First Name:	
Last Name:	
Matriculation Number:	

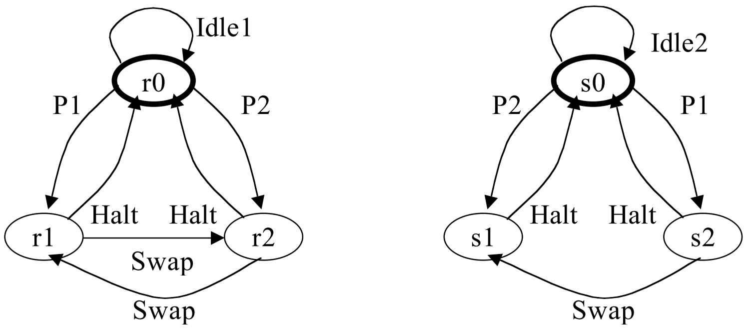
Leopold-Franzens Universität Innsbruck
Fakultät für Informatik
Dr. B. Schätz

SS 2005
Final Exam
06. July 2005

Final Exam " Specification of Distributed Systems"

<u>Question 1</u> Labeled Transition System (9 credits)

The LTS for Process 1 and Process 2, resp., each describe a process participating in a leader selection protocol. Action "P1" selects Process 1, "P2" selects Process 2, "Halt" nullifies the election, "Swap" swaps leadership between the processes. Process 1 has alphabet { Idle1, P1, P2, Halt, Swap}, Process 2 has alphabet { Idle2, P1, P2, Halt, Swap }. r0 and s0, resp., are the initial states of Process 1 and Process2, resp.



Process 1 Process 2

i. (3 credits) Mark each of the following statements as either true ("T") or false ("F"):

a) ☐T ☐F The sequence " Idle1 Idle2 P1 Swap Swap" is an observation trace of the synchronized product transition system of Process 1 and Process 2

False; "Idle1 Idle2 P1 Swap Swap" is not a possible observation trace of the product automaton, since "Idle2 P1 Swap Swap" is not a trace of Process 2.

b) ☐T ☐F The causal order of each event structure describing an execution of the synchronized product transition system is linear.

False; Each event structure containing concurrent actions Idle1 and Idle2 is not linear.

c) ☐T ☐F The synchronized product transition system is deterministic.

True, since the transition systems of each process are deterministic.

ii. (6 credits) Define the synchronized product transition system (S,A,S_0,T) of Process 1 and Process 2
LTS = (S,A,S_0,T) with

- S = { r0s0, r1s2,r2s1 } (1 credit)
- A = { Idle1, Idle2, P1, P2, Hold, Swap} (1 credit)
- S_0 = { r0s0 } (1 credit)

- T = { (r0s0,Idle1,r0s0), (r0s0,Idle2,r0s0), (r0s0,P1,r1s2), (r0s0,P2, r2s1), (r1s2,Hold,r0s0), (r2s1,Hold,r0s0), (r1s2,Swapr2s1) } (3 credits with 1 credit for each class of transition, -1 credits for transition (r2s1,Swap,r1s2), 0.5 credits for any other missing transition per class)

<u>Question 2</u> Data Flow Systems (4 credits)

A demux component takes an integer value on its input channel, and distributes it via one of its two output channels. Using untimed structured behavior, it can be described as the largest set DEMUX $\subseteq int^{\omega} \times int^{\omega} \times int^{\omega}$ with

$$
\begin{aligned}
(x,y.z) \in \text{DEMUX} \;\Leftrightarrow\; & (x = \langle\rangle \wedge y = \langle\rangle \wedge z = \langle\rangle) \vee \\
& (x = a \bullet r \wedge y = a \bullet s \wedge z = t \wedge (r,s,t) \in \text{DEMUX}) \vee \\
& (x = a \bullet r \wedge y = s \wedge z = a \bullet t \wedge (r,s,t) \in \text{DEMUX}) \vee
\end{aligned}
$$

i. (4 credits) Mark each of the following statements as either true ("T") or false ("F"):

a) ☐T ☐F (1 • 2 • 3, 1• 3, 2) ∈ DEMUX and (1• 2 • 3, 1, 2 • 3) ∈ DEMUX

True

b) ☐T ☐F DEMUX is deterministic

False, e.g. (1,1,<>) and (1,<>,1) ∈ DEMUX

c) ☐T ☐F DEMUX is input-enabled

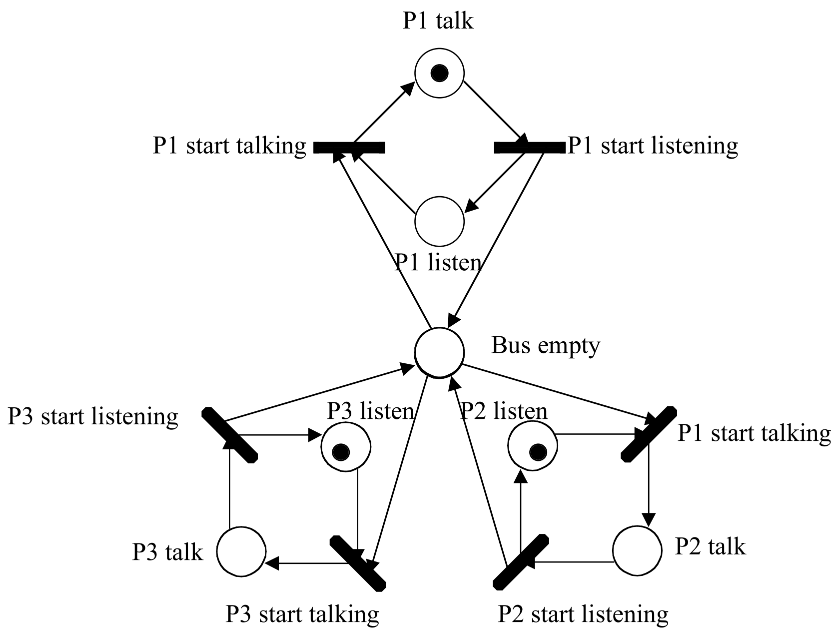
True.

d) ☐T ☐F DEMUX can be equivalently described by a Kahn function

False, since non-deterministic behavior can only be equivalently be described by a set of Kahn functions

<u>Question 3</u> Petri Nets (7 credits)

Consider the following Condition Event Net describing three components broad-casting over a common bus.



i. (3 credits) Mark each of the following statements as either true ("T") or false ("F"):

First Name:	
Last Name:	
Matriculation Number:	

a) ☐T ☐F The number of marks is constant throughout the execution of the net

False, since the configuration obtained by firing transition "P1 start listening" leads to 4 marks.

b) ☐T ☐F The marking {P1talk, P2talk, P3talk} is reachable from the initial marking.

False, since the marking is not reachable since all transitions "PX start talking" are in conflict.

c) ☐T ☐F The marking {P1 listen, P2 listen, P3 listen} is reachable from the initial marking.

The marking is not reachable since it must include the "Bus empty" mark.

ii. (2 credits) What is the smallest set of places including P1talk, P2talk and P3talk forming a place invariant cornering all transitions?
The set must additionally include "Bus empty" (1 credit)
The set may not contain any other location (1 credit)

iii. (1 credit) What is the number of marks in this set?
The number of marks is 1. (1 credit)

iv. (1 credits) Give an interpretation of the invariant concerning the collision of talking processes.
At most one process is talking: either exactly one process is talking or the bus is empty.

Question 4 Process Algebra (9 credits)

The following process terms with alphabet A = {PIN, open, close, lock, unlock} specify the lock/unlock functions of a digital vault:

- `S1(P)= PIN → unlock → open → P, S2(P) = PIN → lock → close → P`
- `V = μ P. (S1(P) | S2(P))`
- `S(Q) = PIN → ((unlock → open → Q) | (lock → close → Q))`
- `W = μ Q. S(Q)`

The additional process term specifies a user never using the lock function of the vault.

- `T(R) = PIN → unlock → open → R`
- `U = μ R . T(R)`

i. (3 credits) Mark each of the following statements as either true ("T") or false ("F"):

a) ☐T ☐F `PIN • unlock • open` is a complete interace trace of `V || U`

False; it is a partial interaction trace, but not a complete interaction trace, since action P is possible before deadlock.

b) ☐T ☐F The processes `V` and `W` have the same set of complete interaction traces.

True, the processes do have the same set of complete interaction traces.

c) ☐T ☐F `V || U` and `W || U` have the same set of complete interaction traces.

False; the combined processes do not have the same set of complete interaction traces, since `PIN` is a complete interaction trace of the former but not of the later.

ii. (4 credits) Show `Z = T(Z)` for `Z = W || U` by algebraic reasoning, including the law `(P | Q) || (a → R) = (P||a → R) | (Q||a → R)`.
First, we show that `Z = R(Z)` via

```
Z = (via. Def. Z)
W || U = (via Def. W, U)
(μ Q. (S(Q))||(μ R . T(R)) = (via μ P. F(P) = F(μ P. F(P)) )
S(μ Q. (S(Q))||T(μ R . T(R)) (via Def. W, U)
S(W) || T(U) =  (via Def. S, T)
(PIN → (unlock → open → W)|(lock → close → W))||(PIN → unlock → open
→ U) = (via (a → P)||(a → Q) = a → P||Q)
PIN → (((unlock → open → W)|(lock → close → W))||(unlock → open →
U)) = (via above law)
PIN → (((unlock → open → W)||(unlock → open → U))|((lock → close →
W)||(unlock → open → U)))= (via a → P||b → Q = a → P||Q for a ≠ b)
PIN → ((unlock → open → (W||U))|STOP) = (via P | STOP = P)
PIN → unlock → open → (W||U) = (via Def. Z)
PIN → unlock → open → Z = (via Def. T)
T(Z)
```

iii. (2 credits) Show `U = W || U` by algebraic reasoning using the result of ii.

```
Z = T(Z) ⇒ (via law of induction)
Z = μ R. T(R) ⇒ (via Def. Z)
W || U = μ R. T(R) ⇒ (via Def. U)
W || U = U
```

Question 5   Temporal Logics (11 credits)

To define properties about the ledership-election process in Question 2 the relevant propositions are „P1 is leader", „P2 is leader", „P1 is ready", „P2 is ready".A process is ready if it can react on selection actions „P1" and „P2".

ii. (6 credits) Define a Kripke structure K = (S,S_0,T,P,O)
K = (S,S_0,T,P,O) with
- S = { r0s0, r1s2,r2s1 } (0.5 credit)
- S_0 = { r0s0 } (0.5 credit)
- T = { (r0s0,r0s0), (r0s0,r1s2), (r0s0,r2s1), (r1s2,r0s0), (r2s1,r0s0), (r1s2,r2s1) } (2 credits with 0.5 credit for each class of transition, -0.5 credits for transition (r2s1,r1s2))
- P = {"P1 is leader", "P2 is leader", "P1 is ready", "P2 is ready"} (1 credit)
- O = { (r0s0, {"P1 is ready", "P2 is ready"}), (r1s2,"P1 is leader"), (r2s1,"P2 is leader")} (2 credit)

iii. (3 credits) Mark each of the following statements as either true ("T") or false ("F"):

a) ☐T ☐F (GF "P1 is ready") ⇒ (GF "P1 is leader") holds in K

False. Since the path consisting of only of states (r0s0) contradicts that: since G "P1 is ready" holds, but G (¬ "P1 is leader")

b) ☐T ☐F AG ¬ ("P1 is leader" ∧ "P2 is leader") holds in K

First Name:	
Last Name:	
Matriculation Number:	

True, since the observation for all states is either “P1 is ready” ∧ “P2 is ready”, “P1 is leader”, or “P2 is leader”.

c) ☐T ☐F A(GF “P1 is leader”) ⇒ GF(“P2 is leader”∨ “P1 is ready”))) holds in K

True, since “P1 is leader” ⇒ X(“P2 is leader”∨ “P1 is ready”)

iv. (2 credits) Define the property “During some execution, both processes will always eventually become leader”.
E(GF “P1 is leader) ∧ (GF “P2 is leader”)) (1 credit for correct use of E, 1 credit for GF)