Embedded System Design and Modeling

EE382N.23, Fall 2015

Homework #2

Models of Computation

Assigned: September 30, 2015
Due: October 14, 2015

Instructions:

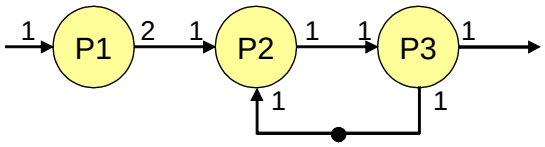
- Please submit your solutions via Canvas. Submissions should include a single PDF with the writeup and a single Zip or Tar archive for any supplementary files (e.g. source files, which has to be compilable by simply running 'make' and should include a README with instructions for running each model).
- You may discuss the problems with your classmates but make sure to submit your own independent and individual solutions.
- Some questions might not have a clearly correct or wrong answer. In general, grading is based on your arguments and reasoning for arriving at a solution.

Problem 2.1: Process Models

(a) Every KNP can be transformed into a (dynamic) dataflow model, and vice versa. If so, what is really the key difference between the two models, and how does that translate into any differences in their implementability?

(b) In class, we mentioned that Parks' algorithm is not guaranteed to find a bounded and complete (and thus non-terminating) schedule even if one exists. Show an example of a KPN where such a schedule exists but Parks' algorithm fails to find it. Hint: in the KPN example presented in class, think about token patterns that can happen on the P2→P3 edge.

(c) In the following synchronous dataflow graph, which pairs of connected actors, if any, can be composed into a hierarchical super-actor to give a valid SDF graph that consists of the super-actor and the remaining actor (show the corresponding hierarchical graph)? For any invalid pairs, show a valid hierarchical graph in which the composite super-actor is converted into an appropriate CSDF actor instead.

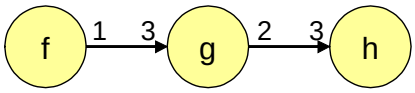


(d) Are KPN models composable? In the example above, when executed as KPN, can every pair of connected processes be composed into a hierarchical super-process while maintaining KPN semantics?

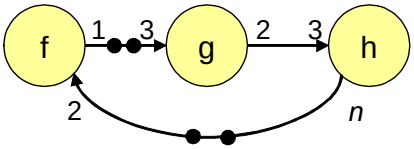
Problem 2.2: Synchronous Dataflow (SDF)

For each of the following SDF models, determine whether a valid schedule exists that can be executed repeatedly and indefinitely in bounded buffer memory. Write down the balance equations, determine if and when the graph is consistent and give the repetition vector for a minimal (least integer) schedule. If it exists, write down a schedule that minimizes buffer sizes. How much buffer space is needed on each communication channel?

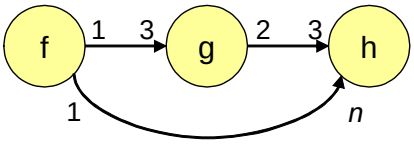
(a)



(b) *n* is an integer parameter.



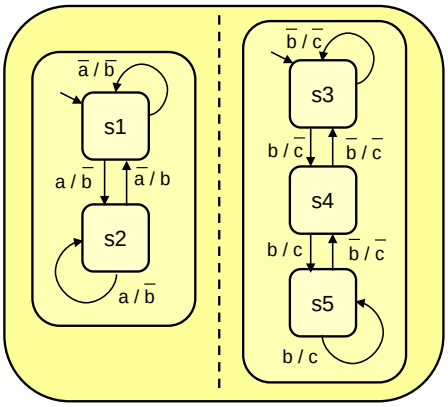
(c) *n* is an integer parameter.



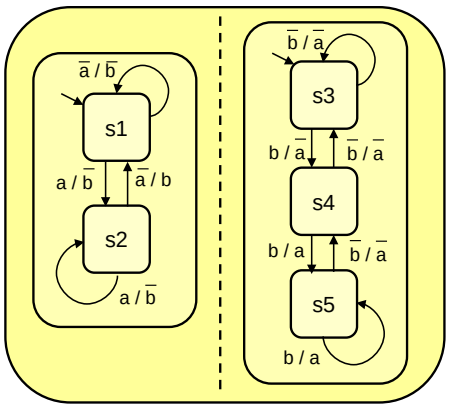
Problem 2.3: State Machine Models

Convert the following Mealy HCFSM compositions into an equivalent single, flat FSM, if possible. Indicate which states of the composition are unreachable, if any. Note that presence/absence of signals as indicated by the x / $\overline{x}$ notation is really the same as setting their value to '1' or '0', respectively.

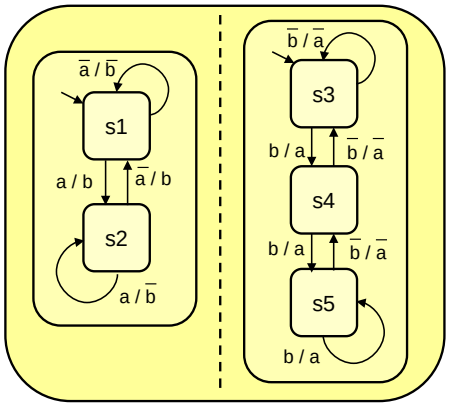
(a)



(b)



(c)



(d) The Statecharts realization of HCFMs has an additional construct called *history transitions* (marked by an 'H'), which, when taken, resume a hierarchical destination in whatever state it was last in (or its initial state on the first entry). Convert the following history-based hierarchical FSM into an equivalent regular flat FSM: