Concurrency Theory, January–April 2018

Assignment 2, 8 March, 2018
Due: 18 March, 2018

Note: Only electronic submissions accepted, via Moodle.

Notation

- For $w \in \Sigma^*$ and $X \subseteq \Sigma$, $w{\downarrow_X}$ denotes the projection of w with respect to X—that is, the word obtained by erasing all letters not in X from w. Formally,

$$\begin{array}{lcll} \varepsilon{\downarrow_X} & = & \varepsilon & \\ wa{\downarrow_X} & = & w{\downarrow_X} \cdot a, & \text{if } a \in X \\ & & w{\downarrow_X}, & \text{otherwise} \end{array}$$

- Given a distributed alphabet $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$,
  - $\Sigma = \bigcup_{i \in \{1,2,\ldots,k\}} \Sigma_i$.
  - For $a \in \Sigma$, $\mathrm{loc}(a) = \{i \mid a \in \Sigma_i\}$.
  - $I_{\mathrm{loc}} = \{(a,b) \mid \mathrm{loc}(a) \cap \mathrm{loc}(b) = \emptyset\}$ is the independence relation, usually denoted just I.
  - $D_{\mathrm{loc}} = (\Sigma \times \Sigma) \setminus I$ is the dependence relation, usually denoted just D.
  - $w \mathrel{\dot\sim} w'$ if $w = uabv$ and $w' = ubav$ for some $(a,b) \in I$.
  - $w \sim w'$ if there exists $m \geq 0$ and words $w_0, w_1, \ldots, w_m$ such that $w = w_0 \mathrel{\dot\sim} w_1 \mathrel{\dot\sim} \cdots \mathrel{\dot\sim} w_m = w'$.

Questions

1. Given a distributed alphabet $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$ and a pair of words u, v, prove that $u \sim v$ if and only if $u{\downarrow_{\{a,b\}}} = v{\downarrow_{\{a,b\}}}$ for every pair of letters $(a,b) \in D$.

2. Suppose we have two distributed alphabets $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$ and $(\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$ such that $\Sigma = \bigcup_{i \in \{1,2,\ldots,k\}} \Sigma_i = \bigcup_{j \in \{1,2,\ldots,m\}} \Sigma'_j$ and both distributed alphabets induce the same independence relation on Σ.

   We would like to check whether the two distributed alphabets are equivalent with respect to distributed implementations in the form of direct products and sycnhronized products.

   Define $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k) \preceq (\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$ if for each $i \in \{1, 2, \ldots, k\}$, there exists $j \in \{1, 2, \ldots, m\}$ such that $\Sigma_i \subseteq \Sigma'_j$.

   (a) Suppose $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k) \preceq (\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$ and L is a direct product language over $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$. Show that L is also a direct product language over $(\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$.

   Is the converse true? If L is a direct product language over $(\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$, is it always implementable as a direct product language over $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$?

   (b) Suppose $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k) \preceq (\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$ and L is a synchronized product language over $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$. Show that L is also a synchronized product language over $(\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$.

   Is the converse true? If L is a synchronized product language over $(\Sigma'_1, \Sigma'_2, \ldots, \Sigma'_m)$, is it always implementable as a synchronized product language over $(\Sigma_1, \Sigma_2, \ldots, \Sigma_k)$?

3. We can extend the notion of primary and secondary information to arbitrary depth: primary information consists of events of the form $\mathit{latest}_{i \to j_1}(I)$, secondary information consists of events of the form $\mathit{latest}_{i \to j_1 \to j_2}(I)$, ..., m-ary information consists of events of the form $\mathit{latest}_{i \to j_1 \to j_2 \to \cdots \to j_m}(I)$.

   The update algorithm for primary information uses secondary information to maintain a consistent labelling across all primary events, but does not guarantee that secondary events are themselves consistently labelled. Show that we can construct an asynchronous automaton that maintains a consistent labelling across all m-ary events, for any choice of m.