Specification and Verification of Distributed Systems

January 18, 2007

Exercise 1

Let $\widetilde{\Sigma} = (\{a,b\},\{b,c\},\{d\})$ be a distributed alphabet. For the following languages $L_i \subseteq \Sigma^*$, determine if $L_i = L(\mathcal{A})$ (or, if not, $[L_i]_{\sim_{\widetilde{\Sigma}}} = L(\mathcal{A})$) for some AA $\mathcal{A}$ over $\widetilde{\Sigma}$.

$L_1 = (abcd)^*$

$L_2 = (d^*ad^*bd^*cd^*)^*$

$L_3 = (ac)^*$

$L_4 = (a+c)^*$

$L_5 = c^*a^*c^*$

$L_6 = c^*a^*c^*a^*$

$L_7 = a((ac+ca)b)^*$

Which of the above languages is accepted by some (locally accepting) PA?

Exercise 2

Build a det-AA $\mathcal{A}$ over $\widetilde{\Sigma} = (\{a,b\},\{b,c\})$ such that

$$L(\mathcal{A}) = (a+b+c)^*(ac+ca)(a+b+c)^*.$$

Exercise 3

Let $\widetilde{\Sigma}$ be a distributed alphabet.

(a) Show that, for any $w \in \Sigma^*$, $tr(w)$ is a trace over $\widetilde{\Sigma}$.

(b) Show that, for any $u,v \in \Sigma^*$ and $(a,b) \in I_{\widetilde{\Sigma}}$, we have $tr(uabv) = tr(ubav)$.

Exercise 4

Let $\widetilde{\Sigma}$ be a distributed alphabet. Consider the following definitions wrt. $\widetilde{\Sigma}$:

- A *safe asynchronous automaton* is an AA $((S_p)_{p\in Proc},(\Delta_a)_{a\in\Sigma},\imath,F)$ such that $F = \prod_{p\in Proc} S_p$.
- A *weakly safe asynchronous automaton* is an AA $\mathcal{A} = ((S_p)_{p\in Proc},(\Delta_a)_{a\in\Sigma},\imath,F)$ such that, for any $u \in \Sigma^*$ and any run ρ of $\mathcal{A}$ on u, there are $u' \in \Sigma^*$ and $\rho' \in (S_{\mathcal{A}})^*$ such that $\rho\rho'$ is an accepting run on uu'.

Compare the expressiveness of (weakly) safe asynchronous automata with that of AA, la-AA, PA, and la-PA. How about decidability of the implementability problems?