Confluence of Equational Generalized Term Rewriting Systems with Application to Rewrite Theories and Maude

– Tutorial –

Salvador Lucas

DSIC & VRAIN, Universitat Politècnica de València, Spain

16th International Workshop on
Rewriting Logic and its Applications

WRLA 2026

```
fmod ListsP is
  sorts Nat ListN .
  subsorts Nat < ListN .

  op _++_ : ListN ListN -> ListN [strat (0) assoc] .
  op 0 : -> Nat .        op s : Nat -> Nat .
  ops _+_ _*_ : Nat Nat -> Nat [strat (1 0)] .
  op prod : ListN -> Nat .

  vars m n : Nat .     vars ms ns : ListN .

  eq 0 + n = n .
  eq s(m) + n = s(m + n) .
  eq 0 * n = 0 .
  eq s(m) * n = (m * n) + n .
  eq prod(m) = m .
  eq prod(n ++ ns) = n * prod(ns) .
endfm
```

Equational Generalized Term Rewriting System $\mathcal{R}$ (EGTRS) [Luc26]:

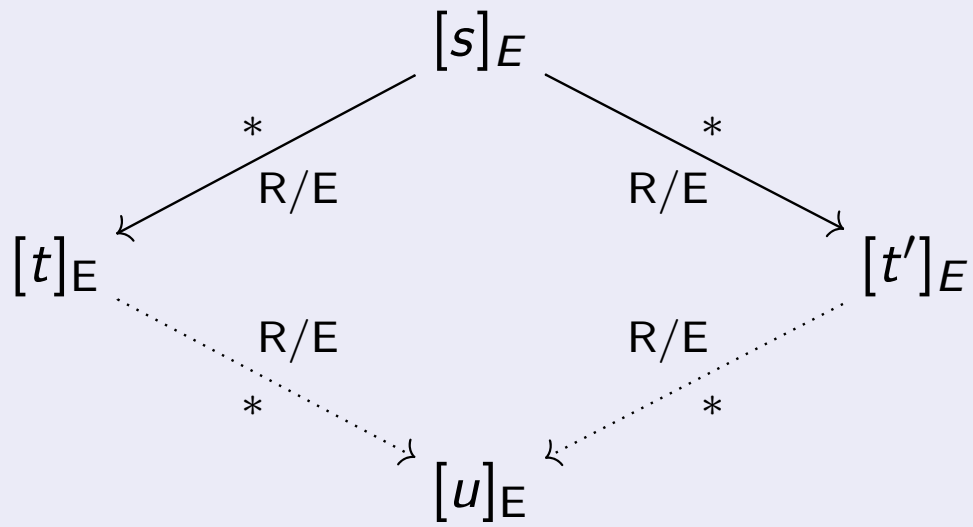
- A signature of *functions* ($\mathcal{F}$)
- A signature of *predicates* (Π)
- A *replacement map* (μ)
- Conditional *equations* (E)
- *Horn clauses* (H)
- Conditional *rewrite rules* (R)

$$xs +\!\!+ (ys +\!\!+ zs) = (xs +\!\!+ ys) +\!\!+ zs$$
$$\mathrm{ListN}(n) \Leftarrow \mathrm{Nat}(n)$$
$$\mathrm{ListN}(xs +\!\!+ ys) \Leftarrow \mathrm{ListN}(xs), \mathrm{ListN}(ys)$$
$$\mathrm{Nat}(0)$$
$$\mathrm{Nat}(\mathrm{s}(n)) \Leftarrow \mathrm{Nat}(n)$$
$$\mathrm{Nat}(m + n) \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n)$$
$$\mathrm{Nat}(m * n) \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n)$$
$$\mathrm{Nat}(\mathrm{prod}(xs)) \Leftarrow \mathrm{ListN}(xs)$$

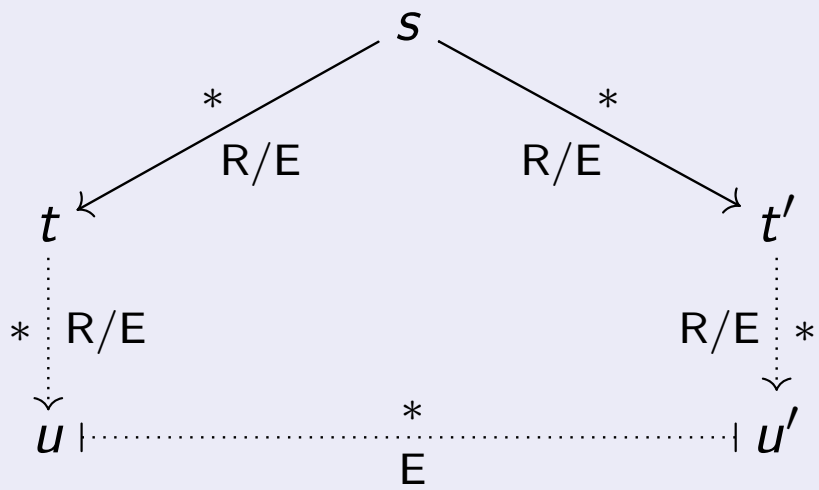
$$0 + n \to n \Leftarrow \mathrm{Nat}(n)$$
$$\mathrm{s}(m) + n \to \mathrm{s}(m + n) \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n)$$
$$0 * n \to 0 \Leftarrow \mathrm{Nat}(n)$$
$$\mathrm{s}(m) * n \to \mathrm{s}(m * n) + n \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n)$$
$$\mathrm{prod}(n) \to n \Leftarrow \mathrm{Nat}(n)$$
$$\mathrm{prod}(n +\!\!+ ns) \to n * \mathrm{prod}(ns) \Leftarrow \mathrm{Nat}(n)$$

$$\mathrm{prod}(\overbrace{0 +\!\!+ \mathrm{s}(0) +\!\!+ \mathrm{s}(\mathrm{s}(0))}^{\underbrace{[0 +\!\!+ \mathrm{s}(0)] +\!\!+ \mathrm{s}(\mathrm{s}(0))}_{0 +\!\!+ [\mathrm{s}(0) +\!\!+ \mathrm{s}(\mathrm{s}(0))]}}) \to_{\mathcal{R}/E} \underline{0 * \mathrm{prod}(\mathrm{s}(0) +\!\!+ \mathrm{s}(\mathrm{s}(0)))} \to_{\mathcal{R}} 0$$

E-confluence (CR(R/E)) is confluence of rewriting on *equivalence classes*



Equivalent to the commutation of the following diagram on *terms*



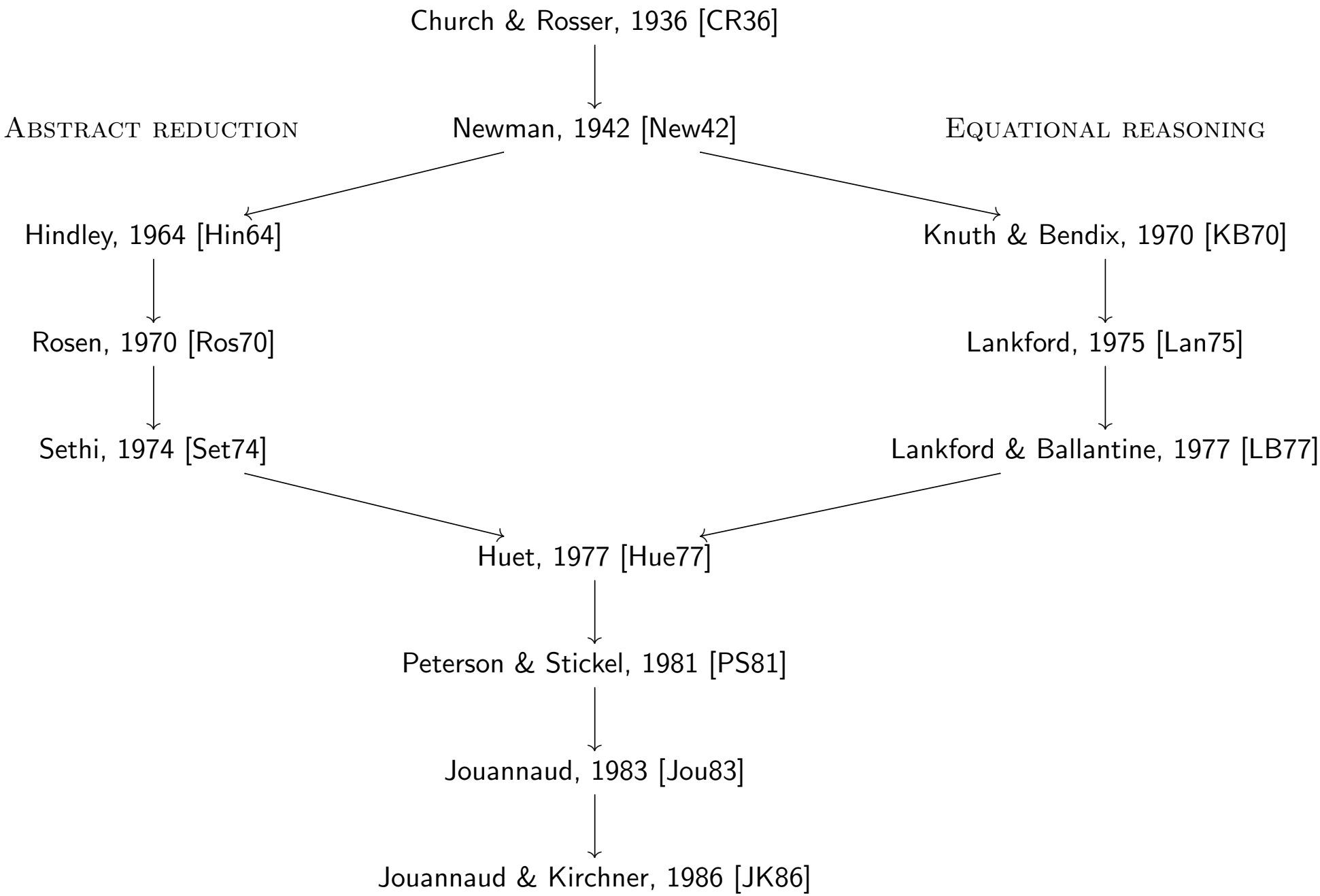
- Lambda calculus (β-*reduction* + α-*conversion*) [CR36, Hin64, Hin69]
- Code optimization [ASU72, Set74]
- Rewriting-based *equational reasoning* [KB70, LB77, Hue80, PS81, Jou83]
- Theorem proving *modulo equations* [RW69, Sla74, RW83, Dow99, DHK03]
- Programming languages implementing *rewriting on equivalence classes* (e.g., Maude) [Mes92, Mes12, CDE$^+$07]
- Extensions to Logically Constrained Term Rewriting Systems [ANS24], Nominal Rewriting [FNSS25], etc.
- ...

Part I

Confluence Modulo of Equational Generalized Term Rewriting Systems

1. Jouannaud and Kirchner's abstract framework
2. Equational Generalized Term Rewriting Systems
3. Application of Jouannaud and Kirchner's framework to EGTRSs
4. Critical and variable peaks and their conditional pairs
5. Proving and disproving confluence modulo of EGTRSs ($\widetilde{\downarrow}_{\mathcal{R}}$)
6. E-critical and E-variable peaks and their conditional pairs
7. Proving and disproving confluence modulo of EGTRSs ($\widetilde{\downarrow}_{\mathcal{R},E}$)

Jouannaud and Kirchner's abstract framework


Church & Rosser, 1936 [CR36]
Abstract reduction
Newman, 1942 [New42]
Equational reasoning
Hindley, 1964 [Hin64]
Knuth & Bendix, 1970 [KB70]
Rosen, 1970 [Ros70]
Lankford, 1975 [Lan75]
Sethi, 1974 [Set74]
Lankford & Ballantine, 1977 [LB77]
Huet, 1977 [Hue77]
Peterson & Stickel, 1981 [PS81]
Jouannaud, 1983 [Jou83]
Jouannaud & Kirchner, 1986 [JK86]

Jouannaud [Jou83] and then Jouannaud and Kirchner [JK86] generalized Huet's abstract framework [Hue80]

Definitions [JK86]:

Let $\vdash_E$ be a *symmetric* relation on A

Let R and R^E be reduction relations on A

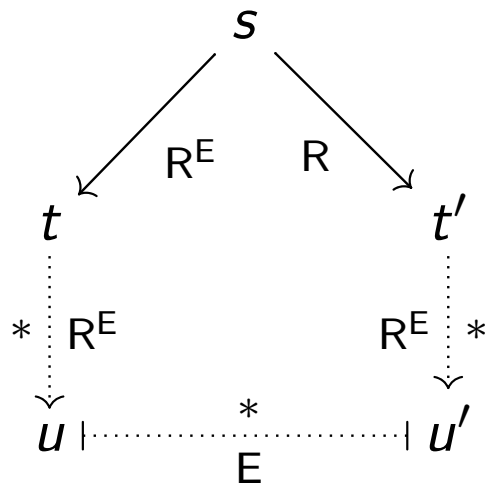
Assumptions:

J&K1: $\sim_E$ is $\vdash_E^*$

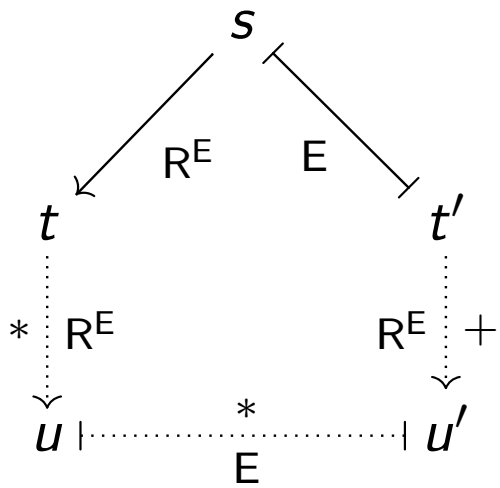
J&K2: $\rightarrow_{R/E} = \sim_E \circ \rightarrow_R \circ \sim_E$

Fundamental assumption J&K3

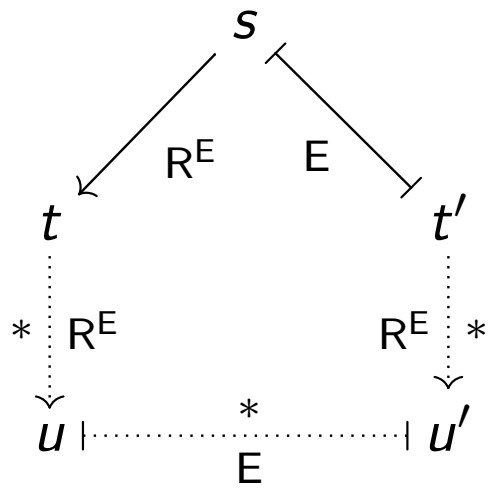
$$\rightarrow_R \subseteq \rightarrow_{R^E} \subseteq \rightarrow_{R/E}$$



Local Confluence
of R^{E} modulo E with R
$\mathsf{LConf}_{\mathsf{E}}(\mathsf{R}^{\mathsf{E}}, \mathsf{R})$



Local Coherence
of R^{E} modulo E
$\mathsf{LCoh}_{\mathsf{E}}(\mathsf{R}^{\mathsf{E}})$



Local Coherence
of R^{E} modulo E
$\mathsf{LCoh}_{\mathsf{E}}(\mathsf{R}^{\mathsf{E}})$
(assume SN(R/E))

Main result cf. [JK86, Theorem 5]: An E-terminating relation R is E-*confluent* if

$$\mathsf{LConf}_{\mathsf{E}}(\mathsf{R}^{\mathsf{E}}, \mathsf{R}) \quad \text{and} \quad \mathsf{LCoh}_{\mathsf{E}}(\mathsf{R}^{\mathsf{E}})$$

hold

Note: it is just a *sufficient condition* for E-confluence!

Let $s \to_{\mathcal{R},E} t$ be *Peterson & Stickel's reduction*, where $s|_p =_E \sigma(\ell)$ for some position p, rule $\ell \to r$, and substitution σ such that $t = s[\sigma(r)]_p$.

The following ETRS [Luc26, Example 4.14]:

$$\begin{array}{lcll} a & = & f(b) & (1) \\ b & \to & d & (2) \end{array} \qquad \begin{array}{lcll} c & \to & a & (3) \\ c & \to & f(d) & (4) \end{array}$$

is E-confluent, but *neither* $\mathrm{LConf}_E(\to_{\mathcal{R},E}, \to_{\mathcal{R}})$ *nor* $\mathrm{LCoh}_E(\to_{\mathcal{R},E})$ hold

$\mathrm{LConf}_E(\to_{\mathcal{R},E}, \to_{\mathcal{R}})$ fails:

$a \;{}_{(3)}\!\leftarrow \; c \;\to_{(4)} f(d)$

is *not* $\widetilde{\downarrow}_{\mathcal{R},E}$-joinable

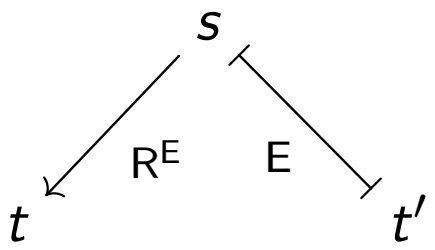
$\mathrm{LCoh}_E(\to_{\mathcal{R},E})$ fails:

$f(d) \;{}_{(2)}\!\leftarrow \; f(b) \;\to_{\blacktriangleleft(1)} a$

is *not* $\widetilde{\downarrow}_{\mathcal{R},E}$-joinable,

For *disproving E-confluence*, use non-$\widetilde{\downarrow}_{R/E}$-joinability.

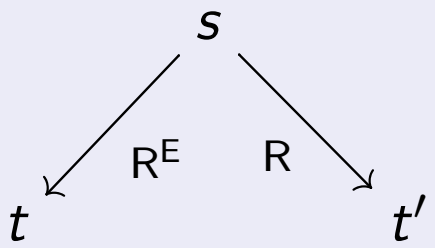
Local E-coherence peaks



are always $\widetilde{\downarrow}_{R/E}$-joinable (as $t' \rightarrow_{R/E} t$)!

Non-E-confluence criterion

If there is a *non-$\widetilde{\downarrow}_{R/E}$-joinable* local E-confluence peak

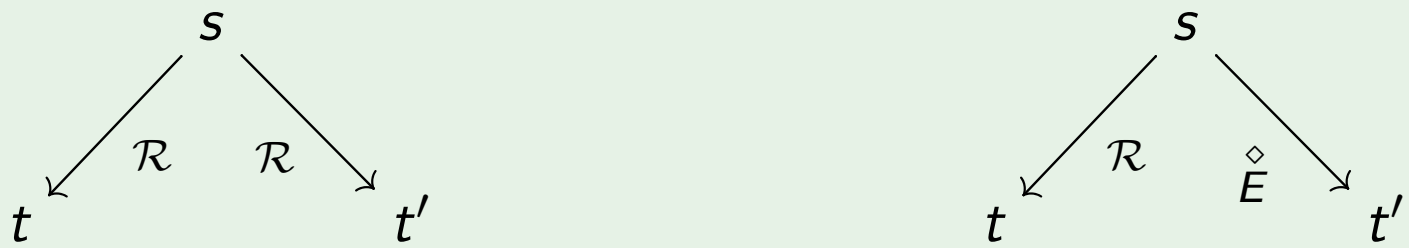


then R is *not E-confluent*.

Let $\mathcal{R} = (\mathcal{F}, E, R)$ be an *Equational Term Rewriting System* (ETRS)

Huet [Hue80]

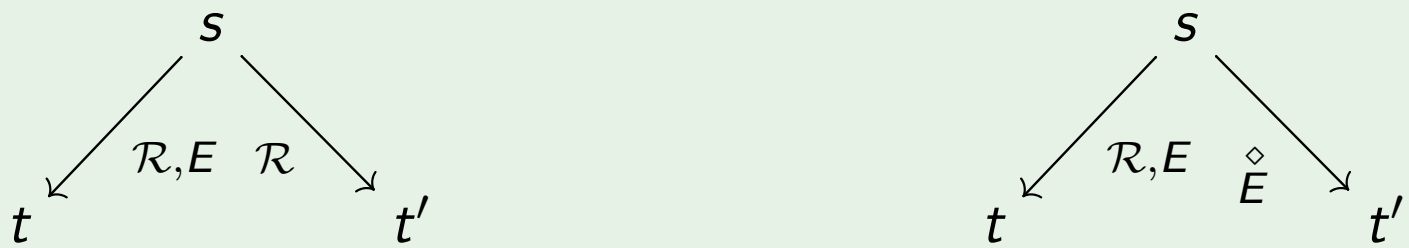
Local confluence peak and **Local coherence peaks**



Critical pairs are used to investigate E-confluence

Jouannaud [Jou83], Jouannaud and Kirchner [JK86]

Local E-confluence peak and **Local E-coherence peaks**

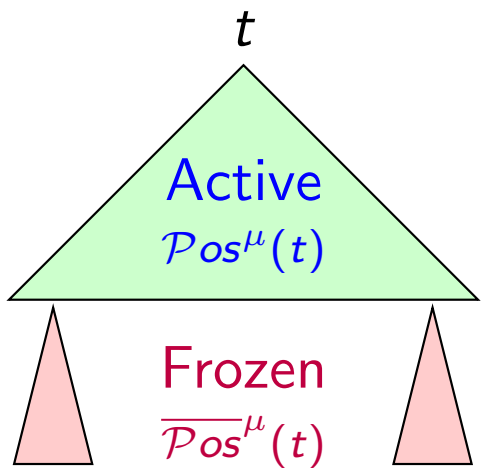


E-critical pairs are used to investigate E-confluence

Equational Generalized Term Rewriting Systems

An *EGTRS* is a tuple $\mathcal{R} = (\mathcal{F}, \Pi, \mu, E, H, R)$ where

- $\mathcal{F}$ is a signature of function symbols,
- Π is a signature of predicate symbols, including $=$, $\to$, and $\to^*$
- μ is a *replacement map* for $\mathcal{F}$, decomposing positions as follows:



Besides,

- E is a set of conditional equations $s = t \Leftarrow c$, for terms s and t;
- H is a set of definite Horn clauses $A \Leftarrow c$ where $A = P(t_1, \ldots, t_n)$ for some terms $t_1, \ldots, t_n$, $n \geq 0$, is such that $P \notin \{=, \to, \to^*\}$; and
- R is a set of conditional rules $\ell \to r \Leftarrow c$ for terms $\ell \notin \mathcal{X}$ and r.

where c is a sequence of atoms.

Computational relations are described by Horn theories Th and defined by

Logical consequence:

$$\mathsf{Th}_E \models s = t$$
$$\mathsf{Th}_{\mathcal{R}} \models s \to t$$
$$\mathsf{Th}_{\mathcal{R},E} \models s \xrightarrow{ps} t$$
$$\mathsf{Th}_{\mathcal{R}/E} \models s \xrightarrow{rm} t$$

Provability in Elementary Inference Systems:

$$\vdash_{\mathcal{I}(\mathsf{Th}_E)} s = t$$
$$\vdash_{\mathcal{I}(\mathsf{Th}_{\mathcal{R}})} s \to t$$
$$\vdash_{\mathcal{I}(\mathsf{Th}_{\mathcal{R},E})} s \xrightarrow{ps} t$$
$$\vdash_{\mathcal{I}(\mathsf{Th}_{\mathcal{R}/E})} s \xrightarrow{rm} t$$

Each sentence

$$(\forall \vec{x})\, A_1 \wedge \cdots \wedge A_n \Rightarrow A$$

in a Horn theory Th is viewed as an elementary inference rule

$$\frac{A_1 \quad \cdots \quad A_n}{A}$$

in $\mathcal{I}(\mathsf{Th})$ and vice versa

For *atoms* A, $\mathsf{Th} \models (\forall \vec{x})A$ iff $\vdash_{\mathcal{I}(\mathsf{Th})} A$ [Luc25, Proposition 14]

Each theory/inference system is obtained from generic sentences/rules:

Label	Purpose	Label	Purpose
$(\mathrm{Rf})^{\bowtie}$	$\bowtie$ is reflexive	$(\mathrm{Tr})^{\bowtie}$	$\bowtie$ is transitive
$(\mathrm{Sy})^{\bowtie}$	$\bowtie$ is symmetric	$(\mathrm{Co})^{\bowtie}$	$\bowtie$ and then $\bowtie^*$ is in $\bowtie^*$
$(\mathrm{Pr})^{\bowtie}_{f,i}$	$\bowtie$ propagated in terms	$(\mathrm{HC})_{A \Leftarrow A_1,\ldots,A_n}$	Clause as sentence/rule

$(\mathrm{R,E})_{\ell \to r \Leftarrow A_1,\ldots,A_n}$ Peterson & Stickel:

sentence: $(\forall x, \vec{x})\ x = \ell \wedge A_1 \wedge \cdots \wedge A_n \Rightarrow x \xrightarrow{ps} r$

inference rule: $\dfrac{x = \ell \quad A_1 \quad \cdots \quad A_n}{x \xrightarrow{ps} r}$

(R/E) Rewriting modulo:

$(\forall x, x', y, y')\ x = x' \wedge x' \to y' \wedge y' = y \Rightarrow x \xrightarrow{rm} y$

$$\frac{x = x' \quad x' \to y' \quad y' = y}{x \xrightarrow{rm} y}$$

Th_E defines $=$ as an equivalence relation, congruent according to μ.

Predicates defined in H which are referred in E are given sentences in Th_E.

$\mathsf{Th}_{\mathcal{R}}$ includes Th_E and defines $\rightarrow$ as rewriting using *pattern matching* against the left-hand sides ℓ of rules $\ell \rightarrow r \Leftarrow c$ in R.

Rewritings are propagated on the structure of terms according to μ.

Predicates referred in c are given sentences in $\mathsf{Th}_{\mathcal{R}}$ using H.

Given $\alpha : A \Leftarrow A_1, \ldots, A_n$, we obtain α^{ps} (resp. α^{rm}) by replacing

1. any occurrence of $\approx$ in A by $\approx_{ps}$ (resp. $\approx_{rm}$) and
2. all occurrences of $\approx$, $\to$ and $\to^*$ in $A_1, \ldots, A_n$ by $\approx_{ps}$, $\xrightarrow{ps}$, and $\xrightarrow{ps}{}^*$ (resp. $\approx_{rm}$, $\xrightarrow{rm}$, and $\xrightarrow{rm}{}^*$).

H^{ps} and R^{ps} (resp. H^{rm} and R^{rm}) are obtained from H and R by replacing each $\alpha \in H \cup R$ by α^{ps} (resp. α^{rm}).

$\mathrm{Th}_{\mathcal{R},E}$ includes Th_E and defines $\xrightarrow{ps}$ as rewriting using *E-pattern matching* against the left-hand sides ℓ of rules $\alpha : \ell \to r \Leftarrow c$ in R^{ps} using $(\mathrm{R,E})_\alpha$.

Rewritings with $\xrightarrow{ps}$ are propagated according to μ.

$\mathrm{Th}_{\mathcal{R}/E}$ includes Th_E and defines $\to$ by *pattern matching* against the left-hand sides ℓ of rules $\ell \to r \Leftarrow c$ in R^{rm}.

Such rewritings are propagated according to μ.

Then, $\xrightarrow{rm}$ is defined as $= \circ \to \circ =$ with $(\mathrm{R/E})$.

Application of Jouannaud & Kirchner's framework to EGTRSs

Application of J&K'86 to Equational Term Rewriting Systems (ETRSs):

- $\vdash_{\mathsf{E}}$ is $\to_{\overset{\diamond}{E}}$ for $\overset{\diamond}{E}$ consisting of $s \to t$ and $t \to s$ for each $s = t$ in E
- $\sim_{\mathsf{E}}$ is $=_E$, i.e., $\to^*_{\overset{\diamond}{E}}$
- R is $\to_{\mathcal{R}}$
- R^{E} is $s \to_{\mathcal{R},E} t$, i.e., *Peterson & Stickel* reduction [PS81]:
  - $s|_p =_E \sigma(\ell)$ for some rule $\ell \to r$ in $\mathcal{R}$ and substitution σ and
  - $t = s[\sigma(r)]_p$
- R/E is $\to_{\mathcal{R}/E}$, i.e., $=_E \circ \to_{\mathcal{R}} \circ =_E$

Extending J&K'86 abstract approach for ETRSs to EGTRSs?

Abstract reduction:	$\vdash_{\mathsf{E}}$	$\sim_{\mathsf{E}}$	$\to_{\mathsf{R}}$	$\to_{\mathsf{R}^{\mathsf{E}}}$	$\to_{\mathsf{R/E}}$
Application to ETRSs $\mathcal{R}$:	$\to_{\overset{\diamond}{E}}$	$\to^*_{\overset{\diamond}{E}}$ is $=_E$	$\to_{\mathcal{R}}$	$\to_{\mathcal{R},E}$	$\to_{\mathcal{R}/E}$

Problem: (J&K2) fails for EGTRSs: $\to_{\mathcal{R}/E}$ is not $=_E \circ \to_{\mathcal{R}} \circ =_E$

$$a = b \qquad a \to c \qquad a \to d \Leftarrow b \to^* c$$

We have

- $\to_{\mathcal{R}} = \{(a, c)\}$ and
- $(=_E \circ \to_{\mathcal{R}} \circ =_E) = \{(a, c), (b, c)\}$, but
- $\to_{\mathcal{R}/E} = \{(a, c), (b, c), (a, d), (b, d)\}$.

Noticed by Meseguer [Mes17, Section 4.3]. His solution: $R = R^E = \to_{\mathcal{R},E}$

Only ${}_{\mathcal{R},E}\!\leftarrow \circ \to_{\mathcal{R},E}$ and ${}_{\mathcal{R},E}\!\leftarrow \circ \vdash_E$ peaks are considered

E-confluence analysis based on *ECCPs*, computed by *E-unification*

In the *CR-theory* $\overline{\mathcal{R}^{\mathrm{CR}}}$ of an EGTRS,

goals $s \to t$ (resp $s \to^* t$)
in conditions of rules or Horn clauses
are *always* evaluated using $\to_{\mathcal{R}/E}$ (resp. $\to^*_{\mathcal{R}/E}$).

Then,

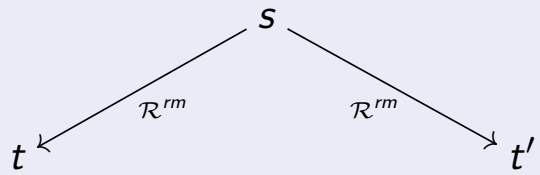
- $\overline{\mathcal{R}^{\mathrm{CR}}} \models s = t$ and $\overline{\mathcal{R}^{\mathrm{CR}}} \models s \overset{rm}{\to} t$ remain as $s =_E t$ and $s \to_{\mathcal{R}/E} t$.
- $\overline{\mathcal{R}^{\mathrm{CR}}} \models s \to t$ is denoted now as $s \to_{\mathcal{R}^{rm}} t$.
- $\overline{\mathcal{R}^{\mathrm{CR}}} \models s \overset{ps}{\to} t$ is denoted now as $s \to_{\mathcal{R}^{rm},E} t$.

Use of Jouannaud & Kirchner's framework with EGTRSs

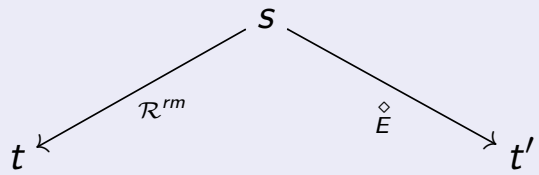
Abstract reduction:	$\vdash_{\mathsf{E}}$	$\sim_{\mathsf{E}}$	$\to_{\mathsf{R}}$	$\to_{\mathsf{R}^{\mathsf{E}}}$	$\to_{\mathsf{R/E}}$
Application to EGTRSs $\mathcal{R}$:	$\to_{\overset{\diamond}{E}}$	$=_E$	$\to_{\mathcal{R}^{rm}}$	$\to_{\mathcal{R}^{rm},E}$	$\to_{\mathcal{R}/E}$

$\vdash_{\mathsf{E}}$ as $\rightarrow_{\overset{\diamond}{E}}$, and both R and R^{E} as $\rightarrow_{\mathcal{R}^{rm}}$

LCON-peak



LCOH-peak

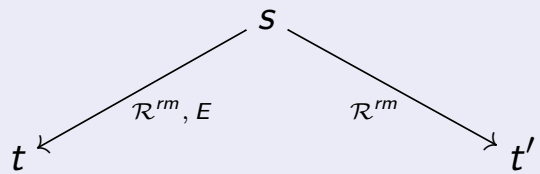


Joinability

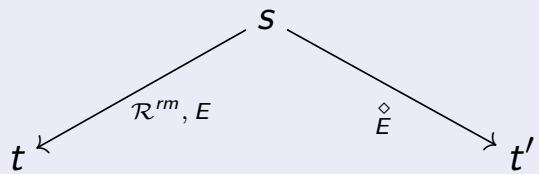
$\widetilde{\downarrow}_{\mathcal{R}^{rm}}$

$\vdash_{\mathsf{E}}$ as $\rightarrow_{\overset{\diamond}{E}}$, R as $\rightarrow_{\mathcal{R}^{rm}}$, and R^{E} as $\rightarrow_{\mathcal{R}^{rm},E}$

E-LCON-peak



E-LCOH-peak

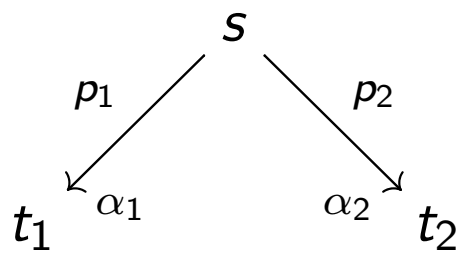


Joinability

$\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$

Critical and variable peaks and their conditional pairs

The peak



1. is a *local confluence peak*, if $\alpha_1, \alpha_2 \in \mathcal{R}^{rm}$
2. is a *local coherence peak*, if either
   1. $\alpha_1 \in \mathcal{R}^{rm}$ and $\alpha_2 \in \overset{\diamond}{E}$ or
   2. $\alpha_1 \in \overset{\diamond}{E}$ and $\alpha_2 \in \mathcal{R}^{rm}$

If $p_1 \parallel p_2$, then t_1 and t_2 are trivially $\widetilde{\downarrow}_{\mathcal{R}}$ *-joinable*

Local confluence peaks

as

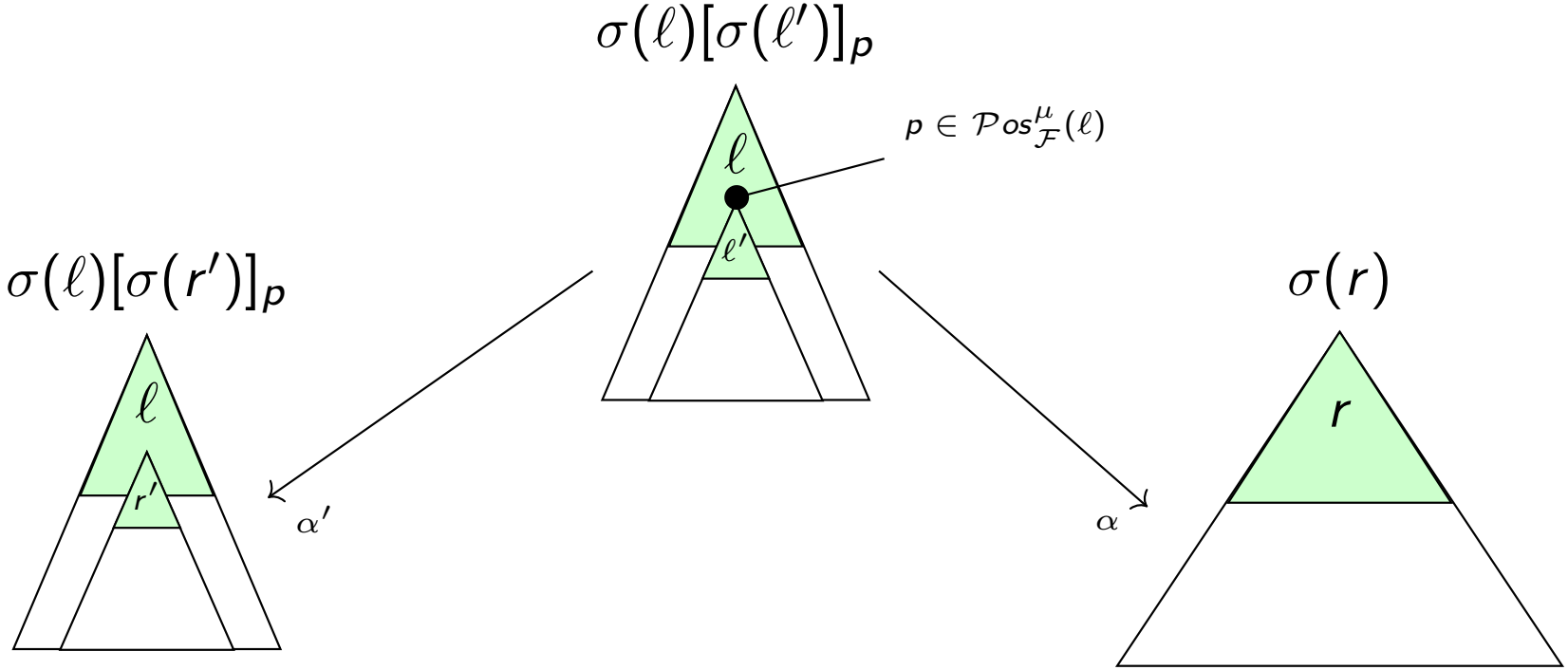
Conditional Critical Pairs

or

Conditional Variable Pairs

LCON critical peaks ($\alpha \in \mathcal{R}^{rm}$, with $\alpha' \in \mathcal{R}^{rm}$ *overlapping* α)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$ and $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$

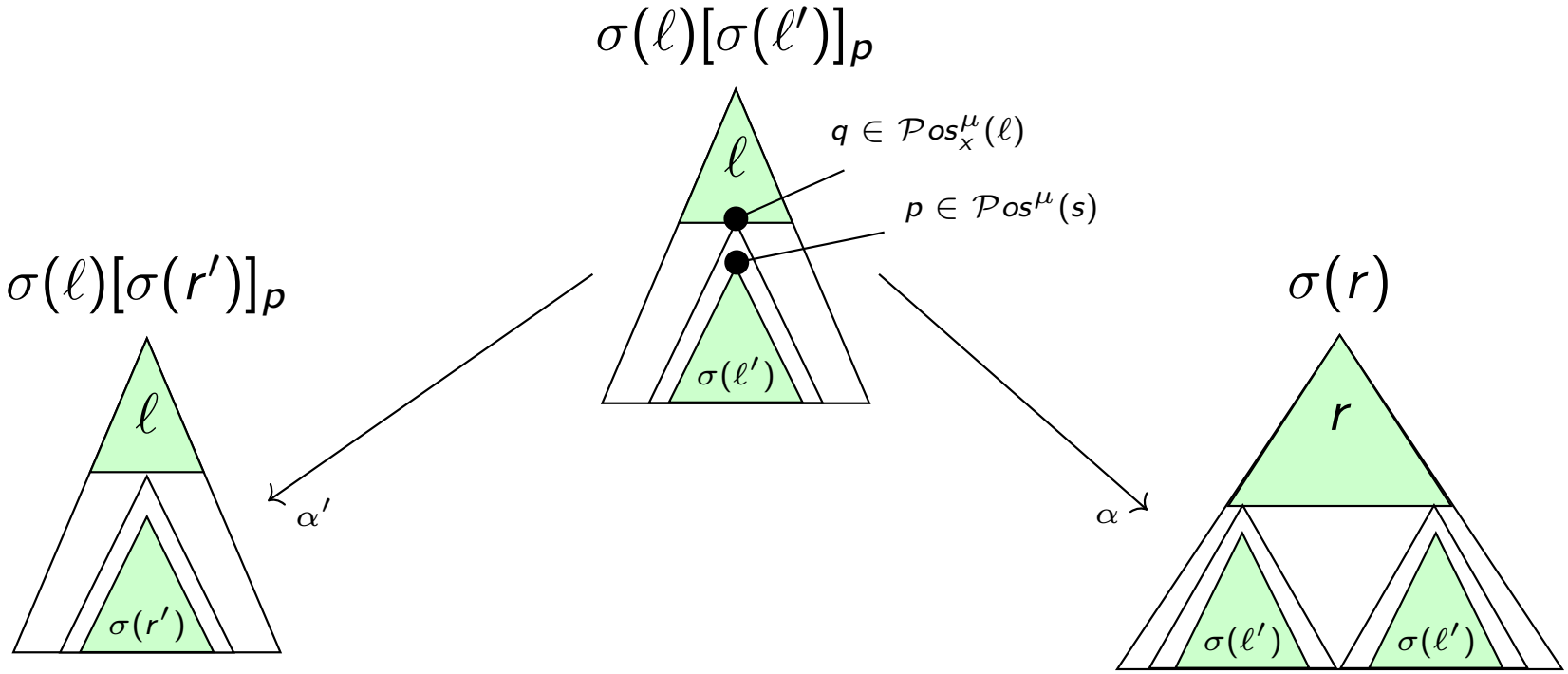


Conditional Critical Pairs CCP($\mathcal{R}$)

$\pi_{\alpha,p,\alpha'} : \langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle \Leftarrow \theta(c), \theta(c')$ where $\ell|_p =^?_\theta \ell'$

LCON variable peaks ($\alpha, \alpha' \in \mathcal{R}^{rm}$, *no overlap*)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$ and $p \notin \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$



Conditional $\to$-Variable Pairs $\mathrm{CVP}^{\to}(\mathcal{R})$

$\pi^{\to}_{\alpha,x,q} : \langle \ell[x']_q, r' \rangle \Leftarrow x \to x', c$ $\quad x \in \mathcal{V}ar^{\mu}(\ell)$, $q \in \mathcal{P}os^{\mu}_{x}(\ell)$, x' fresh

For *Equational TRSs*, these CVPs are $\downarrow_{\mathcal{R}}$-*joinable*, cf. [Hue80]

[Luc26, Proposition 8.3(1)]

Let $\mathcal{R} = (\mathcal{F}, \Pi, \mu, E, H, R)$ be an EGTRS. Then,

$\rightarrow_{\mathcal{R}^{rm}}$ is locally confluent modulo E with $\rightarrow_{\mathcal{R}^{rm}}$

iff all conditional pairs in

$\mathsf{CCP}(\mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(\mathcal{R})$

are $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$-joinable.

Local coherence peaks

as

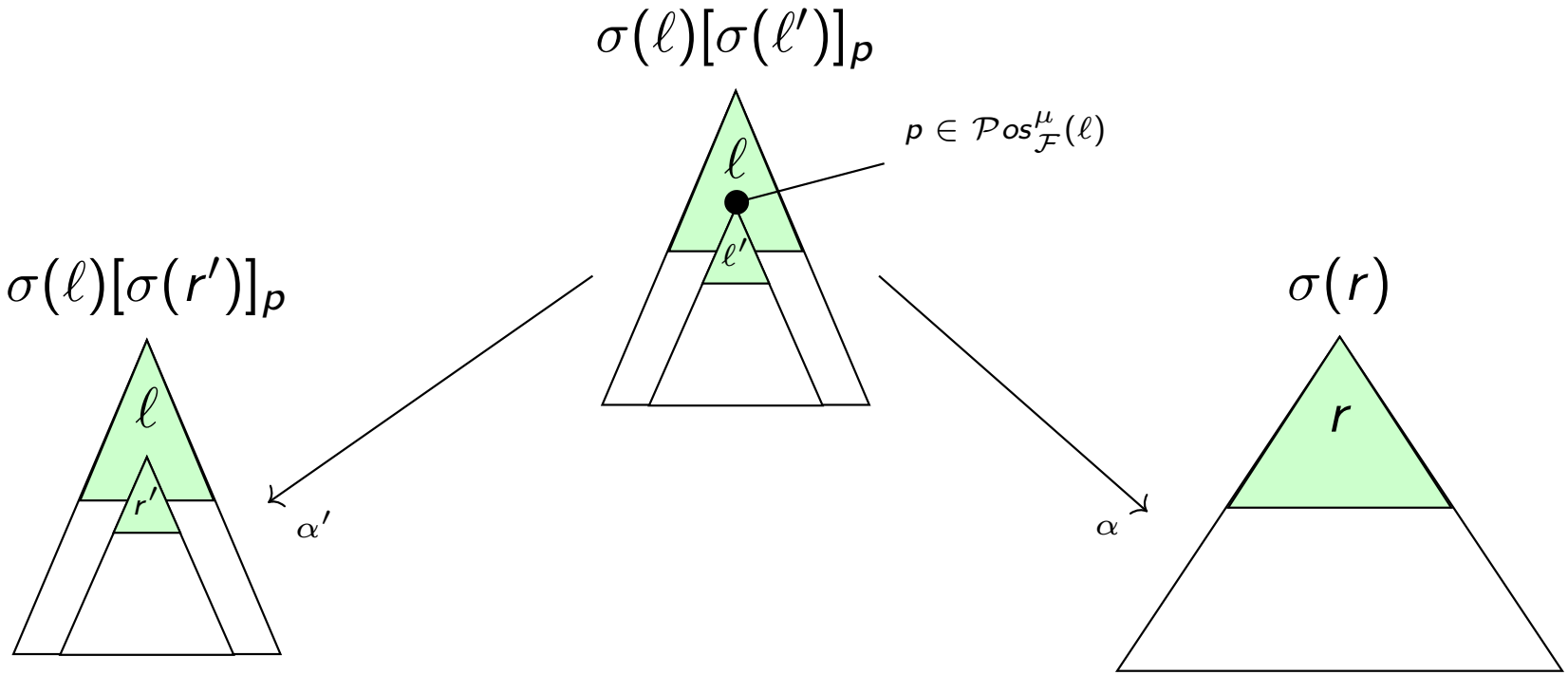
Conditional Critical Pairs

or

Conditional Variable Pairs

LCOH critical peaks ($\alpha \in \mathcal{R}^{rm}$, with $\alpha' \in \overset{\diamond}{E}$ overlapping α)

For $\alpha : \ell \to r \Leftarrow c \in \mathcal{R}^{rm}$, $\alpha' : \ell' \to r' \Leftarrow c' \in \overset{\diamond}{E}$ and $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$

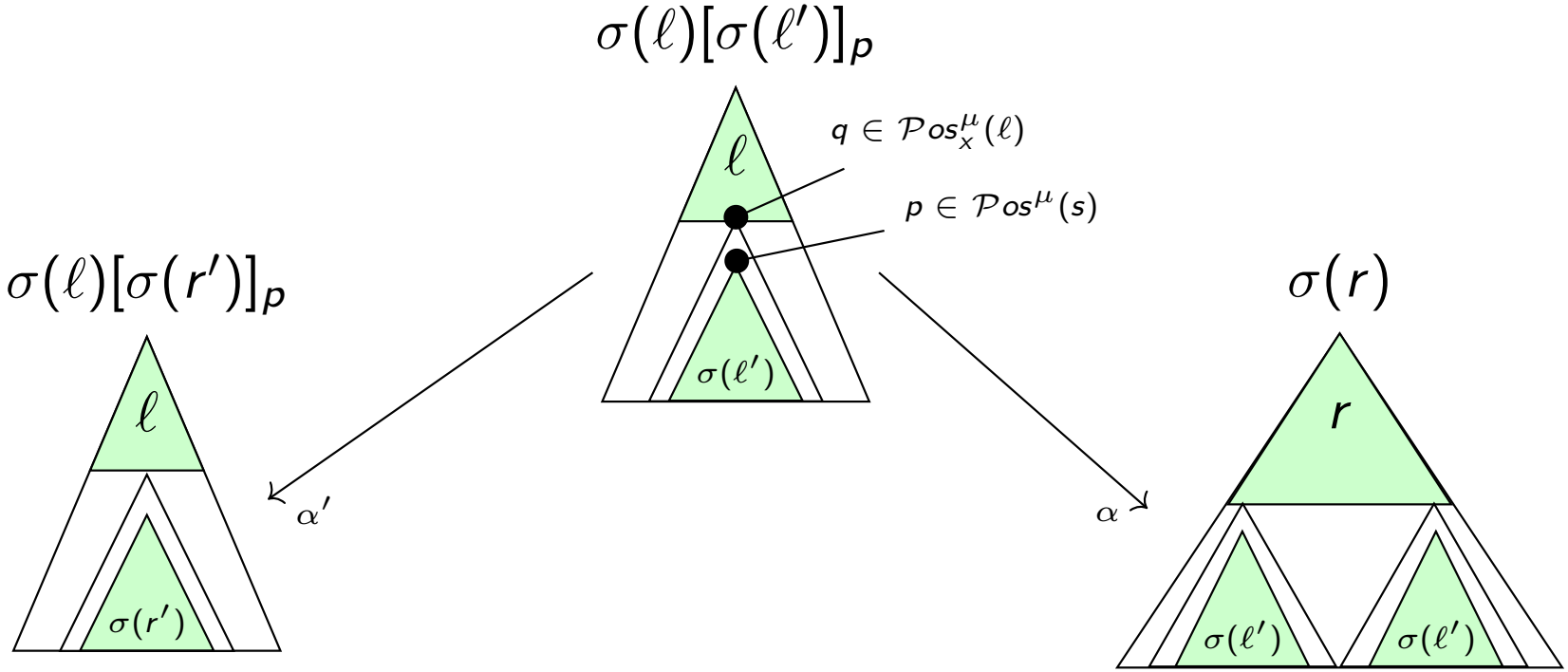


Conditional Critical Pairs $\mathrm{CCP}(\mathcal{R}, E)$

$\pi_{\alpha,p,\alpha'} : \langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle \Leftarrow \theta(c), \theta(c')$ where $\ell|_p =^?_\theta \ell'$

LCOH variable peaks ($\alpha \in \mathcal{R}^{rm}$ and $\alpha' \in \overset{\diamond}{E}$, no overlap)

For $\alpha : \ell \to r \in \mathcal{R}$, $\alpha' : \ell' \to r' \in \overset{\diamond}{E}$ and $p \notin \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$



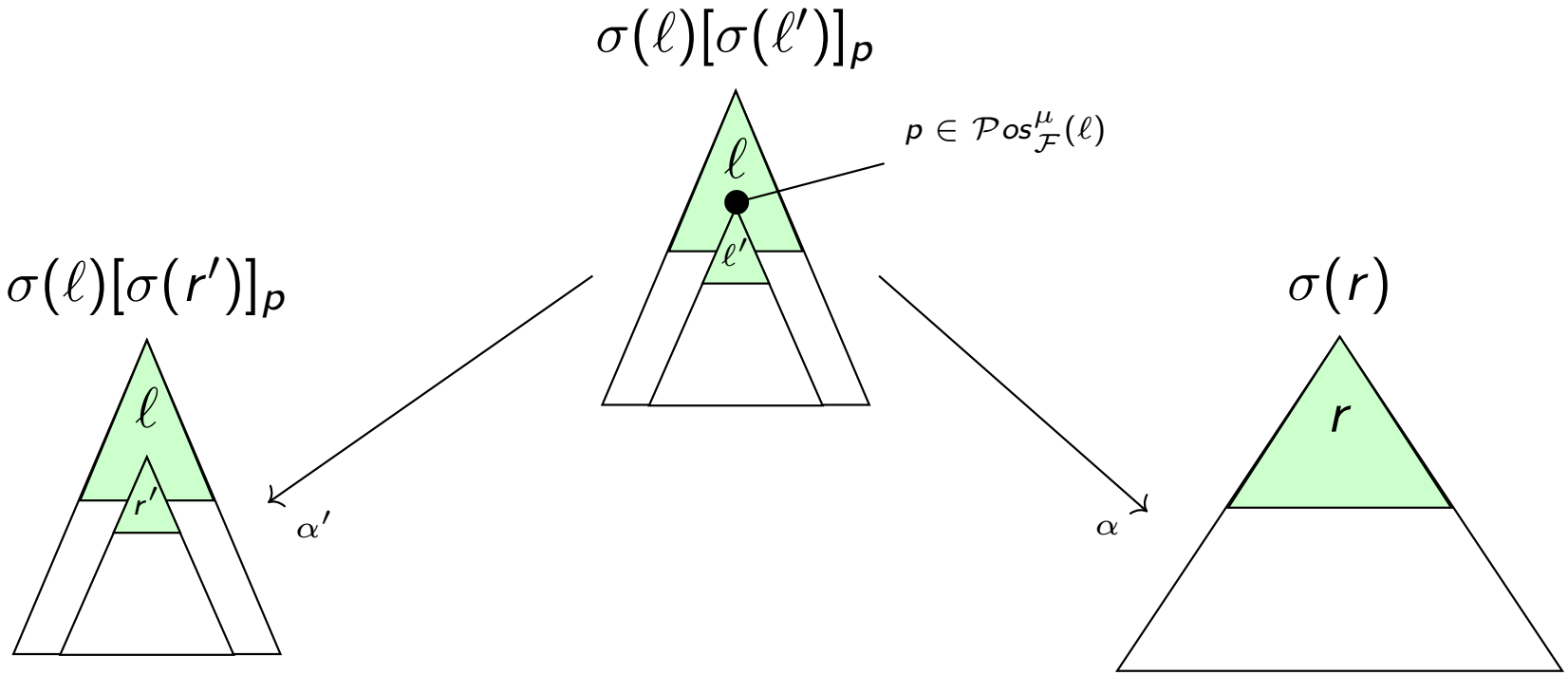
Conditional $\vdash$-Variable Pairs $\mathrm{CVP}^{\vdash}(\mathcal{R})$

$\pi^{\vdash}_{\alpha,x,q} : \langle \ell[x']_q, r' \rangle \Leftarrow x \vdash x', c$ $\quad x \in \mathcal{V}ar^{\mu}(\ell)$, $q \in \mathcal{P}os^{\mu}_{x}(\ell)$, x' fresh

For Equational, *left-linear* TRSs, these CVPs are $\widetilde{\downarrow}_{\mathcal{R}}$ *-joinable*, cf. [Hue80]

LCOH critical peaks ($\alpha \in \overset{\diamond}{E}$, with $\alpha' \in \mathcal{R}^{rm}$ overlapping α)

For $\alpha : \ell \to r \Leftarrow c \in \overset{\diamond}{E}$, $\alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$ and $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$

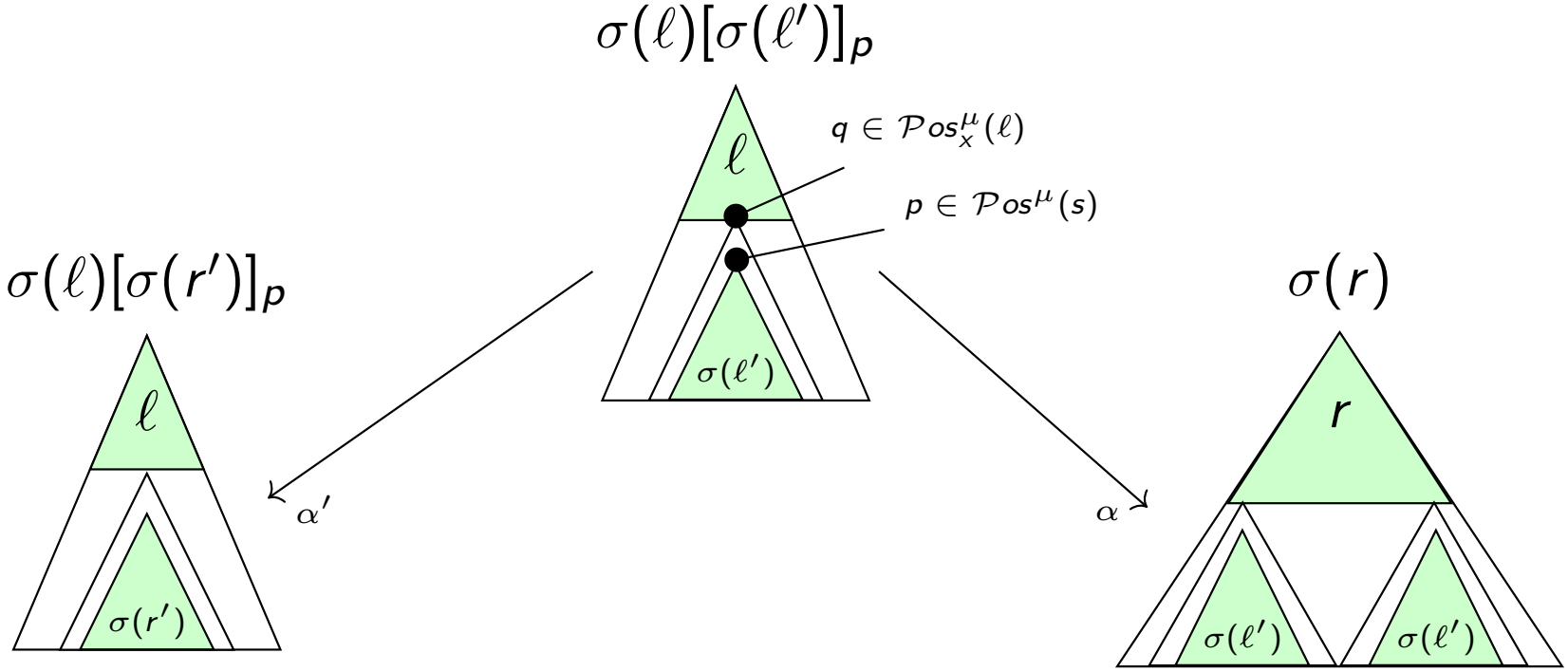


Conditional Critical Pairs $\mathrm{CCP}(E, \mathcal{R})$

$\pi_{\alpha,p,\alpha'} : \langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle \Leftarrow \theta(c), \theta(c')$ where $\ell|_p =^{?}_{\theta} \ell'$

LCOH variable peaks ($\alpha \in \overset{\diamond}{E}$, and $\alpha' \in \mathcal{R}^{rm}$, no overlap)

For $\alpha : \ell \to r \Leftarrow c \in \overset{\diamond}{E}$, $\alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$ and $p \notin \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$



Conditional $\to$-Variable Pairs $\mathrm{CVP}^{\to}(E)$

$\pi^{\to}_{\alpha,x,q} : \langle \ell[x']_q, r' \rangle \Leftarrow x \to x', c$ $\quad x \in \mathcal{V}ar^{\mu}(\ell)$, $q \in \mathcal{P}os^{\mu}_{x}(\ell)$, x' fresh

For Equational TRSs, these CVPs are $\widetilde{\downarrow_{\mathcal{R}}}$ *-joinable*, cf. [Hue80]

[Luc26, Proposition 8.3(2)]

Let $\mathcal{R} = (\mathcal{F}, \Pi, \mu, E, H, R)$ be an EGTRS. Then,

$\rightarrow_{\mathcal{R}^{rm}}$ is locally coherent modulo E

iff all conditional pairs in

$$\mathsf{CCP}(\mathcal{R}, E) \cup \mathsf{CCP}(E, \mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(E) \cup \mathsf{CVP}^{\vdash}(\mathcal{R})$$

are $\widetilde{\downarrow}^{\mathsf{rs}}_{\mathcal{R}^{rm}}$-joinable (or $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$-joinable if $\mathcal{R}$ is E-terminating)

E-CONFLUENCE AS $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$-JOINABILITY OF CCPS AND CVPS

A rule $\ell \to r \Leftarrow c$ is

- *μ-left-linear* if no active variable occurs twice in ℓ
- *μ-left-homogeneous* if no active variable in ℓ is frozen in ℓ
- *μ-compatible* if no active variable in ℓ is frozen in r and no active variable in ℓ occurs in c

Joinability of CVPs for *conditional* rules $\alpha \in \mathcal{R}^{rm}$ and $\beta \in \hat{E}^{\diamond}$

If α and β are left μ-homogeneous and μ-compatible, then

CVP	Joinable with
$\pi^{\to}_{\alpha,x,p}$	$\downarrow_{\mathcal{R}^{rm}}$
$\pi^{\to}_{\beta,x,p}$	$\widetilde{\downarrow}_{\mathcal{R}^{rm}}$
$\pi^{\mapsto}_{\alpha,x,p}$	$\widetilde{\downarrow}_{\mathcal{R}^{rm}}$, if α is μ-left-linear

Theorem (E-Confluence of EGTRSs $\mathcal{R}$ using $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinability)

1. *An E-terminating EGTRS $\mathcal{R}$ is E-confluent if all pairs in*

$$\underbrace{\mathsf{CCP}(\mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(\mathcal{R})}_{\text{LCON-peaks}} \cup \underbrace{\mathsf{CCP}(\mathcal{R}, E) \cup \mathsf{CVP}^{\mapsto}(\mathcal{R}) \cup \mathsf{CCP}(E, \mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(E)}_{\text{LCOH-peaks}}$$

   are $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinable.

2. *An E-terminating left-linear ETRS $\mathcal{R}$ is E-confluent if all pairs in*

$$\mathsf{CP}(\mathcal{R}) \cup \mathsf{CP}(\mathcal{R}, E) \cup \mathsf{CP}(E, \mathcal{R})$$

   are $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinable [Hue80, Theorem 3.3].

Theorem (Non-E-confluence of EGTRSs $\mathcal{R}$)

If there is a non-$\widetilde{\downarrow}_{\mathcal{R}/E}$ -joinable $\pi \in \mathsf{CCP}(\mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(\mathcal{R})$ then $\mathcal{R}$ is not E-confluent.

40

$$\mathsf{s}(\mathsf{s}(x)) = x \Leftarrow x \geq \mathsf{s}(0) \quad (5) \qquad \mathsf{test}(x) \to \mathsf{pev}(x) \Leftarrow x = \mathsf{s}(\mathsf{s}(0)) \quad (8)$$
$$x \geq 0 \quad (6) \qquad \mathsf{test}(x) \to \mathsf{odd}(x) \Leftarrow x = \mathsf{s}(0) \quad (9)$$
$$\mathsf{s}(x) \geq \mathsf{s}(y) \Leftarrow x \geq y \quad (7) \qquad \mathsf{test}(x) \to \mathsf{zero}(x) \Leftarrow x = 0 \quad (10)$$

- $\mathrm{CCP}(\mathcal{R}) = \emptyset$: iCCPs $\downarrow_{\mathcal{R}}$-joinable (2-rules!); pCCPs *infeasible*; e.g.,

$$\pi_{(8),\Lambda,(9)} : \langle \mathsf{odd}(x), \mathsf{pev}(x) \rangle \Leftarrow x = \mathsf{s}(\mathsf{s}(0)), x = \mathsf{s}(0)$$

- $\mathrm{CVP}^{\to}(\mathcal{R}) = \emptyset$: $\to$-CVPs of R are *infeasible*; e.g.,

$$\pi^{\to}_{(8),x,1} : \langle \mathsf{test}(x'), \mathsf{pev}(x) \rangle \Leftarrow x \to x', x = \mathsf{s}(\mathsf{s}(0))$$

- $\mathrm{CCP}(\mathcal{R}, E) = \emptyset$: all CCPs are *infeasible*; e.g.,

$$\pi_{(8),\Lambda,\overleftarrow{(5)}} : \langle \mathsf{s}(\mathsf{s}(x)), \mathsf{pev}(x) \rangle \Leftarrow x = \mathsf{s}(\mathsf{s}(0)), \mathsf{test}(x) \geq \mathsf{s}(0)$$

- $\mathrm{CCP}(E, R) = \emptyset$: no rule in R overlaps any rule in $\overset{\diamond}{E}$.
- $\mathrm{CVP}^{\to}(E) = \emptyset$: $\to$-CVPs of E are infeasible.
- $\mathrm{CVP}^{\vdash}(\mathcal{R})$ consists of $\widetilde{\downarrow}_{\mathcal{R}}$ -joinable CVPs (see [Luc26, Example 8.5]).

Being clearly E-terminating, $\mathcal{R}$ is proved E-confluent

Limitations of $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinability of CCPs and CVPs

As for the running example, R^{rm} coincides with R, i.e.,

$$0 + n \to n \Leftarrow \mathsf{Nat}(n) \tag{11}$$
$$\mathsf{s}(m) + n \to \mathsf{s}(m + n) \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{12}$$
$$0 * n \to 0 \Leftarrow \mathsf{Nat}(n) \tag{13}$$
$$\mathsf{s}(m) * n \to \mathsf{s}(m * n) + n \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{14}$$
$$\mathsf{prod}(n) \to n \Leftarrow \mathsf{Nat}(n) \tag{15}$$
$$\mathsf{prod}(n +\!\!+ ns) \to n * \mathsf{prod}(ns) \Leftarrow \mathsf{Nat}(n), \mathsf{ListN}(ns) \tag{16}$$

We have that $\pi_{(16),1,(17)} \in \mathsf{CCP}(E, \mathcal{R})$, i.e.,

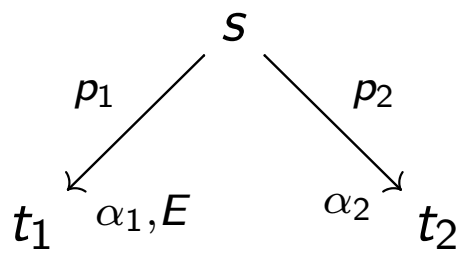
$$\langle \mathsf{prod}((xs +\!\!+ ys) +\!\!+ zs), xs * \mathsf{prod}(ys +\!\!+ zs) \rangle \Leftarrow \mathsf{Nat}(xs), \mathsf{ListN}(ys +\!\!+ zs)$$

is not $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinable: $\sigma = \{xs \mapsto 0, ys \mapsto 0, zs \mapsto 0\}$ satisfies $\mathsf{Nat}(xs)$ and $\mathsf{ListN}(ys +\!\!+ zs)$. However, $\sigma(\mathsf{prod}((xs +\!\!+ ys) +\!\!+ zs))$ and $\sigma(xs * \mathsf{prod}(ys +\!\!+ zs))$ are *not* $\widetilde{\downarrow}_{\mathcal{R}}$ -joinable.

E-confluence cannot be proved. We prove $\mathcal{R}$ *is E-confluent* below.

E-critical and E-variable peaks and their conditional pairs

The following peak, where $\alpha_1 \in \mathcal{R}^{rm}$,



1. is a *local E-confluence peak*, if $\alpha_2 \in \mathcal{R}^{rm}$
2. is a *local E-coherence peak*, if $\alpha_2 \in \overset{\diamond}{E}$

If $p_1 \parallel p_2$, then t_1 and t_2 are trivially $\widetilde{\downarrow}_{\mathcal{R},E}$ *-joinable*

Local E-confluence peaks

as

Logic-Based Conditional Critical Pairs

or

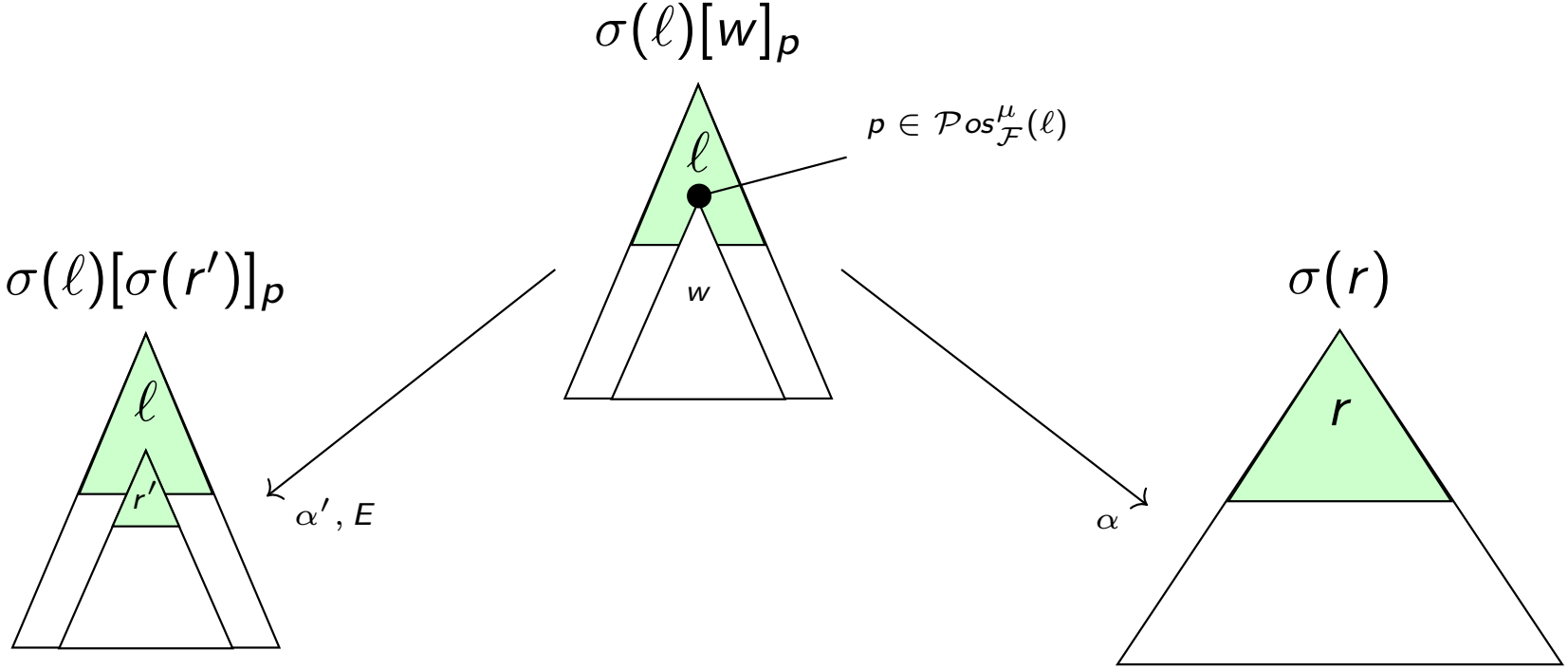
Conditional Variable Pairs

or

Down Conditional Pairs

LCON *E*-critical peaks ($\alpha \in \mathcal{R}^{rm}$, with $\alpha' \in \mathcal{R}^{rm}$ *E-overlapping* α)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$, $w =_E \sigma(\ell')$

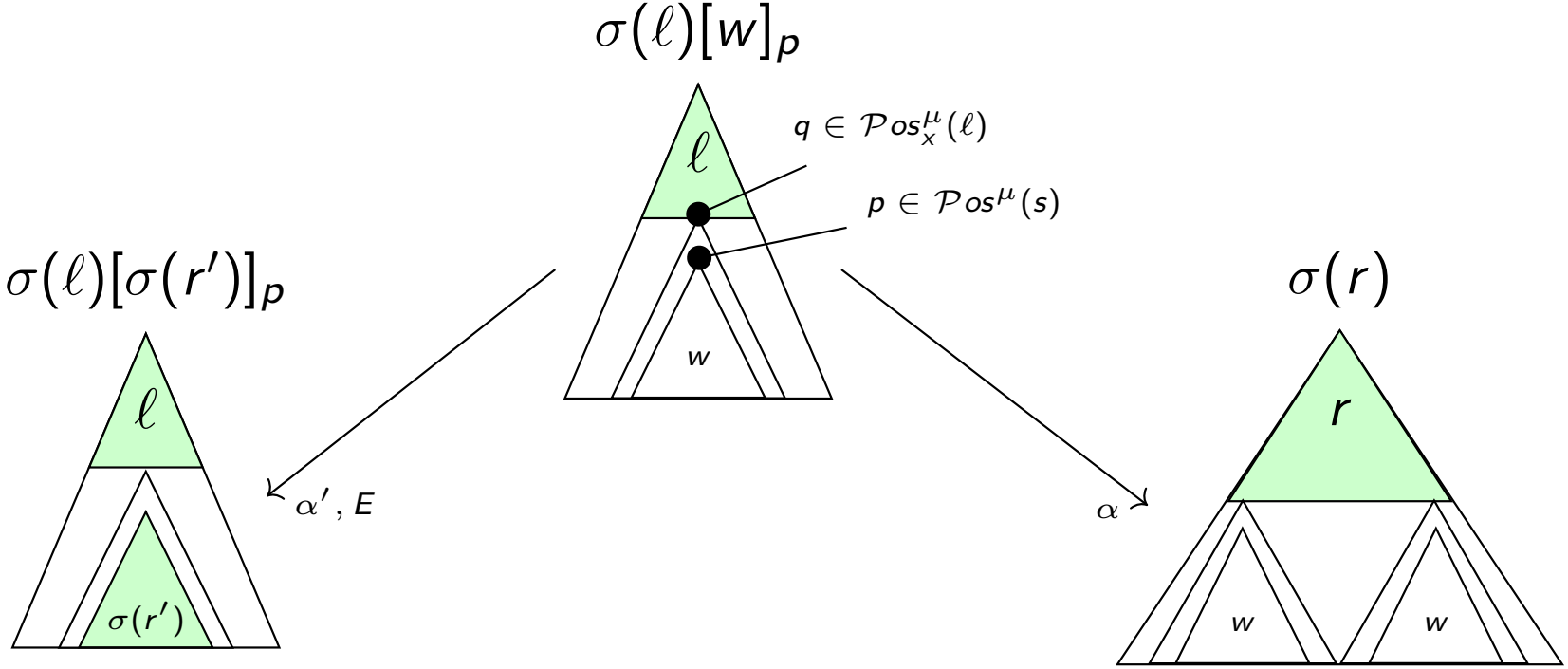


Logic-Based Conditional Critical Pairs LCCP($\mathcal{R}$)

$\pi^{\mathrm{LCCP}}_{\alpha,p,\alpha'} : \langle \ell[r']_p, r \rangle \Leftarrow \ell|_p = \ell', c, c'$

LCON E-variable peaks ($\alpha, \alpha' \in \mathcal{R}^{rm}$, *no E-overlap*)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p \notin \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$, $w =_E \sigma(\ell')$

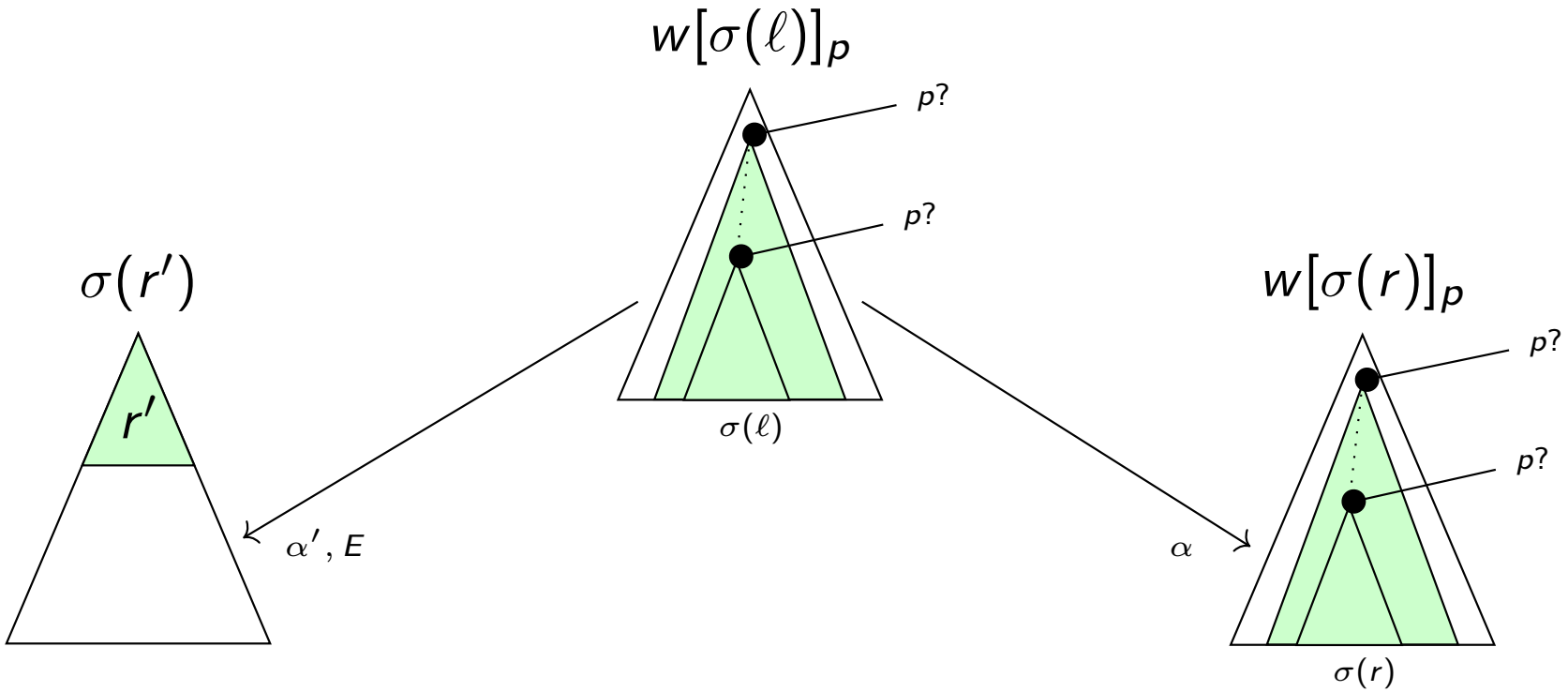


Conditional $\overset{ps}{\to}$-Variable Pairs $\mathrm{CVP}^{\overset{ps}{\to}}(\mathcal{R})$

$\pi^{\overset{ps}{\to}}_{\alpha,x,q} : \langle \ell[x']_q, r' \rangle \Leftarrow x \overset{ps}{\to} x', c$ $\quad x \in \mathcal{V}ar^{\mu}(\ell)$, $q \in \mathcal{P}os^{\mu}_{x}(\ell)$, x' fresh

LCON E-down peaks ($\alpha \in \mathcal{R}^{rm}$ and root PS-step with $\alpha' \in \mathcal{R}^{rm}$)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p > \Lambda$, $w[\sigma(\ell)]_p =_E \sigma(\ell')$



Down Conditional Pairs DCP($\mathcal{R}$)

$$\pi^{\mathrm{DCP}}_{\alpha'} : \langle r', x' \rangle \Leftarrow x = \ell', x \xrightarrow{>\Lambda} x', c'$$

[Luc26, Theorem 10.2(1)]

Let $\mathcal{R} = (\mathcal{F}, \Pi, \mu, E, H, R)$ be an EGTRS. Then,

$\rightarrow_{\mathcal{R}^{rm},E}$ is locally confluent modulo E with $\rightarrow_{\mathcal{R}^{rm}}$

iff all conditional pairs in

$$\mathsf{LCCP}(\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\rightarrow}}(\mathcal{R}) \cup \mathsf{DCP}(\mathcal{R})$$

are $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$ -joinable.

Local E-coherence peaks

as

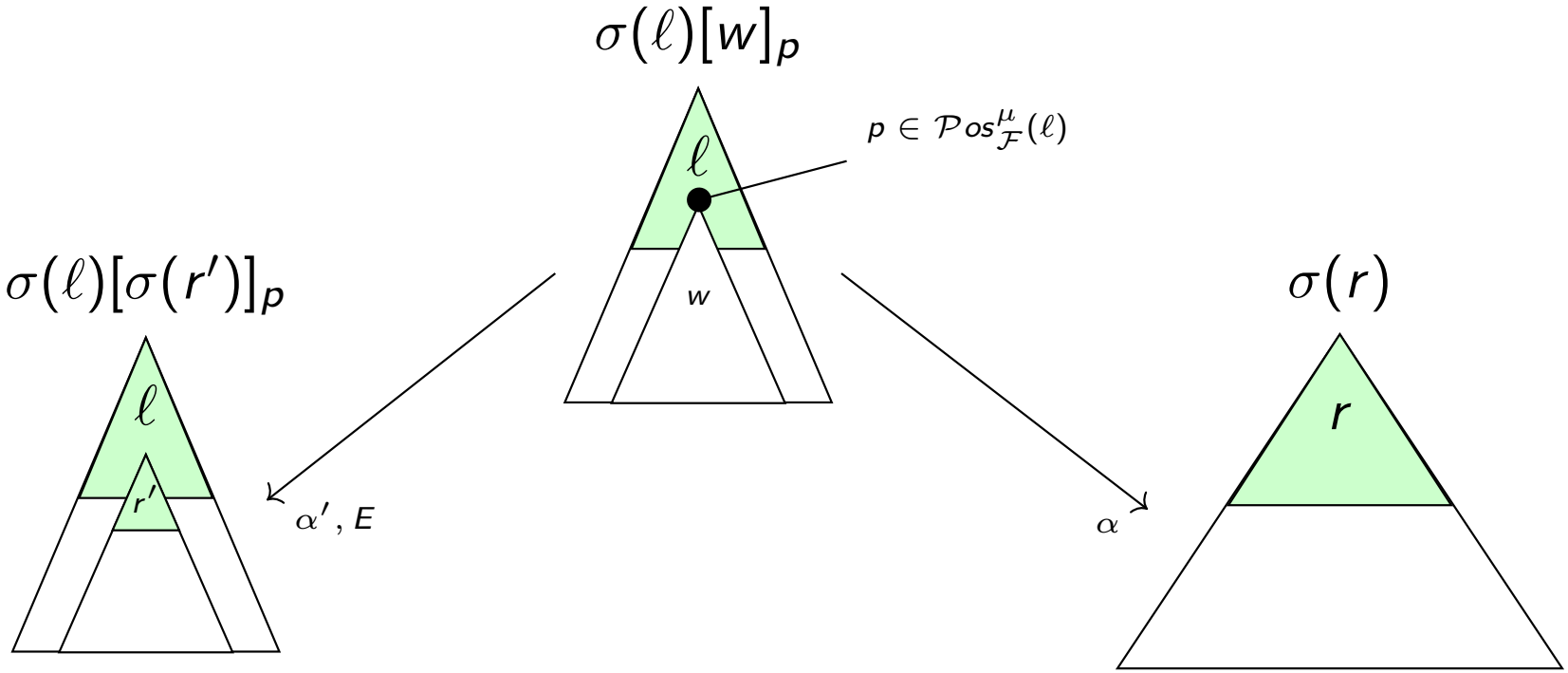
Logic-Based Conditional Critical Pairs

or

Conditional Variable Pairs

LCOH E-critical peaks ($\alpha \in \overset{\diamond}{E}$, with $\alpha' \in \mathcal{R}^{rm}$ E-overlapping α)

For $\alpha : \ell \to r \Leftarrow c \in \overset{\diamond}{E}$, $\alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p \in \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$, $w =_E \sigma(\ell')$

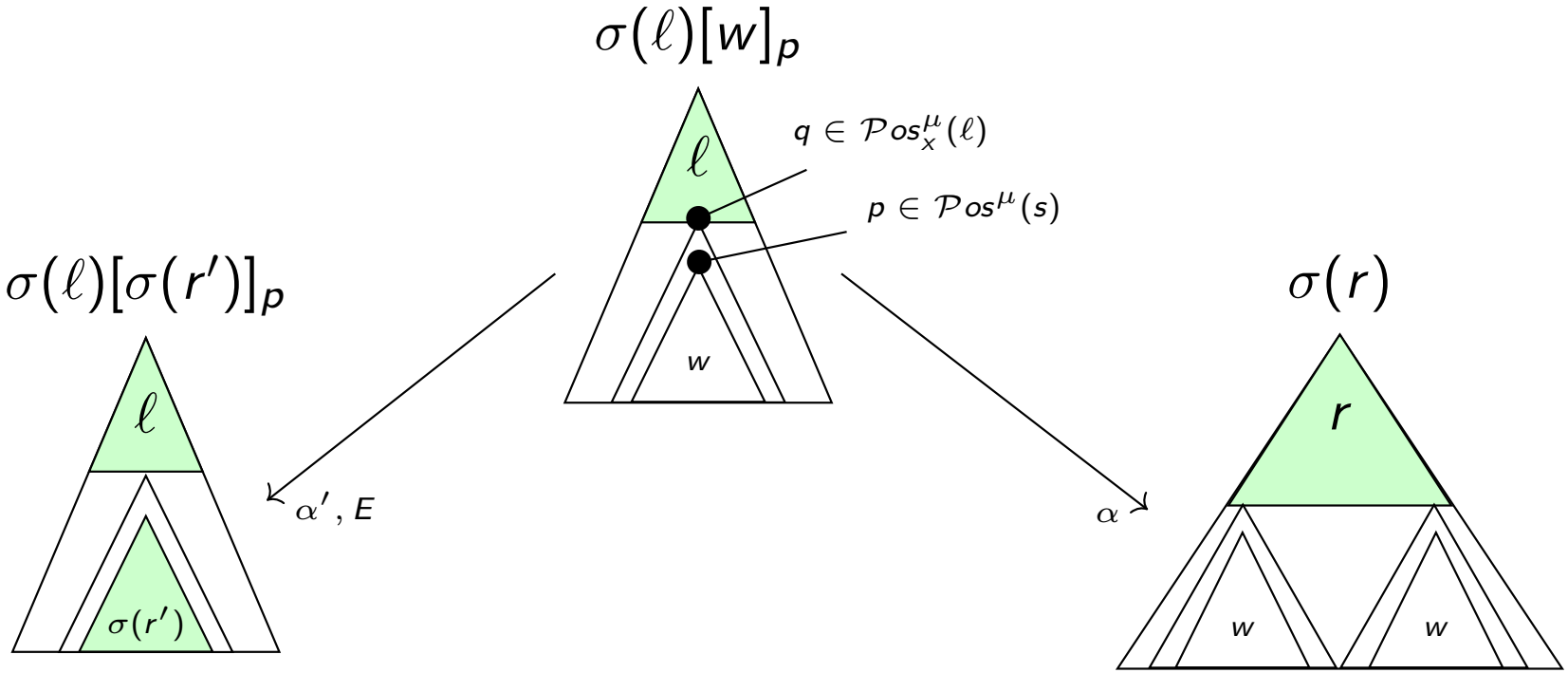


Logic-Based Conditional Critical Pairs LCCP$(E, \mathcal{R})$

$\pi^{\mathrm{LCCP}}_{\alpha,p,\alpha'} : \langle \ell[r']_p, r \rangle \Leftarrow \ell|_p = \ell', c, c'$ where $p > \Lambda$

LCOH E-variable peaks ($\alpha \in \overset{\diamond}{E}$, $\alpha' \in \mathcal{R}^{rm}$, *no E-overlap*)

For $\alpha : \ell \to r \Leftarrow c \in \overset{\diamond}{E}$, $\alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p \notin \mathcal{P}os^{\mu}_{\mathcal{F}}(\ell)$, $w =_E \sigma(\ell')$

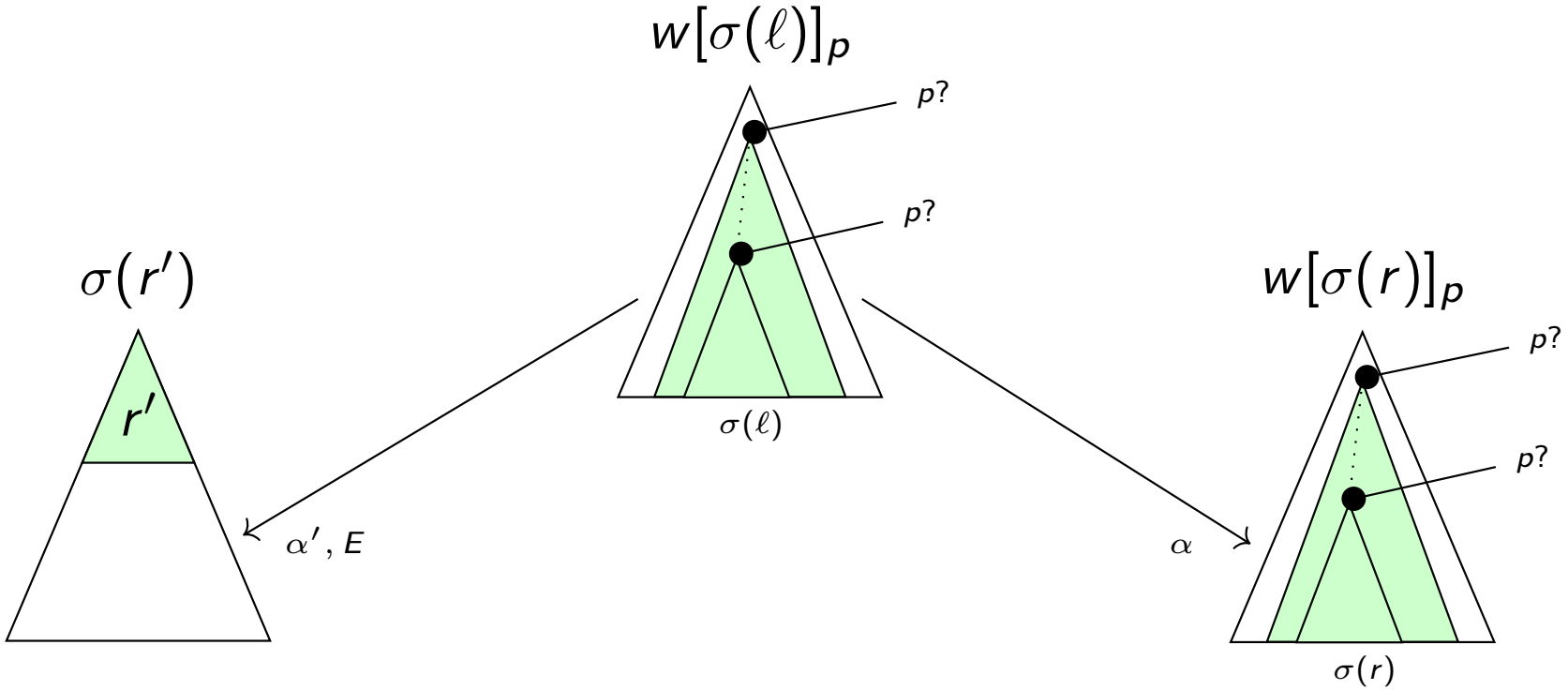


Conditional $\overset{ps}{\to}$-Variable Pairs $\mathrm{CVP}^{\overset{ps}{\to}}(E)$

$\pi^{\overset{ps}{\to}}_{\alpha,x,q} : \langle \ell[x']_q, r' \rangle \Leftarrow x \overset{ps}{\to} x', c$ $\quad$ $x \in \mathcal{V}ar^{\mu}(\ell)$, $q \in \mathcal{P}os^{\mu}_{x}(\ell)$, x' fresh

LCOH E-down peaks ($\alpha \in \overset{\diamond}{E}$ and root PS-step with $\alpha' \in \mathcal{R}^{rm}$)

For $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$, $p > \Lambda$, $w[\sigma(\ell)]_p =_E \sigma(\ell')$



Always $\downarrow_{\mathcal{R}^{rm},E}$-joinable

$$\sigma(r') \;{}_{\alpha'}\!\leftarrow \sigma(\ell') =_E w[\sigma(\ell)]_p \;{}_{\alpha^{-1}}\!\leftarrow w[\sigma(r)]_p$$

[Luc26, Theorem 10.2(2)]

Let $\mathcal{R} = (\mathcal{F}, \Pi, \mu, E, H, R)$ be an EGTRS. Then,

$\rightarrow_{\mathcal{R}^{rm},E}$ is locally coherent modulo E

iff all conditional pairs in

$$\mathrm{LCCP}(E,\mathcal{R}) \cup \mathrm{CVP}^{\overset{ps}{\rightarrow}}(E)$$

are $\widetilde{\downarrow}^{\mathsf{rs}}_{\mathcal{R}^{rm},E}$-joinable (or $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$ -joinable if $\mathcal{R}$ is E-terminating)

E-CONFLUENCE AS $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$ -JOINABILITY OF LCCPS, CVPS, AND DCPS

A rule $\ell \to r \Leftarrow c$ is

- *μ-left-homogeneous* if no active variable in ℓ is frozen in ℓ
- *μ-compatible* if no active variable in ℓ is frozen in r and no active variable in ℓ occurs in c

Joinability of CVPs for *conditional* rules $\alpha \in \mathcal{R}^{rm}$ and $\beta \in \overset{\diamond}{E}$

If α and β are left μ-homogeneous and μ-compatible, then

CVP	Joinable with
$\pi^{ps}_{\alpha,x,p}(\to)$	$\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$
$\pi^{ps}_{\beta,x,p}(\to)$	$\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$

Theorem (Confluence of EGTRSs $\mathcal{R}$ by $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$-joinability)

$\mathcal{R}$ is E-confluent if it is E-terminating and one of ❶ or ❷ holds:

❶ *all conditional pairs in the following set are $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$-joinable:*

$$\mathsf{LCCP}(\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\rightarrow}}(\mathcal{R}) \cup \mathsf{LCCP}(E,\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\rightarrow}}(E)$$

❷ *$R \cup \overset{\diamond}{E}$ is μ-left-homogeneous and μ-compatible and all conditional pairs in the following set are $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$-joinable:*

$$\mathsf{LCCP}(\mathcal{R}) \cup \mathsf{LCCP}(E,\mathcal{R})$$

Theorem (Non-E-confluence)

If there is a non-$\widetilde{\downarrow}_{\mathcal{R}/E}$-joinable $\pi \in \mathsf{LCCP}(\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\rightarrow}}(\mathcal{R}) \cup \mathsf{DCP}(\mathcal{R})$, then $\mathcal{R}$ is not E-confluent

Subsumes non-$\widetilde{\downarrow}_{\mathcal{R}/E}$-joinability of $\pi \in \mathsf{CCP}(\mathcal{R}) \cup \mathsf{CVP}^{\rightarrow}(\mathcal{R})$

58

E-CONFLUENCE OF THE RUNNING EXAMPLE

As for the running example, R^{rm} coincides with R, i.e.,

$$0 + n \to n \Leftarrow \mathsf{Nat}(n) \tag{11}$$
$$\mathsf{s}(m) + n \to \mathsf{s}(m + n) \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{12}$$
$$0 * n \to 0 \Leftarrow \mathsf{Nat}(n) \tag{13}$$
$$\mathsf{s}(m) * n \to \mathsf{s}(m * n) + n \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{14}$$
$$\mathsf{prod}(n) \to n \Leftarrow \mathsf{Nat}(n) \tag{15}$$
$$\mathsf{prod}(n +\!\!+ ns) \to n * \mathsf{prod}(ns) \Leftarrow \mathsf{Nat}(n), \mathsf{ListN}(ns) \tag{16}$$

LCCP($\mathcal{R}$) $= \emptyset$ as all conditions $\ell|_p = \ell', c, c'$ for $\pi^{\mathrm{LCCP}}_{\alpha,p,\alpha'}$ obtained from $\alpha : \ell \to r \Leftarrow c, \alpha' : \ell' \to r' \Leftarrow c' \in \mathcal{R}^{rm}$ are *infeasible*. E.g., $\pi^{\mathrm{LCCP}}_{(15),\Lambda,(16)}$:

$$\langle n' * \mathsf{prod}(ns'), n\rangle \Leftarrow \mathsf{prod}(n) = \mathsf{prod}(n' +\!\!+ ns'), \mathsf{Nat}(n), \mathsf{Nat}(n'), \mathsf{ListN}(ns')$$

is infeasible: if $\sigma(\mathsf{prod}(n)) =_E \sigma(\mathsf{prod}(n' +\!\!+ ns'))$ holds, then $\sigma(n) =_E \sigma(n' +\!\!+ ns')$, i.e., $root(\sigma(n))$ is $+\!\!+$, and $\sigma(\mathsf{Nat}(n))$ does *not* hold.

Variables in atoms A of the conditional part c of rules $\ell \to r \Leftarrow c \in \mathcal{R}$ of the running example

$$0 + n \to n \Leftarrow \mathsf{Nat}(n) \tag{11}$$
$$\mathsf{s}(m) + n \to \mathsf{s}(m + n) \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{12}$$
$$0 * n \to 0 \Leftarrow \mathsf{Nat}(n) \tag{13}$$
$$\mathsf{s}(m) * n \to \mathsf{s}(m * n) + n \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n) \tag{14}$$
$$\mathsf{prod}(n) \to n \Leftarrow \mathsf{Nat}(n) \tag{15}$$
$$\mathsf{prod}(n +\!\!+ ns) \to n * \mathsf{prod}(ns) \Leftarrow \mathsf{Nat}(n), \mathsf{ListN}(ns) \tag{16}$$

are all preserved under $\to_{\mathcal{R},E}$-reductions in the sense of [Luc24b, page 17].

By [Luc24b, Proposition 61(2)], all CVPs in $\mathrm{CVP}^{\overset{ps}{\to}}(\mathcal{R})$ are $\downarrow_{\mathcal{R},E}$-joinable. For instance,

$$\pi^{\overset{ps}{\to}}_{(15),n,1} : \langle \mathsf{prod}(n'), n\rangle \Leftarrow n \overset{ps}{\to} n', \mathsf{Nat}(n)$$

is $\downarrow_{\mathcal{R},E}$-joinable because $\mathsf{Nat}(n)$ is preserved under $\to_{\mathcal{R},E}$-reductions: if $\mathsf{Nat}(s)$ holds for some term s, and $s \to_{\mathcal{R},E} t$, then $\mathsf{Nat}(t)$ holds as well.

The rules in $\overset{\diamond}{E} = \{\overrightarrow{(1)}, \overleftarrow{(1)}\} = \{(17), (18)\}$ are

$$xs +\!\!+ (ys +\!\!+ zs) \to (xs +\!\!+ ys) +\!\!+ zs \tag{17}$$

$$(xs +\!\!+ ys) +\!\!+ zs \to xs +\!\!+ (ys +\!\!+ zs) \tag{18}$$

LCCP$(E,\mathcal{R}) = \emptyset$ as all positions $p > \Lambda$ in the left-hand sides of rules (17) and (18) are frozen.

All variables in rules of $\overset{\diamond}{E}$:

$$xs +\!\!+ (ys +\!\!+ zs) \rightarrow (xs +\!\!+ ys) +\!\!+ zs \tag{17}$$

$$(xs +\!\!+ ys) +\!\!+ zs \rightarrow xs +\!\!+ (ys +\!\!+ zs) \tag{18}$$

are frozen. Hence, $\mathrm{CVP}^{\overset{ps}{\rightarrow}}(E) = \emptyset$.

After removing *sorts* and *replacement restrictions*, we obtain a TRS $\widehat{\mathcal{R}}$

$$0 + n \to n \tag{19}$$
$$\mathsf{s}(m) + n \to \mathsf{s}(m + n) \tag{20}$$
$$0 * n \to 0 \tag{21}$$
$$\mathsf{s}(m) * n \to \mathsf{s}(m * n) + n \tag{22}$$
$$\mathsf{prod}(n) \to n \tag{23}$$
$$\mathsf{prod}(n +\!\!+ ns) \to n * \mathsf{prod}(ns) \tag{24}$$

where $+\!\!+$ is *associative*. The A-termination of $\widehat{\mathcal{R}}$ can be proved by using AProVE or MU-TERM. Since $\to_{\mathcal{R}/E} \subseteq \to_{\widehat{\mathcal{R}}/A}$, $\mathcal{R}$ *is E-terminating*.

Since all conditional pairs in

$$\mathsf{LCCP}(\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\to}}(\mathcal{R}) \cup \mathsf{LCCP}(E,\mathcal{R}) \cup \mathsf{CVP}^{\overset{ps}{\to}}(E)$$

are $\widetilde{\downarrow}_{\mathcal{R},E}$-joinable and $\mathcal{R}$ is E-terminating, E-confluence of $\mathcal{R}$ follows.

DCPs in proofs of non-E-confluence

DCPs in proofs of non-E-confluence

Consider the ETRS $\mathcal{R}$ [Luc24a, Example 14]:

$$\mathsf{b} = \mathsf{f}(\mathsf{a}) \quad (25)$$
$$\mathsf{a} = \mathsf{c} \quad (26)$$
$$\mathsf{c} \to \mathsf{d} \quad (27)$$
$$\mathsf{b} \to \mathsf{d} \quad (28)$$

All pairs in $\mathrm{LCCP}(\mathcal{R})$ are trivial, i.e., joinable. And $\mathrm{CVP}^{\overset{ps}{\to}}(\mathcal{R}) = \emptyset$. But

$$\pi^{\mathrm{DCP}}_{(28)} : \langle \mathsf{d}, x' \rangle \Leftarrow x = \mathsf{b}, x \xrightarrow{>\Lambda} x'$$

is not $\widetilde{\downarrow}_{\mathcal{R}/E}$-joinable: $\sigma = \{x \mapsto \mathsf{f}(\mathsf{c}), x' \mapsto \mathsf{f}(\mathsf{d})\}$ satisfies $x = \mathsf{b}, x \xrightarrow{>\Lambda} x'$, but d and $\sigma(x') = \mathsf{f}(\mathsf{d})$ are not $\mathcal{R}/E$-joinable.

Thus, $\mathcal{R}$ is *not E-confluent*

Limitations of $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$ -joinability

of

LCCPs, CVPs, and DCPs

The previously discussed ETRS [Luc26, Example 4.14]:

$$\begin{array}{llll} a = f(b) & (1) & c \to a & (3) \\ b \to d & (2) & c \to f(d) & (4) \end{array}$$

is E-confluent, but *neither* $\mathsf{LConf}_E(\to_{\mathcal{R},E}, \to_{\mathcal{R}})$ *nor* $\mathsf{LCoh}_E(\to_{\mathcal{R},E})$ hold

$\mathsf{LConf}_E(\to_{\mathcal{R},E}, \to_{\mathcal{R}})$ fails:

$a \;{}_{(3)}\!\leftarrow\; c \;\to_{(4)} f(d)$

is *not* $\widetilde{\downarrow}_{\mathcal{R},E}$-joinable

(but $a =_E f(\underline{b}) \to_{(2)} f(d)$)

$\mathsf{LCoh}_E(\to_{\mathcal{R},E})$ fails:

$f(d) \;{}_{(2)}\!\leftarrow\; f(b) \;\to_{\blacktriangleleft(1)} a$

is *not* $\widetilde{\downarrow}_{\mathcal{R},E}$-joinable

Future work

Improve the analysis of E-confluence by using other approaches

Part II

Application to Rewrite Theories and Maude

1. Maude functional modules
2. Maude functional modules as EGTRSs
3. Maude functional modules as EGTRSs: mismatches
4. Maude system modules
5. Maude system modules as EGTRSs
6. Maude system modules as EGTRSs: mismatches
7. Related work

Maude functional modules

$\mathcal{M} = (\Sigma, M \cup B \cup E)$ is a Maude functional module, where:

1. Σ is a signature;
2. M is a collection of (possibly conditional) memberships;
3. B is a collection of (unconditional) equational axioms;
4. E is a collection of (possibly conditional) equations.

Σ is a signature consisting of:

1. a set S of sorts (or types),
2. a subsort ordering relation $\leq$: $s < s'$ means that s is a (strict) *subsort* of s'. Tuples: $s_1 \cdots s_n \leq s'_1 \cdots s'_n$ iff $s_i \leq s'_i$ for all $1 \leq i \leq n$.
3. ranked function symbols $f : w \to s$ with $w = s_1 \cdots s_k$ for some $k \geq 0$, denoted $f \in \Sigma_{w,s}$;
4. operator evaluation strategies $\varphi(f)$ for k-ary functions f as sequences of numbers $0, \ldots, k$. E.g., for `if_then_else_` it is `(1 0 2 3 0)`.

Top sort

$(S, \leq)$ may contain a greatest element, i.e., a top sort $\top \in S$ such that for all $s \in S$, $s \leq \top$.

Overloading

Function symbols can be overloaded, i.e., $f \in \Sigma_{w,s} \cap \Sigma_{w',s'}$, provided that $w \leq w'$ implies $s \leq s'$ (monotonicity condition [GM92])

M is a set of (possibly conditional) memberships $t : s \Leftarrow c$ where

- t is a term,
- s is a sort,
- c consists of
  - membership statements $t_i : s_i$ and/or
  - equational goals, which can be of three types:
    1. ordinary equations $u = v$,
    2. matching equations $p := t$,
    3. abbreviated Boolean equations of the form t (instead of $t = \mathsf{true}$).

B is a collection of (unconditional) equational axioms satisfied by some of the (infix) binary operators $\oplus : s\ s \to s$ in Σ for some $s \in S$:

- associativity: $x \oplus (y \oplus z) = (x \oplus y) \oplus z$,
- commutativity: $x \oplus y = y \oplus x$,
- idempotency: $x \oplus x = x$
- (left/right) identity w.r.t. some term t: $t \oplus x = x$ / $x \oplus t = x$.

These axioms are associated to particular binary symbols as operator attributes

`assoc`, `comm`, `idem`, *and* (`left`/`right`) `id:`

in the signature declaration.

For instance

```
op _+_ : Nat Nat -> Nat [assoc] .
```

establishes the addition operator as associative.

E is a collection of (possibly conditional) equations: $u = v \Leftarrow c$, where

- u and v are terms of the same kind or connected component $[s]$ for some $s \in S$,
- c consists of
  - membership statements $t_i : s_i$ and/or
  - equational goals, which can be of three types:
    1. ordinary equations $u = v$,
    2. matching equations $p := t$,
    3. abbreviated Boolean equations of the form t (instead of $t = \mathsf{true}$).

Equations as rewrite rules

Conditional equations $u = v \Leftarrow c$ in E are used as left-to-right rewrite rules $u \to v \Leftarrow c$ to evaluate terms.

Furthermore, such equations are sort decreasing, i.e., for all substitutions σ $\mathsf{ls}(\sigma(u)) \geq \mathsf{ls}(\sigma(v))$, where, given a term t, $\mathsf{ls}(t)$ is the *least sort* of t.

Maude functional modules as EGTRSs

Translating Maude functional modules

$$\mathcal{M} = ((S, \leq, \Sigma_{w \in S^+}, \varphi), M \cup B \cup E)$$

into an EGTRSs

$$\mathcal{R}_{\mathcal{M}} = (\mathcal{F}_{\mathcal{M}}, \Pi_{\mathcal{M}}, \mu_{\mathcal{M}}, E_{\mathcal{M}}, H_{\mathcal{M}}, R_{\mathcal{M}})$$

1. $\mathcal{F}_{\mathcal{M}}$ includes a function symbol f for each $f \in \Sigma_{w \in S^+}$.
2. $\Pi_{\mathcal{M}}$ includes predicates $P_{s/1}$ for each $s \in S$.
3. $\Pi_{\mathcal{M}}$ includes $\_\downarrow\_$ if equations are used in the conditions of equations.
4. $\mu_{\mathcal{M}}$ is given by $\mu_{\mathcal{M}}(f) = \{i \in \varphi(f) \mid i > 0\}$.
5. $H_{\mathcal{M}}$ includes $P_{s'}(x) \Leftarrow P_s(x)$ for each subsort statement $s < s'$, if s' is a non-top sort.
6. $H_{\mathcal{M}}$ includes $P_s(f(x_1, \ldots, x_k)) \Leftarrow P_{s_1}(x_1), \ldots, P_{s_k}(x_k)$ for each $f : s_1 \cdots s_k \to s$ in Σ. $P_{s_i}(x_i)$ is included only if s_i is a non-top sort.

Encoding the signature of ListsP

- $\mathcal{F}_{\mathcal{M}} = \{\ \ 0,\ \ \ s_{/1},\ \ \ \_ +\!+ \_,\ \ \ \_ + \_,\ \ \ \_ * \_,\ \ \ prod_{/1}\ \ \ \}$
- $\Pi_{\mathcal{M}} = \{\ \ \mathsf{Nat}_{/1},\ \ \ \mathsf{ListN}_{/1}\ \ \}$
- $\mu_{\mathcal{M}}(\ +\!+\ ) = \emptyset, \quad \mu(\mathsf{s}) = \mu(+) = \mu(*) = \mu(\mathsf{prod}) = \{1\}$
- $H_{\mathcal{M}}$ includes the following clauses:

$$\mathsf{ListN}(xs +\!+ ys) \qquad \mathsf{Nat}(m + n) \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n)$$
$$\mathsf{Nat}(0) \qquad \mathsf{Nat}(m * n) \Leftarrow \mathsf{Nat}(m), \mathsf{Nat}(n)$$
$$\mathsf{Nat}(\mathsf{s}(n)) \Leftarrow \mathsf{Nat}(n) \qquad \mathsf{Nat}(\mathsf{prod}(xs))$$

79

1. $\Pi_{\mathcal{M}}$ includes a a binary predicate $\_ \downarrow \_$ if ordinary equations are used in the conditional part of memberships.
2. $H_{\mathcal{M}}$ includes $P_s(t) \Leftarrow c'$ for each $t : s \Leftarrow c$ in M, where c' is obtained from c by
   1. adding $P_s(x)$ for each variable x of a non-top sort s occurring in t or c;
   2. replacing $t' : s'$ in c by $P_{s'}(t')$;
   3. replacing $u = v$ in c, by $u \downarrow v$;
   4. replacing $p := t$ in c, by $t \rightarrow^* p$, and
   5. replacing abbreviated boolean equations t in c, by $t \rightarrow^* \mathit{true}$.
3. $H_{\mathcal{M}}$ includes $P_s(x) \Leftarrow x \rightarrow y, P_s(y)$, where x and y are variables, for each $s \in S$ with a membership $t : s \Leftarrow c$ (*Subject reduction rules*).
4. $H_{\mathcal{M}}$ includes $x \downarrow y \Leftarrow x \rightarrow^* z, y \rightarrow^* z$ if ordinary equations are used in the conditional part of memberships.

Let B_f be the set of equations associated to all attributes `assoc`, `comm`, etc., for f. Then, $B = \bigcup_{f \in \Sigma} B_f$. Accordingly,

$$E_{\mathcal{M}} \text{ is } B'$$

where each $u = v \in B$, with variables x_i of sort s_i, becomes $u = v \Leftarrow c$ in B', where c includes an atom $P_{s_i}(x_i)$ if s_i is a non-top sort.

Encoding the axiom of `ListsP`

Since $++ : \text{ListN ListN} \to \text{ListN}$ and ListN is the top sort of $(S, \leq)$,

$$E_{\mathcal{M}} = \{xs ++ (ys ++ zs) = (xs ++ ys) ++ zs\}$$

Functional modules as GTRSs

If $B = \emptyset$ (no axioms are used), then $\mathcal{R}_{\mathcal{M}}$ is a GTRS. Thus,

existing methods and tools for proving confluence and termination of GTRSs (CONFident and MU-TERM-GTRS) apply to functional modules.

1. $\Pi_{\mathcal{M}}$ includes a a binary predicate $\_ \downarrow \_$ if ordinary equations are used in the conditional part of equations in E.
2. $H_{\mathcal{M}}$ includes $x \downarrow y \Leftarrow x \to^* z, y \to^* z$ if ordinary equations are used in the conditional part of equations in E.
3. $R_{\mathcal{M}}$ consists of rules $\ell \to r \Leftarrow c'$ for each $\ell = r \Leftarrow c$ in E, where c' is obtained from c by
   1. adding $P_s(x)$ for each variable x of a non-top sort s occurring in ℓ, r, or c;
   2. replacing $t : s$ in c by $P_s(t)$;
   3. replacing $u = v$ in c, by $u \downarrow v$;
   4. replacing $p := t$ in c, by $t \to^* p$; and
   5. replacing abbreviated boolean equations t in c, by $t \to^* \mathit{true}$.

Encoding the equations of `ListsP`

- $R_{\mathcal{M}}$ consists of the following rules:

$$0 + n \to n \Leftarrow \mathrm{Nat}(n) \qquad \mathrm{s}(m) * n \to \mathrm{s}(m * n) + n \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n)$$

$$\mathrm{s}(m) + n \to \mathrm{s}(m + n) \Leftarrow \mathrm{Nat}(m), \mathrm{Nat}(n) \qquad \mathrm{prod}(n) \to n \Leftarrow \mathrm{Nat}(n)$$

$$0 * n \to 0 \Leftarrow \mathrm{Nat}(n) \qquad \mathrm{prod}(n +\!\!+ ns) \to n * \mathrm{prod}(ns) \Leftarrow \mathrm{Nat}(n)$$

Maude functional modules as EGTRSs: Mismatches

```
fmod AddPalindromes is
  sorts Nat Pal ListN .
  subsorts Nat < Pal < ListN .
  op _++_ : ListN ListN -> ListN
      [strat (0) assoc] .
  op 0 : -> Nat .
  ops s double : Nat -> Nat .
  op _+_ : Nat Nat -> Nat .
  op sum : ListN -> Nat .
  vars m n : Nat .
  var p : Pal .
  var ns : ListN .
  mb n ++ n : Pal .
  mb n ++ p ++ n : Pal .
  eq 0 + n = n .
  eq s(m) + n = s(m + n) .
  eq double(0) = 0 .
  eq double(s(n)) = s(s(double(n))) .
  eq sum(n) = n .
  eq sum(n ++ n) = double(n) .
  eq sum(n ++ p ++ n) = double(n) + sum(p) .
endfm
```

```
Maude> red sum((0 ++ s(0)) ++ (s(0) ++ 0))  .
...
result Nat: s(s(0))
```

In the EGTRS encoding, this is not possible:

$$\begin{aligned} sum(\underline{(0 \mathbin{+\!\!+} s(0)) \mathbin{+\!\!+} (s(0) \mathbin{+\!\!+} 0)}) &=_A sum(0 \mathbin{+\!\!+} \underline{(s(0) \mathbin{+\!\!+} (s(0) \mathbin{+\!\!+} 0))}) \\ &\neq_A sum(0 \mathbin{+\!\!+} ((s(0) + +s(0)) \mathbin{+\!\!+} 0)) \end{aligned}$$

The last step fails due to $\mu(\mathbin{+\!\!+}) = \emptyset$.

If replacement restrictions are desirable, we can avoid modifying equivalence classes (for associativity, commutativity, etc.) by adding conditional (propagation) equations

$$x \oplus y = x' \oplus y \quad \Leftarrow \quad x = x' \tag{29}$$
$$x \oplus y = x \oplus y' \quad \Leftarrow \quad y = y' \tag{30}$$

to the E component of $\mathcal{R}_{\mathcal{M}}$ for operators $\oplus$ endowed with empty strategies `strat (0)`.

In the case above, we add

$$x +\!\!+ y = x' +\!\!+ y \quad \Leftarrow \quad x = x' \tag{31}$$
$$x +\!\!+ y = x +\!\!+ y' \quad \Leftarrow \quad y = y' \tag{32}$$

```
fmod Example is
  sort S .
  op 0 : -> S .
  op s : S -> S .
  op _+_ : S S -> S [strat (0) assoc] .
  var x : S .
  eq 0 + x = x .
endfm
```

```
(REPLACEMENT-MAP
  (+ )
)
(VAR x y z)
(EQUATIONS
  x + (y + z) = (x + y) + z
)
(RULES
  0 + x -> x
)
```

```
Maude> set trace on .
Maude> red (s(0) + 0) + 0 .
reduce in Example : (s(0) + 0) + 0 .
********** equation
eq 0 + x = x .
x --> 0
s(0) + 0 + 0
--->
s(0) + 0
...
result S: s(0) + 0
```

$s(0) + (0 + 0)$ is $\to_{\mathcal{R}/E}$-irreducible:

$s(0) + (0 + 0) =_A (s(0) + 0) + 0$

However, $0 + x \to x$ does not apply to any of them

Frozen subterm $0+0$ is rewritten!

```
fmod ExampleA is
    sort S .
    ops a b c : -> S .
    op f : S -> S .
    eq a = c .
    eq f(c) = a .
    eq b = c .
    ceq b = a if a = f(b) .
endfm
```

```
fmod ExampleB is
    sort S .
    ops a b c : -> S .
    op f : S -> S .
    eq a = c .
    eq f(c) = a .
    ceq b = a if a = f(b) .
    eq b = c .
endfm
```

```
Maude> red b .
reduce in ExampleA : b .
...
result S: c
```

```
Maude> red b .
reduce in ExampleB : b .

Fatal error: stack overflow.
This can happen because you have
an infinite computation, ...
```

ExampleB is as ExampleA up to permutation of the last two rules

- Both are confluent and terminating; a and f(b) are normalized into c.
- Condition a = f(b) is treated by joinability to normal form, but fails
- Rules are attempted in the order they are written.

87

```
fmod EqConditions is
  sort S .
  ops 0 1 2 : -> S .
  op f : S -> S .
  op _+_ : S S -> S [assoc] .
  var x : S .
  eq 0 + 0 = 0 .
  eq 1 + 0 + 1 = 2 .
  ceq f(x) = x if (1 + x) + (x + 1) = 2 .
endfm
```

```
Maude> red f(0) .
reduce in EqConditions : f(0) .
...
result S: 0
```

```
fmod EqConditionsIDEM is
  sort S .
  ops 0 1 2 : -> S .
  op f : S -> S .
  op _+_ : S S -> S [assoc idem] .
  var x : S .
  eq 1 + 0 + 1 = 2 .
  ceq f(x) = x if (1 + x) + (x + 1) = 2 .
endfm
```

```
Maude> red f(0) .
reduce in EqConditionsIDEM : f(0) .
...
result S: f(0)
```

Evaluating (1 + 0) + (0 + 1) using $\to_{\mathcal{R}/E}$:

$$\begin{array}{lll} \underline{(1+0)+(0+1)} & =_A & 1+(\underline{0+(0+1)}) \\ & =_A & 1+(\underline{0+0})+1 \\ & \to & \underline{1+0+1} \\ & \to & 2 \end{array}$$

$$\begin{array}{lll} \underline{(1+0)+(0+1)} & =_A & 1+(\underline{0+(0+1)}) \\ & =_A & 1+(\underline{0+0})+1 \\ & =_{Idem} & \underline{1+0+1} \\ & \to & 2 \end{array}$$

Maude system modules

$\mathcal{M} = (\Sigma, M \cup B \cup E, R, \phi)$ is a Maude system module, where:

1. Σ is a signature;
2. M is a collection of (possibly conditional) memberships;
3. B is a collection of (unconditional) equational axioms;
4. E is a collection of (possibly conditional) equations;
5. R is a collection of (possibly conditional) rewrite rules; and
6. ϕ is a frozenness map.

System modules embed a functional module

If $\mathcal{M} = (\Sigma, M \cup B \cup E, R, \phi)$ is a Maude system module, then $(\Sigma, M \cup B \cup E)$ is a functional module.

Besides,

1. R is a collection of (possibly conditional) rules: $\ell \to r \Leftarrow c$, where
   - ℓ and r are terms of the same kind.
   - c consists of
     - memberships $t : s$ and equational goals $u = v$ as in functional modules; also
     - rewriting goals $u \to v$ which become reachability goals $\sigma(u) \to^* \sigma(v)$ for some (matching) substitution σ.
2. ϕ is a mapping that, for each k-ary symbol $f \in \Sigma$, returns the subset $\phi(f) \subseteq \{1, \ldots, k\}$ of frozen arguments of f, on which rewriting with rules is forbidden.

Maude system modules as EGTRSs

1. $\mu_{\mathcal{M}}$ is given by $\mu_{\mathcal{M}}(f) = \{1, \ldots, k\} - \phi(f)$ for each k-ary $f \in \Sigma$.

What about evaluation strategies φ?

2. $R_{\mathcal{M}}$ consists of rules $\ell \to r \Leftarrow c'$ for each $\ell \to r \Leftarrow c \in \vec{E} \cup R$, where c' is obtained from c by
   1. adding $P_s(x)$ for each variable x of a non-top sort s occurring in ℓ, r, or c;
   2. replacing $t : s$ in c by $P_s(t)$;
   3. replacing $u = v$ in c, by $u \downarrow v$;
   4. replacing $p := t$ in c, by $t \to^* p$; and
   5. replacing abbreviated boolean equations t in c, by $t \to^* \mathit{true}$.
   6. replacing $u \to v$ in c, by $u \to^* v$.

Are rules from $\vec{E}$ and rules from R used in the same way?

Maude system modules as EGTRSs: Mismatches

```
mod Example is
  sort S .
  ops a b c d : -> S .
  op f : S -> S .
  eq a = f(b) .
  rl b => d .
  rl c => a .
  rl c => f(d) .
endm
```

```
Maude> search c =>! Y:S .
search in Example : c =>! Y:S .

Solution 1 (state 2)
...
Y:S --> f(d)
```

Can be proved confluent

```
mod ExampleBis is
  sort S .
  ops a b c d : -> S .
  op f : S -> S .
  eq f(b) = a .
  rl b => d .
  rl c => a .
  rl c => f(d) .
endm
```

```
Maude> search c =>! Y:S .
search in ExampleBis : c =>! Y:S .

Solution 1 (state 1)
...
Y:S --> a

Solution 2 (state 2)
...
Y:S --> f(d)
```

Confluence can be disproved

```
mod StratFrozen is
  sort S .
  op 0 : -> S .
  op s : S -> S .
  op _+_ : S S -> S
     [strat (1 2 0) frozen] .
  var x : S .
  eq 0 + x = x .
  rl s(x) => x .
endm
```

```
mod StratFrozen2 is
  sort S .
  op 0 : -> S .
  op s : S -> S .
  op _+_ : S S -> S
     [strat (1 2 0) frozen] .
  var x : S .
  rl 0 + x => x .
  eq s(x) = x .
endm
```

```
Maude> rew 0 + (s(0) + 0) .
rewrite in StratFrozen :
...
result S: s(0) + 0

Maude> red 0 + (s(0) + 0) .
reduce in StratFrozen :
...
result S: s(0) + 0
```

```
Maude> rew 0 + (s(0) + 0) .
rewrite in StratFrozen2 :
...
result S: 0

Maude> red 0 + (s(0) + 0) .
reduce in StratFrozen2 :
...
result S: 0 + (0 + 0)
```

The use of evaluation strategies and frozenness annotations depends on *using* rules from $\vec{E}$ or from R

```
mod TEST-E is
    sort S .
    op a : -> S .
    op b : -> S .
    op c : -> S .
    op f : S -> S .
    var X : S .
    eq a = f(b) .
    rl b => c .
    crl b => X if a => X .
endm
```

```
Maude> search [4] b =>* Y:S .
search [4] in TEST-E : b =>* Y:S .
...
Y:S --> b
...
Y:S --> c
...
Y:S --> f(b)
...
Y:S --> f(c)
```

```
mod TEST-R is
    sort S .
    op a : -> S .
    op b : -> S .
    op c : -> S .
    op f : S -> S .
    var X : S .
    rl a => f(b) .
    rl b => c .
    crl b => X if a => X .
endm
```

```
Maude> search [4] b =>* Y:S .
search [4] in TEST-R : b =>* Y:S .
...
Y:S --> b
...
Y:S --> c
...
Y:S --> a
...
Y:S --> f(b)
```

Evaluation of expressions (also in conditions of rules) uses $\to^{!}_{\vec{E},B} \circ \to_{R,B}$

97

```
mod TEST is
    sort S .
    ops a b c d : -> S .
    var X : S .
    eq a = b .
    rl a => c .
    crl d => X if a => X .
endm
```

```
(VAR x)
(RULES
    a -> b
    a -> c
    d -> x | a ->* x
)
```

```
Maude> search a =>* Y:S .
Y:S --> b
No more solutions.

Maude> search b =>* Y:S .
Y:S --> b
No more solutions.

Maude> search c =>* Y:S .
Y:S --> c
No more solutions.
```

```
Maude> search d =>* Y:S .
Y:S --> d
Y:S --> b
No more solutions.
```

```
Maude> search a =>* c .
search in TEST : a =>* c .
No solution.
```

TEST is confluent but $\mathcal{R}_{\text{TEST}}$ is proved non-confluent

Rule a => b is never used

$\mathcal{R}_{\text{TEST}}$ without $a \to b$ is proved *confluent*

Related work

Mapping Maude programs into existing rewriting-based systems to investigate confluence or termination is not new.

In proofs of *operational termination* [LMM05], direct approaches, e.g., [BJM00] use term orderings to compare components of equations and rules.

Such techniques are *weaker* than state-of-the-art techniques like the DP-framework [GTS04, GTSF06].

For functional modules, operational termination has been treated as

- Termination of CS-TRSs [DLM+04, DLM+08a].
- Termination of OS-CS-TRSs [LM09].

The tool MTT [DLM08b] implements such transformations [DLM09].

Encodings into unconditional rewrite rules may lead to

- Incompleteness (e.g., when investigating termination of conditional systems by unraveling them into TRSs).
- Inefficiencies (by introducing *additional* symbols).

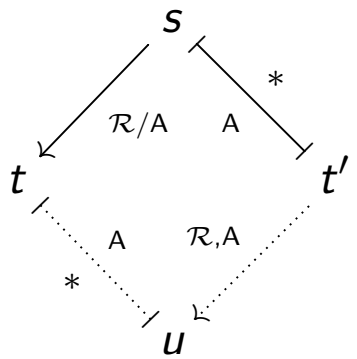
A DP-framework for proving termination of GTRSs [Luc24c], is implemented in the tool MU-TERM-GTRS [GL25]

For functional modules without axioms, confluence has been investigated [BJM00] as

- joinability of (order-sorted) conditional critical pairs and
- sort decreasingness of critical memberships.

For conditional order-sorted rewrite theories (Σ, A, R), such that

1. A is a set of *linear and regular unconditional* equations;
2. R is *strongly E-coherent* [DM12, page 819]:



3. the rules in R are strongly deterministic and
4. $\mathcal{R}$ is quasi-decreasing.

E-confluence is *characterized* as the $\widetilde{\downarrow}_{\mathcal{R},A}$-joinability of each ECCP

$$\langle \theta(\ell)[\theta(r')]_p, \theta(r) \rangle \Leftarrow \theta(c), \theta(c')$$

where $p \in \mathcal{P}os_{\mathcal{F}}(\ell)$ and $\ell|_p =^?_{A,\theta} \ell'$

The tool CRC [DM10] implements these techniques.

Conclusions and future work

E-confluence of E-terminating EGTRSs $\mathcal{R}$ can be *proved* by checking:

- $\widetilde{\downarrow}_{\mathcal{R}^{rm}}$ -joinability of
  - *Conditional Critical Pairs* in $\mathrm{CCP}(\mathcal{R})$, $\mathrm{CCP}(E,\mathcal{R})$, $\mathrm{CCP}(\mathcal{R},E)$, and
  - *Conditional Variable Pairs* in $\mathrm{CVP}^{\rightarrow}(\mathcal{R})$, $\mathrm{CVP}^{\rightarrow}(E)$, and $\mathrm{CVP}^{\vdash}(\mathcal{R})$
- $\widetilde{\downarrow}_{\mathcal{R}^{rm},E}$ -joinability of
  - *Logic-based Conditional Critical Pairs* in $\mathrm{LCCP}(\mathcal{R})$ and $\mathrm{LCCP}(E,\mathcal{R})$, and
  - *Conditional Variable Pairs* in $\mathrm{CVP}^{\overset{ps}{\rightarrow}}(\mathcal{R})$ and $\mathrm{CVP}^{\overset{ps}{\rightarrow}}(E)$

Simplifications possible depending on the *structure of equations and rules*

Our methods apply to *ETRSs* and *Conditional Rewrite Theories* as particular cases of EGTRSs

Non-E-confluence of EGTRSs can be proved as the non- $\widetilde{\downarrow}_{\mathcal{R}/E}$ -joinability of some conditional pair in $\mathrm{LCCP}(\mathcal{R}) \cup \mathrm{CVP}^{\overset{ps}{\rightarrow}}(\mathcal{R}) \cup \mathrm{DCP}(\mathcal{R})$

Implementation

Envisaged implementation in our confluence tool CONFident, including:

- Improving MU-TERM-GTRS to prove E-termination of EGTRSs
- Improving infChecker to deal with (in)feasibility proofs with EGTRSs

Theoretical developments

Conditions to use $\widetilde{\downarrow}_{\mathcal{R},E}$-joinability of CCPs instead of ECCPs or LCCPs

Develop methods to prove E-termination of EGTRSs

Use abstract results of E-confluence not requiring E-termination

Extension to other rewriting-based frameworks

- Logically-Constrained Term Rewriting Systems (LCTRSs),
- Higher-Order Rewriting,
- Nominal Rewriting

Order-Sorted EGTRSs

As for Maude functional modules

- A translation into EGTRSs is possible
- Some features are not easily captured

As for Maude system modules

- The functional module $(\Sigma, M \cup B \cup E)$ included in a system module $(\Sigma, M \cup B \cup E, R, \phi)$ is required to be confluent and terminating. This can be investigated using EGTRSs
- Encoding into EGTRSs more difficult due to the use of $\to^{!}_{\vec{E},B} \circ \to_{R,B}$

Theoretical developments

Restrictions on functional and system modules leading to faithful simulations as EGTRSs should be investigated.

Implementation

Envisaged development of a Maude confluence and (operational) termination analyzer based on CONFident and MU-TERM-GTRS

Thanks!

Confluence of Equational Generalized Term Rewriting Systems with Application to Rewrite Theories and Maude

– Tutorial –

Salvador Lucas

DSIC & VRAIN, Universitat Politècnica de València, Spain

16th International Workshop on Rewriting Logic and its Applications

WRLA 2026


Takahito Aoto, Naoki Nishida, and Jonas Schöpf.
Equational Theories and Validity for Logically Constrained Term Rewriting.
In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction, FSCD 2024, July 10-13, 2024, Tallinn, Estonia*, volume 299 of *LIPIcs*, pages 31:1–31:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.FSCD.2024.31.


Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman.
Code Optimization And Finite Church-Rosser Theorems.
In Randall Rustin, editor, *Design and Optimization of Compilers, New York, March 29-30 1971*, volume 5 of *Courant Computer Science Symposium*, pages 89–105. Prentice-Hall, 1972.


Adel Bouhoula, Jean-Pierre Jouannaud, and José Meseguer.
Specification and proof in membership equational logic.
Theor. Comput. Sci., 236(1-2):35–132, 2000.


Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott.
All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic, volume 4350 of *Lecture Notes in Computer Science*.
Springer, 2007.
doi:10.1007/978-3-540-71999-1.


Alonzo Church and J. B. Rosser.
Some Properties of Conversion.
Transactions of the American Mathematical Society, 39(3):472–482, 1936.
URL: http://www.jstor.org/stable/1989762.


Francisco Durán, Steven Eker, Santiago Escobar, Narciso Martí-Oliet, José Meseguer, Rubén Rubio, and Carolyn L. Talcott.
Programming and symbolic computation in Maude.
J. Log. Algebr. Meth. Program., 110, 2020.
doi:10.1016/j.jlamp.2019.100497.


Gilles Dowek, Thérèse Hardin, and Claude Kirchner.
Theorem Proving Modulo.
J. Autom. Reason., 31(1):33–72, 2003.
doi:10.1023/A:1027357912519.


Francisco Durán, Salvador Lucas, José Meseguer, Claude Marché, and Xavier Urbain.
Proving termination of membership equational programs.
In Nevin Heintze and Peter Sestoft, editors, *Proceedings of the 2004 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-based Program Manipulation, 2004*, pages 147–158. ACM, 2004.
doi:10.1145/1014007.1014022.


Francisco Durán, Salvador Lucas, Claude Marché, José Meseguer, and Xavier Urbain.
Proving operational termination of membership equational programs.
High. Order Symb. Comput., 21(1-2):59–88, 2008.
doi:10.1007/s10990-008-9028-2.


Francisco Durán, Salvador Lucas, and José Meseguer.
MTT: The Maude Termination Tool (System Description).
In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, volume 5195 of *Lecture Notes in Computer Science*, pages 313–319. Springer, 2008.
doi:10.1007/978-3-540-71070-7_27.


Francisco Durán, Salvador Lucas, and José Meseguer.
Methods for Proving Termination of Rewriting-based Programming Languages by Transformation.
Electr. Notes Theor. Comput. Sci., 248:93–113, 2009.


Francisco Durán and José Meseguer.
A Church-Rosser Checker Tool for Conditional Order-Sorted Equational Maude Specifications", booktitle="Rewriting Logic and Its Applications.
pages 69–85, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
doi:10.1007/978-3-642-16310-4_6.


Francisco Durán and José Meseguer.
On the Church-Rosser and coherence properties of conditional order-sorted rewrite theories.
J. Log. Algebraic Methods Program., 81(7-8):816–850, 2012.
doi:10.1016/j.jlap.2011.12.004.


Gilles Dowek.
La part du calcul (Mèmoire d'Habilitation).
1999.
URL: https://tel.archives-ouvertes.fr/tel-04114581.


Maribel Fernández, Daniele Nantes-Sobrinho, and Daniella Santaguida.
A Completion Procedure for Equational Rewriting Systems with Binders.
In Santiago Escobar and Laura Titolo, editors, *Logic-Based Program Synthesis and Transformation - 35th International Symposium, LOPSTR 2025, Proceedings*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2025.


Raúl Gutiérrez and Salvador Lucas.
MU-TERM-GTRS: A tool for proving termination of Generalized Term Rewriting Systems.
In Dieter Hofbauer and Johannes Waldmann, editors, *20th International Workshop on Termination, WST 2025*, pages 73–78, 2025.


Joseph A. Goguen and José Meseguer.
Order-sorted algebra I: equational deduction for multiple inheritance, overloading, exceptions and partial operations.
Theor. Comput. Sci., 105(2):217–273, 1992.
doi:10.1016/0304-3975(92)90302-V.


Jürgen Giesl, René Thiemann, and Peter Schneider-Kamp.
The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs.
In Franz Baader and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 11th International Conference, LPAR 2004, Proceedings*, volume 3452 of *Lecture Notes in Computer Science*, pages 301–331. Springer, 2004.
doi:10.1007/978-3-540-32275-7_21.


Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke.
Mechanizing and Improving Dependency Pairs.
J. Autom. Reasoning, 37(3):155–203, 2006.
doi:10.1007/s10817-006-9057-7.


James Roger Hindley.
The Church-Rosser Property and a result in Combinatory Logic.
PhD thesis, University of Newcastle upon Tyne, July 1964.
URL: http://www.ens-lyon.fr/LIP/REWRITING/MISC/these-roger-hindley.pdf.


J. Roger Hindley.
An Abstract Form of the Church-Rosser Theorem. I.
J. Symb. Log., 34(4):545–560, 1969.
doi:10.1017/S0022481200128439.


Gérard P. Huet.
Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems.
In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 30–45. IEEE Computer Society, 1977.
doi:10.1109/SFCS.1977.9.


Gérard P. Huet.
Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems.
J. ACM, 27(4):797–821, 1980.
doi:10.1145/322217.322230.


Jean-Pierre Jouannaud and Hélène Kirchner.
Completion of a Set of Rules Modulo a Set of Equations.
SIAM J. Comput., 15(4):1155–1194, 1986.
doi:10.1137/0215084.


Jean-Pierre Jouannaud.
Confluent and Coherent Equational Term Rewriting Systems: Application to Proofs in Abstract Data Types.
In Giorgio Ausiello and Marco Protasi, editors, *CAAP'83, Trees in Algebra and Programming, 8th Colloquium, Proceedings*, volume 159 of *Lecture Notes in Computer Science*, pages 269–283. Springer, 1983.

doi:10.1007/3-540-12727-5_16.


Donald E. Knuth and Peter E. Bendix.
Simple Word Problems in Universal Algebra.
In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.


Dallas S. Lankford.
Canonical algebraic simplification in computational logic.
Technical report, University of Texas, Austin, TX, May 1975.
report ATP-25.


Dallas S. Lankford and A. Michael Ballantine.
Decision procedures for simple equational theories with permutative axioms: complete sets of permutative reductions.
Technical report, University of Texas, Austin, TX, April 1977.
report ATP-37.


Salvador Lucas and José Meseguer.
Operational Termination of Membership Equational Programs: the Order-Sorted Way.
Electr. Notes Theor. Comput. Sci., 238(3):207–225, 2009.
doi:10.1016/j.entcs.2009.05.021.


Salvador Lucas, Claude Marché, and José Meseguer.
Operational termination of conditional term rewriting systems.
Inf. Process. Lett., 95(4):446–453, 2005.


Salvador Lucas.
Confluence of Conditional Rewriting Modulo.
In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:21, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
doi:10.4230/LIPIcs.CSL.2024.37.


Salvador Lucas.
Local confluence of conditional and generalized term rewriting systems.
Journal of Logical and Algebraic Methods in Programming, 136:paper 100926, pages 1–23, 2024.
doi:10.1016/j.jlamp.2023.100926.


Salvador Lucas.
Termination of Generalized Term Rewriting Systems.
In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, volume 299 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
doi:10.4230/LIPIcs.FSCD.2024.29.


Salvador Lucas.
Semantic Properties of Computations Defined by Elementary Inference Systems.
Electronic Proceedings in Theoretical Computer Science, 434:10–26, October 2025.
doi:10.4204/eptcs.434.4.


Salvador Lucas.
Confluence of conditional rewriting modulo.
CoRR, abs/2504.01847v3, 2026.
arXiv:2504.01847, doi:10.48550/ARXIV.2504.01847.


José Meseguer.
Conditional rewriting logic as a unified model of concurrency.
Theor. Comput. Sci., 96(1):73–155, 1992.
doi:10.1016/0304-3975(92)90182-F.


José Meseguer.
Twenty years of rewriting logic.
J. Log. Algebr. Program., 81(7-8):721–781, 2012.
doi:10.1016/j.jlap.2012.06.003.


José Meseguer.
Strict coherence of conditional rewriting modulo axioms.
Theor. Comput. Sci., 672:1–35, 2017.
doi:10.1016/J.TCS.2016.12.026.


M. H. A. Newman.
On Theories with a Combinatorial Definition of "Equivalence".
Annals of Mathematics, 43(2):pp. 223–243, 1942.
URL: http://www.jstor.org/stable/1968867.


Gerald E. Peterson and Mark E. Stickel.
Complete Sets of Reductions for Some Equational Theories.
J. ACM, 28(2):233–264, 1981.
doi:10.1145/322248.322251.


Barry K. Rosen.
Tree-Manipulating Systems and Church-Rosser Theorems.
In Patrick C. Fischer, Robert Fabian, Jeffrey D. Ullman, and Richard M. Karp, editors, *Proceedings of the 2nd Annual ACM Symposium on Theory of Computing, May 4-6, 1970, Northampton, Massachusetts, USA*, pages 117–127. ACM, 1970.
doi:10.1145/800161.805157.


George A. Robinson and Larry Wos.
Paramodulation and theorem-proving in first-order theories with equality.
Machine Intelligence, 4:135–150, 1969.


G. Robinson and L. Wos.
Paramodulation and Theorem-Proving in First-Order Theories with Equality, pages 298–313.
Springer Berlin Heidelberg, Berlin, Heidelberg, 1983.
doi:10.1007/978-3-642-81955-1_19.


Ravi Sethi.
Testing for the Church-Rosser Property.
J. ACM, 21(4):671–679, 1974.
doi:10.1145/321850.321862.


James R. Slagle.
Automated Theorem-Proving for Theories with Simplifiers Commutativity, and Associativity.
J. ACM, 21(4):622–642, 1974.
doi:10.1145/321850.321859.