Formalising Confluence

Mauricio Ayala-Rincón

Joint work with: Ana Cristina Rocha Oliveira & André Luiz Galdino

Universidade de Brasília, Brazil

Research funded by CNPq and CAPES

Developments in Computational Models - DCM 2015

PU Javeriana, Cali, 28 October 2015

Table of contents

Universidade de Brasília

Confluence

- Related with *non-ambiguity* or *determinism* of processes.
  - As termination, confluence is undecidable.
- Criteria:
  - Under termination: Newman's Lemma, Knuth-Bendix(-Huet) Critical Pairs Lemma
  - Without termination: orthogonality.
- Analytical proofs
  - CP criterion: Knuth-Bendix (1969), Huet (1980).
  - Confluence of orthogonal systems: Rosen (1973).

  $\Rightarrow$ Further, several styles of proof were given as surveyed in TeRese textbook.

Universidade de Brasília

Knuth-Bendix Critical Pair criterion

$$R: \quad \begin{array}{lcl} \text{odd} + \text{even} \rightarrow \text{even} + \text{odd} & \qquad & \text{even} + \text{odd} \rightarrow \text{odd} \\ \text{odd} + \text{odd} \rightarrow \text{even} + \text{even} + \text{even} & \qquad & \text{even} + \text{even} \rightarrow \text{even} \end{array}$$

Since it's terminating (why?), it's only necessary to prove all its *critical* divergences are joinable:

$$\begin{tikzcd} & \text{odd} + \text{even} + \text{even} \arrow[dl, "R"'] \arrow[dr, "R"] & \\ \text{even} + \text{odd} + \text{even} & & \text{odd} + \text{even} \end{tikzcd}$$

Knuth-Bendix Critical Pair criterion

$$
\begin{array}{llll}
R: & \text{odd} + \text{even} \to \text{even} + \text{odd} & & \text{even} + \text{odd} \to \text{odd} \\
 & \text{odd} + \text{odd} \to \text{even} + \text{even} + \text{even} & & \text{even} + \text{even} \to \text{even}
\end{array}
$$

Since it's terminating (why?), it's only necessary to prove all its *critical* divergences are joinable:

$$
\begin{tikzcd}
 & \text{odd} + \text{even} + \text{even} \arrow[dl, "R"'] \arrow[dr, "R"] & \\
\text{even} + \text{odd} + \text{even} & & \text{odd} + \text{even}
\end{tikzcd}
$$

Knuth-Bendix Critical Pair criterion

$$R: \quad \text{odd} + \text{even} \rightarrow \text{even} + \text{odd} \qquad \text{even} + \text{odd} \rightarrow \text{odd}$$
$$\text{odd} + \text{odd} \rightarrow \text{even} + \text{even} + \text{even} \qquad \text{even} + \text{even} \rightarrow \text{even}$$

Since it's terminating (why?), it's only necessary to prove all its *critical* divergences are joinable:

$$\begin{array}{ccc} & \text{odd} + \text{even} + \text{even} & \\ \swarrow_{R} & & \searrow^{R} \\ \text{even} + \text{odd} + \text{even} & & \text{odd} + \text{even} \end{array}$$

Universidade de Brasília

Knuth-Bendix Critical Pair criterion

$$R: \quad \begin{array}{ll} odd + even \to even + odd & even + odd \to odd \\ odd + odd \to even + even + even & even + even \to even \end{array}$$

Since it's terminating (why?), it's only necessary to prove all its *critical* divergences are joinable:

$$\begin{tikzcd} & odd + even + even \arrow[dl, "R"'] \arrow[dr, "R"] & \\ even + odd + even \arrow[rr, "R"] & & odd + even \end{tikzcd}$$

Universidade de Brasília

Confluence through Orthogonality

- *Functional programs* can be viewed as **orthogonal** TRSs:
  - Left linear
  - Without critical pairs

Confluence through Orthogonality

$$
\begin{array}{l}
Ack(0, n) \rightarrow s(n) \\
Ack(s(m), 0) \rightarrow Ack(m, s(0)) \\
Ack(s(m), s(n)) \rightarrow Ack(m, Ack(s(m), n))
\end{array}
$$

Knuth-Bendix Critical Pairs criterion implies confluence, since it is terminating (Why?).
Well, also orthogonality implies confluence.

$$
\begin{array}{l}
Ack'(0, n) \rightarrow s(n) \\
Ack'(s(m), 0) \rightarrow Ack'(m, s(0)) \\
Ack'(s(m), s(n)) \rightarrow Ack(Ack(m, s(n)), n)
\end{array}
$$

Knuth-Bendix CP criterion does not applies. It is not terminating.
But, by orthogonality, it is also confluent.

Confluence - undecidability

(Tseiten 1956) The semigroup over the alphabet $\Sigma = \{a, b, c, d, e\}$ with congruence given by equations below has a **undecidable** *word problem*:

$$E = \left\{ \begin{array}{ll} ac = ca, & ad = da, \\ bc = cb, & bd = db, \\ ce = eca, & de = edb, \\ cdca = cdcae & \end{array} \right\}$$

- For two words $u, v \in \Sigma^*$, the question $u =_E v$? is undecidable.
- "$\to_E$", defined as the symmetric closure of E, is confluent.
- For $u, v \in \Sigma^*$, to decide if $\to_{uv} \; = \; \to_E \cup \{i \to u, i \to v\}$ is confluent corresponds to decide if $u =_E v$:

$$u \downarrow_{uv} v \textbf{ iff } u =_E v$$

Universidade de Brasília

Rewriting notation

Given T and a binary relation $\rightarrow \subseteq T \times T$.

$a \rightarrow b$		$(a, b) \in \rightarrow$.
$\leftarrow$		the inverse of $\rightarrow$: $b \leftarrow a$ iff $a \rightarrow b$.
$\leftrightarrow$		denotes $\leftarrow \cup \rightarrow$.
$\rightarrow^{=}$	denote	reflexive closure of $\rightarrow$: $\rightarrow \cup =$.
$\rightarrow^{*}$		reflexive transitive closure of $\rightarrow$.
$\leftrightarrow^{*}$		equivalence closure of $\rightarrow$.
$\downarrow$		joinability: $\rightarrow^{*} \circ {}^{*}\!\leftarrow$.

Local Confluence		$\leftarrow \circ \rightarrow \; \subseteq \; \downarrow \; (= \; \rightarrow^{*} \circ {}^{*}\!\leftarrow)$
Confluence	are defined as	${}^{*}\!\leftarrow \circ \rightarrow^{*} \; \subseteq \; \downarrow \; (= \; \rightarrow^{*} \circ {}^{*}\!\leftarrow)$
Church-Rosser		$\leftrightarrow^{*} \; \subseteq \; \downarrow \; (= \; \rightarrow^{*} \circ {}^{*}\!\leftarrow)$
Termination		$\neg\exists \; : \; a_0 \rightarrow a_1 \rightarrow \cdots$

Rewriting notation

Abstract reduction relations are extended to terms in a signature $\Sigma(T, V)$. Given a relation on terms R, the induced (term) rewriting relation $\to_R$ is given by

$$s \to_R t \textbf{ iff } \begin{cases} \exists (l, r) \in R, \\ s|_\pi = l\sigma \\ t = s[\pi \leftarrow r\sigma] \end{cases}$$

where,

- $s|_\pi$ denotes the subterm of s at position π and
- $s[\pi \leftarrow r\sigma]$ the term resulting from replacing the subterm at position π of s by $r\sigma$.

Universidade de Brasília

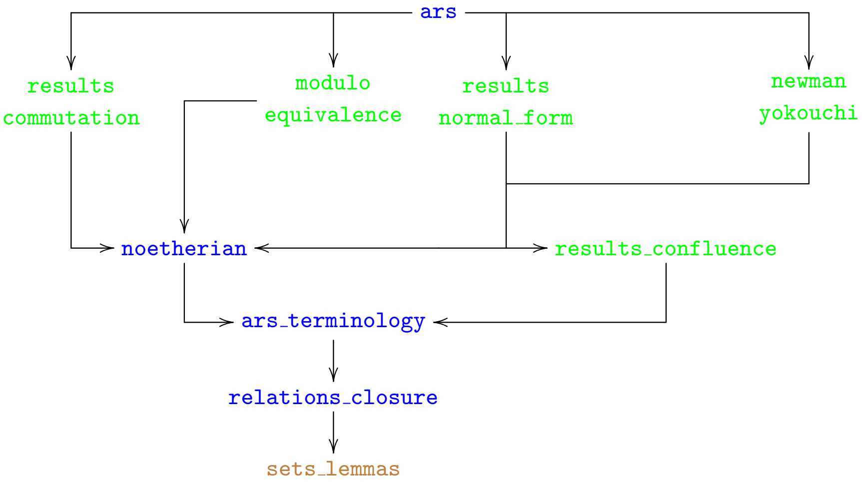
The Prototype Verification System - PVS

PVS is a verification system, developed by the SRI International Computer Science Laboratory, which consists of

1. a specification language:
   - based on higher-order logic;
   - a type system based on Church's simple theory of types augmented with subtypes and dependent types.
2. an interactive theorem prover:
   - based on sequent calculus; that is, goals in PVS are sequents of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences of formulae, with the usual Gentzen semantics.

Universidade de Brasília

Hierarchy of the ars theory



Available: NASA LaRC PVS library or trs.cic.unb.br.

Universidade de Brasília

ARS specification - Relations

- An ARS $(A, \rightarrow)$ is specified as
  - a uninterpreted type `T` and
  - a binary relation `R` that is a predicate:

```
PRED: TYPE = [[T,T] -> bool]
```

- Closure relations are specified using the `iterate` function. For example the reflexive-transitive closure

$$\rightarrow^* = \cup_{n \geq 0} \rightarrow^n$$

is specified in the ars theory as

```
RTC(R): reflexive_transitive =
                    IUnion(LAMBDA n:  iterate(R, n))
```

Universidade de Brasília

ARS specification - Rewriting Properties

Specifying other properties

```
joinable?(R)(x,y): bool = EXISTS z: RTC(R)(x,z) & RTC(R)(y, z)

church_rosser?(R): bool = FORALL x,y:
                           EC(R)(x,y) => joinable?(R)(x,y)

confluent?(R): bool = FORALL x,y,z:
                       RTC(R)(x,y) & RTC(R)(x,z)
                        =>
                          joinable?(R)(y,z)

commute?(R1,R2): bool = FORALL x,y,z:
                         RTC(R1)(x,y) & RTC(R2)(x,z)
                          =>
                            EXISTS r: RTC(R2)(y,r) & RTC(R1)(z,r)
```

Newman's Lemma

R noetherian

$$
\begin{tikzcd}
\cdot \arrow[r, "R"] \arrow[d, "R"'] \arrow[dr, phantom, "LC"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}
\quad \Leftrightarrow \quad
\begin{tikzcd}
\cdot \arrow[r, "R^*"] \arrow[d, "R^*"'] \arrow[dr, phantom, "C"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}
$$

LocalConfluent *Confluent*

Newman's Lemma Specification

```
Newman_lemma: THEOREM noetherian?(R) =>
                     (confluent?(R) <=> local_confluent?(R))
```

Proof: By noetherian induction with the predicate

$$P(x) = \forall y, z.\ y \ {}^*\!\!\leftarrow x \rightarrow^* z \Rightarrow y \downarrow z$$

Newman's Lemma

R noetherian

$$
\begin{tikzcd}
\cdot \arrow[r, "R"] \arrow[d, "R"'] \arrow[dr, phantom, "LC"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}
\quad \Leftrightarrow \quad
\begin{tikzcd}
\cdot \arrow[r, "R^*"] \arrow[d, "R^*"'] \arrow[dr, phantom, "C"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}
$$

LocalConfluent *Confluent*

Newman's Lemma Specification

```
Newman_lemma: THEOREM noetherian?(R) =>
                      (confluent?(R) <=> local_confluent?(R))
```

Proof: By noetherian induction with the predicate

$$P(x) = \forall y, z.\ y \ {}^*\!\!\leftarrow x \rightarrow^* z \Rightarrow y \downarrow z$$

Newman's Lemma

In the `ars` *theory* properties are formalised in an *"almost diagramatic style"* as it is desirable in rewriting theory.

Geometric sketch of Newman's Lemma formalisation

$$
\begin{tikzcd}
{} \arrow[r, "R"] \arrow[d, "R"'] \arrow[dr, phantom, "LC"] & {} \arrow[r, "R^*"] \arrow[d, dotted, "R^*"] \arrow[ddr, phantom, "IH"] & {} \arrow[dd, dotted, "R^*"] \\
{} \arrow[r, dotted, "R^*"'] \arrow[d, "R^*"'] \arrow[dr, phantom, "IH"] & {} \arrow[d, dotted, "R^*"] & \\
{} \arrow[r, dotted, "R^*"'] & {} \arrow[r, dotted, "R^*"'] & {}
\end{tikzcd}
$$

Yokouchi's Lemma

Specification

```
Yokouchi_lemma: THEOREM
   (noetherian?(R) & confluent?(R) & diamond_property?(S) &
    (FORALL x,y,z: S(x,y) & R(x,z) =>
       EXISTS (u:T): RTC(R)(y,u) & (RTC(R) o S o RTC(R))(z,u)))
    => diamond_property?(RTC(R) o S o RTC(R))
```

R noetherian

$$
\begin{tikzcd}
\cdot \arrow[r, "R^*"] \arrow[d, "R^*"'] \arrow[dr, phantom, "C"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}
\quad \& \quad
\begin{tikzcd}
\cdot \arrow[r, "S"] \arrow[d, "S"'] \arrow[dr, phantom, "\Diamond"] & \cdot \arrow[d, dotted, "S"] \\
\cdot \arrow[r, dotted, "S"'] & \cdot
\end{tikzcd}
\quad \& \quad
\begin{tikzcd}
\cdot \arrow[r, "S"] \arrow[d, "R"'] \arrow[dr, phantom, "D"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^* \circ S \circ R^*"'] & \cdot
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
\cdot \arrow[r, "R^* \circ S \circ R^*"] \arrow[d, "R^* \circ S \circ R^*"'] \arrow[dr, phantom, "\Diamond"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] \\
\cdot \arrow[r, dotted, "R^* \circ S \circ R^*"'] & \cdot
\end{tikzcd}
$$

Universidade de Brasília

Yokouchi's Lemma

Generalisation of D as D'

```
Yokouchi_lemma_ax1: LEMMA
  (noetherian?(R) & confluent?(R) &
  (FORALL x,y,z: S(x,y) & R(x,z) =>
     EXISTS (u:T): RTC(R)(y,u) & (RTC(R) o S o RTC(R))(z,u)))
    => (FORALL x,y,z: S(x,y) & RTC(R)(x,z) =>
        EXISTS (w:T): RTC(R)(y,w) & (RTC(R) o S o RTC(R))(z,w))
```

$$\begin{tikzcd} \cdot \arrow[r, "R"] \arrow[d, "S"'] \arrow[dr, phantom, "D"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] \\ \cdot \arrow[r, dotted, "R^*"'] & \cdot \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} \cdot \arrow[r, "R^*"] \arrow[d, "S"'] \arrow[dr, phantom, "D'"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] \\ \cdot \arrow[r, dotted, "R^*"'] & \cdot \end{tikzcd}$$

Proof: By noetherian induction with the predicate

$$P(x) := \forall y, z.\ xR^*z \ \wedge\ xSy \ \Rightarrow\ \exists u.(yR^*u \ \wedge\ zR^* \circ S \circ R^* u)$$

Yokouchi's Lemma

Generalisation of D as D'

```
Yokouchi_lemma_ax1: LEMMA
  (noetherian?(R) & confluent?(R) &
  (FORALL x,y,z: S(x,y) & R(x,z) =>
     EXISTS (u:T): RTC(R)(y,u) & (RTC(R) o S o RTC(R))(z,u)))
    => (FORALL x,y,z: S(x,y) & RTC(R)(x,z) =>
         EXISTS (w:T): RTC(R)(y,w) & (RTC(R) o S o RTC(R))(z,w))
```

$$\begin{tikzcd} \cdot \arrow[r, "R"] \arrow[d, "S"'] \arrow[dr, phantom, "D"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] & & \cdot \arrow[r, "R^*"] \arrow[d, "S"'] \arrow[dr, phantom, "D'"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] \\ \cdot \arrow[r, dotted, "R^*"'] & \cdot & \Rightarrow & \cdot \arrow[r, dotted, "R^*"'] & \cdot \end{tikzcd}$$

Proof: By noetherian induction with the predicate

$$P(x) := \forall y, z.\ xR^*z \ \wedge \ xSy \ \Rightarrow \ \exists u.(yR^*u \ \wedge \ zR^* \circ S \circ R^* u)$$

Yokouchi's Lemma

Proof

Then, to prove that $R^* \circ S \circ R^*$ has the diamond property, one also proceeds by noetherian induction but this time using the predicate

$$\begin{array}{rl} P'(x) := & \forall y, z.\ xR^* \circ S \circ R^* y \ \wedge\ xR^* \circ S \circ R^* z \\ & \Rightarrow\ \exists u.(yR^* \circ S \circ R^* u \ \wedge\ zR^* \circ S \circ R^* u) \end{array}$$

One distinguishes between the cases

$$R^0 \circ S \circ R^*$$

and

$$R^+ \circ S \circ R^*$$

Yokouchi's Lemma

Geometric Sketch: Case $R^0 \circ S \circ R^*$

$$\begin{tikzcd}
\cdot \arrow[r, "S"] \arrow[d, "S"'] \arrow[dr, phantom, "\Diamond"] & \cdot \arrow[r, "R^*"] \arrow[d, dotted, "S"] \arrow[dr, phantom, "D'"] & \cdot \arrow[d, dotted, "R^* \circ S \circ R^*"] \\
\cdot \arrow[r, dotted, "S"'] \arrow[d, "R^*"'] \arrow[dr, phantom, "D'"] & \cdot \arrow[r, dotted, "R^*"'] \arrow[d, dotted, "R^*"] \arrow[dr, phantom, "C"] & \cdot \arrow[d, dotted, "R^*"] \\
\cdot \arrow[r, dotted, "R^* \circ S \circ R^*"'] & \cdot \arrow[r, dotted, "R^*"'] & \cdot
\end{tikzcd}$$

Yokouchi's Lemma

Geometric Sketch: Case $R^{+} \circ S \circ R^{*}$

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
{} \arrow[r, "R^*"] \arrow[d, "R"'] \arrow[dr, phantom, "C"] & {} \arrow[r, "S"] \arrow[d, dotted, "R^*"'] \arrow[dr, phantom, "D'"] & {} \arrow[r, "R^*"] \arrow[d, dotted, "R^*"] \arrow[dr, phantom, "C"] & {} \arrow[d, dotted, "R^*"] \\
{} \arrow[r, dotted, "R^*"'] \arrow[d, "R^*\circ S\circ R^*"'] & {} \arrow[r, dotted, "R^*\circ S\circ R^*"'] \arrow[d, phantom, "IH"] & {} \arrow[r, dotted, "R^*"'] & {} \arrow[d, dotted, "R^*\circ S\circ R^*"] \\
{} \arrow[rrr, dotted, "R^*\circ S\circ R^*"'] & & & {}
\end{tikzcd}
$$

Hierarchy of the trs *theory*

$$
\begin{tikzcd}
\texttt{reduction} \arrow[d] & \texttt{critical\_pairs} \arrow[l] & \texttt{trs} \arrow[l] \\
\texttt{rewrite\_rules} \arrow[r] & \texttt{substitution} \arrow[d] & \texttt{orthogonality} \arrow[ul] \\
\texttt{replacement} \arrow[d] & \texttt{compatibility} \arrow[l] \arrow[r] & \texttt{ars[term]} \\
\texttt{subterm} \arrow[r] & \texttt{positions} \arrow[r] \arrow[d] & \texttt{finite\_sequences\_extras} \arrow[d] \\
 & \texttt{term} \arrow[r] & \texttt{finite\_sequences}
\end{tikzcd}
$$

Available: NASA LaRC PVS library or `trs.cic.unb.br`.

Universidade de Brasília

TRS specification - Terms

The set of terms

```
term[variable: TYPE+, symbol: TYPE+] : DATATYPE
BEGIN

IMPORTING arity[symbol]

vars(v: variable): vars?

app(f:symbol,
    args:{args:finite_sequence[term] | length(args)=arity(f)}): app?

END term
```

TRS specification - Other key basic concepts

Positions and Subterms

- The *set of positions* of the term t, denoted by $Pos(t)$, is inductively defined as follows:
  (a) If $t = x \in V$, then $Pos(t) := \epsilon$, where ϵ denotes the empty string.
  (b) If $t = f(t_1, \cdots, t_n)$, then

$$Pos(t) := \{\epsilon\} \cup \bigcup_{i=1}^{n} \{ip \mid p \in Pos(t_i)\}$$

- The *subterm* of a term s at position $p \in Pos(s)$, denoted by $s|_p$, is inductively defined on the length of p as follows:

$$\begin{array}{rcl} s|_\epsilon & := & s \\ f(s_1, \ldots, s_n)|_{iq} & := & s_i|_q \end{array}$$

Universidade de Brasília

TRS specification - Replacement

Replacing the subterm of s at position $p \in Pos(s)$ by t: $s[p \leftarrow t]$

```
replaceTerm(t: term, s: term, (p: positions?(s))): RECURSIVE term =
    (IF length(p) = 0
     THEN t
     ELSE LET st = args(s),
               i = first(p),
               q = rest(p),
             rst = replace(replaceTerm(t, st(i-1), q), st,i-1) IN
          app(f(s), rst)
     ENDIF)
  MEASURE length(p)
```

Usefull properties

Let s, t, r be terms. If p and q are parallel positions in s, then

(a) $s[p \leftarrow t]|_q = s|_q$ persistence

(b) $s[p \leftarrow t][q \leftarrow r] = s[q \leftarrow r][p \leftarrow t]$ commutativity

Universidade de Brasília

TRS specification - Substitution and Renaming

Substitution

(a) The substitutions are built as functions from variables to terms

```
sig:  [V -> term]
```

whose domain is finite:

```
Sub?(sig):  bool = is_finite(Dom(sig))
```

(b) The homomorphic extension `ext(sig)` of a substitution `sig` is specified inductively over the structure of terms.

Renaming

```
Ren?(sig): bool = subset?(Ran(sig),V) &
                  (bijective?[(Dom(sig)),(Ran(sig))])(sig)
```

Universidade de Brasília

TRS specification - Rewrite Rules and Reduction Relation

Rewrite Rules

```
rewrite_rule?(l,r): bool = (NOT vars?(l)) & subset?(Vars(r), Vars(l))

rewrite_rule: TYPE = (rewrite_rule?)
```

Reduction Relation

```
reduction?(E)(s,t): bool =
   EXISTS ( (e | member(e, E)), sig, (p: positions?(s)) ):
                subtermOF(s, p) = ext(sig)(lhs(e)) &
                             t = replaceTerm(ext(sig)(rhs(e)), s, p)
```

Lemma

Let `E` be a set of rewrite rules. The reduction relation `reduction?(E)` is closed under substitutions and compatible with operations (structure of terms).

TRS specification - Critical Pairs

Critical Pairs - Analytic Definition

Let $l_i \to r_i$, $i = 1, 2$, be two rules whose "variables have been renamed" such that $Var(l_1) \cap Var(l_2) = \emptyset$. Let $p \in Pos(l_1)$ be such that $l_1|_p$ is not a variable and let $\sigma = mgu(l_1|_p, l_2)$. This determines a *critical pair* $\langle t_1, t_2 \rangle$:

$$\begin{array}{rcl} t_1 & = & \sigma(r_1) \\ t_2 & = & \sigma(l_1)[p \leftarrow \sigma(r_2)] \end{array}$$

Critical Pairs - Specification

```
CP?(E)(t1, t2): bool =
 EXISTS (sigma, rho, (e1 | member(e1, E)), (e2p | member(e2p, E)),
       (p: positions?(lhs(e1)))):
        LET e2 = (# lhs := ext(rho)(lhs(e2p)),
                    rhs := ext(rho)(rhs(e2p)) #) IN
        disjoint?(Vars(lhs(e1)),Vars(lhs(e2)))                    &
        NOT vars?(subtermOF(lhs(e1), p))                          &
        mgu(sigma)(subtermOF(lhs(e1), p), lhs(e2))                &
        t1 = ext(sigma)(rhs(e1))                                  &
        t2 = replaceTerm(ext(sigma)(rhs(e2)), ext(sigma)(lhs(e1)), p)
```

Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

Specification

```
CP_Theorem: THEOREM
 FORALL E:
          local_confluent?(reduction?(E))
                        <=>
(FORALL t1, t2: CP?(E)(t1, t2) => joinable?(reduction?(E))(t1,t2))
```

Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation

Let s be a term of divergence such that

$$\begin{tikzcd} & s \arrow[dl, "l_1 \to r_1"'] \arrow[dr, "l_2 \to r_2"] & \\ s_1 & & s_2 \end{tikzcd}$$

that is, there are positions $p_1, p_2 \in \texttt{positions?}(s)$, rules $l_1 \to r_1,\ l_2 \to r_2 \in \mathrm{E}$, and substitutions σ_1, σ_2, such that

$$s|_{p_1} = \sigma_1(l_1) \qquad \& \qquad s_1 = s[p_1 \leftarrow \sigma_1(r_1)]$$

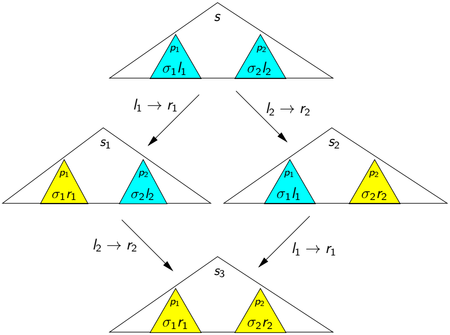
$$s|_{p_2} = \sigma_2(l_2) \qquad \& \qquad s_2 = s[p_2 \leftarrow \sigma_2(r_2)]$$

Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation: Disjoint positions

p_1 and p_2 are in separate subtrees, i.e., p_1 and p_2 are parallel positions in s.

Case 1: Disjoint positions



Case 1: Disjoint positions

- Persistence
- Commutativity

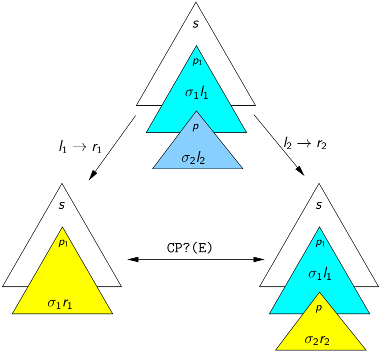
Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation: Critical overlap

$p \in \texttt{positions?}(l_1)$, $l_1|_p$ is not a variable and $\sigma_1(l_1|_p) = \sigma_2(l_2)$.

Case 2: Either $p_1 \leq p_2$ or $p_2 \leq p_1$ - $p_2 = p_1 p$



Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

Case 2: The divergence corresponds to an instance of a critical pair $\langle t_1, t_2 \rangle$

```
CP_lemma_aux1: LEMMA
 FORALL E, (e1 | member(e1, E)), (e2 | member(e2, E)), (p: position):
  positionsOF(lhs(e1))(p)                                              &
  NOT vars?(subtermOF(lhs(e1), p))                                     &
  ext(sg1)(subtermOF(lhs(e1), p)) = ext(sg2)(lhs(e2))
=>
  EXISTS t1, t2, delta:
  CP?(E)(t1, t2)                                                       &
  ext(delta)(t1) = ext(sg1)(rhs(e1))                                   &
  ext(delta)(t2) = replaceTerm(ext(sg2)(rhs(e2)), ext(sg1)(lhs(e1)), p)
```

Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

In general the critical overlap case is proved in textbooks by assuming that the rewriting rules $l_1 \to r_1$ and $l_2 \to r_2$ are renamed such that $\mathtt{Vars}(l_1) \cap \mathtt{Vars}(l_2) = \emptyset$.

Case 2: Auxiliary properties

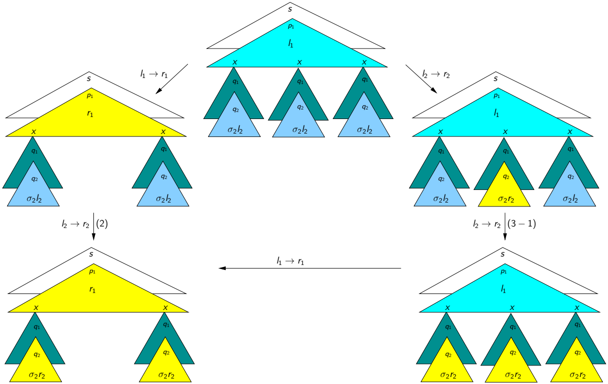
```
CP_lemma_aux1a: LEMMA
 FORALL E, (e1 | member(e1, E)), (e2 | member(e2, E)), (p: position):
  positionsOF(lhs(e1))(p)                                           &
  NOT vars?(subtermOF(lhs(e1), p))                                  &
  ext(sg1)(subtermOF(lhs(e1), p)) = ext(sg2)(lhs(e2)) )
=>
  EXISTS alpha, rho:
  disjoint?(Vars(lhs(e1)), Vars(ext(rho)(lhs(e2))))                 &
  ext(sg1)(subtermOF(lhs(e1), p)) = ext(comp(alpha, rho))(lhs(e2))
```

Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation: Non-critical overlap

$p = q_1 q_2$, for q_2 possibly empty, such that q_1 is a position of variable in l_1 and $\sigma_2(l_2) = \sigma_1(l_1|_{q_1})|_{q_2}$.



Knuth-Bendix Critical Pair Theorem

Case 3: Auxiliary lemma

Let $\rightarrow$ be a relation compatible with the structure of terms, x be a variable, and σ_1 and σ_2 be substitutions such that:

$$\begin{array}{lcl} \sigma_1(x) & \rightarrow & \sigma_2(x) \text{ and} \\ \sigma_1(y) & = & \sigma_2(y), \text{ for all } y \neq x. \end{array}$$

Let t be an arbitrary term, and $p_1, \ldots, p_n \in \texttt{positions?}(t)$ be all the occurrences of x in t. Define $t_0 = \sigma_1(t)$ and $t_i = t_{i-1}[p_i \leftarrow \sigma_2(x)]$, for $1 \leq i \leq n$. Then $t_i \rightarrow^{n-i} \sigma_2(t)$, for $0 \leq i \leq n$. In particular, $\sigma_1(t) \rightarrow^n \sigma_2(t)$.

Case 3: Auxiliary constructors

```
replace_pos(t, s, (fssp:SPP(s)) ): RECURSIVE term =
    IF length(fssp) = 0 THEN  s
    ELSE replace_pos(t,replaceTerm(t, s, fssp(0)), rest(fssp)) ENDIF
  MEASURE length(fssp)

RSigma(R, sg1, sg2, x): bool = FORALL (y: (V)):
   IF y /= x THEN sg1(y) = sg2(y) ELSE R(sg1(x), sg2(x)) ENDIF
```

Universidade de Brasília

Knuth-Bendix Critical Pair Theorem

Case 3: The variable $l_1|_{q_1}$ can occur repeatedly in both sides of the rule $l_1 \to r_1$

```
CP_lemma_aux2: LEMMA
 FORALL R, t, x, sg1, sg2:
  LET Posv = Pos_var(t, x),
      seqv = set2seq(Posv) IN
  comp_cont?(R) & RSigma(R, sg1, sg2, x)
=>
  FORALL (i: below[length(seqv)]):
   RTC(R)(replace_pos(ext(sg2)(x),ext(sg1)(t), #(seqv(i))),ext(sg2)(t))
                                    &
                  RTC(R)(ext(sg1)(t), ext(sg2)(t))
```

Universidade de Brasília

The PVS theory orthogonality

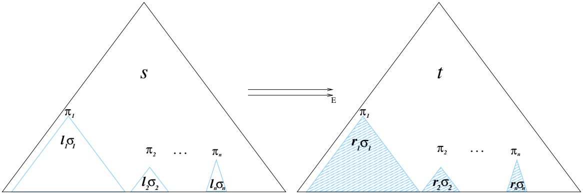
- The PVS theory `orthogonality` substantially enlarges the theory `trs` including several notions and formalisations related with the specification of orthogonal TRSs.

$\Rightarrow$ `orthogonality` includes a formalisation of the theorem of confluence of orthogonal TRSs according to:

  - use of the parallel reduction relation and
  - an inductive construction of terms of joinability for parallel divergences through the Parallel Moves Lemma.

Available: NASA LaRC PVS library or `trs.cic.unb.br`.

Parallel Rewriting



$$s \rightrightarrows t$$

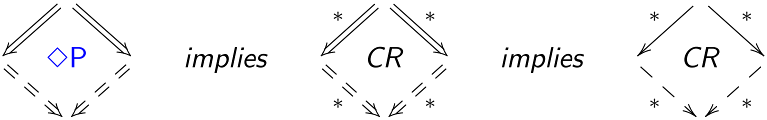
```
⇉(E)(s,t) : bool = ∃ (Π : SPP(t1), Γ : Seq[E], Σ :
Seq[Subs]) :    ...
                  t = replace_par_pos(s, Π, sigma_rhs(Σ,Γ))
```

Universidade de Brasília

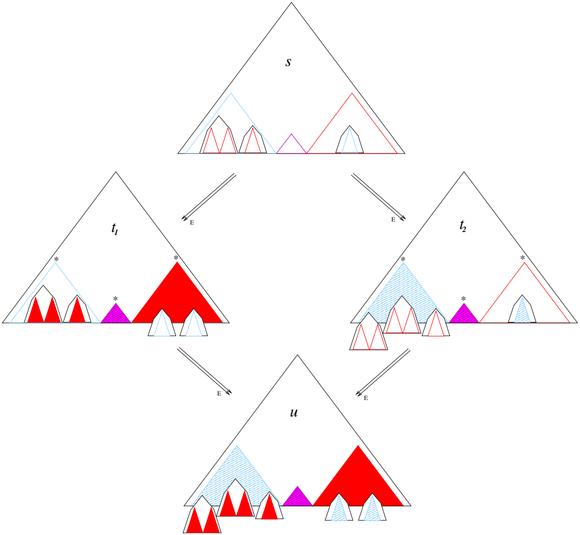
Theorem [Confluence of Orthogonal TRSs]
Orthogonality $\Rightarrow$ confluence

One has to prove:

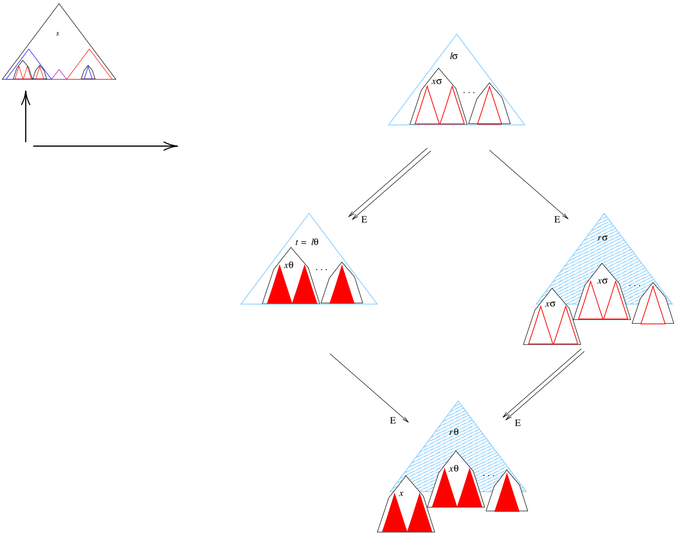
- the **diamond property ($\Diamond$P) for $\rightrightarrows$**;
- $\rightarrow \;\subseteq\; \rightrightarrows \;\subseteq\; \rightarrow^*$ implies $\rightrightarrows^* \;\equiv\; \rightarrow^*$;
- $\rightrightarrows$ confluent, implies $\rightarrow$ confluent.



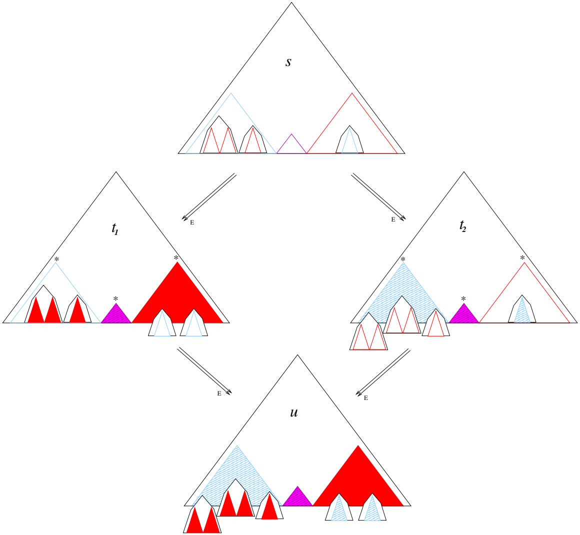
`Orthogonal?(E) => diamond_property?(parallel_reduction?(E))`



Building the joinability term: the Parallel Moves Lemma



Joinability requires synchronised applications of PML



Universidade de Brasília

Formalisation: Orthogonal_implies_confluent

Lemma (Specification of Orthogonality implies Confluence)

```
Orthogonal_implies_confluent:  LEMMA
     FORALL (E : Orthogonal) :
        confluent?(reduction?(E))
```

Formalisation: parallel_reduction_has_DP

Lemma (Specification of Orthogonality of $\rightarrow$ implies $\diamond$P of $\rightrightarrows$)

```
parallel_reduction_has_DP: LEMMA
     Orthogonal?(E) =>
          diamond_property?(⇉(E))
```

Formalisation: divergence_in_Pos_Over

divergence_in_Pos_Over: LEMMA

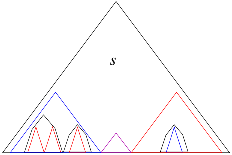
$\rightrightarrows$(E)(s,t1,Π_1) $\wedge$ $\rightrightarrows$(E)(s,t2,Π_2) $\wedge$ $\pi \in \mathsf{Pos\_Over}(\Pi_1, \Pi_2)$

=>

LET Π = complement_pos(π, Π_2) IN
$\exists$ ($(l, r) \in$ E , σ) :
subtermOF(s, π) = $l\sigma$ $\wedge$
subtermOF(t1, π) = $r\sigma$ $\wedge$
$\rightrightarrows$(E)(subtermOF(s, π),subtermOF(t2,π), Π)

Universidade de Brasília

Formalisation: subterm_joinability



subterm_joinability: LEMMA

Orthogonal?(E) $\wedge$ $\rightrightarrows$(E)(s,t1,Π_1) $\wedge$ $\rightrightarrows$(E)(s,t2,Π_2) $\wedge$
Π = Pos_Over(Π_1,Π_2) o Pos_Over(Π_2,Π_1) o Pos_Equal(Π_1,Π_2)

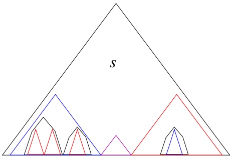
=>

$\forall i <\mid \Pi \mid$:

$\exists u_i$: $\rightrightarrows$(E)(subtermOF(t1, Π(i)), u_i) $\wedge$

$\rightrightarrows$(E)(subtermOF(t2, Π(i)), u_i)

Formalisation: subterms_joinability



subterms_joinability: LEMMA

Orthogonal?(E) $\wedge$ $\rightrightarrows$(E)(s,t1,Π_1) $\wedge$ $\rightrightarrows$(E)(s,t2,Π_2) $\wedge$
Π = Pos_Over(Π_1,Π_2) o Pos_Over(Π_2,Π_1) o Pos_Equal(Π_1,Π_2)

=>

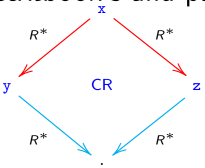
$\exists$U : | U | = | Π | $\wedge$

$\forall$ i : $\rightrightarrows$(E)(subtermOF(t1, Π(i)), U(i)) $\wedge$

$\rightrightarrows$(E)(subtermOF(t2, Π(i)), U(i))

Conclusion and Future Work

- `trs` and `orthogonality` provide elegant formalisations close to textbook's and paper's proofs.



$$\texttt{confluent?(R): bool} = \forall(\texttt{ x, y, z}):$$
$$\rightarrow^*\texttt{(R)(x,y)} \wedge \rightarrow^*\texttt{(R)(x,z)}$$
$$\texttt{=>} \downarrow\texttt{(R)(y,z)}$$

$\Rightarrow$ First straightforward complete formalisation of Knuth-Bendix CP Th.
$\Rightarrow$ A complete formalisation of Rosen's confluence of orthogonal TRS's.

- Precise discrimination of notions and properties:
  - $\diamond$ property implies non termination.
  - proof's analogies fail: a whole new development of parallel rewriting concepts was necessary to formalise confluence of orthogonal TRS's.
- Clarity about adaptation of results in other contexts: confluence in *nominal rewriting*.

Universidade de Brasília

Conclusion and Future Work

- Applications to certify confluence of orthogonal specifications, variants of lambda calculus, nominal rewriting.
- Adaptation of the proof in Takahashi's style.
- Formalisations using other styles of proof. Van Oostrom's developments, for instance.

Universidade de Brasília

References

M. Ayala-Rincón, M. Fernández, M. J. Gabbay, and A. C. Rocha Oliveira.
Checking Overlaps of Nominal Rewriting Rules.
In *Pre-proc. Logical and Semantic Frameworks with Applications LSFA*, 2015.

M. Ayala-Rincón, M. Fernández, and A. C. Rocha Oliveira.
Completeness in PVS of a Nominal Unification Algorithm.
In *Pre-proc. Logical and Semantic Frameworks with Applications LSFA*, 2015.

A. L. Galdino and M. Ayala-Rincón.
A Formalization of Newman's and Yokouchi Lemmas in a Higher-Order Language.
J. of Formalized Reasoning, 1(1):39–50, 2008.

A. L. Galdino and M. Ayala-Rincón.
A Formalization of the Knuth-Bendix(-Huet) Critical Pair Theorem.
J. of Automated Reasoning, 45(3):301–325, 2010.

A. C. Rocha Oliveira and M. Ayala-Rincón.
Formalizing the confluence of orthogonal rewriting systems.
CoRR, abs/1303.7335, 2013.

Universidade de Brasília