# Church-Rosser Properties of Terminating First-Order and Higher-Order Rewriting Relations

Jean-Pierre Jouannaud
LIX, École Polytechnique, Palaiseau

Joint work with Femke van Raamsdonk
Faculty of Sciences, Vrije Universiteit, Amsterdam

ICMS, Edinburgh, May 26-28, 2007

# Outline

Examples and questions

# Plain first-order rewriting

rule (associativity):

$$(x + y) + z \to x + (y + z)$$

rewrite step:

$$(1 + 2) + 3 \to 1 + (2 + 3)$$

this is called plain rewriting

rule (inverse):

$$x + (-x) \to 0$$

equations (commutativity and associativity):

$$\begin{aligned} x + y &= y + x \\ (x + y) + z &= x + (y + z) \end{aligned}$$

rewrite step:

$$-x+(x+y) = (-x+x)+y = (x+-x)+y \to 0+y$$

this is called class rewriting

# Plain Higher-Order Rewriting

rules (recursor and beta):

$$
\begin{array}{rcl}
rec(0, u, f) & \to & u \\
rec(s(y), u, f) & \to & @(f, y, rec(y, u, f)) \\
@(\lambda z.u, v) & \to & u\{z \mapsto v\}
\end{array}
$$

rewrite step:

$$
\begin{array}{l}
rec(s(0), 1, \lambda xy. + (x, y)) \to \\
@(\lambda xy. + (x, y), 0, rec(0, 1, \lambda xy. + (x, y))) \to \\
@(\lambda xy. + (x, y), 0, 1) \to +(0, 1) \to 1
\end{array}
$$

Uses plain pattern matching in presence of binders

# Plain Higher-Order Rewriting continued

[Barendregt and Klop]:

$$\begin{array}{rcl} \omega\,1 & = & (\lambda x\,.\,x\,x)(\lambda s.\lambda z.s\,z) \\ & \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\,z) \\ & \longrightarrow & \lambda z.(\lambda s.\lambda z.s\,z)z \\ & \overset{\Lambda}{\underset{\alpha}{\longleftrightarrow}} & \lambda z'.(\lambda s.\lambda z.s\,z)z' \\ & \overset{1}{\underset{\beta}{\longrightarrow}} & \lambda z'.(\lambda z.z'\,z) \end{array}$$

Plain HOR is a form of
class rewriting modulo $\alpha$-conversion

rules (differentiation):

$$\begin{aligned} \mathit{diff}(\lambda x.\mathit{sin}(f(x))) &\rightarrow \lambda x.\mathit{cos}(f(x)) * \mathit{diff}(f) \\ \mathit{diff}(\lambda x.x) &\rightarrow 1 \end{aligned}$$

rewrite step:

$$\begin{aligned} \mathit{diff}(\lambda x.\mathit{sin}(x)) &\overset{\Lambda}{\underset{\beta}{\longleftrightarrow}} \mathit{diff}(\lambda x.\mathit{sin}(@(\lambda x.x, x))) \\ &\longrightarrow \lambda x.\mathit{cos}(x) * \mathit{diff}(\lambda x.x) \\ &\longrightarrow \lambda x.\mathit{cos}(x) * \mathit{diff}(\lambda x.x) \\ &\longrightarrow \lambda x.\mathit{cos}(x) \end{aligned}$$

Higher-order rewriting is another form of class rewriting modulo alpha, beta and eta.

# Questions

1. is my calculus terminating ?
2. is my calculus confluent ?

We focus on

- Confluence assuming termination
- General abstract results
- A treatment of binders as a particular case
- Application to higher-order rewriting

A Review

# Example of confluence

rule:

$$(x + y) + z \to x + (y + z)$$

converging divergence:

$$
\begin{array}{lll}
 & \to & 1 + (2 + (3 + 4)) \\
 & \to & (1 + 2) + (3 + 4) \\
((1 + 2) + 3) + 4 & & \\
 & \to & (1 + (2 + 3)) + 4 \\
 & \to & 1 + ((2 + 3) + 4) \\
 & \to & 1 + (2 + (3 + 4))
\end{array}
$$

# Example of non-confluence

rules:

$$\begin{array}{rcl} (x + y) + z & \rightarrow & x + (y + z) \\ x + 0 & \rightarrow & x \end{array}$$

Non-converging divergence:

$$(1 + 0) + 3 \quad \begin{array}{l} \rightarrow \quad 1 + (0 + 3) \\ \\ \rightarrow \quad 1 + 3 \end{array}$$

# Definitions

Divergence: $t_1 \longleftarrow^* s \longrightarrow^* t_2$

Local divergence: $t_1 \longleftarrow s \longrightarrow t_2$

Joinability: $t_1 \longrightarrow^* u \longleftarrow^* t_2$

Confluence:
every divergence is joinable.

Local confluence:
every local divergence is joinable.

# Proving confluence

1. via orthogonality (see [Terese])
2. via local confluence and termination:
   (i) confluence reduces to local confluence
   (ii) local confluence reduces to the joinability of critical pairs

# Example of critical pairs for plain first-order rewriting

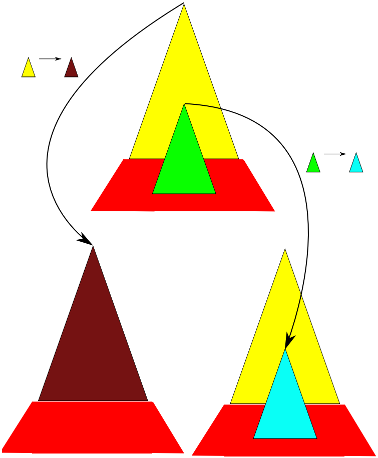
rules:

$$\begin{array}{rcl} (x + y) + z & \to & x + (y + z) \\ x + 0 & \to & x \end{array}$$

critical pair: most general divergence

$$x + (0 + z) \leftarrow (x + 0) + z \to x + z$$

# Critical pairs for plain first-order rewriting



# Critical pairs for plain first-order rewriting

Assume

*rules* $l \rightarrow r$ *and* $g \rightarrow d$

*non-variable position* $p$ *in* $l$

*mgu* $\sigma$ *such that* $l\sigma|_p = g\sigma$

then:

$$r\sigma \xleftarrow{\Lambda} l\sigma = l\sigma[g\sigma]_p \xrightarrow{p} l\sigma[d\sigma]_p$$

$\sigma$ is the mgu of $l|_p = g$ because $l\sigma|_p = l|_p\sigma$ in the absence of binders

With binders, discard mgus of $l|_p = g$ such that $l\sigma|_p \neq g\sigma$.

# Critical pairs for plain first-order rewriting

Assume

*rules* $l \to r$ *and* $g \to d$

*non-variable position p in l*

*mgu* $\sigma$ *such that* $l\sigma|_p = g\sigma$

then:

$$r\sigma \xleftarrow{\Lambda} l\sigma = l\sigma[g\sigma]_p \xrightarrow{p} l\sigma[d\sigma]_p$$

$\sigma$ is the mgu of $l|_p = g$ because

$l\sigma|_p = l|_p\sigma$ in the absence of binders

With binders, discard mgus of $l|_p = g$ such that $l\sigma|_p \neq g\sigma$.

# Critical pairs for plain first-order rewriting

Assume

*rules* $l \to r$ *and* $g \to d$

*non-variable position p in l*

*mgu* $\sigma$ *such that* $l\sigma|_p = g\sigma$

then:

$$r\sigma \overset{\Lambda}{\longleftarrow} l\sigma = l\sigma[g\sigma]_p \overset{p}{\longrightarrow} l\sigma[d\sigma]_p$$

$\sigma$ is the mgu of $l|_p = g$ because
$l\sigma|_p = l|_p\sigma$ in the absence of binders

With binders, discard mgus of $l|_p = g$
such that $l\sigma|_p \neq g\sigma$.

# Rewriting modulo

# Rewriting modulo given: $S$, $R$, $\longrightarrow_{RS}$

**Proof:** $t_1 \longleftrightarrow^*_{R \cup S} t_2$

**Joinability:** $t_1 \longrightarrow^*_{RS} u \longleftrightarrow^*_{S} v \longleftarrow^*_{RS} t_2$

**Church-Rosser:** every proof is joinable.

**Local confluence:** every local divergence
$t_1 \longleftarrow_{RS} s \longrightarrow_{RS} t_2$ is joinable

**Local coherence:** every local semi-divergence
$t_1 \longleftarrow_{RS} s \longleftrightarrow_{S} t_2$ is joinable

Church-Rosser reduces to both local properties

## Rewriting modulo given: $S$, $R$, $\longrightarrow_{RS}$

Proof: $t_1 \longleftrightarrow^*_{R \cup S} t_2$

Joinability: $t_1 \longrightarrow^*_{RS} u \longleftrightarrow^*_{S} v \longleftarrow^*_{RS} t_2$

Church-Rosser: every proof is joinable.

Local confluence: every local divergence
$t_1 \longleftarrow_{RS} s \longrightarrow_{RS} t_2$ is joinable

Local coherence: every local semi-divergence
$t_1 \longleftarrow_{RS} s \longleftrightarrow_{S} t_2$ is joinable

Church-Rosser reduces to both local properties

# Rewriting modulo given: $S$, $R$, $\longrightarrow_{RS}$

Proof: $t_1 \longleftrightarrow^*_{R \cup S} t_2$

Joinability: $t_1 \longrightarrow^*_{RS} u \longleftrightarrow^*_{S} v \longleftarrow^*_{RS} t_2$

Church-Rosser: every proof is joinable.

Local confluence: every local divergence
$t_1 \longleftarrow_{RS} s \longrightarrow_{RS} t_2$ is joinable

Local coherence: every local semi-divergence
$t_1 \longleftarrow_{RS} s \longleftrightarrow_{S} t_2$ is joinable

Church-Rosser reduces to both local properties

# Class rewriting [Lankford & Ballantyne]

$$s \rightarrow_{RS} t$$

if there is some $s'$ such that

$$s \longleftrightarrow^{*}_{S} s' \longrightarrow^{p}_{R} t$$

Set of rules $R = \{x + (-x) \rightarrow 0\}$

Set of equations $S = \{(x + y) + z = x + (y + z)\}$

Rewrite step:

$$(1 + 2) + (-2) \xleftrightarrow[S]{\Lambda} 1 + (2 + (-2)) \xrightarrow[R]{2} 1 + 0$$

The equality step occurs above the rewrite step

# Class rewriting continued

| | |
|---|---|
| rule | $b \to c$ |
| equation | $a = f(b)$ |
| rewrite step | $a \longleftrightarrow^{\Lambda} f(b) \longrightarrow^{1} f(c)$ |

Is it a rewrite at position 1 in $a$ ?

Makes sense for very specific theories :

1. permutative equations
2. associativity
3. alpha-conversion
4. their combinations

# Class rewriting continued

| | |
|---|---|
| rule | $b \rightarrow c$ |
| equation | $a = f(b)$ |
| rewrite step | $a \longleftrightarrow^{\Lambda} f(b) \longrightarrow^{1} f(c)$ |

Is it a rewrite at position 1 in $a$ ?

Makes sense for very specific theories :

1. permutative equations
2. associativity
3. alpha-conversion
4. their combinations

# Plain higher-order rewriting

is class rewriting modulo alpha-conversion [Barendregt and Klop]:

$$\begin{array}{rcl} \omega\, 1 & = & (\lambda x.\, x\, x)(\lambda s.\lambda z.s\, z) \\ & \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\, z) \\ & \longrightarrow & \lambda z.(\lambda s.\lambda z.s\, z)z \\ & \xleftrightarrow[\alpha]{\Lambda} & \lambda z'.(\lambda s.\lambda z.s\, z)z' \\ & \xrightarrow[\beta]{1} & \lambda z'.(\lambda z.z'\, z) \end{array}$$

Binders: requires a non-variable capturing substitution and unification modulo the theory of binders for computing critical pairs;

General case: complete sets of $S$-unifiers needed;

All previous theories have finite CSUs.

# Plain higher-order rewriting

is class rewriting modulo alpha-conversion [Barendregt and Klop]:

$$
\begin{array}{rcl}
\omega\, 1 & = & (\lambda x.\, x\, x)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & \lambda z.(\lambda s.\lambda z.s\, z)z \\
& \overset{\Lambda}{\underset{\alpha}{\longleftrightarrow}} & \lambda z'.(\lambda s.\lambda z.s\, z)z' \\
& \overset{1}{\underset{\beta}{\longrightarrow}} & \lambda z'.(\lambda z.z'\, z)
\end{array}
$$

Binders: requires a non-variable capturing substitution and unification modulo the theory of binders for computing critical pairs;
General case: complete sets of $S$-unifiers needed;
All previous theories have finite CSUs.

# Plain higher-order rewriting

is class rewriting modulo alpha-conversion [Barendregt and Klop]:

$$
\begin{array}{rcl}
\omega\, 1 & = & (\lambda x . x\, x)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & \lambda z.(\lambda s.\lambda z.s\, z)z \\
& \overset{\Lambda}{\underset{\alpha}{\longleftrightarrow}} & \lambda z'.(\lambda s.\lambda z.s\, z)z' \\
& \overset{1}{\underset{\beta}{\longrightarrow}} & \lambda z'.(\lambda z.z'\, z)
\end{array}
$$

Binders: requires a non-variable capturing substitution and unification modulo the theory of binders for computing critical pairs;
General case: complete sets of $S$-unifiers needed;
All previous theories have finite CSUs.

# Plain rewriting modulo [Huet]

$$s \longrightarrow^{p}_{RS} t \text{ iff } s \longrightarrow^{p}_{R} t$$

Restrictions to reduce local properties to critical (confluence and coherence) pairs:

- rewrite rules must be left-linear
- equations must be linear

Does not apply to plain higher-order rewriting because renaming is only possible at the end.

# Rewriting modulo [Peterson and Stickel]

Equalities must occur below the rewrite step:

$$s \xrightarrow[R_S]{p} t$$

iff

$$s \xleftrightarrow[S]{\geq p} s' \xrightarrow[l \to r]{p} t$$

that is

$$\begin{array}{ccc} s|_p & \longleftrightarrow^*_S & l\sigma \\ t & = & s[r\sigma]_p \end{array}$$

S-matching has replaced plain matching.

# Rewriting modulo [Peterson and Stickel]

Equalities must occur below the rewrite step:

$$s \longrightarrow^{p}_{R_S} t$$

iff

$$s \xleftrightarrow[S]{\geq p} s' \xrightarrow[l \to r]{p} t$$

that is

$$\begin{array}{ccc} s|_p & \longleftrightarrow^{*}_{S} & l\sigma \\ t & = & s[r\sigma]_p \end{array}$$

S-matching has replaced plain matching.

# Example

$$\begin{array}{rcl} S & = & \{x + y = y + x, (x + y) + z = x + (y + z)\} \\ R & = & \{x + (-x) \longrightarrow 0\} \end{array}$$

Rewrite step:

$$(-2) + 2 \xleftrightarrow[AC]{\Lambda} 2 + (-2) \xrightarrow[R]{\Lambda} 0$$

Non-rewrite step:

$$(x + y) + (-y) \xleftrightarrow[A]{\Lambda} x + (y + (-y)) \xrightarrow[R]{2} x + 0$$

$$(x + -x) + y \;\rightarrow\; 0 + y$$

to resolve the local semi-divergences:

$$(x + y) + (-y) \xleftrightarrow[S]{\Lambda^2} (y + (-y)) + x \xrightarrow[R]{\Lambda} x + 0$$

$$x + ((-x) + z) \xleftrightarrow[S]{\Lambda} (x + (-x)) + z \xrightarrow[R]{\Lambda} x + 0$$

**Theorem**

*Assuming class-rewriting terminates, S-equivalence classes are size-bounded, and R is closed under extensions, Church-Rosser reduces to joinability of all S-critical pairs.* [Jouannaud and Kirchner]

# Extensions [Stickel, Jouannaud-Kirchner]

$$(x + -x) + y \;\rightarrow\; 0 + y$$

to resolve the local semi-divergences:

$$(x+y)+(-y) \underset{S}{\overset{\Lambda^2}{\longleftrightarrow}} (y+(-y))+x \underset{R}{\overset{\Lambda}{\longrightarrow}} x+0$$

$$x+((-x)+z) \underset{S}{\overset{\Lambda}{\longleftrightarrow}} (x+(-x))+z \underset{R}{\overset{\Lambda}{\longrightarrow}} x+0$$

**Theorem**

*Assuming class-rewriting terminates, S-equivalence classes are size-bounded, and R is closed under extensions, Church-Rosser reduces to joinability of all S-critical pairs. [Jouannaud and Kirchner]*

# Plain higher-order rewriting as rewriting modulo

$$
\begin{array}{rcl}
\omega\, 1 & = & (\lambda x.x\,x)(\lambda s.\lambda z.s\,z) \\
& \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\,z) \\
& \longrightarrow & \lambda z.(\lambda s.\lambda z.s\,z)z \\
& \overset{1\cdot 1}{\underset{\alpha}{\longleftrightarrow}} & \lambda z.(\lambda s.\lambda z'.s\,z')z \\
& \overset{1}{\underset{\beta}{\longrightarrow}} & \lambda z.(\lambda z'.z\,z')
\end{array}
$$

- Non-variable capture taken care of by pattern-matching
- Alpha-extensions are not needed for plain HOR: they are joinable;
- Critical pairs use mgu modulo alpha;
- Yields a clean handling of alpha-conversion.

# Plain higher-order rewriting as rewriting modulo

$$
\begin{array}{rcl}
\omega\, 1 & = & (\lambda x.\, x\, x)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & (\lambda s.\lambda z.sz)(\lambda s.\lambda z.s\, z) \\
& \longrightarrow & \lambda z.(\lambda s.\lambda z.s\, z)z \\
& \xleftrightarrow[\alpha]{1\cdot 1} & \lambda z.(\lambda s.\lambda z'.s\, z')z \\
& \xrightarrow[\beta]{1} & \lambda z.(\lambda z'.z\, z')
\end{array}
$$

- Non-variable capture taken care of by pattern-matching
- Alpha-extensions are not needed for plain HOR: they are joinable;
- Critical pairs use mgu modulo alpha;
- Yields a clean handling of alpha-conversion.

# Normalized rewriting [Marché]

$$s \xrightarrow{p}_{R_S} t$$

iff

$$s \xrightarrow[S_E]{*} s{\downarrow}_{S_E} \xrightarrow[R_E]{p} t$$

- Marché's normalized rewriting normalizes with respect to $S_E$ and rewrites with $R_E$, where $E$ is C or AC.
- Higher-order rewriting [Nipkow] needs normalizing terms (with respect to beta, eta modulo alpha) before rewriting modulo alpha, beta and eta.

# Normal rewriting

# Abstract normal rewriting with

- a set of rules $R$,
- a set of rules $S$ and a set of equations $E$ such that $S$ is Church-Rosser modulo $E$.

Assuming $s = s{\downarrow_{S_E}}$ then $s \xrightarrow[R\downarrow_{S_E}]{p} t$ iff

$$s \xrightarrow[R_{SE}]{p} u \xrightarrow[S_E]{!} u{\downarrow_{S_E}} = t$$

For Nipkow's higher-order rewriting, $E$ is alpha, $S$ is made of beta and eta, and $R$ is made of rules $l \to r$ such that $l$ and $r$ have the same base type and $l$ is a pattern [Miller].

# Abstract normal rewriting with

- a set of rules $R$,
- a set of rules $S$ and a set of equations $E$ such that $S$ is Church-Rosser modulo $E$.

$$\text{Assuming } s = s{\downarrow_{S_E}} \text{ then } s \xrightarrow[R\downarrow_{S_E}]{p} t \quad \text{iff}$$

$$s \xrightarrow[R_{SE}]{p} u \xrightarrow[S_E]{!} u{\downarrow_{S_E}} = t$$

For Nipkow's higher-order rewriting, $E$ is alpha, $S$ is made of beta and eta, and $R$ is made of rules $l \to r$ such that $l$ and $r$ have the same base type and $l$ is a pattern [Miller].

# Abstract normal rewriting with

- a set of rules $R$,
- a set of rules $S$ and a set of equations $E$ such that $S$ is Church-Rosser modulo $E$.

$$\text{Assuming } s = s{\downarrow_{S_E}} \ \text{ then } s \xrightarrow[R\downarrow_{S_E}]{p} t \quad \text{iff}$$

$$s \xrightarrow[R_{SE}]{p} u \xrightarrow[S_E]{!} u{\downarrow_{S_E}} = t$$

For Nipkow's higher-order rewriting, $E$ is alpha, $S$ is made of beta and eta, and $R$ is made of rules $l \rightarrow r$ such that $l$ and $r$ have the same base type and $l$ is a pattern [Miller].

# Examples

- Commutative groups:

$$R = \{x + x^{-1} \to 0\}$$
$$S = \{x + 0 \to x\}$$
$$E = \{(x + y) + z = x + (y + z),\ x + y = y + x\}$$

- Differentiation:

$$R = \{diff(\lambda x.sin(f(x)), y) \to cos(f(y)) * diff(f, y)$$
$$diff(\lambda x.x, y) \to 1\}$$
$$S = \{u \to \lambda x.@(u, x) \mid x \notin \mathcal{V}ar(u),$$
$$@(\lambda x.u, v) \to u\{x \mapsto v\}\}$$
$$E = \{\lambda x.u = \lambda y.u\{x \mapsto y\} \mid y \notin \mathcal{V}ar(\lambda x.u)\}$$

# Examples

- Differentiation 2:

$$R = \{diff(\lambda x.sin(f(x)), y) \to cos(f(y)) * diff(f, y)$$
$$diff(\lambda x.x, y) \to 1\}$$
$$S = \{\lambda x.@(u, x) \to u \mid x \notin \mathcal{V}ar(u),$$
$$@(\lambda x.u, v) \to u\{x \mapsto v\}\}$$
$$E = \{\lambda x.u = \lambda y.u\{x \mapsto y\} \mid y \notin \mathcal{V}ar(\lambda x.u)\}$$

- Differentiation 3:

$$R = \{diff(sin \circ f) \to cos * diff(f)$$
$$diff(\lambda x.x) \to \lambda x.1\}$$
$$S = \{\lambda x.@(u, x) \to u \mid x \notin \mathcal{V}ar(u),$$
$$@(\lambda x.u, v) \to u\{x \mapsto v\}\}$$
$$E = \{\lambda x.u = \lambda y.u\{x \mapsto y\} \mid y \notin \mathcal{V}ar(\lambda x.u)\}$$

# Abstract normal rewriting

# Abstract normal rewriting

## Definition

$$s = s{\downarrow_{S_E}} \xrightarrow[R_{SE}]{p} u \xrightarrow[S_E]{!} u{\downarrow_{S_E}} = t$$

## General Assumptions

- (a) $S$ is a Church-Rosser set of rules mod $E$
- (b) $R_{SE} \cup SE$ is terminating,
- (c) Rules in $R$ are $S_E$-normalized,
- (d) Equations in $E$ are regular.
- (e) $S_1 \cup S_2 = S$ is a *splitting* of $S$, that is $t \longrightarrow^*_{S_1} t{\downarrow_{S_1}} \longrightarrow^*_{S_2} t{\downarrow_S}$

From now on, $E$ is alpha-conversion.

**Proof:** $t_1 \xleftrightarrow[R \cup S \cup E]{*} t_2$

**Joinability:** $t_1 \xrightarrow[S_E]{!} \xrightarrow[R\downarrow_{S_E}]{*} u \xleftrightarrow[E]{*} v \xleftarrow[R\downarrow_{S_E}]{*} \xleftarrow[S_E]{!} t_2$

**Church-Rosser:** every proof is joinable.

**Local confluence:** every local divergence

$t_1 \longleftarrow_{R_{SE}} s \longrightarrow_{R_{SE}} t_2$ is joinable

**Local coherence:** every local semi-divergence

$t_1 \longrightarrow_{R_{SE}} \longleftrightarrow_E t_2$ is joinable

# Properties of normal rewriting modulo given: $R, S, E$

Proof: $t_1 \xleftrightarrow[R\cup S\cup E]{*} t_2$

Joinability: $t_1 \xrightarrow[S_E]{!} \; \xrightarrow[R\downarrow_{S_E}]{*} u \xleftrightarrow[E]{*} v \xleftarrow[R\downarrow_{S_E}]{*} \; \xleftarrow[S_E]{!} t_2$

Church-Rosser: every proof is joinable.

Local confluence: every local divergence
$t_1 \longleftarrow_{R_{SE}} s \longrightarrow_{R_{SE}} t_2$ is joinable

Local coherence: every local semi-divergence
$t_1 \longrightarrow_{R_{SE}} \longleftrightarrow_E t_2$ is joinable

# Abstract results

Let $(R, S, E)$ satisfying (a,b,c,d).

**Theorem**

*Assuming local confluence and local coherence, normal rewriting is Church-Rosser.*

reduces to the joinability (at the root) of S-extensions:

Given $g \rightarrow d \in S$, $p \in \mathcal{FP}os(g) \setminus \{\wedge\}$ and $l \rightarrow r \in R$ such that $l$ and $g|_p$ $ES$-unify:

$$(g[l]_p)\downarrow_{S_E} \rightarrow (g[r]_p)\downarrow_{S_E}$$

Nipkow's rewriting:
No $\eta$-extension because the lefthand side of eta is a variable
No $\beta$-extension because rules are of basic type

reduces to the joinability (at the root) of S-extensions:

Given $g \rightarrow d \in S$, $p \in \mathcal{FP}os(g) \setminus \{\wedge\}$ and $l \rightarrow r \in R$ such that $l$ and $g|_p$ $ES$-unify:

$$(g[l]_p)\downarrow_{S_E} \rightarrow (g[r]_p)\downarrow_{S_E}$$

Nipkow's rewriting:

No $\eta$-extension because the lefthand side of eta is a variable

No $\beta$-extension because rules are of basic type

Nipkow's counter example:

$$R = \{\lambda x.a \rightarrow \lambda x.b\}$$

$$a \underset{\beta}{\overset{\Lambda}{\longleftrightarrow}} (\lambda x.a\ u) \underset{R}{\overset{1}{\longleftrightarrow}} (\lambda x.b\ u) \underset{\beta}{\overset{\Lambda}{\longleftrightarrow}} b$$

The Church-Rosser property is lost!

Explanation: a beta-extension is needed obtained by unifying $\lambda x.a$ with the lhs of beta:

$$@(\lambda x.a, u) \rightarrow @(\lambda x.b, u)$$

and by $\beta$-normalization, we get

$$a \rightarrow b$$

## Nipkow's counter example:

$$R = \{\lambda x.a \rightarrow \lambda x.b\}$$

$$a \overset{\Lambda}{\underset{\beta}{\longleftrightarrow}} (\lambda x.a\ u) \overset{1}{\underset{R}{\longleftrightarrow}} (\lambda x.b\ u) \overset{\Lambda}{\underset{\beta}{\longleftrightarrow}} b$$

The Church-Rosser property is lost!

Explanation: a beta-extension is needed obtained by unifying $\lambda x.a$ with the lhs of beta:

$$@(\lambda x.a, u) \rightarrow @(\lambda x.b, u)$$

and by $\beta$-normalization, we get

$$a \rightarrow b$$

## Nipkow's counter example:

$$R = \{\lambda x.a \rightarrow \lambda x.b\}$$

$$a \overset{\Lambda}{\underset{\beta}{\longleftrightarrow}} (\lambda x.a\ u) \overset{1}{\underset{R}{\longleftrightarrow}} (\lambda x.b\ u) \overset{\Lambda}{\underset{\beta}{\longleftrightarrow}} b$$

The Church-Rosser property is lost!

Explanation: a beta-extension is needed obtained by unifying $\lambda x.a$ with the lhs of beta:

$$@(\lambda x.a, u) \rightarrow @(\lambda x.b, u)$$

and by $\beta$-normalization, we get

$$a \rightarrow b$$

# Local-confluence

This is where splittings come in play. Reducing local-confluence to critical pairs requires three different ingredients:

- Forward extensions;
- Shallow pairs;
- Cririral pairs.

Explain the proof sketch on white board.

# Forward extensions

**Definition**

Given a rule $g \to d \in S_2$, a rule $l \to r \in R$, and a position $p \in \mathcal{FD}om(d) \setminus \{\wedge\}$ such that $l$ $S$-unifies with $d|_p$, then the rule $d[l]_p \to d[r]_p$ is a *forward extension* of $R$ with $S_2$.

Rules of the form $g \to x$ or $g \to f(\overline{x})$ have none.
Forward extensions satisfy their purpose: if $\sigma$ is unifies the equation $l = d|_p$, then

$$g\sigma \xrightarrow[R_S]{\wedge} d\sigma[r\sigma]_p$$

# Forward extensions

## Definition

Given a rule $g \rightarrow d \in S_2$, a rule $l \rightarrow r \in R$, and a position $p \in \mathcal{FD}om(d) \setminus \{\wedge\}$ such that $l$ $S$-unifies with $d|_p$, then the rule $d[l]_p \rightarrow d[r]_p$ is a *forward extension* of $R$ with $S_2$.

Rules of the form $g \rightarrow x$ or $g \rightarrow f(\overline{x})$ have none.

Forward extensions satisfy their purpose: if $\sigma$ is unifies the equation $l = d|_p$, then

$$g\sigma \xrightarrow[R_S]{\wedge} d\sigma[r\sigma]_p$$

**Definition**

Given a rule $g \to d \in S_2$, a rule $l \to r \in R$, and a position $p \in \mathcal{FD}om(d) \setminus \{\wedge\}$ such that $l$ $S$-unifies with $d|_p$, then the rule $d[l]_p \to d[r]_p$ is a *forward extension* of $R$ with $S_2$.

Rules of the form $g \to x$ or $g \to f(\overline{x})$ have none. Forward extensions satisfy their purpose: if $\sigma$ is unifies the equation $l = d|_p$, then

$$g\sigma \xrightarrow[R_S]{\Lambda} d\sigma[r\sigma]_p$$

**Definition**

$l \rightarrow r \in R, \ p \in \mathcal{FP}os(l)$ and $g \rightarrow d \in S_1, \ g \notin \mathcal{X}$
$l|_p = g$ has a most general plain unifier $\sigma$
then $(r\sigma, l\sigma[d\sigma]_p) \in SCP(S_1, R)$ is a *shallow critical pair* of $g \rightarrow d$ onto $l \rightarrow r$ at position $p$.
A shallow pair $(a, b)$ is *strongly joinable* if $b \longrightarrow_S^* \longrightarrow_R^\Lambda c$ and the pair $(a, c)$ is joinable.
A pair $(r\sigma, l\sigma[d\sigma]_p) \in CP_S(R)$ is *reducible* if $l\sigma$ is $S$-reducible.

**Theorem**

*Assume that*
*(i) $S_1$-irreducible pairs in $CP_S(R)$ are joinable,*
*(ii) Normal $S$-extensions are joinable*
*(iii) $S_1$-irreducible pairs in $SCP(S_1, R)$ are strongly joinable,*
*(iv) Forward extensions with $S_2$ are joinable,*
*then local confluence holds.*

**Theorem**

*Let $R, S, E$ satisfying properties (a), (b), (c), (d) and $(S_1, S_2)$ be a splitting of $S$. Assuming that*
*(i) normalized extensions are joinable,*
*(ii) forward pairs with $S_2$ are joinable,*
*(iii) $S_1$-irreducible pairs in $CP_S(R)$ are joinable,*
*(iv) $S_1$-irreducible shallow pairs in $SCP(R, S)$ are strongly joinable,*
*then normal rewriting is Church-Rosser.*

# Conclusion

# Summary

A general clean framework for normal rewriting which applies to

- First-order rewriting (commutative groups)
- Plain higher-order rewriting (such as in Coq)
- Nipkows rewriting
- Variations of Nipkow's rewriting:
  – orienting eta as a reduction (in $S_2$) or expansion (in $S_1$)
  – allowing for rules of arrow type (needs $\beta$-extensions)
  – allowing for associativity and commutativity