

Concurrency Theory

Prof. Dr. Peter Thiemann
Marius Weidner, Leonardo Mieschendahl

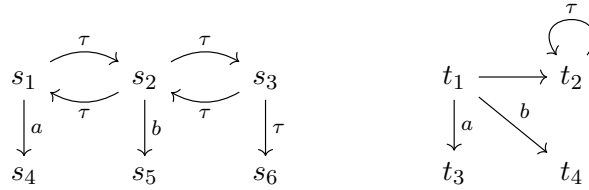
University of Freiburg
Winter 2025

Sheet 3

Due: Monday, 2025-11-17

Exercise 3.1

Consider the two LTS below:



Show that $s_1 \approx t_1$ by constructing a weak bisimulation.

Exercise 3.2

Proof or disprove the statements below:

$$a.\tau.nil \sim \tau.a.nil$$

$$a.\tau.nil \approx \tau.a.nil$$

Exercise 3.3

Prove that $P + \tau.P \approx \tau.P$ for all CCS processes P .

Exercise 3.4

In the weak bisimulation game the attacker is only allowed to use \xrightarrow{a} moves, but the defender is allowed to use \xRightarrow{a} moves. Give an argument why letting the attacker use \xRightarrow{a} moves would not make the attacker more powerful.

Exercise 3.5

Define two CCS processes A and B such that

- $LTS(A)$ has an infinite number of states,
- $LTS(B)$ has a finite number of states
- $A \sim B$ is false (no proof needed)
- $A \approx B$ is true (no proof needed)

If you have questions, please post a message in the dedicated [chat](#).

Bonus: prove that $A \approx B$ is true, but $A \sim B$ is false.

Exercise 3.6 (Bonus)

We want to model a simple communication protocol between a sender and a receiver. The sender accepts data and the receiver delivers the data. We want to implement this specification via a medium that transmits the data between sender and receiver.

Consider the following CCS definition for the specification and implementation of this communication protocol:

$\text{ProtocolSpec} \doteq \text{acc}.\overline{\text{del}}.\text{ProtocolSpec}$

$\text{Protocol} \doteq (\text{Send} \parallel \text{Med} \parallel \text{Rec}) \setminus \{\text{send}, \text{error}, \text{trans}, \text{ack}\}$

$\text{Send} \doteq \text{acc}.\text{Sending}$

$\text{Med} \doteq \text{send}.\text{Trans}$

$\text{Rec} \doteq \text{trans}.\text{Del}$

$\text{Sending} \doteq \overline{\text{send}}.\text{Wait}$

$\text{Trans} \doteq \tau.\text{Err} + \overline{\text{trans}}.\text{Med}$

$\text{Del} \doteq \overline{\text{del}}.\text{Ack}$

$\text{Wait} \doteq \text{ack}.\text{Send} + \text{err}.\text{Sending}$

$\text{Err} \doteq \overline{\text{err}}.\text{Med}$

$\text{Ack} \doteq \overline{\text{ack}}.\text{Rec}$

Show that Protocol is an implementation of ProtocolSpec by showing that $\text{Protocol} \approx \text{ProtocolSpec}$ is true.