Synchronizing Applications of the Parallel Moves Lemma To Formalize Confluence of Orthogonal TRSs in PVS

Ana Cristina Rocha Oliveira Valverde, André Luiz Galdino and Mauricio Ayala-Rincón

Universidade de Brasília and Universidade Federal de Goiás
Braziliië

Research funded by CNPq and CAPES

2^{nd} Int. Workshop on Confluence - IWC 2013

TU/e, 28 juni 2013

Universidade de Brasília

Table of contents

Universidade de Brasília

Orthogonality

- Functional programs can be viewed as **orthogonal** TRSs:
  - Left linear
  - Without critical pairs
- Related with non-ambiguity in functional programming and specification.
- Important in confluence without termination.

Analytical proofs

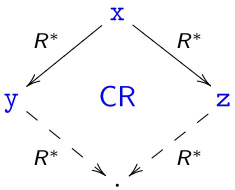
- A first proof of confluence of orthogonal rewriting systems was published by Rosen (1973).
- Further, several styles of proof were given as surveyed in TeRese textbook.

Previous work

Galdino and Ayala-Rincón developed the PVS theory `trs` 2007-10, available as part of the NASA LaRC PVS libraries.

- `trs` provides specified notions and formalized properties used to proof elaborated theorems about TRS's in a natural way as it's done in textbooks on rewriting (Baader&Nipkow).

⇒ The first straightforwardly complete formalization of Knuth-Bendix-Huet CP theorem is inside `trs`.



confluent?(R): bool =
∀(x, y, z):
$\rightarrow^*$(R)(x,y) ∧ $\rightarrow^*$(R)(x,z)
=> ↓(R)(y,z)

Universidade de Brasília

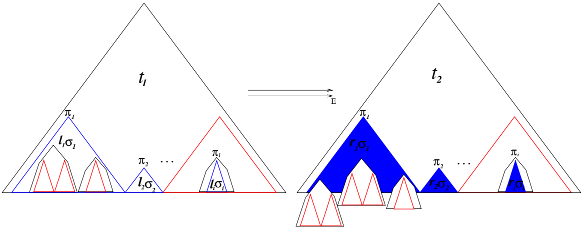
The PVS theory `orthogonality`

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

⇒ `orthogonality` includes a formalization 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.

Parallel Rewriting



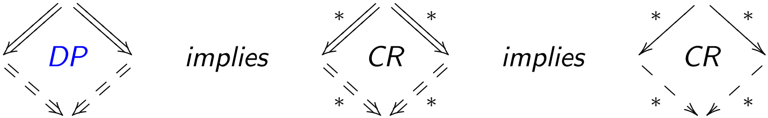
$$t_1 \rightrightarrows t_2$$

$\rightrightarrows (E)(t1, t2) : bool = \exists(\Pi : SPP(t1), \Gamma : Seq[E], \Sigma : Seq[Subs]) :$
`t2 = replace_par_pos(t1, Π, sigma_rhs(Σ,Γ))`

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

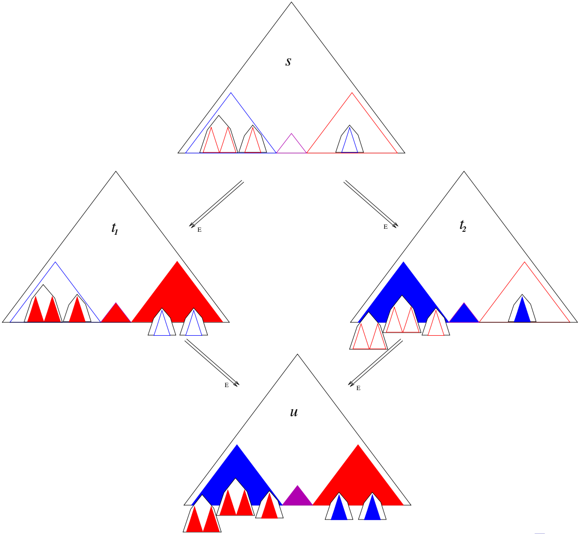
One has to prove:

- the $\diamond$ property for $\rightrightarrows$;
- $\rightarrow \subset \rightrightarrows \subset \rightarrow^*$ implies $\rightrightarrows^* \equiv \rightarrow^*$;
- $\rightrightarrows$ confluent, implies $\rightarrow$ confluent.

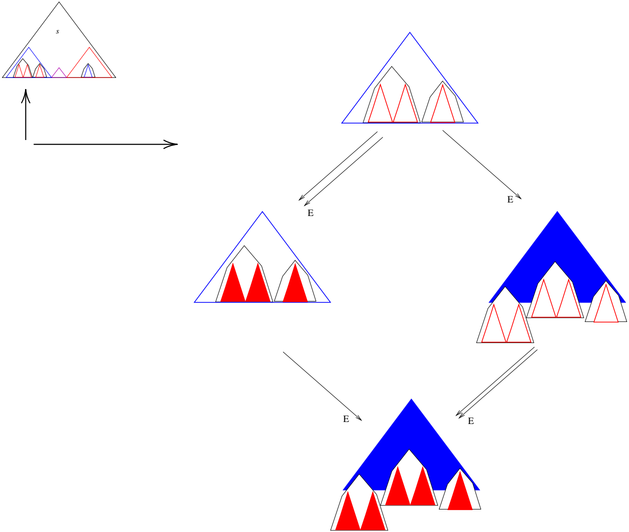


Universidade de Brasília

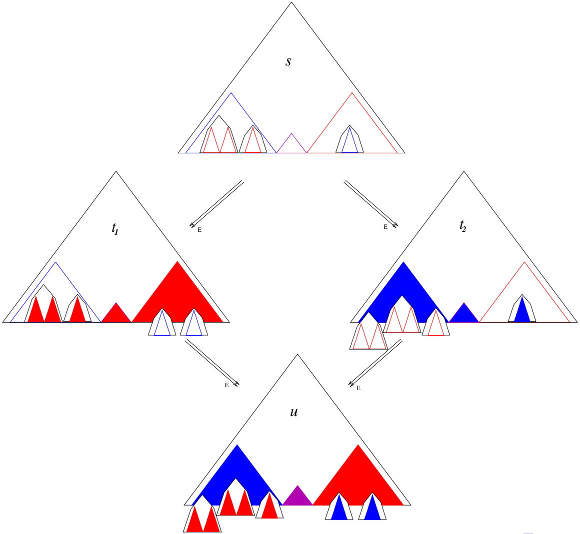
Orthogonal?(E) => diamond_property?(parallel_reduction?(E))



Building the joinability term: the Parallel Moves Lemma



Joinability requires synchronized applications of PML



Formalization: Orthogonal_implies_confluent

Lemma (Specification of Orthogonality implies Confluence)

```
Orthogonal_implies_confluent:  LEMMA

      FORALL (E : Orthogonal) :

         confluent?(reduction?(E))
```

Universidade de Brasília

Formalization: Orthogonal_implies_confluent

```
Orthogonal_implies_confluent :
    [-1] →*(E)(x, y)
    [-2] →*(E)(x, z)
       |-------
    {1} ∃ (z1: term): →*(E)(y, z1) ∧ →*(E)(z, z1)
```

Universidade de Brasília

Formalization: Orthogonal_implies_confluent

```
Orthogonal_implies_confluent :

  {-1} ⇉*(E)(y, z1)
  {-2} ⇉*(E)(z, z1)
  [-3] strong_confluent?(⇉(E))
  [-4] diamond_property?(⇉(E))
  [-5] →*(E) = ⇉*(E)
  [-6] →*(E)(x, y)
  [-7] →*(E)(x, z)
    |-------
  [1] →*(E)(y, z1)   ∧   →*(E)(z, z1)
```

Formalization: parallel_reduction_has_DP

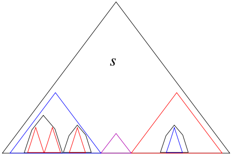
Lemma (Specification of Orthogonality of $\rightarrow$ implies $\diamond$P of $\rightrightarrows$)

```
parallel_reduction_has_DP: LEMMA

      Orthogonal?(E) =>

            diamond_property?(⇉(E))
```

Formalization: subterms_joinability



subterms_joinability: LEMMA

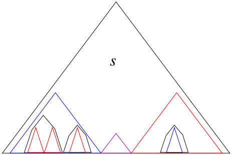
Orthogonal?(E) $\wedge$ $\rightrightarrows$(E)(t,t1,Π_1) $\wedge$ $\rightrightarrows$(E)(t,t2,Π_2) $\wedge$
Π = Pos_Over(Π_1,Π_2) o Pos_Over(Π_2,Π_1) o Pos_Equal(Π_1,Π_2)

=>

$\exists T$: $|T| = |\Pi|$ $\wedge$
$\forall i$: $\rightrightarrows$(E)(subtermOF(t1, Π(i)), T(i)) $\wedge$
$\rightrightarrows$(E)(subtermOF(t2, Π(i)), T(i))

Universidade de Brasília

Formalization: subterm_joinability



subterm_joinability: LEMMA

Orthogonal?(E) $\wedge$ $\rightrightarrows$(E)(t,t1,Π_1) $\wedge$ $\rightrightarrows$(E)(t,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)

=>

$\forall i <| \Pi |$:
$\exists s$: $\rightrightarrows$(E)(subtermOF(t1, Π(i)), s) $\wedge$
$\rightrightarrows$(E)(subtermOF(t2, Π(i)), s)

Formalization: divergence_in_Pos_Over

```
divergence_in_Pos_Over:  LEMMA
⇉(E)(t,t1,Π₁) ∧ ⇉(E)(t,t2,Π₂) ∧  π ∈ Pos_Over(Π₁, Π₂))
=>
        LET Π = complement_pos(π, Π₂) IN
          ∃( (l,r)∈ E , σ )  :
          subtermOF(t, π) = lσ ∧
          subtermOF(t1, π) = rσ ∧
          ⇉(E)(subtermOF(t, π),subtermOF(t2,π), Π)
```

Quantitative data: specification vs Formalization

- Specification: 787 lines/31K
  - ◇ (Contribution to PVS theory `structures`)
- Formalization: 55.077 lines/41M.

The majority of the effort is related with proving mundane but essential properties, as usual.

Conclusion and Future Work

- Contributions for the PVS theory `trs` including parallel rewriting.
- `Orthogonality` contains a formalization of confluence of orthogonal TRS's.
- It uses induction via synchronization of applications of the PML.
- As far as we know, there exist only other formalization in `IsaFoR` by R. Thiemann (IWS'12).

Universidade de Brasília

Conclusion and Future Work

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

References

Theory trs, (consulted March 2013): *Available in the NASA LaRC PVS library*, http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/.

Franz Baader and Tobias Nipkow, *Term rewriting and All That*, Cambridge University Press, 1998.

M. Bezem, J.W. Klop, and R. de Vrijer, *Term rewriting systems by TeReSe*, Cambridge Tracts in Theoretical Computer Science, no. 55, Cambridge University Press, 2003.

A. L. Galdino and M. Ayala-Rincón, *A formalisation of newman's and yokouchi lemmas in a higher-order language*, Journal of Formal Reasonning **1** (2008), no. 1, 39–50.

____________, *A PVS theory for term rewriting systems*, Electronic Notes in Theoretical Computer Science **247** (2009), 67–83, Third Workshop on Logical and Semantic Frameworks with Applications - LSFA 2008.

____________, *A formalisation of the Knut-Bendix(-Huet) critical pair theorem*, Journal of Automated Reasonning **45** (2010), no. 3, 301–325.

R. Thiemann, *Certification of confluence proofs using CeTA*, First International Workshop on Confluence (IWC 2012), p. 45.

Universidade de Brasília