Automated Reasoning 2012 (KB) PROBLEMS 9 Knuth Bendix Procedure

QUICKIES:

1. Apply Knuth Bendix to rules of previous Quick Questions, using equations ordered as follows. Make sure that you justify the direction of each rewrite rule.
(a) (1) s(j(x,y)) => j(y,x), (2) j(x,n) => x.
From Sheet 8 Quickie Q2a you found the critical pair j(n,x) and s(x) from 1+2.
Either order (3) as j(n,x)=>s(x) and use kbo ordered by size
or order (3) as s(x) => j(n,x) and use rpo with s>$_1$j and s>$_1$n.
(b) (1) f(x,x) => e, (2) h(g(x,y)) => g(y,x), (3) g(x,y) => f(y,x).
From sheet 8 Quickie Q2b you found the critical pair h(f(y,x)) and f(x,y), which should be ordered as (3) h(f(y,x)) => f(x,y). Take as the ordering either kbo and size with g >$_1$ f, or rpo with h >$_1$ g >$_1$ f >$_1$ e.

LONGER QUESTIONS:

1. Apply the Knuth Bendix algorithm to the following sets of rules, aiming to derive a confluent set. In some cases the algorithm may not terminate, which should be clear from the structure of the equations being generated. Make sure you justify the selected direction of each rewrite rule. Use the answers from Sheet 8 to find the initial critical pairs for parts a) and c).

(a) (1) f(x, e) => x, (2) f(x, x) => e, (3) g(f(x, y)) => f(g(x),y)

From sheet 8 Q1b you found the critical pair f(g(x),x) and g(e) from 2+3;
order by kbo using length of terms, taking g $>_1$ f $>_1$ e

(b) (1) f(x, x)=>e, (2) h(g(x, y))=>g(y, x),
(3) f(x, y)=>g(y, x), (4) g(z,f(x,y))=>g(g(y,x),z) order by rpo.

2. Use narrowing and the confluent set (1)-(7), to find (several) values for x and y such that min(x, min(y, x)) = x .

(1) min(x,x+y)=>x; (2) min(x+y,x)=>x; (3) x+0=>x; (4) 0+x=>x;
(5) min(x,x)=>x; (6) min(x,0)=>0; (7) min(0,x)=>0.

3(a) (1) g(a) =>a, (2) g(b) =>b, (3) h(a, x) => f(a, x), (4) h(b, x) => f(b, x),
(5) f(h(x, y),z) => f(y, f(x, z)), (6) g(h(x, y)) => h(g(y), g(x))

The critical pairs found in Sheet 8 were f(f(a,x),z)=f(x,f(a,z)) from 3+5, g(f(a,x))=h(g(x),a) from 3+6, and similarly for b using 4+5 and 4+6.
Here, you can't use kbo by counting terms because in (6) the size of RHS is bigger than the size of LHS; also, can't use *rpo* because in (5) the multi-set order would require {h(x,y),z} >> {y,f(x,z)}which fails; however, **can** use lpo and g $>_1$ h $>_1$ f – *lpo* differs from *rpo* only in the way it deals with equal outermost functors; must check left_arglist >lpo right_arglist (lexicographically) <u>and</u> left_side>lpo arg2 of rightside (since max of 2 arguments). (See answers to Sheet 8, part 3e for an application of *lpo* and also the slide 15civ). It is possible to use *kbo* in which all symbols have weight=1 except the unary functor g, which has weight=0, and when the ranking of g is greater than the ranking of any other functor (ie g $>_1$ f, g $>_1$ h, g $>_1$ a, g $>_1$ b). This has been shown by Dershowitz, in Termination of Rewriting, J. Symbolic Computation, 1987.

(b) Show that, if (3) and (4) of part (a) were ordered in the opposite direction as
(7) f(a,x)=>h(a,x) and (8) f(b,x)=>h(b,x),
then rules (1), (2), (5) – (8) can still be ordered by lpo (g $>_1$ f $>_1$ h) and there are no superpositions at all.

(c) Use rewriting and (1), (2), (5) - (8) to rewrite f(f(a, g(b)), g(f(h(a,b),b))).