Analysis of Distributed Systems

Solutions to the Exercises

Máté Tejfel

September 19, 2013

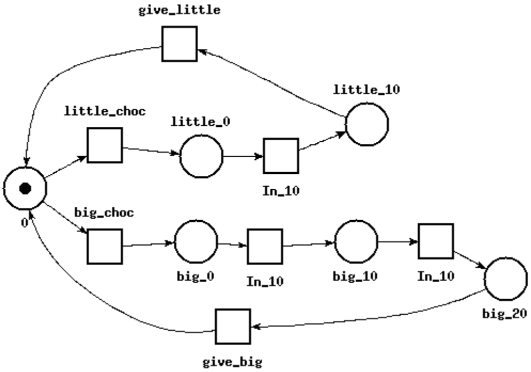
Theme I

Part I/a

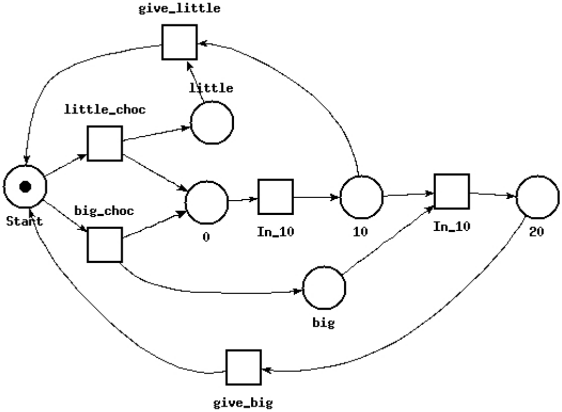
Agenda

Exercise 1.1

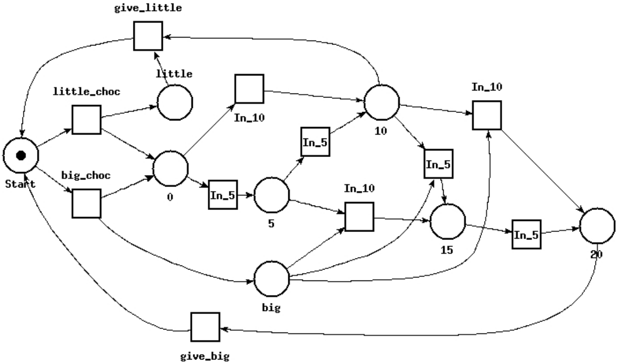
Solution A.



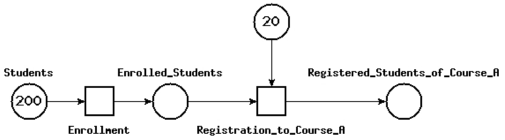
Solution B.


give_little
little
little_choc
Start
big_choc
0
In_10
10
In_10
20
big
give_big

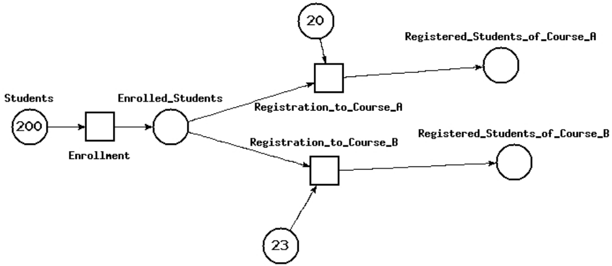
Exercise 1.2


give_little
little
little_choc
In_10
10
In_10
Start
big_choc
0
In_5
5
In_5
In_5
In_10
15
In_5
20
big
give_big

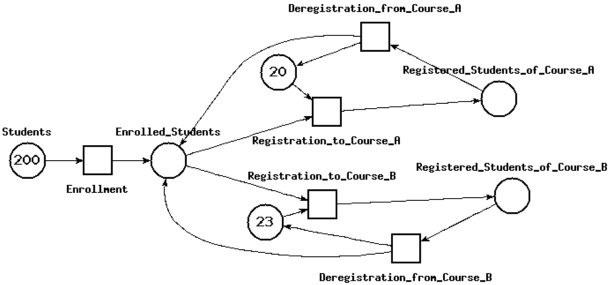
Exercise 1.3



Exercise 1.4



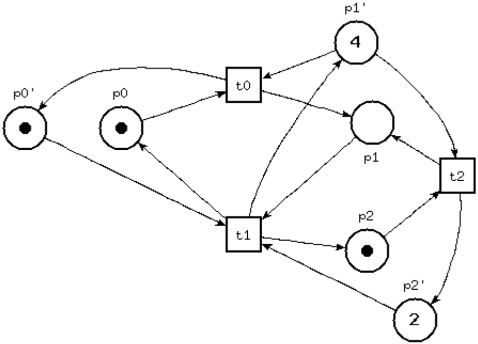
Exercise 1.5


Deregistration_from_Course_A
20
Registered_Students_of_Course_A
Students
Enrolled_Students
Registration_to_Course_A
200
Enrollment
Registration_to_Course_B
Registered_Students_of_Course_B
23
Deregistration_from_Course_B

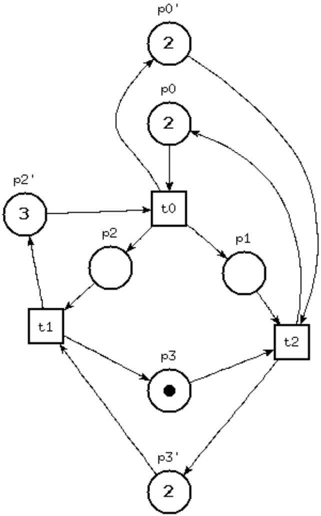
Exercise 1.6

Every prefix of t_0, t_4 and t_1, t_3, t_0, t_4.

Exercise 1.7



Exercise 1.8



Exercise 1.9

Every prefix of t_0, t_1, t_2, t_3, t_0.

Exercise 1.10

Every prefix of $t_1, t_3, t_2, t_0, t_1, t_3$.

Agenda

Exercise 2.1

From A initial marking:

- M_i : $t_1, t_3, t_1, t_3, t_2, t_1, t_3$
- M_j : not reachable

From B initial marking:

- M_i : $t_1, t_3, t_1, t_4, t_2, t_1, t_3, t_1, t_3, t_2, t_1, t_3$
- M_j : $t_1, t_3, t_2, t_1, t_4, t_1, t_3, t_2, t_1, t_1, t_3, t_3, t_1, t_1, t_2, t_2$

Exercise 2.2

Reachable markings:

- $\{1, 0, 1, 0\}$
- $\{1, 0, 0, 1\}$
- $\{0, 0, 1, 1\}$
- $\{1, 1, 0, 0\}$
- $\{0, 1, 1, 0\}$
- $\{0, 1, 0, 1\}$
- $\{0, 0, 0, 2\}$
- $\{0, 0, 2, 0\}$
- $\{2, 0, 0, 0\}$

Exercise 2.3

Ex 2.1, A initial marking

- Boundedness : No
- Safety : No
- Liveness : t_4 is dead

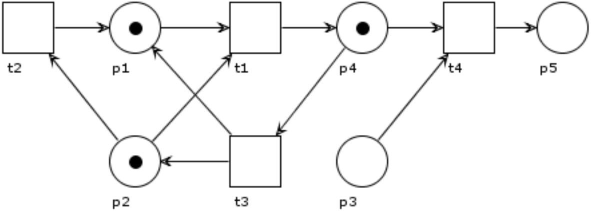
Ex 2.1, B initial marking

- Boundedness : No
- Safety : No
- Liveness : L_1

Ex 2.2

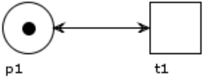
- Boundedness : Yes (k=2)
- Safety : No
- Liveness : L_2

Exercise 2.4

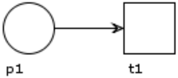


The $p_2 \Rightarrow t_1$ edge was added.

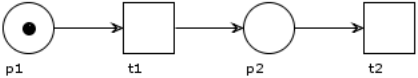
Exercise 2.5



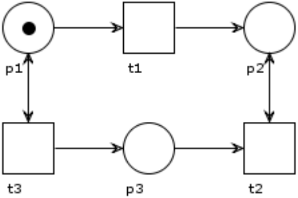
Exercise 2.6



Exercise 2.7

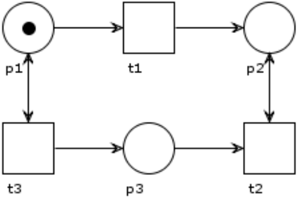


Exercise 2.8



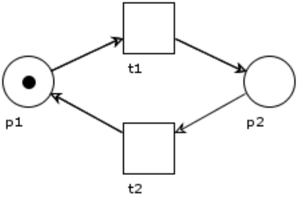
t_3 is L_2 but not L_3

Exercise 2.9



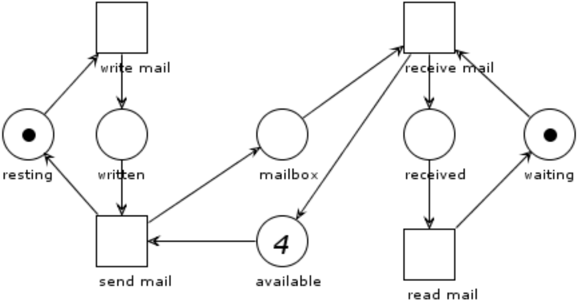
t_1 is L_3 but not L_4

Exercise 2.10



Exercise 2.11

Mailbox capacity is 4.



Agenda

Exercise 3.1

Reversibility: Yes

Home states:

- $\{p_1, p_5\}$
- $\{p_5, p_6\}$
- $\{p_2, p_3\}$
- $\{p_3, p_7\}$

It models mutual exclusion, e.g. an intersection with two traffic lights.

Exercise 3.2

- M_1 : Yes
- M_2 : Yes
- M_3 : Yes
- M_4 : No
- M_5 : No

Exercise 3.3

Exercise 3.1 :

- Not persistent, (t_1, t_2) is the reason.
- Persistent after reversing p_4 all four edges. It is now a counter instead of a mutex!

Exercise 3.2 :

- Not persistent, the reasons are (t_1, t_4), (t_1, t_2).
- Persistent without t_1.

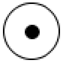
Exercise 3.4

- $M_0 \,:\, d_{1,2} = 1,\; d_{3,2} = 1$
- $M_1 \,:\, d_{1,3} = 1,\; d_{2,3} = 0$
- $M_2 \,:\, d_{2,3} = 2$

Exercise 3.5

Reductions:

- fusion of series places: (yellow1 + yg1 + green1) and (yellow2 + yg2 + green2)
- fusion of series transitions: (ry1 + yellow1green1 + gr1) and (ry2 + yellow2green2 + gr2)
- elimination of self-loop places: (red1) and (red2)
- fusion of series transitions: (ry1yellow1green1gr1 + safe2 + ry2yellow2green2gr2)
- elimination of self loop transitions: (ry1yellow1green1gr1safe2ry2yellow2green2gr2)



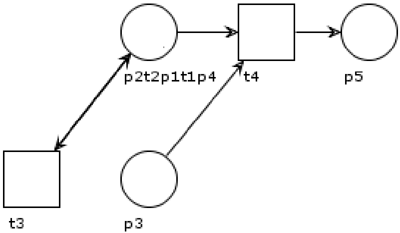
safe1ry1yellow1green1gr1safe2ry2yellow2green2gr2

Exercise 3.6

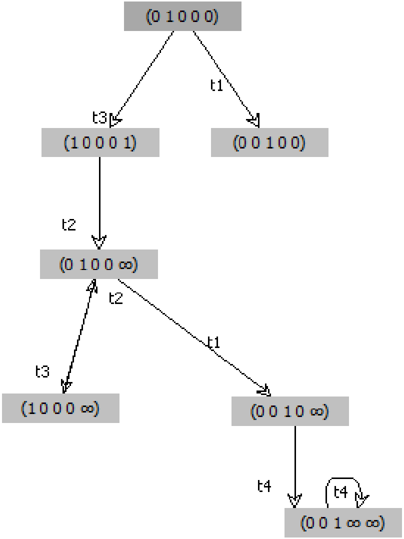
Reductions:

- fusion of series places: (p2 + t2 + p1)
- fusion of series places: (p2t2p1 + t1 + p4)

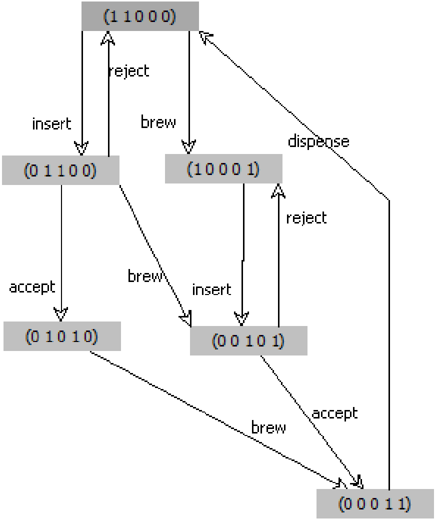
Note: the (t3 ⇒ p2t2p1t1p4) edge is double!



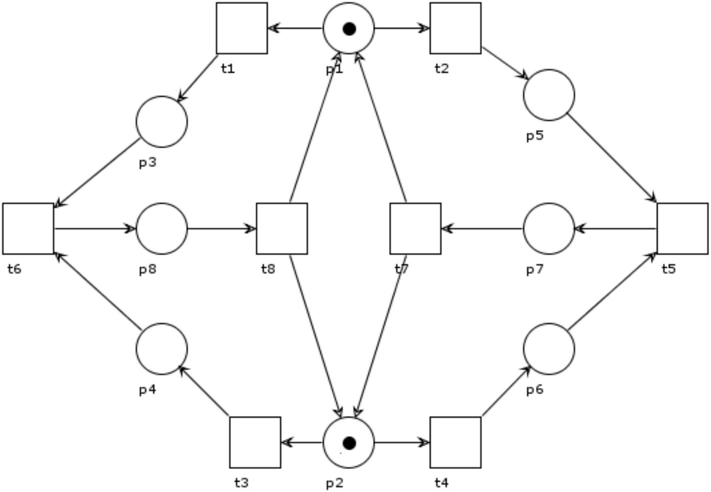
Exercise 3.7



Exercise 3.8


(1 1 0 0 0)
reject
insert
brew
dispense
(0 1 1 0 0)
(1 0 0 0 1)
reject
accept
brew
insert
(0 1 0 1 0)
(0 0 1 0 1)
accept
brew
(0 0 0 1 1)

Exercise 3.9



Exercise 3.10

Yes, because it is live and safe (k=1 bounded).

Agenda

Exercise 4.1

Not AC (p2, p4), not live (t4) and not safe (p3)!

Exercise 4.2

MG, live and not safe (p3)!

Exercise 4.3

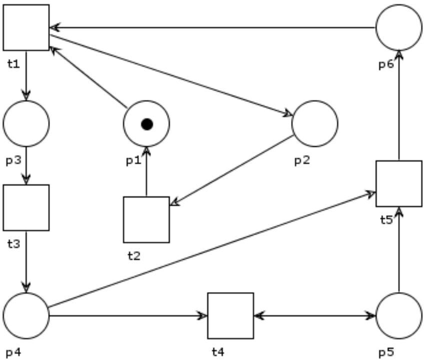
FC, not live and safe!

Exercise 4.4

Modifications:

- t5 $\Rightarrow$ p4 edge reversed
- p5 $\Rightarrow$ t4 edge added

p4, p5 are the interesting ones!

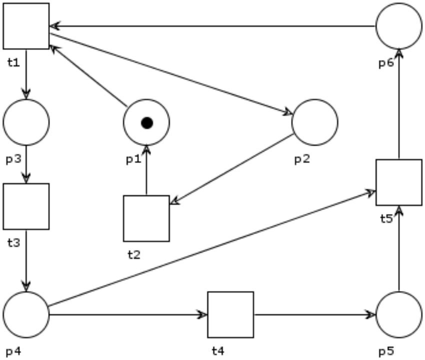


Exercise 4.5

Modifications:

- t5 $\Rightarrow$ p4 edge reversed

p4, p5 are the interesting ones!



Exercise 4.6

- p1 is a source place
- {p2, p3} is a siphon
- {p4, p5} is a trap
- p6 is a sink place

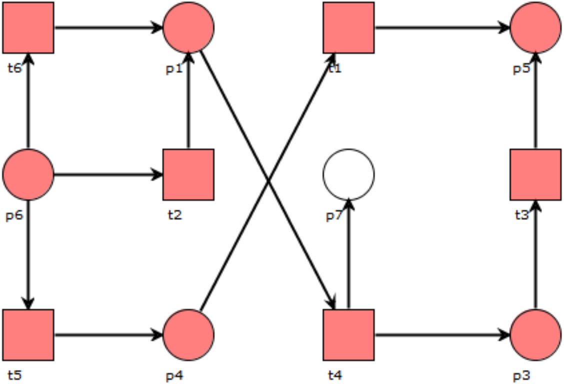
Exercise 4.7

- $\{t2 \Rightarrow p3, t3 \Rightarrow p3\}$ is a minimal FAS
- $\{p1 \Rightarrow t2, t3 \Rightarrow p3\}$ is a minimal FAS
- $\{t2 \Rightarrow p3, p1 \Rightarrow t3\}$ is a minimal FAS
- $\{p1 \Rightarrow t2, t1 \Rightarrow t3\}$ is a minimal FAS
- $p3 \Rightarrow t1$ is a minimum and also a minimal FAS
- $t1 \Rightarrow p1$ is a minimum and also a minimal FAS
- x set of edges is a FAS if x contains at least one of the previous 6 FAS-es

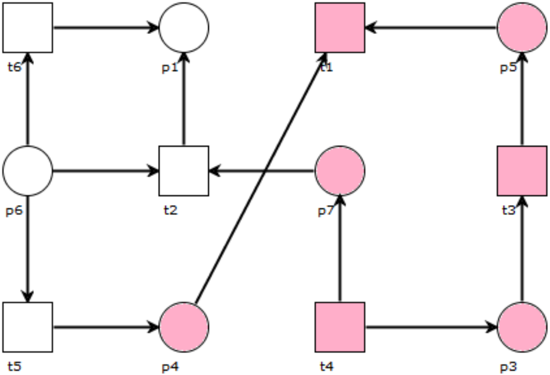
Exercise 4.8

It is safe, because for every reachable marking, the set of marked edges is a minimal FAS.

Exercise 4.9

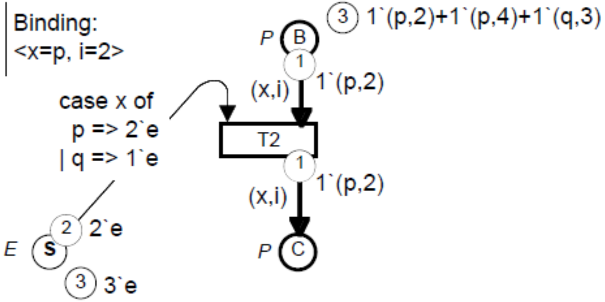


Exercise 4.10

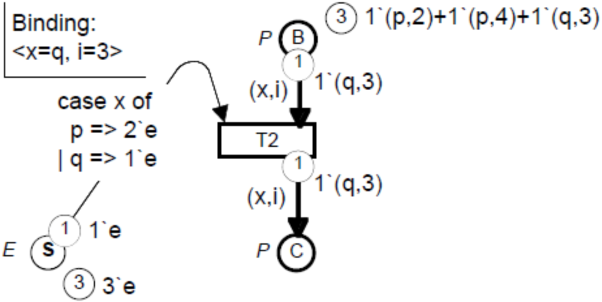


Agenda

Exercise 5.1


Binding:
<x=p, i=2>
case x of
p => 2`e
| q => 1`e
3 1`(p,2)+1`(p,4)+1`(q,3)
P
B
1
(x,i)
1`(p,2)
T2
1
(x,i)
1`(p,2)
P
C
E
S
2
2`e
3
3`e

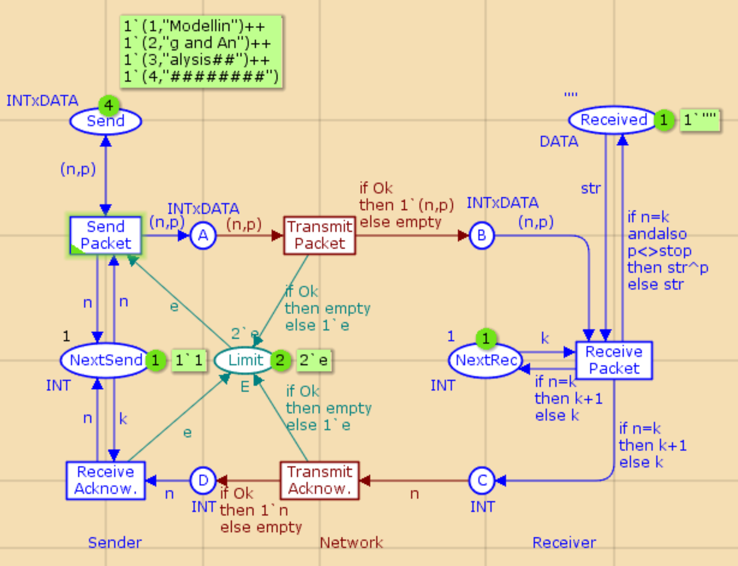
Exercise 5.2


Binding:
<x=q, i=3>
case x of
p => 2`e
| q => 1`e
E
S
1
1`e
3
3`e
P
B
3
1`(p,2)+1`(p,4)+1`(q,3)
1
1`(q,3)
(x,i)
T2
1
1`(q,3)
(x,i)
P
C

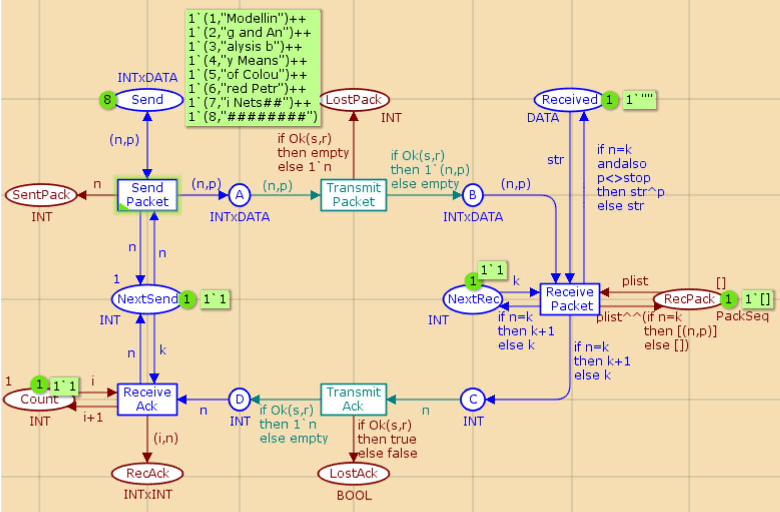
Exercise 5.3

CPN Tools can be dowloaded from http://cpntools.org/

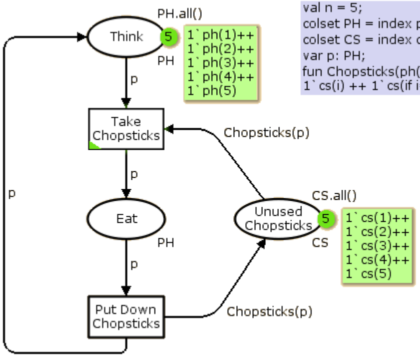
Exercise 5.4


1`(1,"Modellin")++
1`(2,"g and An")++
1`(3,"alysis##")++
1`(4,"########")
INTxDATA
4
Send
(n,p)
Send Packet
INTxDATA
(n,p)
A
(n,p)
Transmit Packet
if Ok
then 1`(n,p)
else empty
INTxDATA
B
(n,p)
""
Received
1
1`""
DATA
str
if n=k
andalso
p<>stop
then str^p
else str
n
n
e
1
NextSend
1
1`1
2`e
Limit
2
2`e
if Ok
then empty
else 1`e
1
1
NextRec
k
Receive Packet
INT
E
INT
if Ok
then empty
else 1`e
if n=k
then k+1
else k
n
k
e
if n=k
then k+1
else k
Receive Acknow.
D
n
INT
if Ok
then 1`n
else empty
Transmit Acknow.
n
C
INT
Sender
Network
Receiver

Exercise 5.5

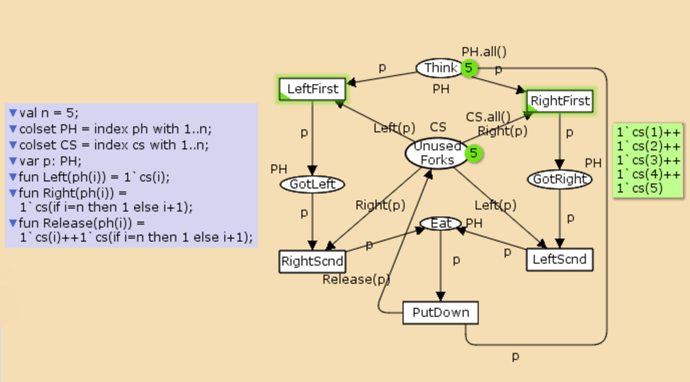

1`(1,"Modellin")++
1`(2,"g and An")++
1`(3,"alysis b")++
1`(4,"y Means")++
1`(5,"of Colou")++
1`(6,"red Petr")++
1`(7,"i Nets##")++
1`(8,"########")
INTxDATA
Send
8
(n,p)
Send Packet
SentPack
n
INT
(n,p)
A
INTxDATA
(n,p)
Transmit Packet
LostPack
INT
if Ok(s,r)
then empty
else 1`n
if Ok(s,r)
then 1`(n,p)
else empty
B
INTxDATA
(n,p)
Received
1
1`""
DATA
str
if n=k
andalso
p<>stop
then str^p
else str
n
n
1
NextSend
1
1`1
INT
n
k
NextRec
1
1`1
k
INT
if n=k
then k+1
else k
Receive Packet
plist
[]
RecPack
1
1`[]
PackSeq
plist^^(if n=k
then [(n,p)]
else [])
if n=k
then k+1
else k
1
Count
1
1`1
i
i+1
INT
Receive Ack
n
D
INT
if Ok(s,r)
then 1`n
else empty
Transmit Ack
n
C
INT
if Ok(s,r)
then true
else false
(i,n)
RecAck
INTxINT
LostAck
BOOL

Exercise 5.6

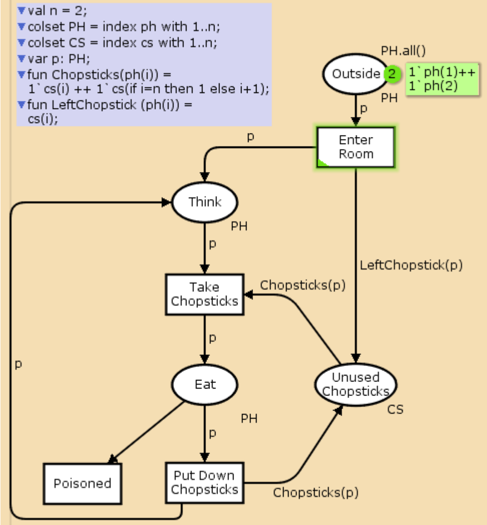



```
val n = 5;
colset PH = index ph with 1..n;
colset CS = index cs with 1..n;
var p: PH;
fun Chopsticks(ph(i)) =
1`cs(i) ++ 1`cs(if i=n then 1 else i+1);
```

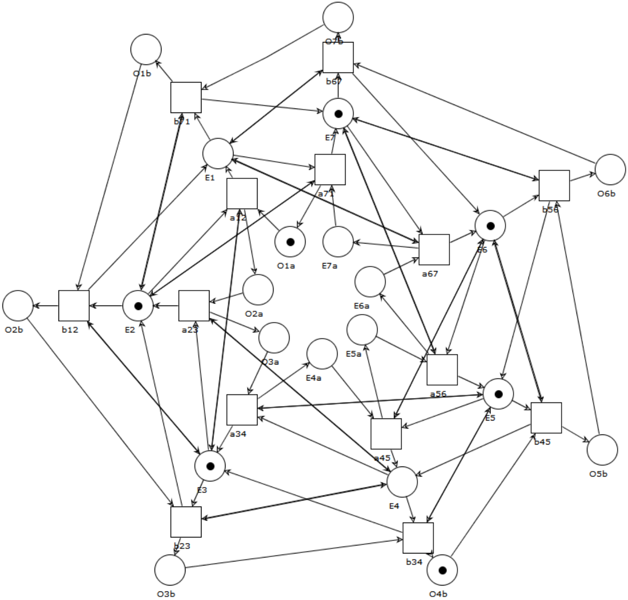
Exercise 5.7


val n = 5;
colset PH = index ph with 1..n;
colset CS = index cs with 1..n;
var p: PH;
fun Left(ph(i)) = 1`cs(i);
fun Right(ph(i)) =
1`cs(if i=n then 1 else i+1);
fun Release(ph(i)) =
1`cs(i)++1`cs(if i=n then 1 else i+1);
PH.all()
Think
5
p
PH
LeftFirst
RightFirst
CS.all()
Left(p)
CS
Right(p)
Unused Forks
GotLeft
GotRight
Right(p)
Left(p)
Eat
RightScnd
LeftScnd
Release(p)
PutDown
1`cs(1)++
1`cs(2)++
1`cs(3)++
1`cs(4)++
1`cs(5)

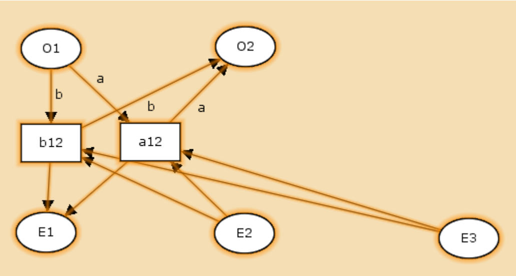
Exercise 5.8


val n = 2;
colset PH = index ph with 1..n;
colset CS = index cs with 1..n;
var p: PH;
fun Chopsticks(ph(i)) =
1`cs(i) ++ 1`cs(if i=n then 1 else i+1);
fun LeftChopstick (ph(i)) =
cs(i);
PH.all()
Outside
2
1`ph(1)++
1`ph(2)
PH
p
Enter Room
p
Think
PH
p
LeftChopstick(p)
Take Chopsticks
Chopsticks(p)
p
p
Eat
Unused Chopsticks
CS
PH
p
Poisoned
Put Down Chopsticks
Chopsticks(p)

Exercise 5.9


O7b
b67
O1b
b71
E7
E1
a71
O6b
b56
a12
O1a
E7a
E6
a67
O2a
E6a
O2b
b12
E2
a23
O3a
E5a
E4a
a56
E5
a34
a45
b45
O5b
E3
E4
b23
b34
O3b
O4b

Exercise 5.10



Theme II

Part I/b

Agenda

Exercise 1.1

$$\Sigma_0 = (S_0, T_0, W_0, \lambda_0, M_0)$$
$$S_0 = \{p_0, p_1\}$$
$$T_0 = \{t_0, t_1, t_2\}$$
$$W_0 = ((TS \cup ST) \times \{1\}) \cup (((S \times T) \setminus ST \cup (T \times S) \setminus TS) \times \{0\})$$
$$\lambda_0 = \{(p_0, i), (p_1, i), (t_0, rr), (t_1, br), (t_2, bb)\}$$
$$M_0 = \{(p_0, 2), (p_1, 1)\}$$

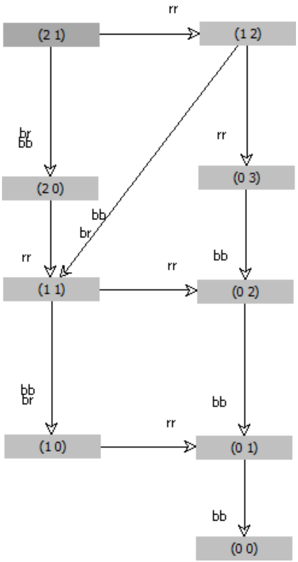
where

$$TS = \{(t_1, p_0), (t_0, p_1)\} \text{ and}$$
$$ST = \{(p_0, t_0), (p_0, t_1), (p_1, t_1), (p_1, t_2)\}$$

Exercise 1.2

The Petri net is 2-bounded, therefore not safe.

Exercise 1.3



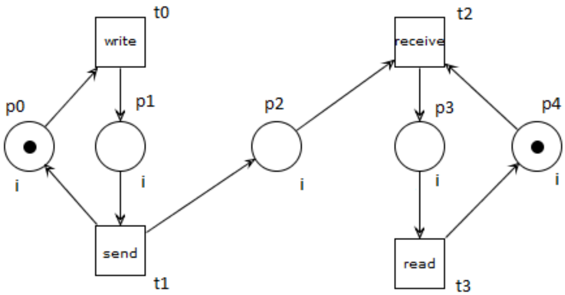
Exercise 1.4

Only the first three step is applicable. Two markings left on p1.

Exercise 1.5

It is applicable. No markings left at the end.

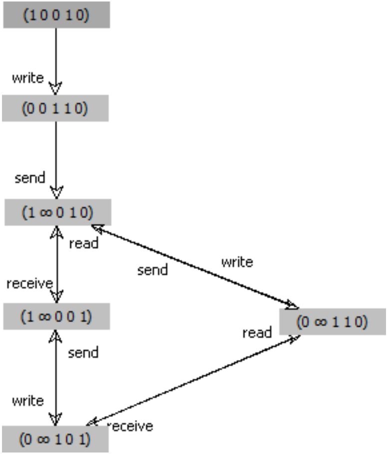
Exercise 1.6



Exercise 1.7

The Petri net is unbounded, therefore not safe, but live.

Exercise 1.8


(1 0 0 1 0)
write
(0 0 1 1 0)
send
(1 ∞ 0 1 0)
read
receive
send
write
(1 ∞ 0 0 1)
(0 ∞ 1 1 0)
read
send
write
receive
(0 ∞ 1 0 1)

Exercise 1.9

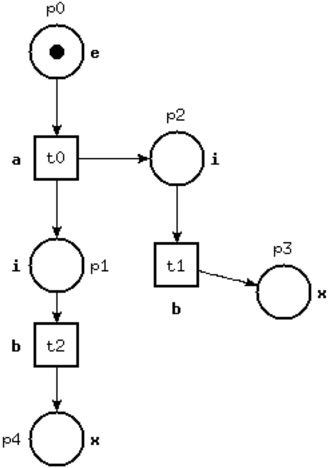
$M_0 = \{(p_1, 1), (p_4, 1)\}$

Exercise 1.10

The last step is not enabled. It would require to have tokens simultaneously on p4 and p5.

Agenda

Exercise 2.1



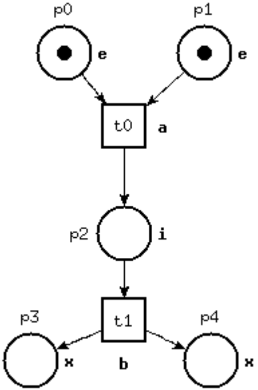
Exercise 2.2

The net is ex-directed but not ex-exclusive. Marking $(0, 0, 0, 0, 0, 1, 1)$ is reachable (applying step sequence $\{t_0\}, \{t_5\}$). The following transitions are independent of transition t_0: t_3, t_6.

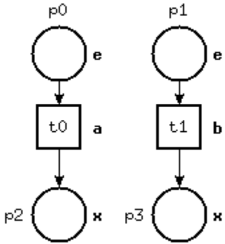
Exercise 2.3

The ex-asymmetric transitions of the net are the following: t_0: t_1.

Exercise 2.4



Exercise 2.5



Exercise 2.6

The net is a Petri box, but the marking $(0, 1, 1, 1)$ (which is not clean) is reachable from $M_{\bullet\Sigma}$, namely, it is not a static box.

Exercise 2.7

The net is a Petri box, but the marking $(0, 0, 0, 2)$ (which is not safe) is reachable from $M_{\bullet\Sigma}$, namely, it is not a dynamic box.

Exercise 2.8

Marking $(0, 1, 0, 1)$ is reachable from the initial marking (applying step $\{t_1\}$), so the net is not ex-exclusive.

Exercise 2.9

The net is an entry box, since it is a Petri box, and every marking reachable from $M_{\bullet\Sigma}$ and $M_{\Sigma\bullet}$ is safe and clean.

Exercise 2.10

Transitions t_1, t_2 are independent. Marking $(0, 1, 0, 0, 1)$ is reachable from the initial marking (applying step sequence $\{t_0\}, \{t_2\}$), so the net is not ex-exclusive.

Agenda

Exercise 3.1

Yes. The following execution is enabled.
$(\{p_0, p_1\}, \emptyset)\ [\{t_0, t_3\} >\ \ (\{p_2, p_3\}, \emptyset)\ [\{t_4\} : \{t_1\}^+ >$
$(\{p_1\}, \{t_1\})\ [\{t_3\} : \{t_1\}^- >\ \ (\{p_0, p_3\}, \emptyset)$.

Exercise 3.2

Yes. The following execution is enabled.
$(\{p_0, p_1\}, \emptyset)\ [\{t_0, t_3\} >\ \ (\{p_2, p_3\}, \emptyset)\ [\{t_2\} : \{t_5\}^+ >$
$(\{p_4, p_5\}, \{t_5\})$.

Exercise 3.3

The following complex markings are directly reachable from $\mathcal{M}'' = (\{p_0\}, \{t_5\})$.
$(\{p_0, p_5\}, \emptyset)$; $(\{p_2, p_5\}, \emptyset)$; $(\{p_2\}, \{t_5\})$; $(\{p_5\}, \{t_0\})$; $(\emptyset, \{t_0, t_5\})$

Exercise 3.4

No. Execution
$(\{p_0\},\emptyset)\ [\{t_2\}>\ (\{p_2,p_3\},\emptyset)\ [\{t_3\}:\{t_0\}^+>\ (\{p_5\},\{t_0\})$
is enabled. However t_1 is not enabled in $(\{p_5\},\{t_0\})$.

Exercise 3.5

Yes. The following execution is enabled.
$(\{p_0\},\emptyset)\ [\{t_2\}>\ (\{p_2,p_3\},\emptyset)\ [\{t_0\}:\{t_3\}^+>$
$(\{p_4\},\{t_3\})\ [\{t_1\}^+:\{t_3\}^->\ (\{p_5\},\{t_1\})$.

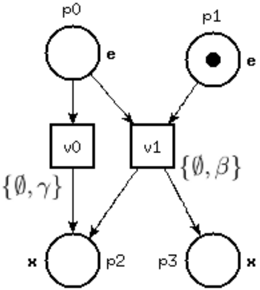
Exercise 3.6

Yes. The following execution is enabled.
$(\{p_0,p_1\},\emptyset)\ [\{t_0\}>\ (\{p_2,p_3\},\emptyset)\ [\{t_3\}:\{t_1\}^+>$
$(\{p_4\},\{t_1\})\ [\{t_2\}^+:\{t_1\}^->\ (\{p_2\},\{t_2\})$.

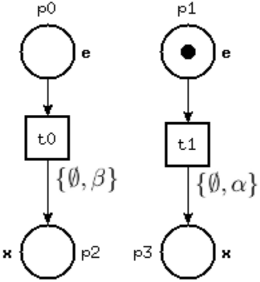
Exercise 3.7

Yes. The following execution is enabled.
$(\{p_0,p_1\},\emptyset)\ [\{t_0\}>\ (\{p_2,p_3\},\emptyset)\ [\{t_3\}:\{t_1\}^+>$
$(\{p_4\},\{t_1\})\ [\{t_2\}^+>\ (\emptyset,\{t_1,t_2\})$.

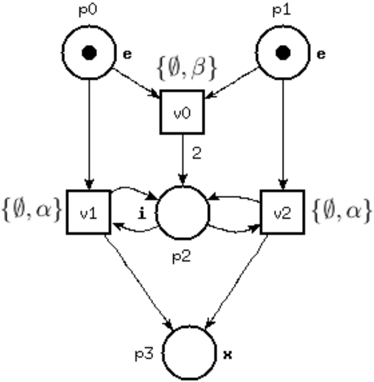
Exercise 3.8



Exercise 3.9

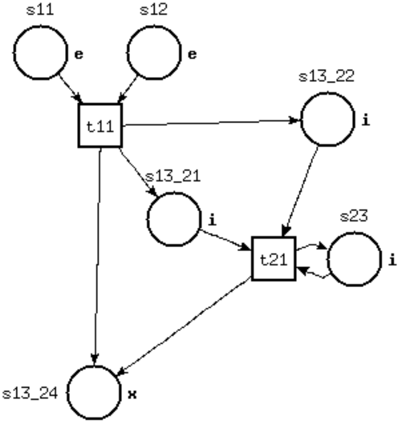


Exercise 3.10

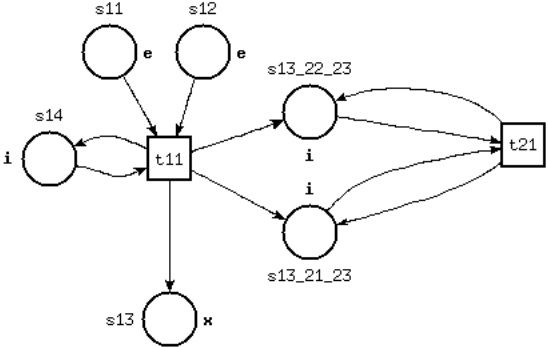


Agenda

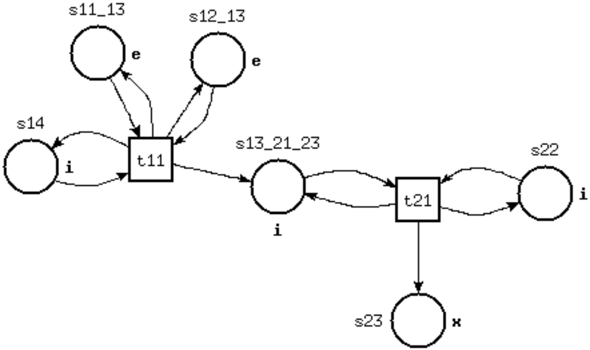
Exercise 4.1



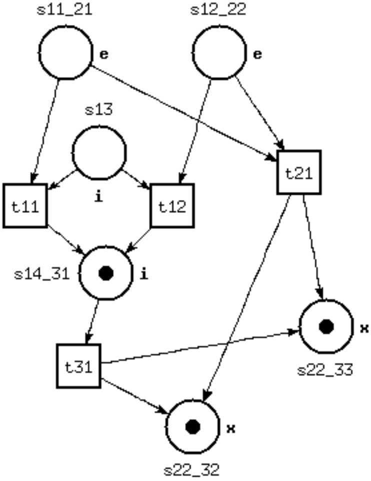
Exercise 4.2



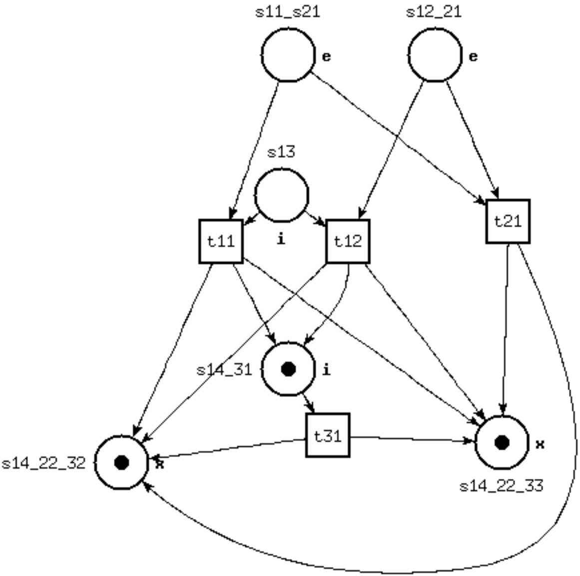
Exercise 4.3



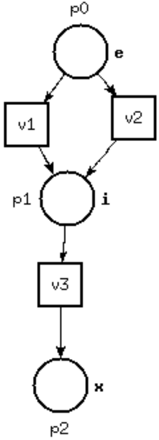
Exercise 4.4



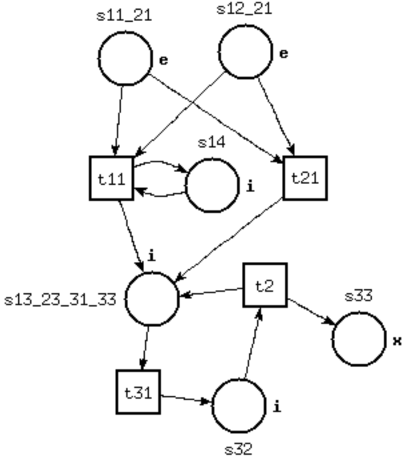
Exercise 4.5


s11_s21
s12_21
e
e
s13
i
t11
t12
t21
s14_31
i
t31
s14_22_32
x
x
s14_22_33

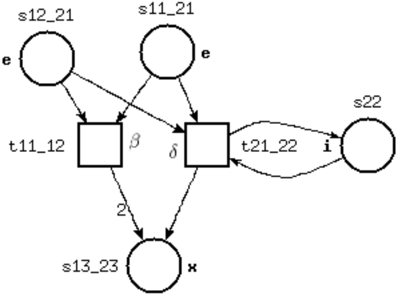
Exercise 4.6



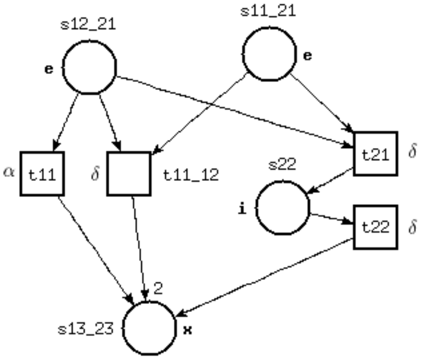
Exercise 4.7



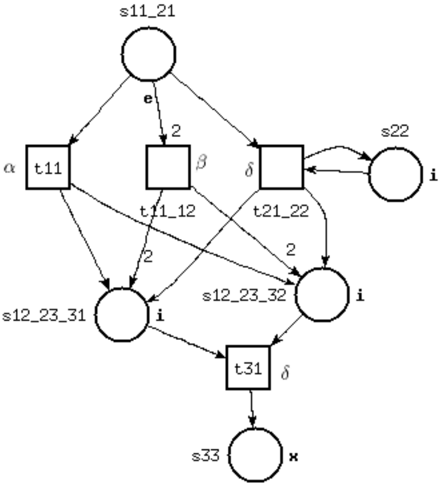
Exercise 4.8



Exercise 4.9



Exercise 4.10



Theme III

Part II

Agenda

Exercise 1.1

$p = a(bnil + cnil)$

Exercise 1.2

$p = a(bnil + cq), \; q = dp$

Exercise 1.3

$a(bnil + c(dnil + bdnil)) \mid\mid acbnil$
$\xrightarrow{a} \; bnil + c(dnil + bdnil) \mid\mid cbnil$
$\xrightarrow{c} \; dnil + bdnil \mid\mid bnil$
$\xrightarrow{b} \; dnil \mid\mid nil$
$\not\rightarrow$
Consequently the process corresponds to the environment.

Exercise 1.4

- $abdnil + a(dnil + cnil) \mid\mid a(bnil + cnil)$
  $\xrightarrow{a}$ $dnil + cnil \mid\mid bnil + cnil$
  $\xrightarrow{c}$ $nil \mid\mid nil$
  $\not\rightarrow$
- $abdnil + a(dnil + cnil) \mid\mid a(bnil + cnil)$
  $\xrightarrow{a}$ $bdnil \mid\mid bnil + cnil$
  $\xrightarrow{b}$ $dnil \mid\mid nil$
  $\not\rightarrow$

  Consequently the process corresponds to the environment.

Exercise 1.5

- $abnil + adnil \mid\mid abnil + adnil$
  $\xrightarrow{a}$ $bnil \mid\mid bnil$
  $\xrightarrow{b}$ $nil \mid\mid nil$
  $\not\rightarrow$
- $abnil + adnil \mid\mid abnil + adnil$
  $\xrightarrow{a}$ $dnil \mid\mid dnil$
  $\xrightarrow{d}$ $nil \mid\mid nil$
  $\not\rightarrow$
- $abnil + adnil \mid\mid abnil + adnil$
  $\xrightarrow{a}$ $bnil \mid\mid dnil$
  $\not\rightarrow$
- $abnil + adnil \mid\mid abnil + adnil$
  $\xrightarrow{a}$ $dnil \mid\mid bnil$
  $\not\rightarrow$
  Consequently the process does not correspond to the environment.

Exercise 1.6

$nil,\ anil,\ a(bnil + dnil),\ anil + cnil,\ a(bnil + dnil + cnil),\ \ldots$

Exercise 1.7

$\tau(a(bnil + c(dnil + bdnil)) = \{ab,\ acd,\ acbd\}$

Exercise 1.8

$\tau'(a(bnil + c(dnil + bdnil)) = \{\epsilon,\ a,\ ab,\ ac,\ acd,\ acb,\ acbd\}$

Exercise 1.9

$abnil + a(cnil + dnil)$
$\stackrel{A5}{=} a(bnil + (cnil + dnil))$
$\stackrel{A1}{=} a((bnil + cnil) + dnil)$
$\stackrel{A5}{=} a(bnil + cnil) + adnil$

Exercise 1.10

$$
\begin{aligned}
& a((bcnil + dnil) + d(cnil + nil)) \\
\stackrel{A4}{=}\ & a((bcnil + dnil) + dcnil) \\
\stackrel{A5}{=}\ & a(bcnil + dnil) + adcnil \\
\stackrel{A5}{=}\ & (abcnil + adnil) + adcnil \\
\stackrel{A1}{=}\ & abcnil + (adnil + adcnil) \\
\stackrel{A4}{=}\ & ab(cnil + nil) + (adnil + adcnil) \\
\stackrel{A5}{=}\ & ab(cnil + nil) + a(dnil + dcnil) \\
\stackrel{A5}{=}\ & ab(cnil + nil) + a(d(nil + cnil))
\end{aligned}
$$

Agenda

Exercise 2.1

$CSP = 10 \rightarrow CS10,$
$CS10 = (5 \rightarrow CS15 \mid little\_choc \rightarrow CS)$
$CS15 = big\_choc \rightarrow CS,$
where $\alpha CSP = \{5, 10, little\_choc, big\_choc\}$

Exercise 2.2

$M = (a \rightarrow M_1 \mid b \rightarrow M_2 \mid c \rightarrow M_3)$
$M_1 = (b \rightarrow M_2 \mid c \rightarrow M_3 \mid d \rightarrow M)$
$M_2 = c \rightarrow M_3$
$M_3 = d \rightarrow M$
where $\alpha M = \{a, b, c, d\}$

Exercise 2.3

$P = AD_0,\ AD_0 = prod1 \to BD_1,$

$BD_i = (prod2 \to AD_i \mid cons1 \to BC_{i-1})$, where $i \in \mathbb{N}$,

$BC_i = (cons2 \to BD_i \mid prod2 \to AC_i)$, where $i \in \mathbb{N}_{\nvdash}$,

$AC_i = (cons2 \to AD_i \mid prod1 \to AC_{i+1})$, where $i \in \mathbb{N}_{\nvdash}$,

$AD_i = (prod1 \to AD_{i+1} \mid cons1 \to AC_{i-1})$, where $i \in \mathbb{N}$,

$BD_0 = prod2 \to AD_0,$

$\alpha P = \{prod1, prod2, cons1, cons2\}$

Exercise 2.4

$P = null \to A_0,$

$A_i = (a \to A_{i+1} \mid null \to B_{i,0}),$
where $i \in \mathbb{N}_0$,

$B_{i,j} = (b \to B_{i-1,j+1} \mid null \to C_{j,0}),$
where $i \in \mathbb{N}, j \in \mathbb{N}_0$, $(i > 0)$,

$B_{0,j} = null \to C_j)$, where $j \in \mathbb{N}_0$,

$C_j = (c \to C_{j-1} \mid null \to STOP)$, where $j \in \mathbb{N}$, $(j > 0)$,

$C_0 = null \to STOP,$

and $\alpha P = \{null, a, b, c\}$

Exercise 2.5

$P \parallel Q$, where

$P = b1 \rightarrow \; meetB \rightarrow meetE \rightarrow e1 \rightarrow P,$

$\quad \alpha P = \{b1, meetB, meetE, e1\},$

$Q = b2 \rightarrow \; meetB \rightarrow meetE \rightarrow e2 \rightarrow Q,$

$\quad \alpha Q = \{b2, meetB, meetE, e2\},$

Exercise 2.6

$(P \parallel Q) \parallel R$, where

$P = b3 \rightarrow \; meet2 \rightarrow meet3 \rightarrow P,$

$\quad \alpha P = \{b3, meet2, meet3\},$

$Q = b1 \rightarrow \; meet\_1B \rightarrow \; meet2 \rightarrow \; meet\_1E \rightarrow meet3 \rightarrow Q,$

$\quad \alpha Q = \{b1, meet\_1B, meet\_1E, meet2, meet3\},$

$R = b2 \rightarrow \; meet1\_B \rightarrow \; meet2 \rightarrow meet1\_E \rightarrow R,$

$\quad \alpha R = \{b2, meet\_1B, meet\_1E, meet2\}.$

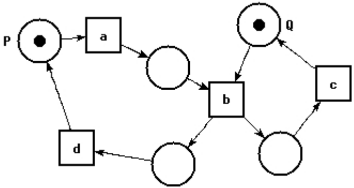
Exercise 2.7

$$P = (a \rightarrow (b \rightarrow STOP \mid c \rightarrow STOP) \mid b \rightarrow d \rightarrow STOP),$$
$$\alpha P = \{a, b, c, d\}$$

Exercise 2.8

$$\begin{aligned} P = (a \rightarrow (b \rightarrow STOP \mid c \rightarrow STOP) \\ \mid d \rightarrow STOP \\ \mid b \rightarrow (c \rightarrow STOP \ \sqcap \ c \rightarrow d \rightarrow STOP)), \end{aligned}$$
$$\alpha P = \{a, b, c, d\}$$

Exercise 2.9



Exercise 2.10

$acbdnil + cabdnil$

Agenda

Exercise 3.1

$P \parallel Q = (a \rightarrow c \rightarrow b \rightarrow (P \parallel Q) \mid c \rightarrow a \rightarrow b \rightarrow (P \parallel Q))$

Exercise 3.2

$P \parallel Q = b \rightarrow (a \rightarrow c \rightarrow (P \parallel Q) \mid c \rightarrow a \rightarrow (P \parallel Q))$

Exercise 3.3

$P \parallel Q = a \rightarrow R$, where
$R = b \rightarrow (a \rightarrow c \rightarrow R \mid c \rightarrow a \rightarrow R)$

Exercise 3.4

$$P \parallel Q = a \to b \to (P \parallel Q)$$

Exercise 3.5

$$\begin{aligned} P \parallel Q = x \to y \to (&w \to z \to y \to (P \parallel Q) \\ &\mid z \to w \to y \to (P \parallel Q)) \end{aligned}$$

Exercise 3.6

$$P \parallel Q = (a \to STOP) \sqcap (a \to b \to (P \parallel Q))$$

Exercise 3.7

$P \setminus \{b\} = (a \to P \mid d \to c \to P)$, where $\alpha P = \{a, c, d\}$

Exercise 3.8

$P \parallel Q = (a \to b \to c \to (P \parallel Q)) \sqcap (a \to c \to d \to (P \parallel Q))$

Exercise 3.9

$P \parallel Q = e \to R$, where
$R = a \to (b \to e \to W \mid e \to b \to W)$, where
$W = c \to (d \to e \to R \mid e \to d \to R)$

Exercise 3.10

$P \parallel Q = (x \to y \to z \to (P \parallel Q)) \sqcap (x \to z \to w \to (P \parallel Q))$

Agenda

Exercise 4.1

1. $traces(R(a)) = \{<>, < b >, < b, c >\}$
2. $traces(R(b)) = \{<>, < d >\}$

$\Rightarrow$ $traces(W)$
$= \{<>, < a >, < a, b >, < a, b, c >, < b >, < b, d >\}$

Exercise 4.2

1. $traces(P) = \{<>, < b >, < b, c >\}$
2. $traces(Q) = \{<>, < d >, < d, c >\}$

$\Rightarrow$ $traces(P \sqcap Q)$
$= \{<>, < b >, < b, c >, < d >, < d, c >\}$

Exercise 4.3

1. $traces(R) = \{<>, < d >, < d, c >\}$
2. $traces(W) = \{<>, < b >, < b, d >, < b, d, c >\}$

$\Rightarrow$ $traces(R \,\Box\, W)$
$= \{<>, < d >, < d, c >, < b >, < b, d >, < b, d, c >\}$

Exercise 4.4

- $traces((P \sqcap Q) \,||\, (R \,\Box\, W))$
  $= traces(P \sqcap Q) \cap traces(R \,\Box\, W)$
  $= \{<>, < b >, < d >, < d, c >\}$

Exercise 4.5

- Here $F(X) = coin \rightarrow choc \rightarrow X$.
- According to the 3. item of definition of function *traces* it is enough to see:
  $\forall n \in \mathbb{N} : traces(F^n(STOP)) = \{s \mid s \ \leq < coin, choc >^n\}$.
  Using induction:

n=0 $traces(F^0(STOP)) = traces(STOP) = \{<>\}$
$= \{s \mid s \ \leq < coin, choc >^0\}$,

n=k+1 $traces(F^{k+1}(STOP_A))$
$= traces(F(F^k(STOP_A)))$
$= traces(coin \rightarrow choc \rightarrow F^k(STOP))$
$= \{<>, < coin >\}$
$\cup \{< coin, choc >^\wedge t \mid t \in traces(F^k(STOP)))\}$
$= \{<>, < coin >\}$
$\cup \{< coin, choc >^\wedge t \mid t \ \leq < coin, choc >^n)\}$
$= \{t \mid t \ \leq < coin, choc >^{n+1}\}$

Exercise 4.6

- $traces(P) = \{<>, < a >, < a, b >, < a, c >\}$,
- $\forall t \in traces(P) : (t \downarrow a) \geq (t \downarrow c)$,

$\Rightarrow$ $P \; sat \; ((tr \downarrow a) \geq (tr \downarrow c))$.

Exercise 4.7

- $P \parallel Q = a \to b \to c \to STOP$,
- $traces(P \parallel Q) = \{<>, < a >, < a, b >, < a, b, c >\}$,
- $\forall t \in traces(P \parallel Q) : ((t \downarrow a) \leq (t \downarrow b))$,

$\Rightarrow$ $(P \parallel Q) \; sat \; ((tr \downarrow a) \leq (tr \downarrow b))$.

Exercise 4.8

- $traces(P \sqcap Q) = \{<>, < b >, < b, c >, < a >, < a, b >\}$,
- $\forall t \in traces(P \sqcap Q) : ((t \downarrow a) + (t \downarrow c) \leq 1)$,

$\Rightarrow$ $(P \sqcap Q) \; sat \; ((tr \downarrow a) + (tr \downarrow c) \leq 1)$.

Exercise 4.9

Let be $MySpec = ((tr \downarrow a) \leq 1 + (tr \downarrow b) + (tr \downarrow d))$.

1. $(tr =<>) \Rightarrow MySpec$

$\Rightarrow$ $STOP\ sat\ (tr =<>) \Longrightarrow STOP\ sat\ MySpec$

2. Suppose $X\ sat\ MySpec$

$$traces(F(X)) = \{<>, <a>, <a,b>, <a,c>\} \cup \{<a,b,c>^\wedge t \mid t \in traces(X)\} \cup \{<a,c,d>^\wedge t \mid t \in traces(X)\}$$

2.a $(<>\downarrow a) = 0,\ (<>\downarrow b) + (<>\downarrow d) = 0,$
$(<a>\downarrow a) = 1,\ (<a>\downarrow b) + (<a>\downarrow d) = 0,$
$(<a,b>\downarrow a) = 1,\ (<a,b>\downarrow b) + (<a,b>\downarrow d) = 1,$
$(<a,c>\downarrow a) = 1,\ (<a,c>\downarrow b) + (<a,c>\downarrow d) = 0,$

2.b $(<a,b,c>^\wedge t \downarrow a) = (t \downarrow a) + 1,$
$(<a,b,c>^\wedge t \downarrow b) = (t \downarrow b) + 1,$
$(<a,b,c>^\wedge t \downarrow d) = (t \downarrow d),$
so $((<a,b,c>^\wedge t \downarrow a) \leq 1 + (<a,b,c>^\wedge t \downarrow b) + (<a,b,c>^\wedge t \downarrow d))$
equivalent with $((t \downarrow a) \leq 1 + (t \downarrow b) + (t \downarrow d))$
(which will hold because of the assumption).

2.c $(< a, c, d >^\wedge t \downarrow a) = (t \downarrow a) + 1,$
$(< a, c, d >^\wedge t \downarrow b) = (t \downarrow b),$
$(< a, c, d >^\wedge t \downarrow d) = (t \downarrow d) + 1,$
so $\big((< a, c, d >^\wedge t \downarrow a)$
$\leq 1 + (< a, c, d >^\wedge t \downarrow b) + (< a, c, d >^\wedge t \downarrow d)\big)$
equivalent with $((t \downarrow a) \leq 1 + (t \downarrow b) + (t \downarrow d))$
(which will hold because of the assumption).

- (2.a) $\wedge$ (2.b) $\wedge$ (2.c) $\Rightarrow F(X)\ sat\ MySpec$.
- (1.) $\wedge$ (2.) $\Rightarrow P\ \ sat\ \ MySpec$.

Exercise 4.10

1. $P \ sat \ (0 \ \leq \ ((tr \downarrow a) - (tr \downarrow b)) \ \leq \ 1)$
2. $Q \ sat \ (0 \ \leq \ ((tr \downarrow b) - (tr \downarrow c)) \ \leq \ 1)$

$$\Rightarrow P \parallel Q \\ sat \ \big((0 \ \leq \ (((tr \uparrow \alpha P) \downarrow a) - ((tr \uparrow \alpha P) \downarrow b)) \ \leq \ 1) \\ \wedge (0 \ \leq \ (((tr \uparrow \alpha Q) \downarrow b) - ((tr \uparrow \alpha Q) \downarrow c)) \ \leq \ 1)\big)$$

$$\Rightarrow P \parallel Q \\ sat \ \big((0 \ \leq \ ((tr \downarrow a) - (tr \downarrow b)) \ \leq \ 1) \\ \wedge (0 \ \leq \ ((tr \downarrow b) - (tr \downarrow c)) \ \leq \ 1)\big)$$

$$\Rightarrow P \parallel Q \ sat \ (0 \ \leq \ ((tr \downarrow a) - (tr \downarrow c)) \ \leq \ 2)$$

Agenda

Exercise 5.1

$$P||Q = c1?x \to c2.(x+5) \to c3!(2*x+10) \to STOP$$

Exercise 5.2

$$(P||Q) \setminus \{c2\} = c1?x \to c3!(2*x+10) \to STOP$$

Exercise 5.3

$$P||Q_{10} = in?x \to c1.x \to c2.(10{*}x) \to out!(10{*}x) \to (P||Q_{10})$$

Exercise 5.4

$$(P||Q_5) \setminus \{c1, c2\} = in?x \to out!(5*x) \to ((P||Q_5) \setminus \{c1, c2\})$$

Exercise 5.5

$$\begin{aligned}
&(P||Q) = \\
&\quad \big(in1?x \to in2?y \\
&\qquad\qquad \to c1.x \to c2.(x+y) \to out!(x+y) \to (P||Q) \\
&\quad\ |in2?y \to in1?x \\
&\qquad\qquad \to c1.x \to c2.(x+y) \to out!(x+y) \to (P||Q)\big)
\end{aligned}$$

Exercise 5.6

$$\begin{aligned}
&(P||Q) = \\
&\; in1?x \rightarrow c1.x \\
&\quad \rightarrow \big(in2?y \rightarrow out!x \rightarrow c2.(x*y) \rightarrow out!(x*y) \rightarrow (P||Q) \\
&\qquad |out!x \rightarrow in2?y \rightarrow c2.(x*y) \rightarrow out!(x*y) \rightarrow (P||Q)\big)
\end{aligned}$$

Exercise 5.7

$$Q \;=\; (in?y \rightarrow out!(y_1 * y_2 + 5) \rightarrow Q)$$

Exercise 5.8

$$Q \;=\; (in?y \rightarrow out!(y_1, (y_1 * y_2 + 3)) \rightarrow Q)$$

Exercise 5.9

$$Q \;=\; (in?y \rightarrow out!(y_1, (y_2 + (y_1 * y_3)), (y_1 * y_3)) \rightarrow Q)$$

Exercise 5.10

$$Q_i \;=\; (in?y \rightarrow out!(y_1, (y_2 + \tfrac{y_1}{i} * y_3), \tfrac{y_1}{i} * y_3) \rightarrow Q_i)$$