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)

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

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)

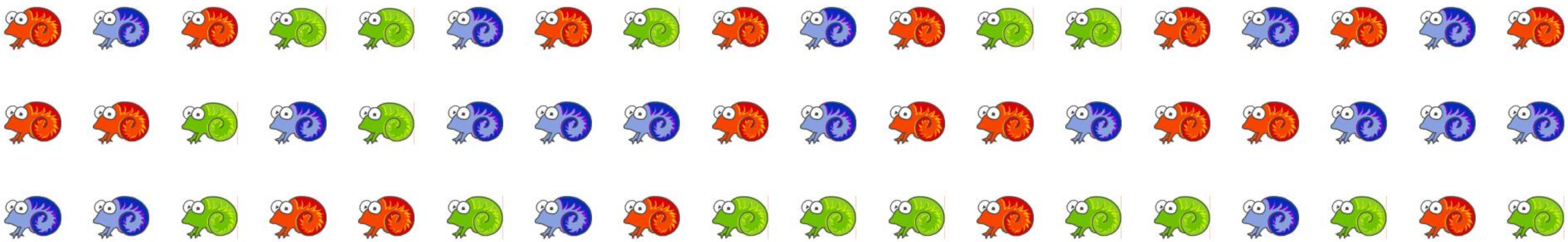
- **Lecture Notes** (by Middeldorp, Laurent, Ong)

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

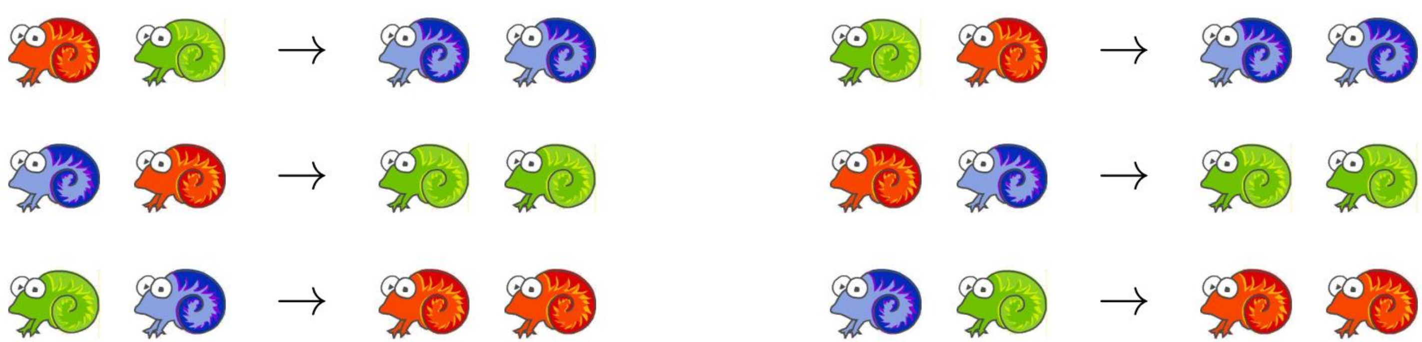
Operational semantics
of formal calculi and programming languages

Rewriting theory

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



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?



Example (Group Theory)

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

rewrite rules $\mathcal{R}$

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

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

Example (Combinatory Logic)

signature: S K I (constants) · (application, binary, infix)

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

rewrite rules:

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

rewriting:

$$((\mathsf{S} \cdot \mathsf{K}) \cdot \mathsf{K}) \cdot x \to (\mathsf{K} \cdot x) \cdot (\mathsf{K} \cdot x) \to x$$

inventor: Moses Schönfinkel (1924)

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

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)

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.

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

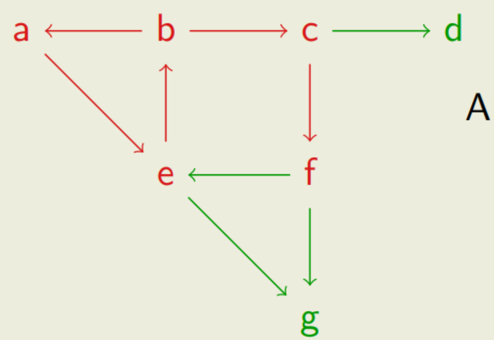
- **independent from structure** of objects that are rewritten
- **uniform** presentation of properties and proofs

Abstract Rewriting

Basic language

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, … 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 → · · ·

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

$$s \leftrightarrow_{\mathcal{R}} t \text{ iff } s \rightarrow_{\mathcal{R}} t \text{ or } t \rightarrow_{\mathcal{R}} s$$

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

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

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

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

Composition

- We denote $\rightarrow^*$ (resp. $\rightarrow^=$) the transitive-reflexive (resp. reflexive) closure of $\rightarrow$;
- If $\rightarrow_1, \rightarrow_2$ are binary relations on A then $\rightarrow_1 \cdot \rightarrow_2$ denotes their composition, *i.e.* $t \rightarrow_1 \cdot \rightarrow_2 s$ iff there exists $u \in A$ such that $t \rightarrow_1 u \rightarrow_2 s$.
- We write $(A, \{\rightarrow_1, \rightarrow_2\})$ to denote the ARS $(A, \rightarrow)$ where $\rightarrow = \rightarrow_1 \cup \rightarrow_2$.

Closure

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

$$\rightarrow \subseteq \rightarrow^*, \qquad (\rightarrow^*)^* = \rightarrow^*, \qquad \rightarrow_1 \subseteq \rightarrow_2 \text{ implies } \rightarrow_1^* \subseteq \rightarrow_2^*$$

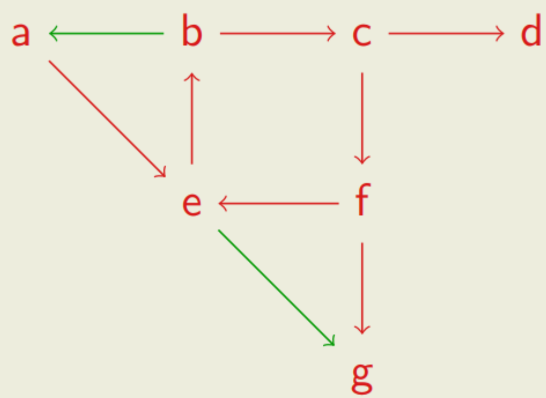
As a consequence

$$(\rightarrow_1 \cup \rightarrow_2)^* = (\rightarrow_1^* \cup \rightarrow_2^*)^*.$$

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

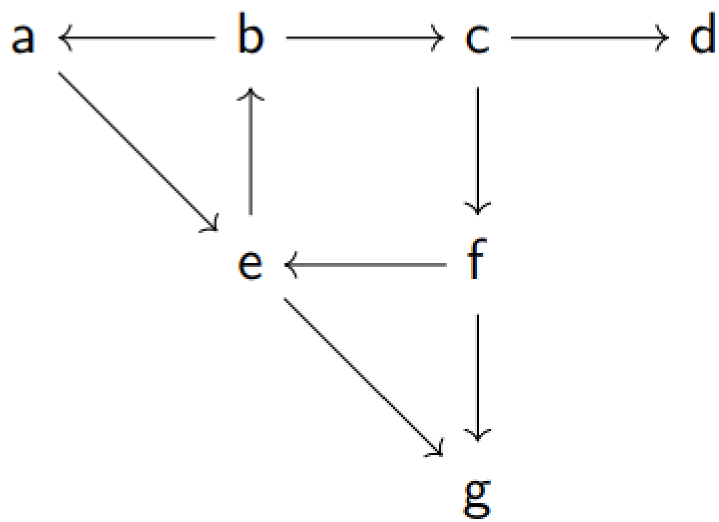
Example



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

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

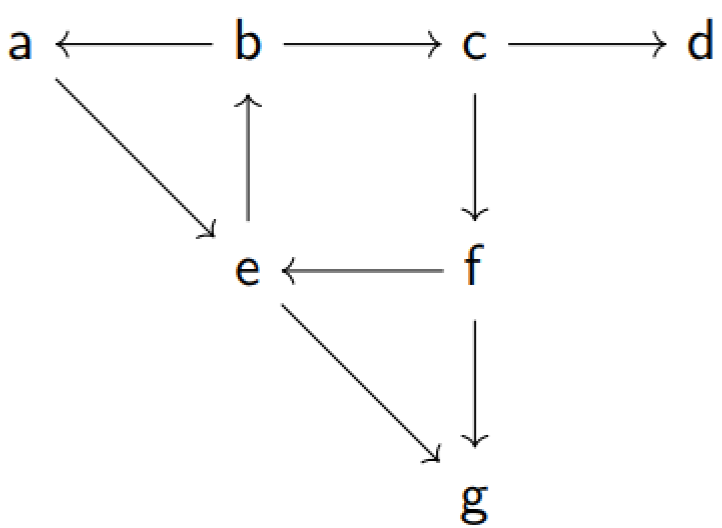
- d is normal form
- $\mathsf{NF}(\mathcal{A}) = \{ d, g \}$
- $b \to^! g$

- 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 $b = c$

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.

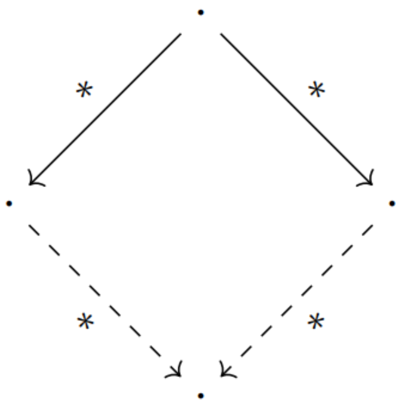


a is WN? SN?
c is WN? SN?
a or c has UN ?

The nf are convertible?

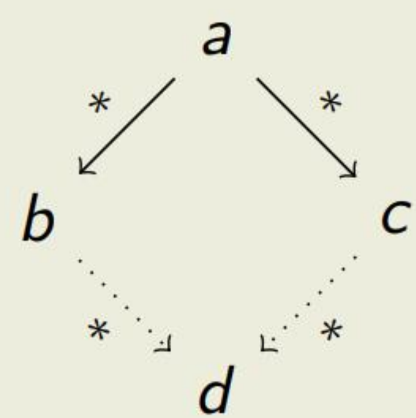
Confluence

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.

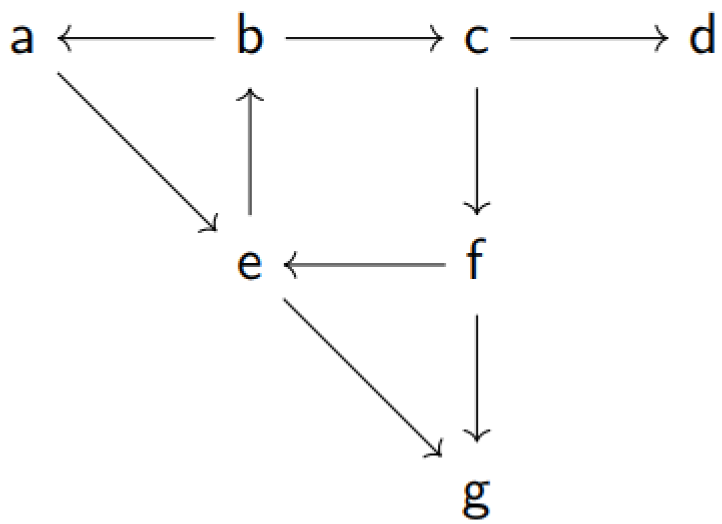


- $\forall\, a, b, c$

  $\exists\, d$



Every confluent ARS *has unique normal forms.*



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 ?**

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?

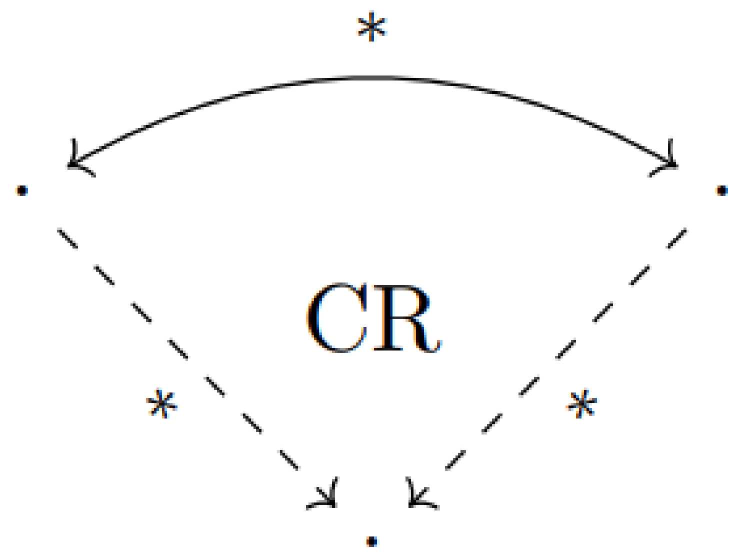
Can you produce two different nf?

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

different meaning for equivalent terms

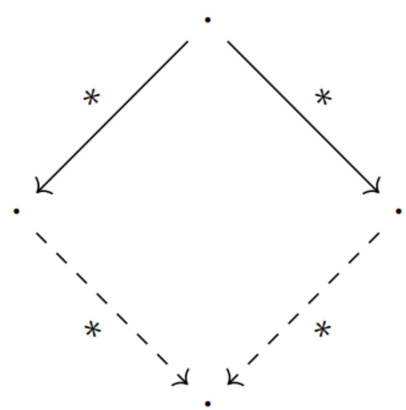
(different meaning for same term!)

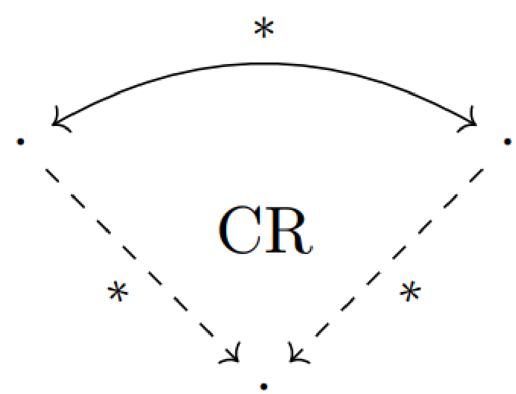
Same meaning for *equivalent* terms



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.



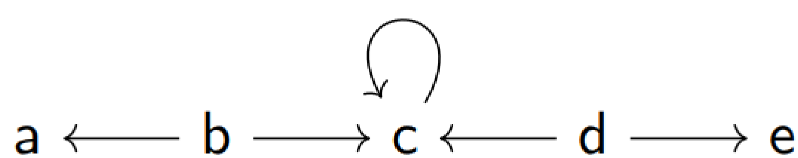


An ARS $\mathcal{A} = \langle A, \rightarrow \rangle$ *is confluent if and only if* $\leftrightarrow^{*} \subseteq \downarrow$.

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?

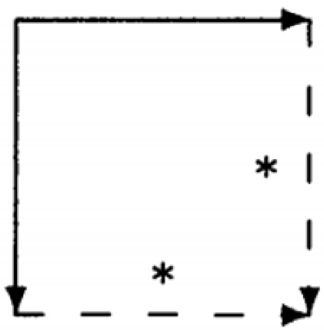


Global vs Local

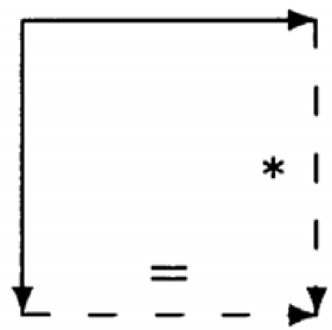
Confluence

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*.

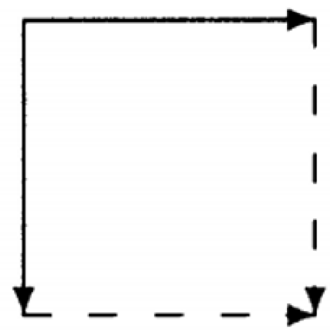
Locally confluent (WCR)



Strongly confluent

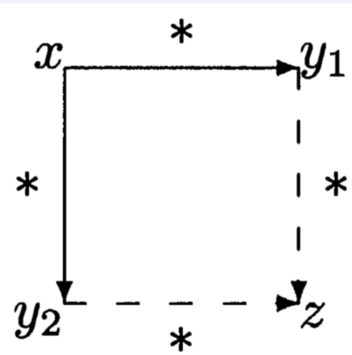


Diamond



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.

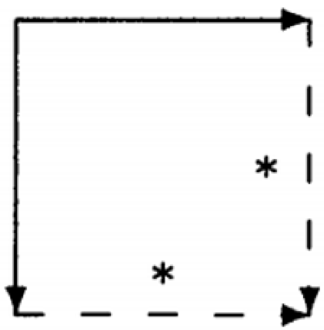
Global property:

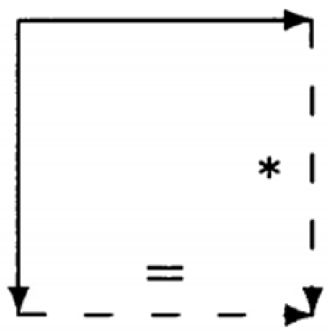


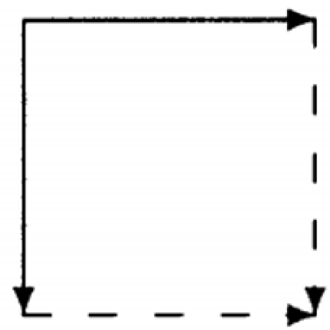
Confluence

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**





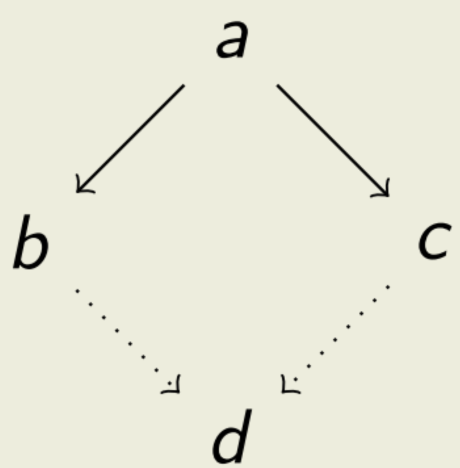


Local confluence Let $\mathcal{A} = \langle A, \to \rangle$ be an ARS. An element $a \in A$ is locally confluent for all elements $b, c \in A$ with $b \leftarrow a \rightarrow c$ we have $b \downarrow c$. The ARS $\mathcal{A}$ is confluent if all its elements are confluent.

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

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

    $\exists\, d$



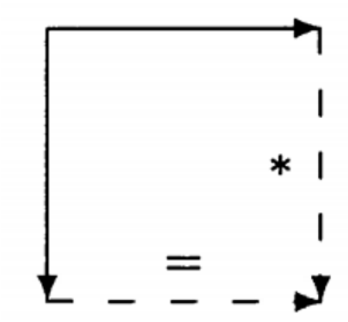
- *every ARS with diamond property is confluent*

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

a Show that every strongly confluent ARS is confluent.

b Does the converse also hold?

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



Which is true?

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

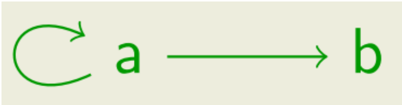
3. Confluence => UN
4. UN => Confluence

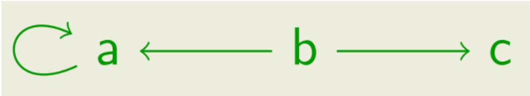
5. Confluence => Local confluence
6. Local confluence => Confluence

7. WN & UN => Confluence

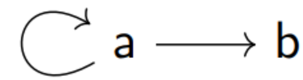
8. WN & Local Conf. => Confluence

9. SN & Local Conf. => Confluence





WN vs SN



(ii) WN $\not\Rightarrow$ SN

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

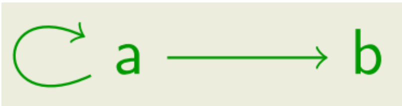
The system is weakly normalising but not strongly normalising:

Can you find an infinite reduction sequence?

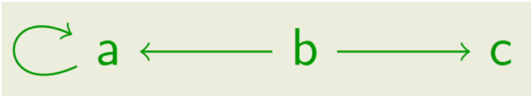
$$f(b) \rightarrow f(a) \rightarrow c$$

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

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

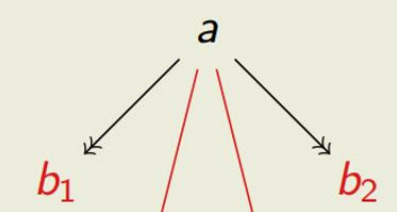
Newman's Lemma

Lemma

WN & UN $\implies$ CR

Proof

- WN $\implies$ $\exists\, n_1, n_2 :\; b_1 \to^! n_1$ and $b_2 \to^! n_2$



- UN $\implies$ $n_1 = n_2$ $\implies$ $b_1 \downarrow b_2$

Newman Lemma

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

By well-founded induction

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

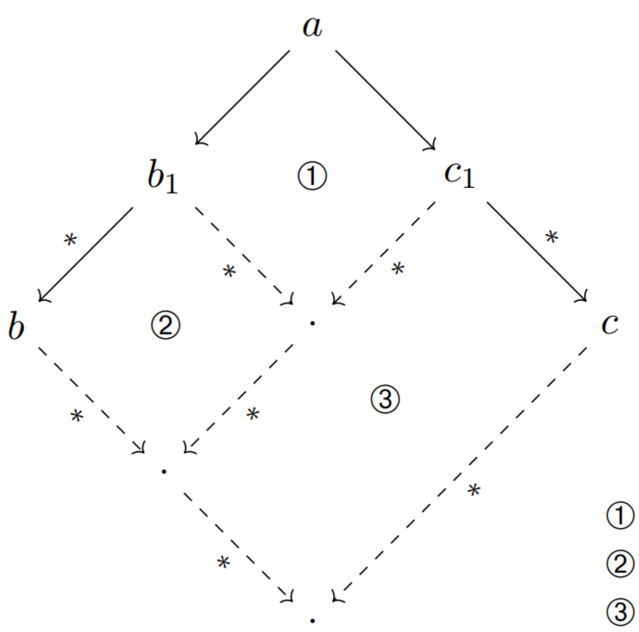
1. pour tout élément **minimal** $m \in \mathscr{M}$ on a **P(m)**
2. le fait que **P(k)** soit vérifiée pour **tout** élément $k < x$ implique **P(x)**

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 $\mathcal{A} = \langle A, \rightarrow \rangle$ 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 $a \rightarrow b$ have the property $\mathcal{P}$. In particular every normal form has to satisfy the property $\mathcal{P}$.

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)

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 $B = \{ a \in A \mid \neg \mathsf{UN}(a) \} \neq \varnothing$
- 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**)