ADDENDUM TO
COMPUTING CONGRUENCES EFFICIENTLY

RALPH FREESE

Two addenda: first we want to point out that Cliff Bergman and Giora Slutzki [1] showed that deciding if (c, d) is in the congruence of $\mathbf{A}$ generated by a set of pairs is nondeterministic log-space complete, **NL**. This implies it is polynomial time, **P**, (but not that it is linear time, of course). It is believed that **NL** is properly contained in **P**.

Secondly Quinn Culver, a graduate student here at the University of Hawaii, gave a better proof of the correctness of the algorithm. While his proof is very similar to the original proof, it has the advantage that one need not assume that the root finding was done without using the collapsing rule, as was done in the original.

Here is his proof: as before we let θ_1 be the final partition of the algorithm. We need to show

$$\text{if } u \ \theta_1 \ v \text{ then } f(u) \ \theta_1 \ f(v) \text{ for } f \in S. \tag{1}$$

As before, if at some point in the procedure (u, v) or (v, u) is on P, (1) will hold. Fix u and let $r_0, r_1, \ldots, r_t$ be the roots for u as the algorithm proceeds. We prove by induction that (1) holds if $v = r_i$. If $u = b$ then $r_0 = a$ and (1) holds since (a, b) gets on P. If $u \neq b$ then $r_0 = u$ and (1) holds trivially. When the root of u changes from r_i to r_{i+1}, r_i is made to be a child of r_{i+1} and (r_i, r_{i+1}) is put on P. So (1) holds for r_i and r_{i+1} and by induction it holds for u and r_i, which implies it holds for u and r_{i+1}.

Now if $u \ \theta_1 \ v$ then they have the same root and it is easy to see (1) holds.

References

[1] Clifford Bergman and Giora Slutzki, *Computational complexity of some problems involving congruences on algebras*, Theoret. Comput. Sci. **270** (2002), no. 1-2, 591–608.

(Ralph Freese) Department of Mathematics, University of Hawaii, Honolulu, Hawaii, 96822 USA

E-mail address, Ralph Freese: `ralph@math.hawaii.edu`

Date: March 24, 2010.