Concurrency Theory

Prof. Dr. Peter Thiemann — University of Freiburg
Marius Weidner, Leonardo Mieschendahl — Winter 2025

Sheet 8

Due: Monday, 2026-01-12

We use the winter holidays to review the topics up to Petri nets using many small and a few larger exam-style exercises. You can either work on them now or use them later as preparation for the exam.

Exercise 8.1 (NL to CCS)

Consider a simple Vending Machine process `VM`. The machine first accepts a `coin`. After the coin is inserted, it allows the user to either choose `water` or `juice`. If `water` is chosen, it performs an internal check and then `dispense`s the water. If `juice` is chosen, it simply `dispense`s the juice. After any dispensing, the machine becomes `idle` and can only be restarted by a `reset` to return to the initial state.

Provide the CCS defining equations for this system.

Exercise 8.2 (CCS to LTS)

Consider the following CCS process definitions:

- $P \doteq a.P_1 + b.Nil$
- $P_1 \doteq \overline{a}.P$

Draw the Labelled Transition System for the process $Q = (P \,||\, a.Nil) \setminus \{a\}$.

Exercise 8.3 (Trace Language)

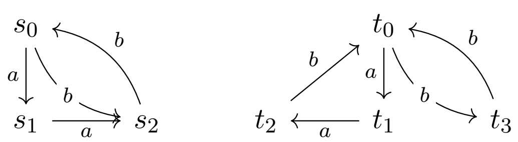
Consider the following CCS definitions involving multiple equations:

$$S \doteq a.A + b.S$$
$$A \doteq c.S + a.Nil$$

State the trace language $Tr(S)$ using regular expression notation.

Exercise 8.4 (Proving Strong Bisimulation using Fixed Point Iteration)

Consider the following LTS:



If you have questions, please post a message in the dedicated chat.

Use the fixed-point characterisation of strong bisimilarity to prove that $s_0 \sim t_0$. For each application of $\mathcal{F}$, it is enough to write down the equivalence classes that the intermediary relations $\sim_i$ induce.

Exercise 8.5 (Disproving Weak Bisimulation using Game Characteristics)

Consider the following two processes:

- $P \doteq a.(\tau.b.Nil + c.Nil)$
- $Q \doteq a.b.Nil + a.c.Nil$

Show that $P \not\approx Q$ by stating a universal winning strategy for the attacker in the weak bisimulation game.

Exercise 8.6 (Prove Bisimulation Law)

For $\beta \in Act$, let $\odot\beta$ be a *new* unary CCS operator with the following semantics:

- **(suff1):** $\dfrac{P \xrightarrow{\alpha} P'}{P \odot \beta \xrightarrow{\alpha} P' \odot \beta}$
- **(suff2):** $\dfrac{P \not\rightarrow}{P \odot \beta \xrightarrow{\beta} Nil}$ where $P \not\rightarrow$ means P has no outgoing transitions.

Prove or disprove: $\odot\beta$ preserves strong bisimilarity, i.e., for any processes S and T with $S \sim T$, it holds that $S \odot \beta \sim T \odot \beta$.

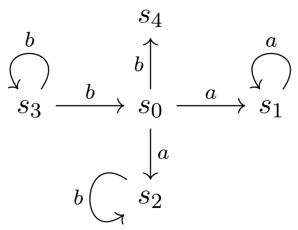
Exercise 8.7 (NL to HML)

Define a set of HML_X equations that model the following statement:

A state can reach a computation path in which for every a-transition a b-transition directly follows.

Please explain the intent of your solution.

Exercise 8.8 (Compute Mutually Recursive HML via Fixed Points)
Consider the following LTS:



Compute the solution of the following set of HML_X equations:

$$X_1 \overset{\min}{=} \langle a \rangle (X_1 \vee X_2)$$
$$X_2 \overset{\min}{=} [b](X_2 \vee X_1)$$

Exercise 8.9 (Define Timed CCS System)
Define a timed CCS process that describes the interaction between a traffic light and a pedestrian:

- The traffic light is initially red.
- Pressing the button will turn it green after 10 seconds.
- The pedestrian presses the button and will then attempt to cross the street after either 3 or 13 seconds.
- The attempt should fail if the traffic light is red, and succeed if it is green (what happens after that is not important).
- The only observable actions should be *try*, *fail* and *succeed*.

Exercise 8.10 (CCS, LTS, Bisimulation and HML)
Consider the following CCS processes:

$A = a.B + a.C$	$D = c.E + b.C$	$G = b.F + a.G + b.H$
$B = b.A + a.C + b.D$	$E = b.B + c.D$	$H = a.G$
$C = b.A + a.B + b.E$	$F = c.F + b.G$	$I = a.b.H + a.G$

(a) Draw $LTS(A)$, $LTS(H)$ and $LTS(I)$, respectively.

(b) Prove or disprove: $A \sim H$, $A \sim I$ and $H \sim I$, where $\sim$ denotes strong bisimilarity. To this end, you may use game characterization or HML formulas.