Rewriting techniques, 6: orthogonality, decreasing diagrams

19–12–2019

Exercise 1:
Are the rewrite relations of the following TRSs confluent?
(a) \( a(x, x) \rightarrow m, \ r \rightarrow s(r), \ a(x, s(x)) \rightarrow p; \)
(b) \( a(x, x) \rightarrow m, \ r \rightarrow s(r), \ s(x) \rightarrow a(x, s(x)) \)

There are many automated tools dealing with rewriting systems. A descriptive list can be found at https://www.jaist.ac.jp/~hirokawa/tool. We can mention

- TTT2 http://cl-informatik.uibk.ac.at/software/ttt2, termination checker,
- mkbTT http://cl-informatik.uibk.ac.at/software/mkbtt performs Knuth-Bendix completion.
- AProVE http://aprove.informatik.rwth-aachen.de is a system for automated termination (can prove termination of Prolog programs, TRSs, Haskell, . . . )
- Dedukti https://deducteam.github.io, a logical framework using rewriting as a computation model.

Termination tools often (at least AProVE and TTT2 do) use the syntax of the termination competition for input TRSs which can be found at https://www.lri.fr/~marche/tpdb/format.html. The main components are, where NL is the newline character,

\[
\begin{align*}
\text{spec ::= } & (\text{ decl }\ )\text{ spec } | (\text{ decl }\ ) \\
\text{decl ::= } & \text{VAR idlist } | \text{RULES listofrules} \\
\text{idlist ::= } & \text{id } | \text{id idlist} \\
\text{listofrules ::= } & \text{rule } | \text{rule NL listofrules} \\
\text{rule ::= } & \text{term }\rightarrow\text{term} \\
\text{term ::= } & \text{id } | \text{id( ) } | \text{id(termlist) } \\
\text{termlist ::= } & \text{term } | \text{term, termlist}
\end{align*}
\]

Exercise 2:
Consider the following rewriting system

\[
\begin{align*}
0 + n &\rightarrow n \quad (1) \\
s \ m + n &\rightarrow s \ (m + n) \quad (2) \\
m + 0 &\rightarrow m \quad (3) \\
m + s \ n &\rightarrow s \ (m + n) \quad (4) \\
\text{triangle } 0 &\rightarrow 0 \quad (5) \\
\text{triangle } (s \ n) &\rightarrow n + (\text{triangle } n) \quad (6)
\end{align*}
\]

in Dedukti3 syntax,

\[
\begin{align*}
\text{constant symbol Nat : TYPE} \\
\text{symbol s : Nat }\Rightarrow\text{ Nat} \\
\text{constant symbol z : Nat} \\
\text{set builtin "0": }\arrow\text{ z} \\
\text{set builtin "+1": }\arrow\text{ s} \\
\text{symbol + : Nat }\Rightarrow\text{ Nat }\Rightarrow\text{ Nat} \\
\text{set infix left 6 "+": }\arrow\text{ add} \\
\text{rule 0 } + \&\text{n }\rightarrow\text{ &n} \\
\text{and } (s \ &\text{n}) + \&\text{n }\rightarrow\text{ s } (\&\text{m + &n)} \\
\text{and } &\text{m } + 0 &\rightarrow\text{ &m}
\end{align*}
\]
and \( \&m + (s \&n) \rightarrow s (\&m + \&n) \)

symbol triangle : Nat \( \Rightarrow \) Nat
rule triangle \( 0 \rightarrow 0 \)
and triangle \( (s \&n) \rightarrow \&n + (\triangle \&n) \)

compute triangle 45

(a) What happens if lines 13 and 14 are placed before line 11 and 12 when the option keep-rule-order is used?

(b) Compute the complexity of triangle \( n \) in each case.

**Exercise 3:**

**Definition 1** (Strong joinability). Two terms \( t \) and \( u \) are strongly joinable if there are terms \( v \) and \( w \) such that \( t \rightarrow^* v \leftarrow^* u \) and \( t \rightarrow^* w \leftarrow^* u \).

(a) Give a TRS that is locally confluent but not confluent.

(b) Why does the critical pairs lemma give only local confluence?

(c) Prove that if every critical pair of a linear TRS \( R \) is strongly joinable, then \( R \) is strongly confluent.

(d) Is the following rewrite system confluent?

\[

g (f (j x) y) \rightarrow (g (f (j x) y)) \quad g (x (k y) z) \rightarrow (i (l y) (h z x)) \\
(f (h x y) \rightarrow (h (f (j y)))) \quad h (x (f (j y))) \rightarrow (h (x y))
\]

(e) Is the system

\[

h f a a \rightarrow g a a \quad a \rightarrow b \\
h g a a \rightarrow h f a a \\
h x y b \rightarrow h x y y
\]

congruent?