β_v-redexes. Assume $M \rightarrow_{\neg w \beta_v} M'$. M is w-normal $\Leftrightarrow$ M' is w-normal.*

79

Back to Factorization

Back to using it

80

From abstract to concrete system

… but using **as-little-of-the-specific-structure** as possible

81

ARS Recipe

- **Property 2** (Criterion). *Given* $\rightarrow = \rightarrow_{e} \cup \rightarrow_{i}$, *e-factorization holds*

$$\rightarrow^* \subseteq \rightarrow_{e}^* \cdot \rightarrow_{i}^*$$

iff exists $\Rightarrow_{i}$

- $\Rightarrow_{i}^* = \rightarrow_{i}^*$ *(same closure)*
- $\Rightarrow_{i} \cdot \rightarrow_{e} \subseteq \rightarrow_{e}^* \cdot \Rightarrow_{i}^=$ *(strong postponement)*

82

Concretely: CbN and Head Factorization

Takahashi method

Indexed parallel β reduction

$$\frac{}{x \overset{0}{\Rightarrow}_\beta x} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t'}{\lambda x.t \overset{n}{\Rightarrow}_\beta \lambda x.t'} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t' \quad s \overset{m}{\Rightarrow}_\beta s'}{ts \overset{n+m}{\Rightarrow}_\beta t's'} \qquad \frac{t \overset{n}{\Rightarrow}_\beta t' \quad s \overset{m}{\Rightarrow}_\beta s'}{(\lambda x.t)s \overset{n+|t'|_x \cdot m+1}{\Rightarrow}_\beta t'\{x \leftarrow s'\}}$$

Parallel $\neg$head reduction

$$\frac{}{x \Rightarrow_{\neg h} x} \qquad \frac{t \Rightarrow_\beta t' \quad s \Rightarrow_\beta s'}{(\lambda x.t)s \Rightarrow_{\neg h} (\lambda x.t')s'} \qquad \frac{t \Rightarrow_{\neg h} t'}{\lambda x.t \Rightarrow_{\neg h} \lambda x.t'} \qquad \frac{t \Rightarrow_{\neg h} t' \quad s \Rightarrow_\beta s'}{ts \Rightarrow_{\neg h} t's'}$$

$$\rightarrow_{\neg h} \subseteq \Rightarrow_{\neg h} \subseteq \rightarrow_{\neg h}^*$$

1. Merge: *if* $t \Rightarrow_{\neg h} \cdot \rightarrow_h u$ *then* $t \Rightarrow_\beta u$.
2. [redacted]
3. Split: *if* $t \Rightarrow_\beta s$ *then* $t \rightarrow_h^* \cdot \Rightarrow_{\neg h} s$.

83

Concretely: CbV and Weak Factorization

Indexed parallel β_v reduction

$$\frac{}{x \overset{0}{\Rightarrow}_{\beta_v} x} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t'}{\lambda x.t \overset{n}{\Rightarrow}_{\beta_v} \lambda x.t'} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t' \quad s \overset{m}{\Rightarrow}_{\beta_v} s'}{ts \overset{n+m}{\Rightarrow}_{\beta_v} t's'} \qquad \frac{t \overset{n}{\Rightarrow}_{\beta_v} t' \quad v \overset{m}{\Rightarrow}_{\beta_v} v'}{(\lambda x.t)v \overset{n+|t'|_x \cdot m+1}{\Rightarrow}_{\beta_v} t'\{x \leftarrow v'\}}$$

Parallel $\neg$weak reduction

$$\frac{}{x \Rightarrow_{\neg w} x} \qquad \frac{t \Rightarrow_{\beta_v} t'}{\lambda x.t \Rightarrow_{\neg w} \lambda x.t'} \qquad \frac{t \Rightarrow_{\neg w} t' \quad s \Rightarrow_{\neg w} s'}{ts \Rightarrow_{\neg w} t's'}$$

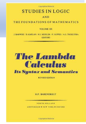
$$\rightarrow_{\neg w} \subseteq \Rightarrow_{\neg w} \subseteq \rightarrow_{\neg w}^*$$

1. Merge: *if* $t \Rightarrow_{\neg w} \cdot \rightarrow_w u$ *then* $t \Rightarrow_{\beta_v} u$.
2. [redacted]
3. Split: *if* $t \Rightarrow_{\beta_v} s$ *then* $t \rightarrow_w^* \cdot \Rightarrow_{\neg w} s$.

84

Recap

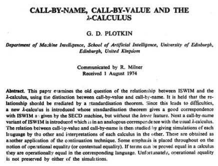
- Classical key result (e.g. in Barendregt 84 book)



in Call-by-Name:

- Head Factorization: $\to_\beta^* \subseteq \underset{h}{\to}_\beta^* \cdot \underset{\neg h}{\to}_\beta^*$.
- Head Normalization: M *has* hnf *if and only if* $M \underset{h}{\to}_\beta^* S$ *(for some* $S \in \mathcal{H}$*)* .

- Classical key result [Plotkin 75]



in Call-by-Value:

Let s ∈ {w,l,r}

- *weak factorization of* $\to_{\beta_v}$: $\to_{\beta_v}^* \subseteq \underset{s}{\to}_{\beta_v}^* \cdot \underset{\neg s}{\to}_{\beta_v}^*$.
- *Convergence:* $T \to_{\beta_v} W (W \in \mathcal{V})$ if and only if $T \underset{s}{\to}_{\beta_v}^* V (V \in \mathcal{V})$

85

You designed a system
You have Factorization
Now what?

From Factorization to **Normalization** (or Standardization)
in a few easy steps [Mitschke 79]

86

ARS:
more abstract tools

Decreasing Diagrams:
Lecture Notes, Chapter 6

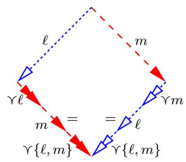
87

Decreasing (Van Oostrom)

Definition 2. $\blacktriangleright, \triangleright$ *is* decreasing, *if* $\triangleright = \bigcup_{\ell \in L} \triangleright_\ell$, $\blacktriangleright = \bigcup_{m \in M} \blacktriangleright_m$

for families of relations $((\triangleright_\ell)_{\ell \in L}, (\blacktriangleright_m)_{m \in M})$

*and some **well-founded strict order < on** the set of labels L ∪ M*
such that for all labels:



where $\curlyvee N = \{n \in L \cup M \mid \exists k \in N\ k \succ n\}$*, and* $\curlyvee n$ *abbreviates* $\curlyvee\{n\}$

88

To commute

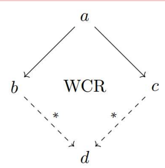
Definition 1. *A pair* $(\triangleright, \blacktriangleright)$ *of rewrite relations* commutes *if* $\lhd\!\!\lhd \cdot \blacktriangleright\!\!\blacktriangleright \subseteq \blacktriangleright\!\!\blacktriangleright \cdot \lhd\!\!\lhd$*, and* commutes locally *if* $\lhd \cdot \blacktriangleright \subseteq \blacktriangleright\!\!\blacktriangleright \cdot \lhd\!\!\lhd$*. A rewrite relation* $\to$ *is* confluent *if* $(\to, \to)$ *commutes, and* locally confluent *if* $(\to, \to)$ *commutes locally.*

Theorem 1 ([1]). *A pair of rewrite relations commutes if it is decreasing. A rewrite relation is confluent if it is decreasing.*

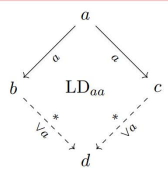
89

Newman Lemma, again

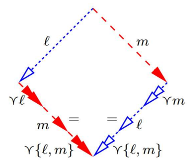
Newman's Lemma. *Every terminating and locally confluent ARS is confluent.*



$\langle A, \to \rangle$



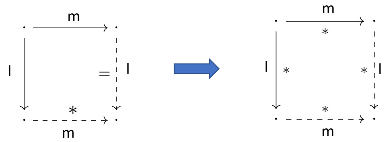
$\langle A, \{\to_a\}_{a \in A} \rangle$

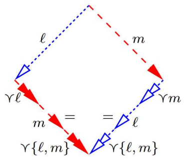


$\curlyvee N = \{n \in L \cup M \mid \exists k \in N\ k \succ n\}$

90

Strong Commutation implies Commutation



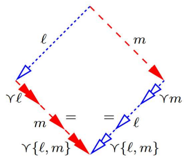


l > m

$\curlyvee N = \{n \in L \cup M \mid \exists k \in N\ k \succ n\}$

91

This is an instance of strong commutation



$\curlyvee N = \{n \in L \cup M \mid \exists k \in N\ k \succ n\}$

92