apl. Prof. Dr. Thomas Noll

Kevin van der Pol, Hao Wu

Exercise 1 (Extended CCS): (6 Points)

Many extended CCS languages are widely used in modelling different systems. Now we focus on an extended CCS which allows

- processes to have parameterized value variables (here we restrict them to be of type integer), and
- such values (= evaluation of variables) can be passed to other processes during synchronization,
- to add a **if** bexp **then** P **else** Q construct such that the process can make decision based on the current values of its variables.

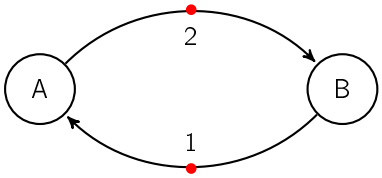
An example of one-place buffer which only store an integer bigger than 10 can be defined by using such extend CCS language as

$$\begin{aligned} B &= \text{input}(x).B(x), \\ B(x) &= \textbf{if } x > 10 \textbf{ then } \overline{\text{output}(x)}.B \textbf{ else } \text{discard}.B. \end{aligned}$$

a) This extension of CCS introduces a *state*, a valuation of variables, in the semantics. What is the new type of the execution relation?

b) Modify the rules in Definition 18.3 to define the semantics of this extended CCS language.

c) Consider the following system. Two processes are connected by two directed channels which can store tokens. As soon as the channel A to B has more than 2 tokens, process B can execute and produce 1 token to the channel from B to A, after and consuming 2 tokens from the channel A to B. In the same way, as soon as the channel A to B has more than 1 token, process A can execute and produce 2 tokens to the channel from A to B, after consuming 1 tokens from the channel B to A. Please model this system by using extended CCS.



d) Now the channels are initialized with 2 tokens in channel A to B and 1 token in channel B to A. Initialize your system and give the corresponding LTS based on your definition.

e) Now we would like to use a model of a clock to measure the time passing (one tick represents one second). We define it as follows:

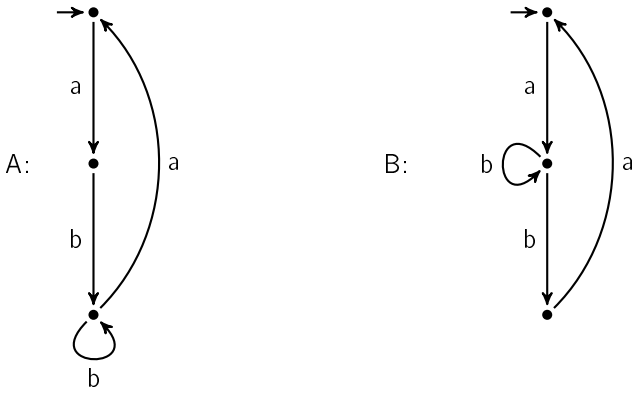
$$\text{Clock} = \overline{\text{tick}}.\text{Clock}$$

Now we assume A's execution takes one second and B's execution takes two seconds, and synchronization between processes happens immediately. Why can this not be modeled in the extended CCS?

f) Change the extended CCS semantics to fix this problem. Add the clock to the model such that the processes correctly synchronize.

Exercise 2 (Bisimulation in CCS): (1+1+2 Points)

Assume following LTSs according to two CCS processes are given:



a) Are following two LTSs A and B trace equivalent?

b) Are they bisimilar? Justify your answers.

c) Give an LTS as a triple $(S, Act, \longrightarrow)$ where S is a nonempty set of state (i.e. processes in CCS), Act is a set of actions, and $\longrightarrow \subseteq S \times Act \times S$ is a set transitions. A so-called Hennessy-Milner logic (HML) defines a set of formulae $\mathcal{M}$ over Act to describe the properties in LTS as:

$$F, G ::== \text{true} \mid \text{false} \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F \quad (\text{where} a \in Act)$$

Let F be a formula in $\mathcal{M}$, a state $p \in S$ satisfies F (denote as $p \vDash F$) is defined as follows:

- $p \vDash \text{true}$ *iff* $p \in S$,
- $p \vDash \text{false}$ *iff* $p \in \varnothing$,
- $p \vDash \langle a \rangle$ *iff* $\exists q \vDash F . p \xrightarrow{a} q$,
- $p \vDash [a]$ *iff* $\forall q \in S . p \xrightarrow{a} q \Rightarrow q \vDash F$.

Give a formula F in HML which can distinguish these two LTSs.