TP Programmation

L3

30 November 2010

We will implemet the termination detection algorithm using RPOs in this session.

Recursive path ordering (given a strict order $>$ on the finite signature)
$s >_{rpo} t$ iff

case1 $t \in Var(s)$ and $s \neq t$, or

case2 $s = f(s_1, \ldots, s_m), t = g((t_1, \ldots, t_n)$ and

- **case2.1** there exists $i, 1 \leq i \leq m$, with $s_i \geq_{rpo} t$, or
- **case2.2** $f > g$ and $s >_{rpo} t_j$ for all $j, 1 \leq j \leq n$, or
- **case2.3** $f = g, s >_{rpo} t_j$ for all $j, 1 \leq j \leq n$, and $[s_1, \ldots s_n]$ greater than $[t_1, \ldots, t_n]$ according to some (non arbitrary) ordering (like lexicographic ordering or multiset ordering)

We will first define the order types we need.

1. Define a suitable type `order` for pre orders.
2. Define a function `lex` for lexicographic ordering.
3. Define a function `mul` for multiset ordering.
4. Implement `lpo` - the lexicographic path ordering. This uses lexicographic ordering in case 2.3 of the definition of RPO.
5. Write a function to check if a term rewriting system terminates with LPO.

In RPO with status, status of a function symbol says what order (lexicographic or multi set ordering) needs to applied in case 2.3 of the definition of RPO.

6. Implement RPO with status.
7. Write a function to check if a term rewriting system terminates with RPO with status ordering for a given status. (It takes a set of rewrite rules and a status function as input.)
8. Write a function to check if a term rewriting system can be checked for termination using some lexicographic path ordering. (This problem is NP-Complete.)