Notes on the complexity of the word problem for semigroups over invertible transducers

Evan Bergeron

September 15, 2016

I suspect we can solve the word problem in poly time for the following subclass of invertible transducers over the binary alphabet: 1-toggle, 2 SCC, minimal automata whose transduction semigroups are Abelian. Then words over the generators can be expressed as a Parikh vector.

At the very least, I'm convinced that residuation over any word produces polynomially many distinct residuals, with respect to the word length and size of the automata. By Okano, we know that the two connected components must be a direct path to the toggle state and a identity self-loop, respectively (since it's Abelian). So then residuation just rotates the elements of the vector, occasionally subtracting $\lfloor x/2 \rfloor$ or $\lceil x/2 \rceil$ from some element x.

Some computation shows that the size of the transitive closure of these two operations on some integer x is bounded by $2\log_2(x+1)$, which is polynomial in the length of the input.

Question: does this mean that we only get polynomially many residuals? And does having polynomially many residuals lead to a poly time word checking algorithm? I suspect it will, since dynamic programming and DFS through the wreath recursions would do the rest.