Connecting State Machines

Last time we
- Introduced the nondeterministic state machine
- Highlighted differences between deterministic and nondeterministic machines
- Introduced ideas of equivalence between state machines: simulation and bisimulation

Today we will
- Identify issues involved when connecting state machines
- Look at different connection topologies
- Consider implications of connection in mathematical description

Cascade Composition

Here, the output of machine A is the input of machine B. The two machines are running simultaneously (both on step n) Each machine has its own input, current state, and output. The effect of the input $x_A(n)$ propagates instantaneously through the cascade at each step: **synchrony**

We may view the cascade of two machines as a single machine.
Cascade Composition: Definition

Let's define the 5-tuple for the composite machine:

\[ (\text{States}_A, \text{Inputs}_A, \text{Outputs}_A, \text{update}_A, \text{initialState}_A) \]

\[ (\text{States}_B, \text{Inputs}_B, \text{Outputs}_B, \text{update}_B, \text{initialState}_B) \]

Define the 5-tuple for the composite machine:

\[ \text{States} = \text{States}_A \]
\[ \text{Inputs} = \text{States}_B \]
\[ \text{Outputs} = \text{Outputs}_A \]
\[ \text{update} = \text{update}_A \]
\[ \text{initialState} = \text{initialState}_A \]

where \( (s_A(n+1), y_A(n)) = \text{update}_A(s_A(n), x(n)) \)

and \( (s_B(n+1), y(n)) = \text{update}_B(s_B(n), y_A(n)) \)

Note that the “internal” output \( y_A(n) \) is used as the “internal” input \( x_B(n) \) to machine B.

Thus, for the cascade connection to be valid, we must have

\[ \text{Outputs}_A \subseteq \text{Inputs}_B \]
Cascade Composition: Example

Consider the cascade of machine A and machine B below, where the output of machine A is the input of machine B. Find the composite state response and output for input $x = 1 \ 0 \ 0$.

Machine A

Machine B

Cascade Composition: Diagram

Two cascaded machines can be drawn using one state diagram:

1) Draw a circle for each state in $States_A \times States_B$.
2) For each state, consider each possible input to machine A.
   a) Find the corresponding next state in machine A.
   b) Find the output of machine A, which forms the input of machine B.
   c) Find the corresponding next state in machine B.
   d) Find the output of machine B.
   e) Draw the transition arrow to $(s_a(n+1), s_b(n+1))$.
   f) Label the transition arrow with the input to machine A and the output from machine B.
Cascade Composition: Diagram

Try drawing a single state diagram for machines A and B in the previous example:

Can we ever get to state (b, c)?
Can we ever get to states (a, c) or (b, b)?
Can we ever get a nonzero output?

Reachability

On its own, given its entire set of legal inputs, Machine B can reach state c and give an output of 1.
But, in cascade, the inputs of Machine B are limited to the possible outputs of Machine A.
Machine A cannot generate a sequence of outputs that would drive machine B into state c. Such an output is not in Behaviors_A.
More Complicated Connections

Here, we wish to have access to the individual output $y_A$, even when treating the cascade as one big machine.

Sending a signal to more than one destination is called **forking**.

Note also that there is an additional external input into machine B.

When viewing the composite machine as a single entity, with multiple input and/or output ports, we can express the input and output in the expected way using a set product:

$\text{Inputs} = \text{Inputs}_A \times \text{Inputs}_{B,\text{EXT}}$  
$\text{Outputs} = \text{Outputs}_A \times \text{Outputs}_{B}$

The set of inputs or outputs for a particular port (Outputs$_B$, for example) is called a **port alphabet**.