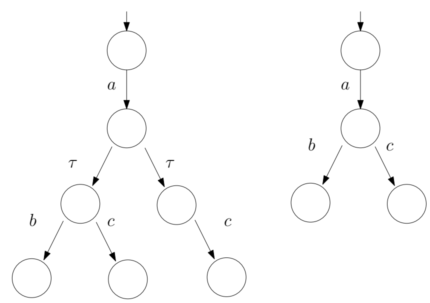
System Validation (IN4387)
November 2, 2012, 14:00-17:00

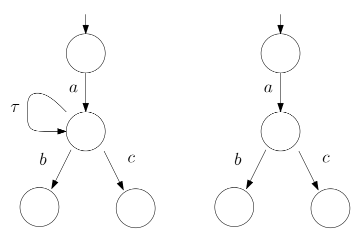
Important Notes. The examination comprises 5 exercises in 4 pages. Please give complete explanation and do not confine yourself to giving the final answer. **Good luck!**

Exercise 1 (20 points) In each of the following items determine whether the specified equivalence holds for the given LTSs. For each and every item provide a complete line of reasoning why a certain equivalence does or does not hold:

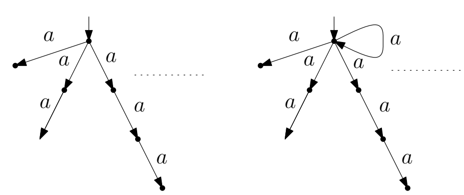
1. Strong bisimilarity:



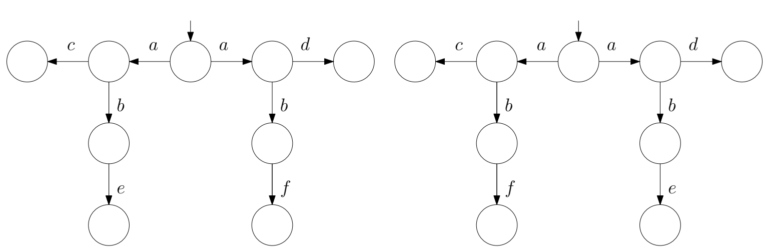
2. Branching bisimilarity:



3. Strong bisimilarity (the dotted lines represent traces of a's with length n, for each and every $n > 3$):



4. Branching bisimilarity:



Exercise 2 (20 points) Consider the following two modal formulae:

$$[request]\langle true^* response\rangle true$$

and

$$[request](\mu X.\langle true\rangle true \wedge [\overline{response}]X)$$

1. Explain in words what each of the two formulae means. **(10 points)**
2. Give a labeled transition system in which one of the two formulae holds and the other one does not hold. (You can freely choose the one to hold.) **(10 points)**

Exercise 3 (20 points) Define a sort (data type) ToDoList, where each element of this sort is either the empty list, or a non-empty list of prioritized tasks. A prioritized task is a pair (i, t) where i is a positive natural number determining the priority and t is an element of a sort $Task$, which contains a constant (constructor) $noTask$ and is not specified any further.

- Give the formal definition of $ToDoList$ and its constructors. **(5 points)**
- Define a function (map) $toDoNow$, which takes a $ToDoList$ as its parameter, and returns the task with the highest priority in the list, if it is non-empty, or $noTask$, otherwise. If needed, you may define and use other auxiliary functions in the definition of $toDoNow$. **(15 points)**

Exercise 4 (20 points) Prove the following equations using the axioms provided in the appendix. Mention the name of the axiom used for each and every step.

1. $(a(1) \mid b(2)) \setminus (c(2) \mid b(3)) = a(1) \mid b(2)$ **(5 points)**,
2. $(a + b) \cdot c \parallel \delta = a \cdot c \cdot \delta + b \cdot c \cdot \delta$ **(5 points)**, and
3. $(c \wedge d) {\rightarrow} a \subseteq c {\rightarrow} a$ (Recall $x \subseteq y$ if and only if $x + y = y$) **(10 points)**.

Exercise 5 (20 points) Specify the following system of two parallel processes:

The first process represents a temperature sensor, which can issue two types of actions: $snd\_temp(n)$ and snd_defect. The sensor can send any natural number between 0 and 200 as the parameter of snd_temp and may non-deterministically choose to send the snd_defect action, after which it deadlocks.

The second process represents a thermostat, which receives the temperature from the sensor and if the received value is in the range between 0 and 50, it issues action *on* to the outside world; if the value is between 51 and 100 it sends action *off* to the outside world; if the received value is outside these ranges, it ignores the value. The thermostat keeps on listening to the sensor at any case. Upon synchronizing with snd_defect, the thermostat will issue an *alarm* action and terminate.

The action names that are not specified in the above-given description can be chosen at will.

MA1	$\alpha\|\beta = \beta\|\alpha$
MA2	$(\alpha\|\beta)\|\gamma = \alpha\|(\beta\|\gamma)$
MA3	$\alpha\|\tau = \alpha$
MD1	$\tau \setminus \alpha = \tau$
MD2	$\alpha \setminus \tau = \alpha$
MD3	$\alpha \setminus (\beta\|\gamma) = (\alpha \setminus \beta) \setminus \gamma$
MD4	$(a(d)\|\alpha) \setminus a(d) = \alpha$
MD5	$(a(d)\|\alpha) \setminus b(e) = a(d)\|(\alpha \setminus b(e)) \quad \text{if } a \not\equiv b \text{ or } d \not\approx e$
MS1	$\tau \sqsubseteq \alpha = \mathit{true}$
MS2	$a \sqsubseteq \tau = \mathit{false}$
MS3	$a(d)\|\alpha \sqsubseteq a(d)\|\beta = \alpha \sqsubseteq \beta$
MS4	$a(d)\|\alpha \sqsubseteq b(e)\|\beta = a(d)\|(\alpha \setminus b(e)) \sqsubseteq \beta \quad \text{if } a \not\equiv b \text{ or } d \not\approx e$
MAN1	$\underline{\tau} = \tau$
MAN2	$\underline{a(d)} = a$
MAN3	$\underline{\alpha\|\beta} = \underline{\alpha}\|\underline{\beta}$

Table 1: Axioms for multi-actions

Note that $a(d)$ and $b(e)$ range over (parameterized) actions, α and β range over (multi)actions and x, y and z range over processes.

A1	$x + y = y + x$
A2	$x + (y + z) = (x + y) + z$
A3	$x + x = x$
A4	$(x + y)\cdot z = x\cdot z + y\cdot z$
A5	$(x\cdot y)\cdot z = x\cdot(y\cdot z)$
A6	$x + \delta = x$
A7	$\delta\cdot x = \delta$
Cond1	$true \rightarrow x \diamond y = x$
Cond2	$false \rightarrow x \diamond y = y$
SUM1	$\sum_{d:D} x = x$
SUM3	$\sum_{d:D} X(d) = X(e) + \sum_{d:D} X(d)$
SUM4	$\sum_{d:D}(X(d) + Y(d)) = \sum_{d:D} X(d) + \sum_{d:D} Y(d)$
SUM5	$(\sum_{d:D} X(d))\cdot y = \sum_{d:D} X(d)\cdot y$

Table 2: Axioms for the basic operators

M	$x \parallel y = x \mathbin{⌊\!⌊} y + y \mathbin{⌊\!⌊} x + x \mid y$
LM1	$\alpha \mathbin{⌊\!⌊} x = \alpha\cdot x$
LM2	$\delta \mathbin{⌊\!⌊} x = \delta$
LM3	$\alpha\cdot x \mathbin{⌊\!⌊} y = \alpha\cdot(x \parallel y)$
LM4	$(x + y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} z + y \mathbin{⌊\!⌊} z$
LM5	$(\sum_{d:D} X(d)) \mathbin{⌊\!⌊} y = \sum_{d:D} X(d) \mathbin{⌊\!⌊} y$
S1	$x \mid y = y \mid x$
S2	$(x \mid y) \mid z = x \mid (y \mid z)$
S3	$x \mid \tau = x$
S4	$\alpha \mid \delta = \delta$
S5	$(\alpha\cdot x) \mid \beta = \alpha \mid \beta\cdot x$
S6	$(\alpha\cdot x) \mid (\beta\cdot y) = \alpha \mid \beta\cdot(x \parallel y)$
S7	$(x + y) \mid z = x \mid z + y \mid z$
S8	$(\sum_{d:D} X(d)) \mid y = \sum_{d:D} X(d) \mid y$
TC1	$(x \mathbin{⌊\!⌊} y) \mathbin{⌊\!⌊} z = x \mathbin{⌊\!⌊} (y \parallel z)$
TC2	$x \mathbin{⌊\!⌊} \delta = x\cdot\delta$
TC3	$(x \mid y) \mathbin{⌊\!⌊} z = x \mid (y \mathbin{⌊\!⌊} z)$

Table 3: Axioms for the parallel composition operators