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
1 Motivation

2 The Prototype Verification System - PVS

3 ars - A PVS theory for abstract reduction systems
   • Formalisations: Newman’s and Yokouchi’s Lemmas

4 trs - A PVS theory for term rewriting systems
   • Formalisation: Knuth-Bendix Critical Pair Theorem

5 orthogonality - Parallel rewriting
   • Formalisation: Confluence of Orthogonal TRSs

6 Conclusion and Future Work
Motivation

Confluence

- Related with *non-ambiguity or determinism* of processes.
  - As termination, confluence is undecidable.
- Criteria:
  - Without termination: orthogonality.
- Analytical proofs
  ⇒ Further, several styles of proof were given as surveyed in TeRese textbook.
Knuth-Bendix Critical Pair criterion

\[ R : \begin{align*}
    \text{odd} + \text{even} & \to \text{even} + \text{odd} \\
    \text{even} + \text{odd} & \to \text{odd} \\
    \text{odd} + \text{odd} & \to \text{even} + \text{even} + \text{even} \\
    \text{even} + \text{even} & \to \text{even}
\end{align*} \]

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

\[ \text{odd} + \text{even} + \text{even} \]

\[ \text{even} + \text{odd} + \text{even} \]

\[ \text{odd} + \text{even} \]
Motivation

Knuth-Bendix Critical Pair criterion

\[ R : \]
\[ \text{odd} + \text{even} \rightarrow \text{even} + \text{odd} \]
\[ \text{odd} + \text{odd} \rightarrow \text{even} + \text{even} + \text{even} \]
\[ \text{even} + \text{odd} \rightarrow \text{odd} \]
\[ \text{even} + \text{even} \rightarrow \text{even} \]

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

\[ \text{odd} + \text{even} + \text{even} \]
\[ \text{even} + \text{odd} + \text{even} \]
\[ \text{odd} + \text{even} \]

M. Ayala-Rincón  DCM 2015
Knuth-Bendix Critical Pair criterion

\[ R : \quad \begin{align*} 
\text{odd} + \text{even} &\rightarrow \text{even} + \text{odd} \\
\text{odd} + \text{odd} &\rightarrow \text{even} + \text{even} + \text{even} \\
\text{even} + \text{odd} + \text{even} &\rightarrow \text{odd}
\end{align*} \]

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

\[ \begin{align*} 
\text{odd} + \text{even} + \text{even} &\rightarrow \text{even} + \text{odd} + \text{even} \\
\text{even} + \text{odd} + \text{even} &\rightarrow \text{odd} + \text{even}
\end{align*} \]
Knuth-Bendix Critical Pair criterion

\[ R : \quad \text{odd} + \text{even} \rightarrow \text{even} + \text{odd} \quad \text{even} + \text{odd} \rightarrow \text{odd} \]

\[ \text{odd} + \text{odd} \rightarrow \text{even} + \text{even} + \text{even} \quad \text{even} + \text{even} \rightarrow \text{even} \]

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

\[ \text{odd} + \text{even} + \text{even} \]

\[ \text{even} + \text{odd} + \text{even} \rightarrow \text{odd} + \text{even} \]
Confluence through Orthogonality

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

\[
\begin{align*}
\text{Ack}(0, n) & \rightarrow s(n) \\
\text{Ack}(s(m), 0) & \rightarrow \text{Ack}(m, s(0)) \\
\text{Ack}(s(m), s(n)) & \rightarrow \text{Ack}(m, \text{Ack}(s(m), n))
\end{align*}
\]

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

\[
\begin{align*}
\text{Ack}'(0, n) & \rightarrow s(n) \\
\text{Ack}'(s(m), 0) & \rightarrow \text{Ack}'(m, s(0)) \\
\text{Ack}'(s(m), s(n)) & \rightarrow \text{Ack}(\text{Ack}(m, s(n)), n)
\end{align*}
\]

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:

\[
\begin{align*}
E &= \left\{ 
\begin{array}{ll}
ac &= ca, & ad &= da, \\
bc &= cb, & bd &= db, \\
ce &= eca, & de &= edb, \\
\end{array}
\right. \\
\text{cdca} &= \text{cdcae}
\end{align*}
\]

- For two words \( u, v \in \Sigma^* \), the question \( u =_E v \) is undecidable.
- \( \rightarrow_E \), defined as the symmetric closure of \( E \), is confluent.
- For \( u, v \in \Sigma^* \), to decide if \( \rightarrow_{uv} = \rightarrow_E \cup \{ i \rightarrow u, i \rightarrow v \} \) is confluent corresponds to decide if \( u =_E v \):

\[ u \downarrow_{uv} v \iff u =_E v \]
Rewriting notation

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

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

Local Confluence, Confluence, Church-Rosser, and Termination are defined as:

\[ \leftarrow \circ \rightarrow \subseteq \downarrow \quad (= \rightarrow^* \circ \leftrightarrow^*) \]
\[ \leftrightarrow^* \circ \rightarrow^* \subseteq \downarrow \quad (= \rightarrow^* \circ \leftrightarrow^*) \]
\[ \leftrightarrow^* \subseteq \downarrow \quad (= \rightarrow^* \circ \leftrightarrow^*) \]
\[-\exists : a_0 \rightarrow a_1 \rightarrow \cdots \]
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 $\rightarrow_R$ is given by

\[
\exists (l, r) \in R, \quad s|_\pi = l\sigma \\
t = s[\pi \leftarrow r\sigma]
\]

where,

- $s|_\pi$ denotes the subterm of $s$ at position $\pi$ and
- $s[\pi \leftarrow r\sigma]$ the term resulting from replacing the subterm at position $\pi$ of $s$ by $r\sigma$. 
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 $\Gamma$ and $\Delta$ are finite sequences of formulae, with the usual Gentzen semantics.
Hierarchy of the ars theory

Available: NASA LaRC PVS library or trs.cic.unb.br.
ARS specification - Relations

- An ARS $(A, \rightarrow)$ is specified as
  - a uninterpreted type $T$ and
  - a binary relation $R$ that is a predicate:
    \[
    \text{PRED: TYPE} = \{[[T, T] \rightarrow \text{bool}]\}
    \]

- Closure relations are specified using the \textit{iterate} function. For example the reflexive-transitive closure
  \[
  \rightarrow^* = \bigcup_{n \geq 0} \rightarrow^n
  \]
  is specified in the \texttt{ars} theory as
  \[
  \text{RTC(R): reflexive\_transitive} = \text{IUnion(LAMBDA n: iterate(R, n))}
  \]
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

![Diagram](image)

**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

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
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))
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))

Proof: By noetherian induction with the predicate
$P(x) := \forall y, z. \ xR^*z \land \ xSy \Rightarrow \exists u. (yR^*u \land zR^* \circ S \circ R^*u)$
Yokouchi’s Lemma

Generalisation of $D$ as $D'$

Yokouchi’s Lemma

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))

Proof: By noetherian induction with the predicate

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

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

One distinguishes between the cases

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

and

$$R^+ \circ S \circ R^*$$
Yokouchi’s Lemma

Geometric Sketch: Case $R^0 \circ S \circ R^*$
Yokouchi’s Lemma

Geometric Sketch: Case $R^+ \circ S \circ R^*$
Hierarchy of the trs theory

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

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

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

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

\[
\begin{align*}
s|_\epsilon & := s \\
f(s_1, \ldots, s_n)|_{iq} & := s_i|_q
\end{align*}
\]
TRS specification - Replacement

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

```pvs
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) \( s[p \leftarrow t]|_q = s|_q \)  \hspace{2cm} \text{persistence}
(b) \( s[p \leftarrow t][q \leftarrow r] = s[q \leftarrow r][p \leftarrow t] \)  \hspace{2cm} \text{commutativity}
TRS specification - Substitution and Renaming

Substitution

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

\[ \text{sig: } [V \rightarrow \text{term}] \]

whose domain is finite:

\[ \text{Sub?(sig): bool } = \text{is_finite(Dom(sig))} \]

(b) The homomorphic extension \( \text{ext(sig)} \) of a substitution \( \text{sig} \) is specified inductively over the structure of terms.

Renaming

\[ \text{Ren?(sig): bool } = \text{subset?(Ran(sig),V) } \& \]

\[ (\text{bijective?[[\text{Dom(sig)},(\text{Ran(sig))}]]})(\text{sig}) \]
TRS specification - Rewrite Rules and Reduction Relation

Rewrite Rules

\[
\text{rewrite_rule?(l,r)}: \text{bool} = (\text{NOT vars?(l)}) \& \text{subset?}(\text{Vars(r)}, \text{Vars(l)})
\]

\[
\text{rewrite_rule}: \text{TYPE} = (\text{rewrite_rule?})
\]

Reduction Relation

\[
\text{reduction?}(E)(s,t): \text{bool} = \\
\text{EXISTS } ( (e \mid \text{member}(e, E)), \text{sig}, (p: \text{positions?}(s)) ) : \\
\text{subtermOF}(s, p) = \text{ext}(\text{sig})(\text{lhs}(e)) \& \\
t = \text{replaceTerm}(\text{ext}(\text{sig})(\text{rhs}(e)), s, p)
\]

Lemma

Let \( E \) be a set of rewrite rules. The reduction relation \( \text{reduction?}(E) \) is closed under substitutions and compatible with operations (structure of terms).
Critical Pairs - Analytic Definition

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

\[
\begin{align*}
t_1 & = \sigma(r_1) \\
t_2 & = \sigma(l_1)[p \leftarrow \sigma(r_2)]
\end{align*}
\]

Critical Pairs - Specification

\[
\text{CP?}(E)(t_1, t_2): \text{bool} =
\begin{align*}
& \exists \sigma, \rho, (e_1 \mid \text{member}(e_1, E)), (e_2p \mid \text{member}(e_2p, E)), \\
& \quad (p: \text{positions?}(\text{lhs}(e_1))): \\
& \quad \text{LET } e_2 = (# \text{ lhs } := \text{ext}(\rho)(\text{lhs}(e_2p)), \\
& \quad \quad \text{ rhs } := \text{ext}(\rho)(\text{rhs}(e_2p)) #) \text{ IN} \\
& \quad \text{disjoint?}(\text{Vars}(\text{lhs}(e_1)), \text{Vars}(\text{lhs}(e_2))) \quad \& \\
& \quad \text{NOT} \text{ vars?}(\text{subtermOF}(\text{lhs}(e_1), p)) \quad \& \\
& \quad \text{mgu}(\sigma)(\text{subtermOF}(\text{lhs}(e_1), p), \text{lhs}(e_2)) \quad \& \\
& \quad t_1 = \text{ext}(\sigma)(\text{rhs}(e_1)) \quad \& \\
& \quad t_2 = \text{replaceTerm}(\text{ext}(\sigma)(\text{rhs}(e_2)), \text{ext}(\sigma)(\text{lhs}(e_1)), p)
\end{align*}
\]
Knuth-Bendix Critical Pair Theorem

Specification

CP_Theorem: THEOREM

\[
\text{FORALL } E: \\
\quad \text{local_confluent?(reduction?(E))} \\
\quad \iff \\
\quad \text{(FORALL } t_1, t_2: \text{CP?(E)}(t_1, t_2) \Rightarrow \text{joinable?(reduction?(E)}(t_1, t_2))
\]
Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation

Let $s$ be a term of divergence such that

\[
\begin{align*}
  l_1 & \rightarrow r_1 \\
  l_2 & \rightarrow r_2 \\
  s_1 & \leftarrow s[p_1 \leftarrow \sigma_1(r_1)] \\
  s_2 & \leftarrow s[p_2 \leftarrow \sigma_2(r_2)]
\end{align*}
\]

that is, there are positions $p_1, p_2 \in \text{positions}(s)$, rules $l_1 \rightarrow r_1, l_2 \rightarrow r_2 \in \mathcal{E}$, and substitutions $\sigma_1, \sigma_2$, such that

\[
\begin{align*}
  s|_{p_1} &= \sigma_1(l_1) & & & & s_1 &= s[p_1 \leftarrow \sigma_1(r_1)] \\
  s|_{p_2} &= \sigma_2(l_2) & & & & s_2 &= s[p_2 \leftarrow \sigma_2(r_2)]
\end{align*}
\]
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

- Persistence
- Commutativity
Knuth-Bendix Critical Pair Theorem

A sketch of the formalisation: Critical overlap

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

Case 2: Either $p_1 \leq p_2$ or $p_2 \leq p_1$ - $p_2 = p_1 \rho$
Knuth-Bendix Critical Pair Theorem

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

\[\text{CPlemma_aux1: LEMMA}\]

\[\begin{align*}
&\text{FORALL } E, (e1 \mid \text{member}(e1, E)), (e2 \mid \text{member}(e2, E)), (p: \text{position}): \\
&\quad \text{positionsOF}(\text{lhs}(e1))(p) \\
&\quad \text{NOT vars?}(\text{subtermOF}(\text{lhs}(e1), p)) \\
&\quad \text{ext}(sg1)(\text{subtermOF}(\text{lhs}(e1), p)) = \text{ext}(sg2)(\text{lhs}(e2)) \\
&\Rightarrow \\
&\quad \text{EXISTS } t_1, t_2, \delta: \\
&\quad \text{CP?}(E)(t_1, t_2) \\
&\quad \text{ext}(\delta)(t_1) = \text{ext}(sg1)(\text{rhs}(e1)) \\
&\quad \text{ext}(\delta)(t_2) = \text{replaceTerm}(\text{ext}(sg2)(\text{rhs}(e2)), \text{ext}(sg1)(\text{lhs}(e1)), p)
\end{align*}\]
In general the critical overlap case is proved in textbooks by assuming that the rewriting rules $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are renamed such that $\text{Vars}(l_1) \cap \text{Vars}(l_2) = \emptyset$.

Case 2: Auxiliary properties

\[ \text{CP\_lemma\_aux1a: LEMMA} \]
\[
\text{FORALL } E, (e_1 \mid \text{member}(e_1, E)), (e_2 \mid \text{member}(e_2, E)), (p: \text{position}): \\
\text{positionsOF}(\text{lhs}(e_1))(p) \\
\text{NOT vars?}(\text{subtermOF}(\text{lhs}(e_1), p)) \\
\text{ext}(\text{sg1})(\text{subtermOF}(\text{lhs}(e_1), p)) = \text{ext}(\text{sg2})(\text{lhs}(e_2)) \\
\Rightarrow \\
\text{EXISTS } \alpha, \rho: \\
\text{disjoint?}(\text{Vars}(\text{lhs}(e_1)), \text{Vars}(\text{ext}(\rho)(\text{lhs}(e_2)))) \\
\text{ext}(\text{sg1})(\text{subtermOF}(\text{lhs}(e_1), p)) = \text{ext}(\text{comp}(\alpha, \rho))(\text{lhs}(e_2)) \]
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 $\sigma_1$ and $\sigma_2$ be substitutions such that:

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

Let $t$ be an arbitrary term, and $p_1, \ldots, p_n \in \text{positions?}(t)$ 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

```plaintext
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)
```

```plaintext
RSigma(R, sg1, sg2, x): bool = FORALL (y: (V)):
    IF y /= x THEN sg1(y) = sg2(y) ELSE R(sg1(x), sg2(x)) ENDIF
```
Knuth-Bendix Critical Pair Theorem

Case 3: The variable \( l_1|q_1 \) can occur repeatedly in both sides of the rule \( l_1 \rightarrow r_1 \)

**CP_lemma_aux2**: LEMMA

\[
\text{FORALL } R, t, x, sg1, sg2:
\text{LET } Posv = Pos\_var(t, x),
\text{seqv = set2seq(Posv) IN}
\text{comp\_cont?(R) \& RSigma(R, sg1, sg2, x)}
\Rightarrow
\text{FORALL } (i: \text{below}[\text{length(seqv)]}):
\text{RTC(R)(replace\_pos(ext(sg2)(x), ext(sg1)(t), #(seqv(i))), ext(sg2)(t))}
\text{\&}
\text{RTC(R)(ext(sg1)(t), ext(sg2)(t))}
\]
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.

⇒ 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

\[ \Rightarrow(E)(s, t) : \text{bool} = \exists (\Pi : \text{SPP}(t_1), \Gamma : \text{Seq}[E], \Sigma : \text{Seq}[\text{Subs}]) : \ldots \]

\[ t = \text{replace-par-pos}(s, \Pi, \text{sigma-rhs}(\Sigma, \Gamma)) \]
Theorem [Confluence of Orthogonal TRSs]
Orthogonality $\Rightarrow$ confluence

One has to prove:
- the **diamond property** ($\Diamond P$) for $\Rightarrow$;
- $\rightarrow \subseteq \Rightarrow \subseteq \rightarrow^* \Rightarrow \rightarrow^*$ implies $\Rightarrow^* \equiv \rightarrow^*$;
- $\Rightarrow$ 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
Lemma (Specification of Orthogonality implies Confluence)

Orthogonal_implies_confluent: LEMMA

\[
\text{FORALL (E : Orthogonal) :}
\]

\[
\text{confluent?(reduction?(E))}
\]
Lemma (Specification of Orthogonality of $\rightarrow$ implies $\Diamond P$ of $\Rightarrow$)

parallel_reduction_has_DP: LEMMA

Orthogonal?(E) =>

diamond_property?(\Rightarrow(E))
Formalisation: divergence_in_Pos_Over

**divergence_in_Pos_Over:** LEMMA

\[ \Rightarrow (E)(s,t_1,\Pi_1) \land \Rightarrow (E)(s,t_2,\Pi_2) \land \pi \in Pos\_Over(\Pi_1, \Pi_2) \]

\[ \Rightarrow \]

\[ \text{LET } \Pi = \text{complement\_pos}(\pi, \Pi_2) \text{ IN} \]

\[ \exists ((l,r) \in E, \sigma) : \]

\[ \text{subtermOF}(s, \pi) = l\sigma \land \]

\[ \text{subtermOF}(t_1, \pi) = r\sigma \land \]

\[ \Rightarrow (E)(\text{subtermOF}(s, \pi),\text{subtermOF}(t_2,\pi), \Pi) \]
Formalisation: subterm_joinability

**subterm_joinability:** LEMMA

\[
\text{Orthogonal?}(E) \land \Rightarrow (E)(s, t_1, \Pi_1) \land \Rightarrow (E)(s, t_2, \Pi_2) \land \\
\Pi = \text{Pos}_\text{Over}(\Pi_1, \Pi_2) \circ \text{Pos}_\text{Over}(\Pi_2, \Pi_1) \circ \text{Pos}_\text{Equal}(\Pi_1, \Pi_2) \\
\Rightarrow \\
\forall i < |\Pi| : \\
\exists u_i : \Rightarrow (E)(\text{subtermOF}(t_1, \Pi(i)), u_i) \land \\
\Rightarrow (E)(\text{subtermOF}(t_2, \Pi(i)), u_i)
\]
subterms_joinability:  LEMMA

Orthogonal?(E) \land \Rightarrow(E)(s,t_1,\Pi_1) \land \Rightarrow(E)(s,t_2,\Pi_2) \land 
\Pi = \text{Pos}_\text{Over}(\Pi_1,\Pi_2) \circ \text{Pos}_\text{Over}(\Pi_2,\Pi_1) \circ \text{Pos}_\text{Equal}(\Pi_1,\Pi_2) 
=>

\exists U: |U| = |\Pi| \land 
\forall i : \Rightarrow(E)(\text{subtermOF}(t_1, \Pi(i)), U(i)) \land 
\Rightarrow(E)(\text{subtermOF}(t_2, \Pi(i)), U(i))
Conclusion and Future Work

- **trs** and **orthogonality** provide elegant formalisations close to textbook’s and paper’s proofs.

\[
\text{confluent?(R): bool = } \forall (x, y, z) : \rightarrow^*(R)(x, y) \land \rightarrow^*(R)(x, z) \\
\Rightarrow \downarrow(R)(y, z)
\]

⇒ First straightforward complete formalisation of Knuth-Bendix CP Th.
⇒ A complete formalisation of Rosen’s confluence of orthogonal TRS’s.

- Precise discrimination of notions and properties:
  - ◊ 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**.
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.
References

Checking Overlaps of Nominal Rewriting Rules.
In *Pre-proc. Logical and Semantic Frameworks with Applications LSFA*, 2015.

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.

A. L. Galdino and M. Ayala-Rincón.
A Formalization of the Knuth-Bendix(-Huet) Critical Pair Theorem.

A. C. Rocha Oliveira and M. Ayala-Rincón.
Formalizing the confluence of orthogonal rewriting systems.