Kapitel 7

Höhere Petrinetze

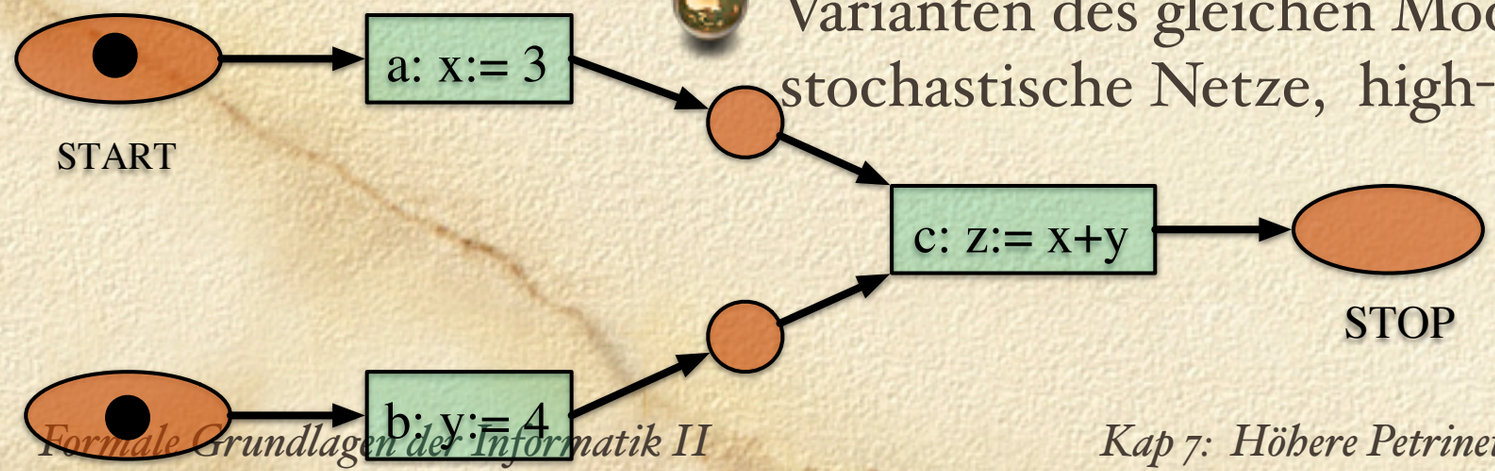
Klausurtermine (ohne Gewähr)

- 1. Klausur: 01.03.2011, 9.30 - 11.30, Audi 1 (Achtung! Zeit geändert!)
- 2. Klausur: 01.04.2011, 9.30 - 11.30, ESA B (Achtung! Zeit geändert!)

Zur Vorbereitung auf die Klausuren werden Repetitorien angeboten.

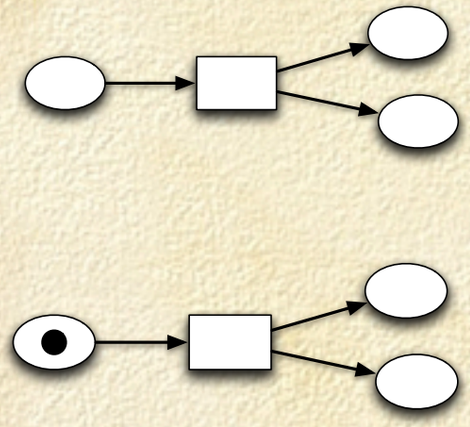
Allgemeine Eigenschaften von Petrinetzen

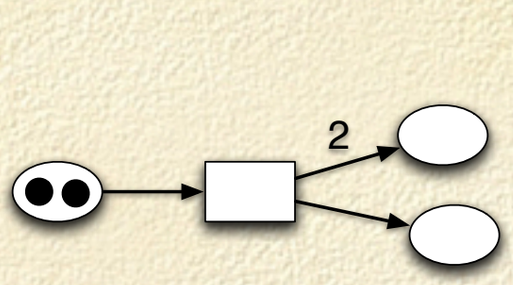
- graphische und äquivalente algebraische/textuelle Darstellung
- (formal abgesicherte) Algorithmen für die Analyse
- Abstraktion und hierarchische Strukturen
- hoch entwickelte Theorie der Nebenläufigkeit (concurrency)
- Rechnerwerkzeuge für Editieren, Simulation und Analyse
- siehe z.B. TGI-Tool RENEW http://www.renew.de/
- Universalität in Anwendbarkeit (Anwendungen in fast allen Gebieten)
- Varianten des gleichen Modellierungskonzeptes (Zeit-Netze, stochastische Netze, high-level, objektorientiert, ...)

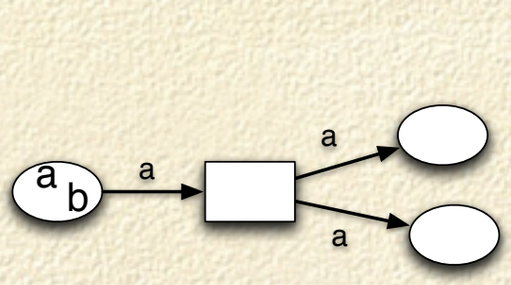


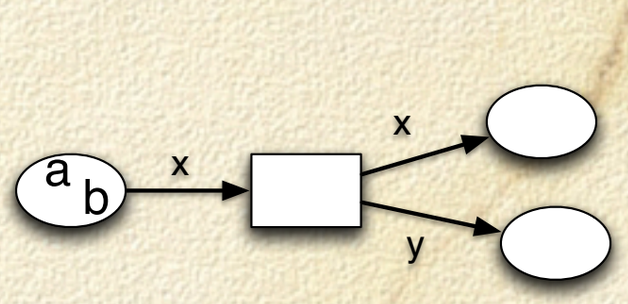
Netzklassen

- *„einfache“ Netze*
- *P/T-Netze*
- *kantenkonstante Netze*
- *gefärbte Netze*



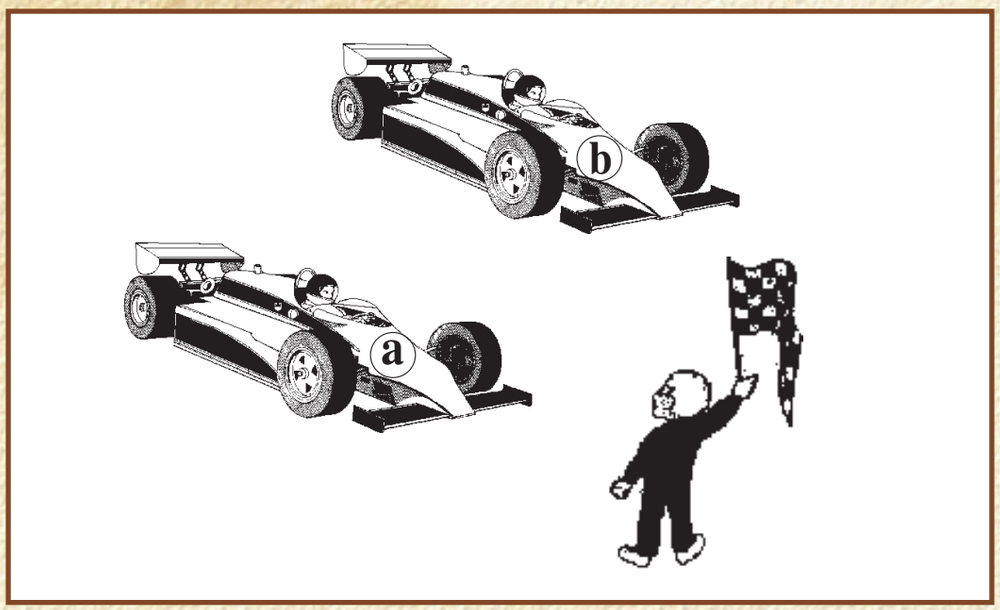


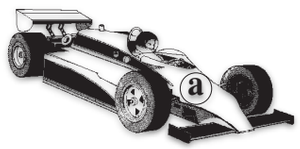


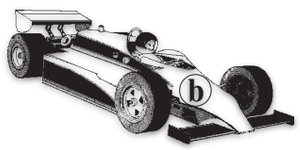


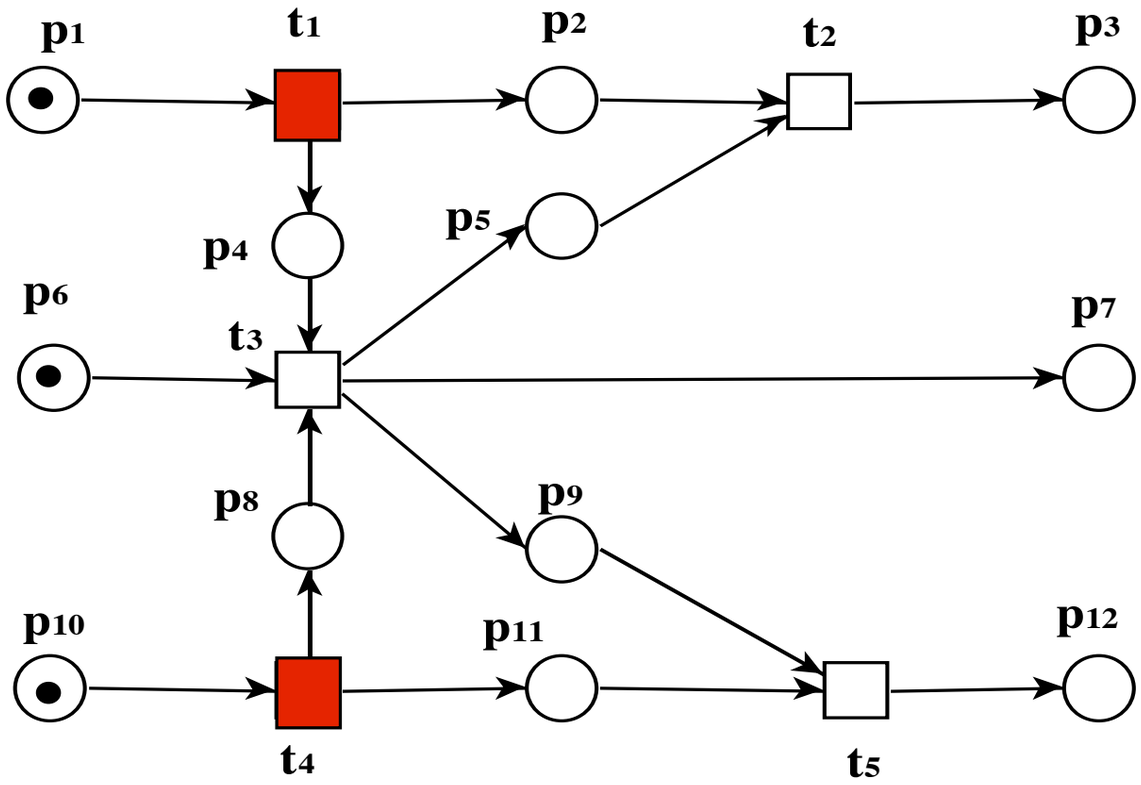
✓ ✓

Start zweier Rennwagen









einfaches Netz

kantenkonstante Netze

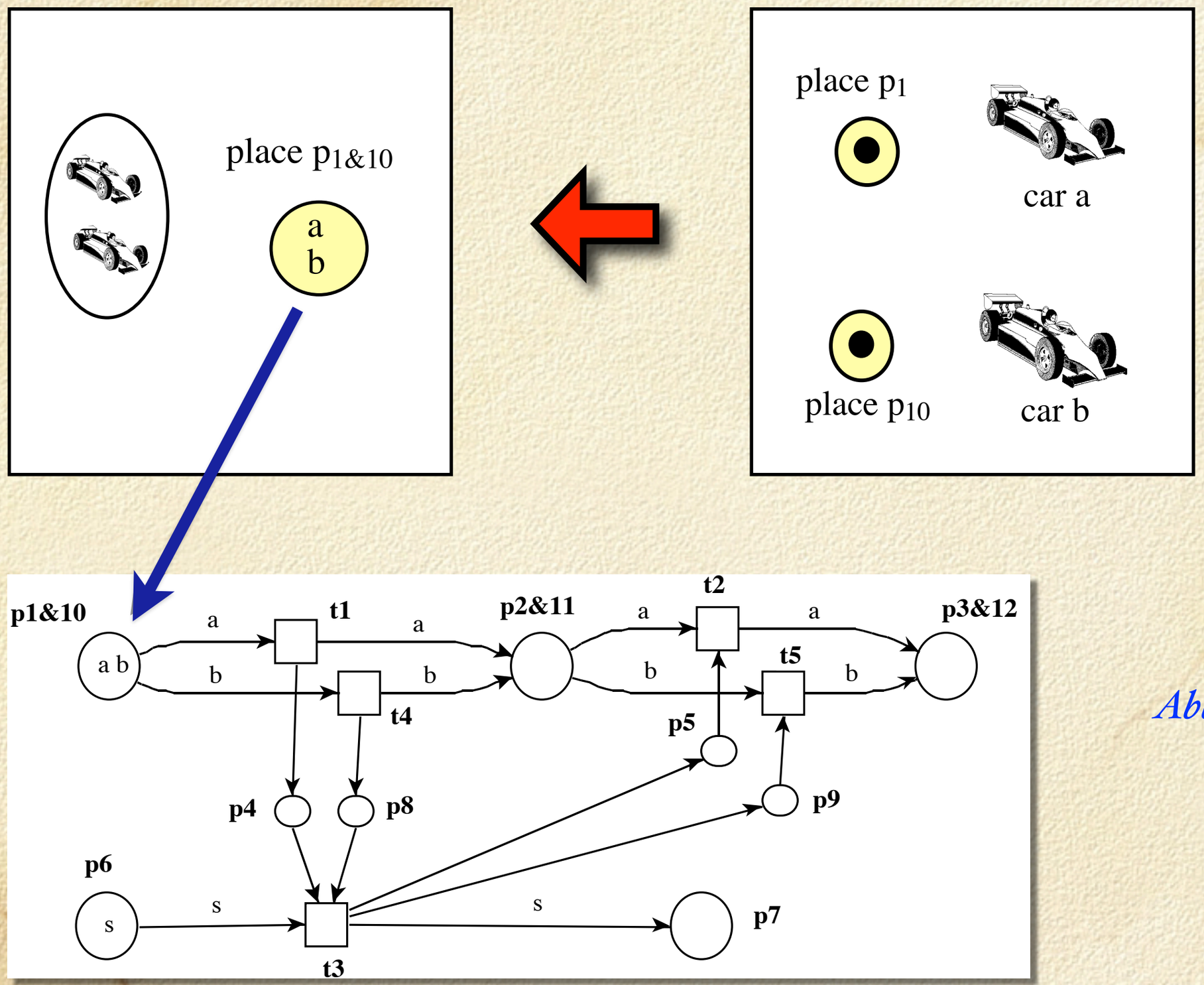


Abb. 7.2

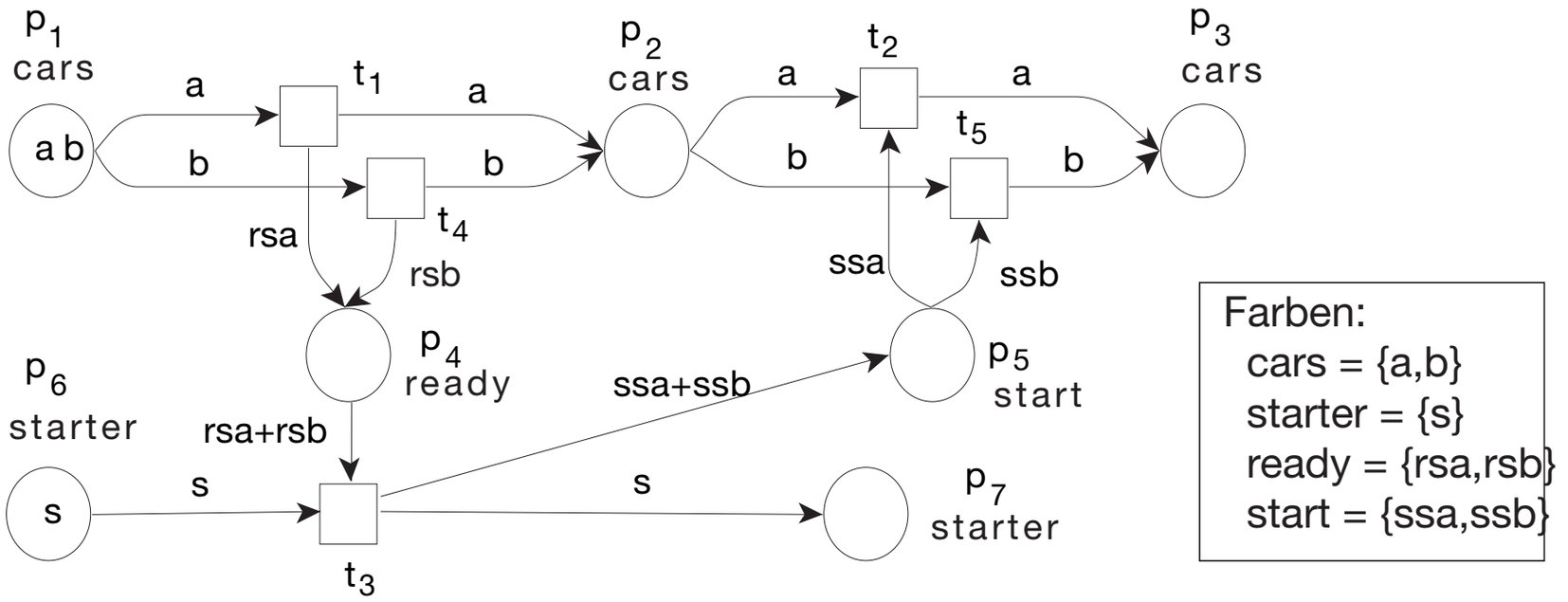


Abbildung *7.3* Das kantenkonstante gefärbte Netz $\mathcal{N}_2$

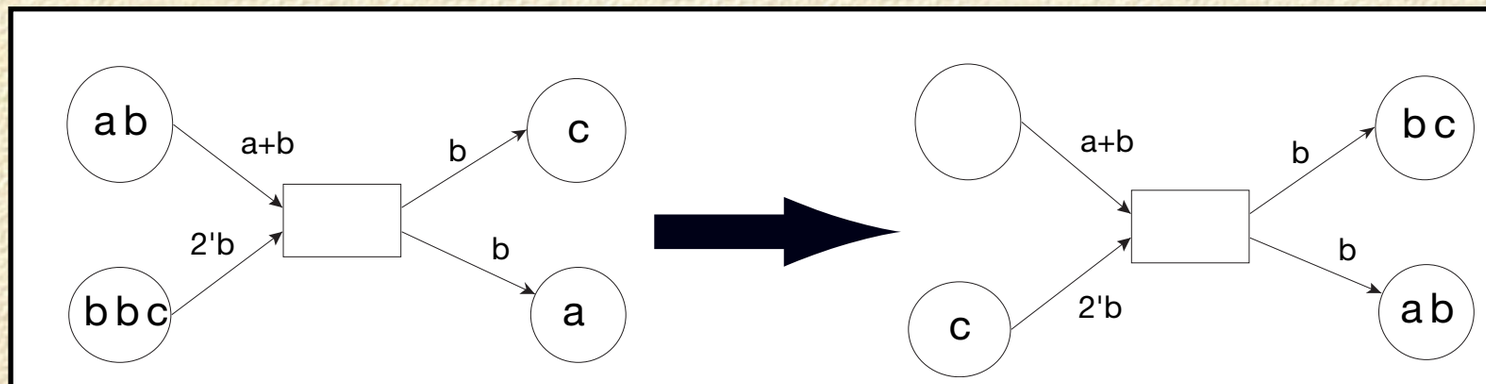
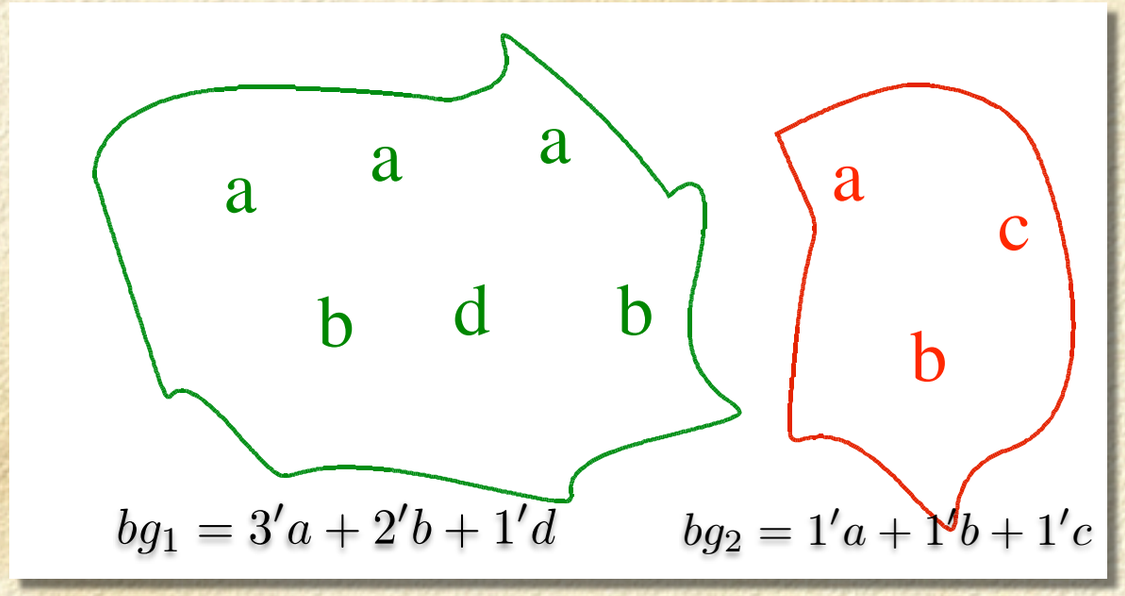


Abbildung *7.4* Schaltregel für kantenkonstante CPN

Multimenge "bag"

(Seite 224) *Multimengen, (multi-sets, bags)*



$\{a,a,a,b,b,d\}_b$

$3'a + 2'b + 0'c + 1'd$

$3'a + 2'b + d$

≠ {a,a,a,b,b,d}
= {a,b,d}

$$bg : A \to \mathbb{N}$$

$$\sum_{a \in A} bg(a)'a$$

- $bg_1 + bg_2 := \sum_{a \in A}(bg_1(a) + bg_2(a))'a$ ≅ *Mengen-Vereinigung* $4'a + 3'b + 1'c + 1'd$
- $bg_1 \leq bg_2 :\Leftrightarrow \forall a \in A : bg_1(a) \leq bg_2(a)$ ≅ *Mengen-Inklusion*
- $bg_1 - bg_2 := \sum_{a \in A}(Max(bg_1(a) - bg_2(a), 0))'a$ *und* ≅ *Mengen-Differenz* $2'a + 1'b + 1'd$
- $|bg| := \sum_{a \in A} bg(a)$ *ist die Mächtigkeit oder Kardinalität von bg (nur definiert, falls die Summe endlich ist) und* $\emptyset$ *bezeichnet die* leere Multimenge *(mit* $|bg| = 0$ *).* ≅ *Mengen-Kardinalität*

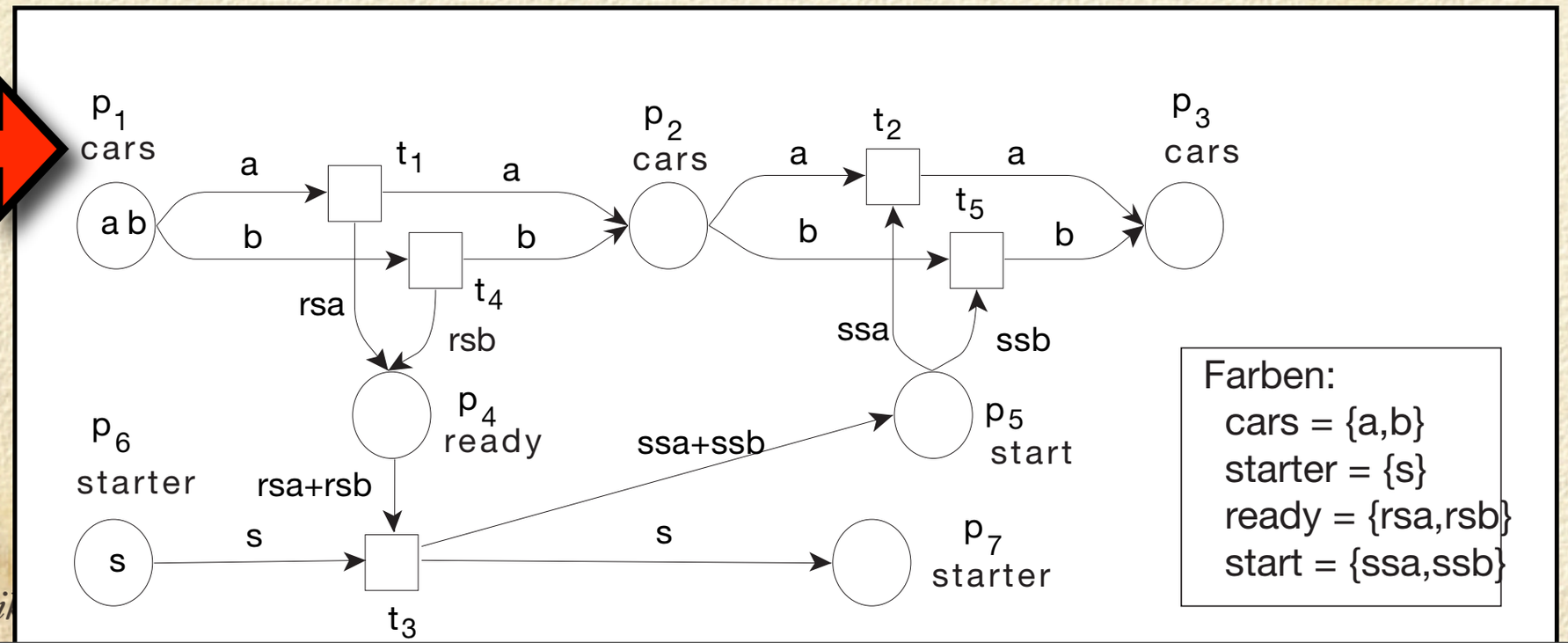
$$|bg_1| = 6$$

Definition 7.2 *Ein* kantenkonstantes gefärbtes Petrinetz*(KKN) wird als Tupel* $\mathcal{N} = \langle P, T, F, \mathcal{C}, cd, W, \mathbf{m}_0 \rangle$ *definiert, wobei*

- (P, T, F) *ein endliches Netz (Def. 3.1) ist ,*
- $\mathcal{C}$ *ist eine Menge von* Farbenmengen,
- cd*:* $P \to \mathcal{C}$ *ist die* Farbzuweisungsabbildung *(colour domain mapping). Sie wird durch* cd*:* $F \to \mathcal{C}$, $cd(x, y) :=$ **if** $x \in P$ **then** $cd(x)$ **else** $cd(y)$ **fi** *auf* F *erweitert.*
- $W : F \to Bag(\bigcup \mathcal{C})$ *mit* $W(x, y) \in Bag(cd(x, y))$ *heißt Kantengewichtung.*
- $\mathbf{m}_0 : P \to Bag(\bigcup \mathcal{C})$ *mit* $\mathbf{m}_0(p) \in Bag(cd(p))$ *für alle* $p \in P$ *ist die Anfangsmarkierung.*

Darstellung als **Abbildung** *oder Vektor möglich*

Typ „Farbe“

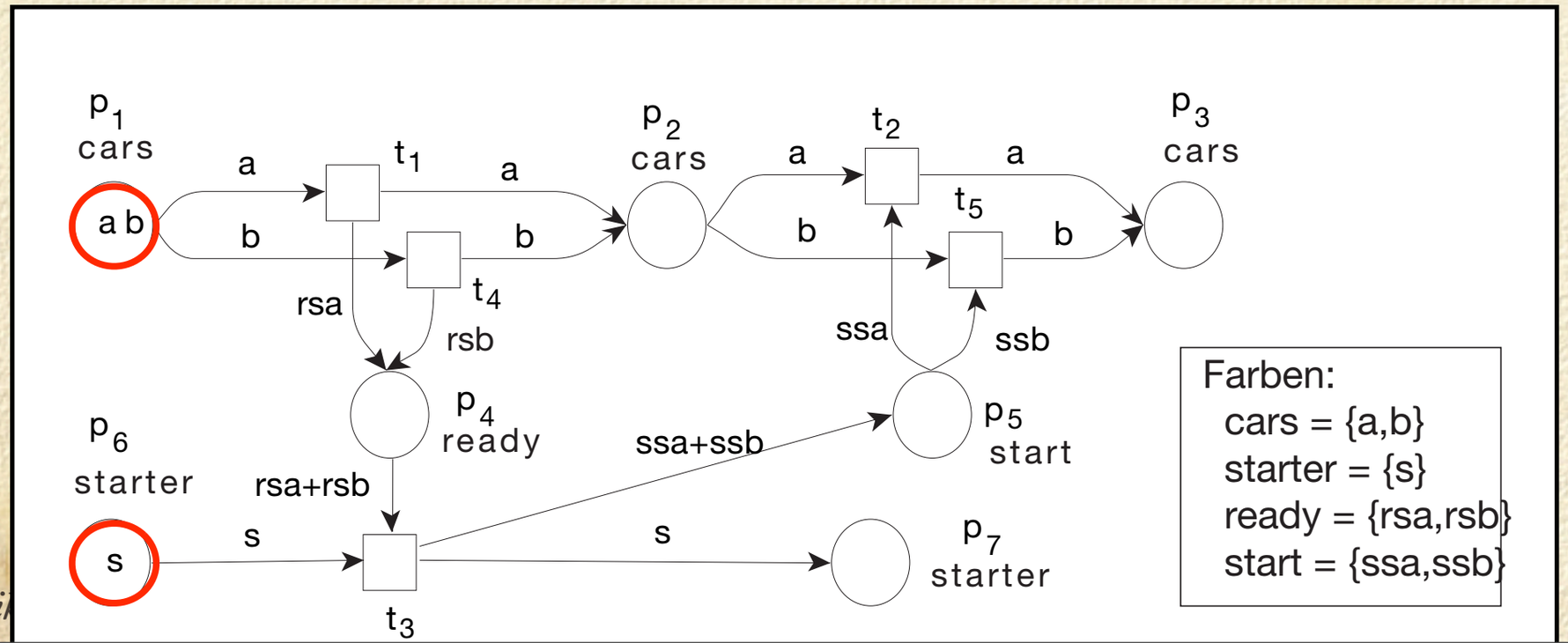


Darstellung als Abbildung:

$$\mathbf{m}_0(p_1) = \{a, b\}_b \qquad \mathbf{m}_0(p_i) = \emptyset$$
$$\mathbf{m}_0(p_6) = \{s\}_b$$

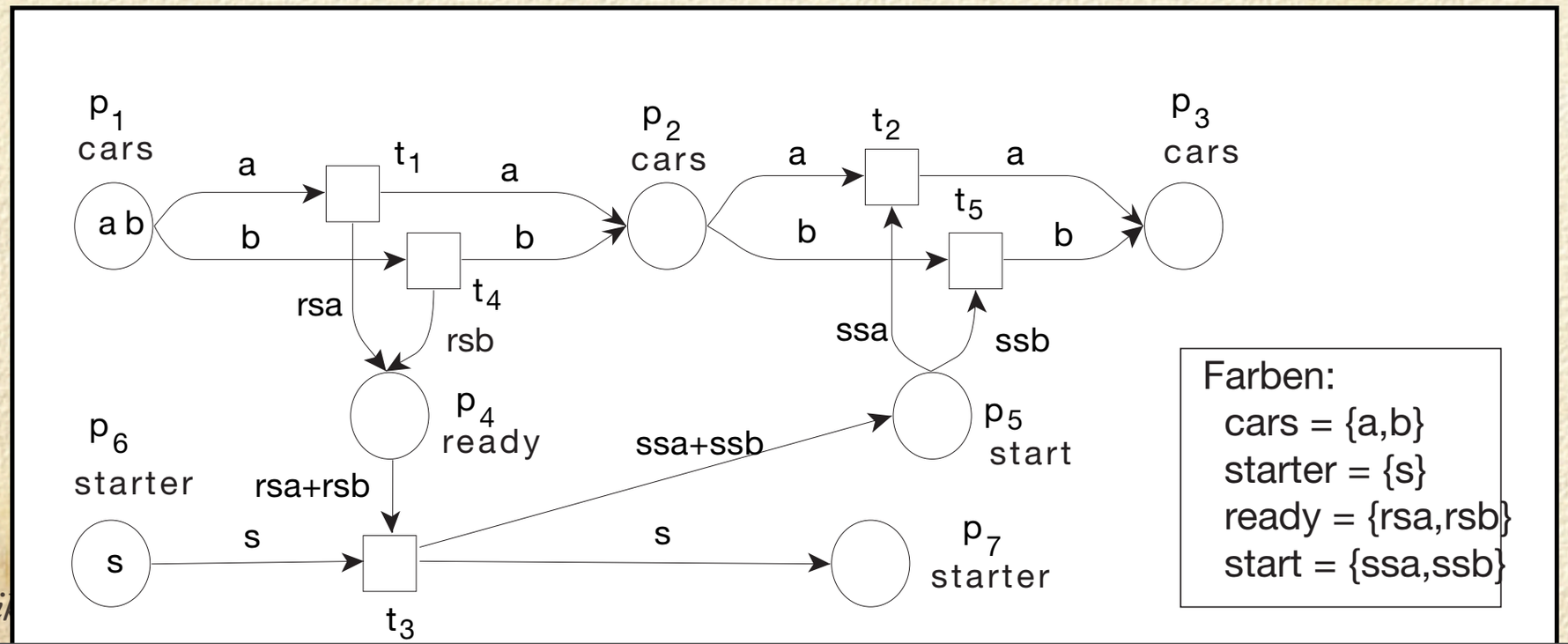
Darstellung als Vektor :

$$\begin{matrix} p_1 \\ \\ \\ \\ \\ \\ p_7 \end{matrix} \begin{pmatrix} \{a,b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_4} \begin{pmatrix} \{a\} \\ \{b\} \\ \emptyset \\ \{rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_1} \begin{pmatrix} \emptyset \\ \{a,b\} \\ \emptyset \\ \{rsa, rsb\} \\ \emptyset \\ \{s\} \\ \emptyset \end{pmatrix} \xrightarrow{t_3} \begin{pmatrix} \emptyset \\ \{a,b\} \\ \emptyset \\ \emptyset \\ \{ssa, ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{t_2} \begin{pmatrix} \emptyset \\ \{b\} \\ \{a\} \\ \emptyset \\ \{ssb\} \\ \emptyset \\ \{s\} \end{pmatrix} \xrightarrow{t_5} \begin{pmatrix} \emptyset \\ \emptyset \\ \{a,b\} \\ \emptyset \\ \emptyset \\ \emptyset \\ \{s\} \end{pmatrix}$$



Definition 7.3 *a) Die Markierung eines KKN* $\mathcal{N} = \langle P, T, F, \mathcal{C}, cd, W, \mathbf{m}_0 \rangle$ *ist ein Vektor* $\mathbf{m}$ *mit* $\mathbf{m}(p) \in Bag(cd(p))$ *für jedes* $p \in P$ *(auch als Abbildung* $\mathbf{m} : P \to Bag(\bigcup \mathcal{C})$ *mit* $\mathbf{m}(p) \in Bag(cd(p))$ *für jedes* $p \in P$ *aufzufassen).*

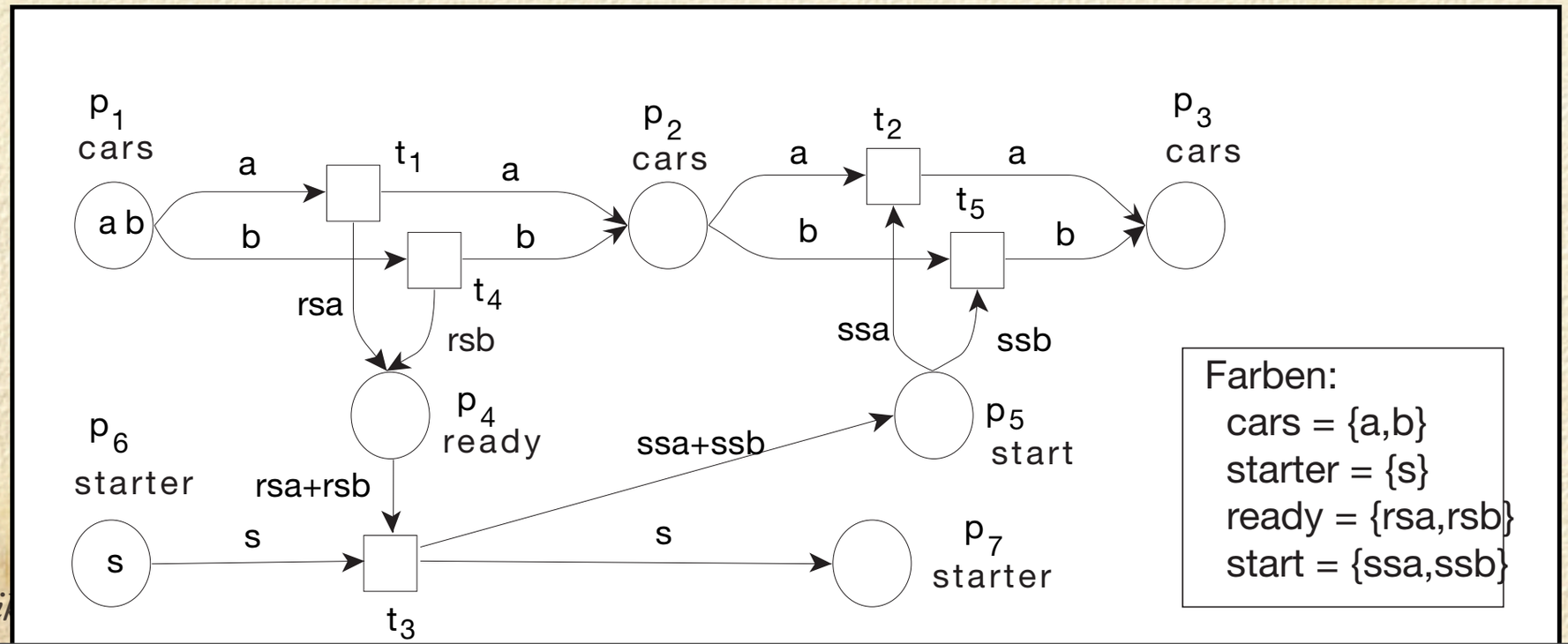
b) Eine Transition $t \in T$ *heißt* aktiviert *in einer Markierung* $\mathbf{m}$ *falls* $\forall p \in {}^\bullet t.\ \mathbf{m}(p) \geq W(p,t)$ *(als Relation:* $\mathbf{m} \xrightarrow{t}$*).*



c) Es sei $\widetilde{W}(p,t) := \begin{cases} W(p,t) & \text{falls } (p,t) \in F, \\ \emptyset & \text{sonst.} \end{cases}$

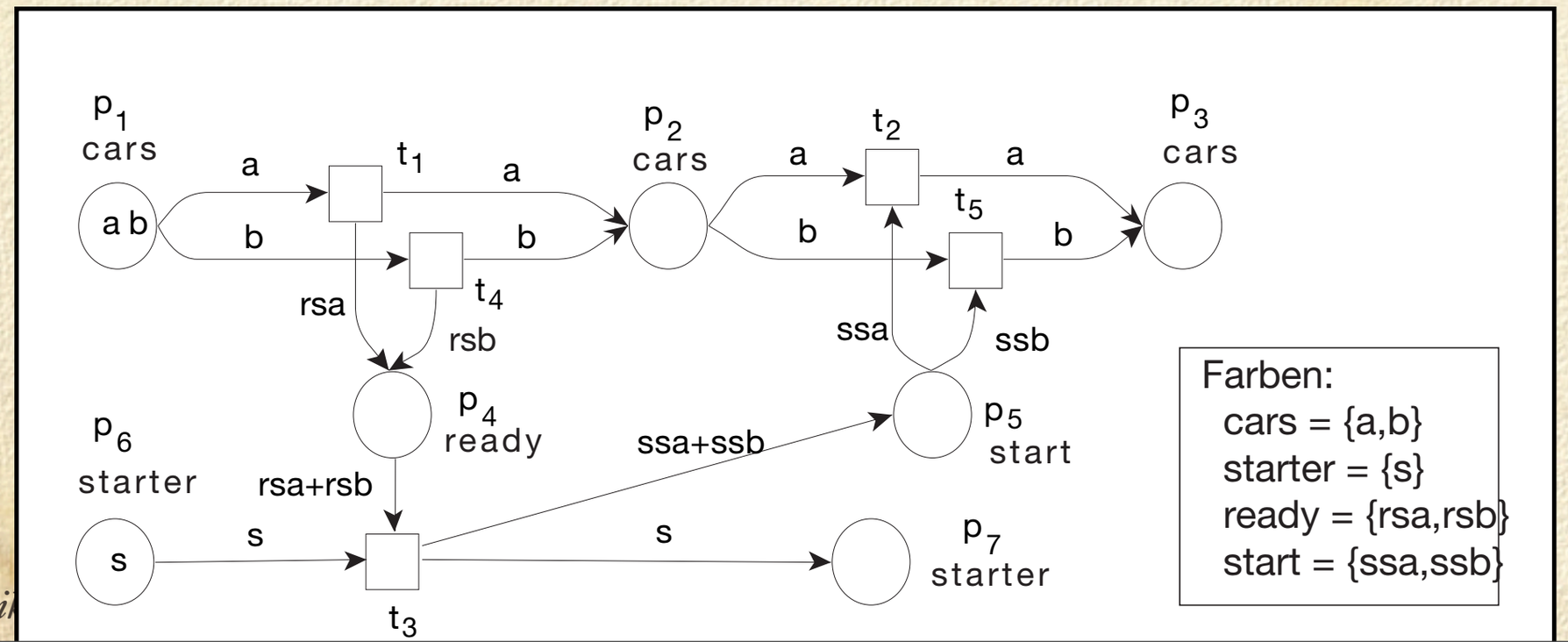
und entsprechend $\widetilde{W}(t,p) := \begin{cases} W(t,p) & \text{falls } (t,p) \in F, \\ \emptyset & \text{sonst.} \end{cases}$

Ist t *in* $\mathbf{m}$ *aktiviert, dann ist die* Nachfolgemarkierung *definiert durch* $\mathbf{m} \xrightarrow{t} \mathbf{m}' \Leftrightarrow \forall p \in P.\ (\mathbf{m}(p) \geq \widetilde{W}(p,t) \wedge \mathbf{m}'(p) = \mathbf{m}(p) - \widetilde{W}(p,t) + \widetilde{W}(t,p))$*. (Beachte, dass es sich um Multimengenoperationen handelt!).*



d) Definiert man $W(\bullet, t) := (\widetilde{W}(p_1, t), \ldots, \widetilde{W}(p_{|P|}, t))$ *als Vektor der Länge* $|P|$ *(und sinngemäß ebenso* $W(t, \bullet)$ *), dann kann die Nachfolgemarkierung kürzer durch Vektoren definiert werden:*
$\mathbf{m} \xrightarrow{t} \mathbf{m}' \Leftrightarrow \mathbf{m} \geq W(\bullet, t) \wedge \mathbf{m}' = \mathbf{m} - W(\bullet, t) + W(t, \bullet)$.

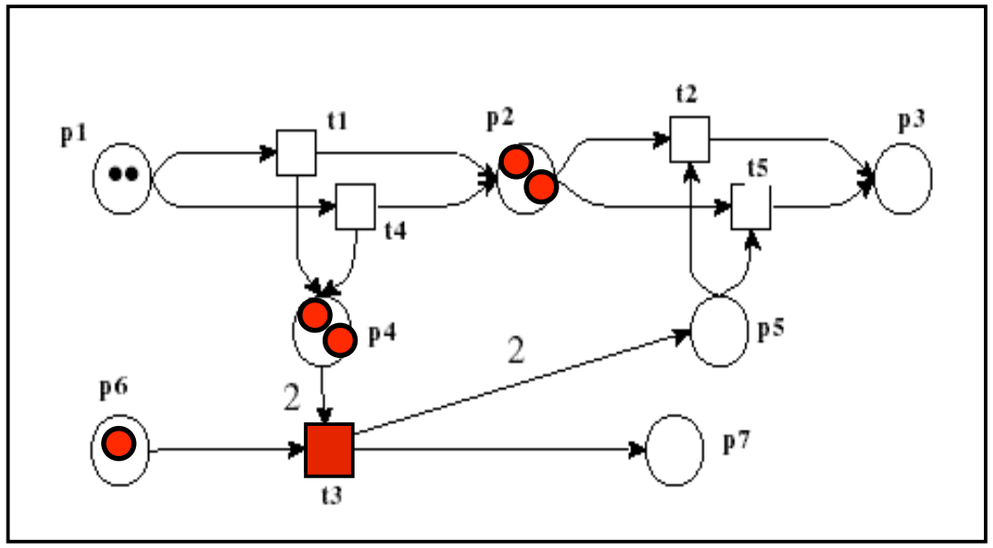
Dabei sind die Multimengenoperatoren komponentenweise auf Vektoren zu erweitern.



W(•,t_3)≤m

	t_1	t_2	t_3
p_1			
p_2			
p_3			
p_4			2
p_5			
p_6			1
p_7			

≤



$$\begin{pmatrix}2\\0\\0\\0\\0\\1\\0\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}1\\1\\0\\1\\0\\1\\0\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}0\\2\\0\\2\\0\\1\\0\end{pmatrix} \xrightarrow{t_3} \begin{pmatrix}0\\2\\0\\0\\2\\0\\1\end{pmatrix} \xrightarrow{t_2} \begin{pmatrix}0\\1\\1\\0\\1\\0\\1\end{pmatrix} \xrightarrow{t_2} \begin{pmatrix}0\\0\\2\\0\\0\\0\\1\end{pmatrix}$$

m' = m - W(•,t_3) + W(t_3,•)

	t_1	t_2	t_3
p_1			
p_2			
p_3			
p_4			2
p_5			
p_6			1
p_7			

	t_1	t_2	t_3
p_1			
p_2			
p_3			
p_4			
p_5			2
p_6			
p_7			1

$\mathbf{m \geq W(\bullet, t_3)}$

Pre	t_1	t_2	t_3	t_4
p_1:				
p_2:				
p_3:				
p_4:			rsa+rsb	
p_5:				
p_6:			s	

M

$$\begin{pmatrix}\{a,b\}\\ \emptyset\\ \emptyset\\ \emptyset\\ \emptyset\\ \{s\}\\ \emptyset\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}\{a\}\\ \{b\}\\ \emptyset\\ \{rsb\}\\ \emptyset\\ \{s\}\\ \emptyset\end{pmatrix} \xrightarrow{t_1} \begin{pmatrix}\emptyset\\ \{a,b\}\\ \emptyset\\ \{rsa,rsb\}\\ \emptyset\\ \{s\}\\ \emptyset\end{pmatrix} \xrightarrow{t_3} \begin{pmatrix}\emptyset\\ \{a,b\}\\ \emptyset\\ \emptyset\\ \{ssa,ssb\}\\ \emptyset\\ \{s\}\end{pmatrix} \xrightarrow{t_2} \begin{pmatrix}\emptyset\\ \{b\}\\ \{a\}\\ \emptyset\\ \{ssb\}\\ \emptyset\\ \{s\}\end{pmatrix} \xrightarrow{t_2} \begin{pmatrix}\\ \\ \\ \\ \\ \\ \end{pmatrix}$$

m

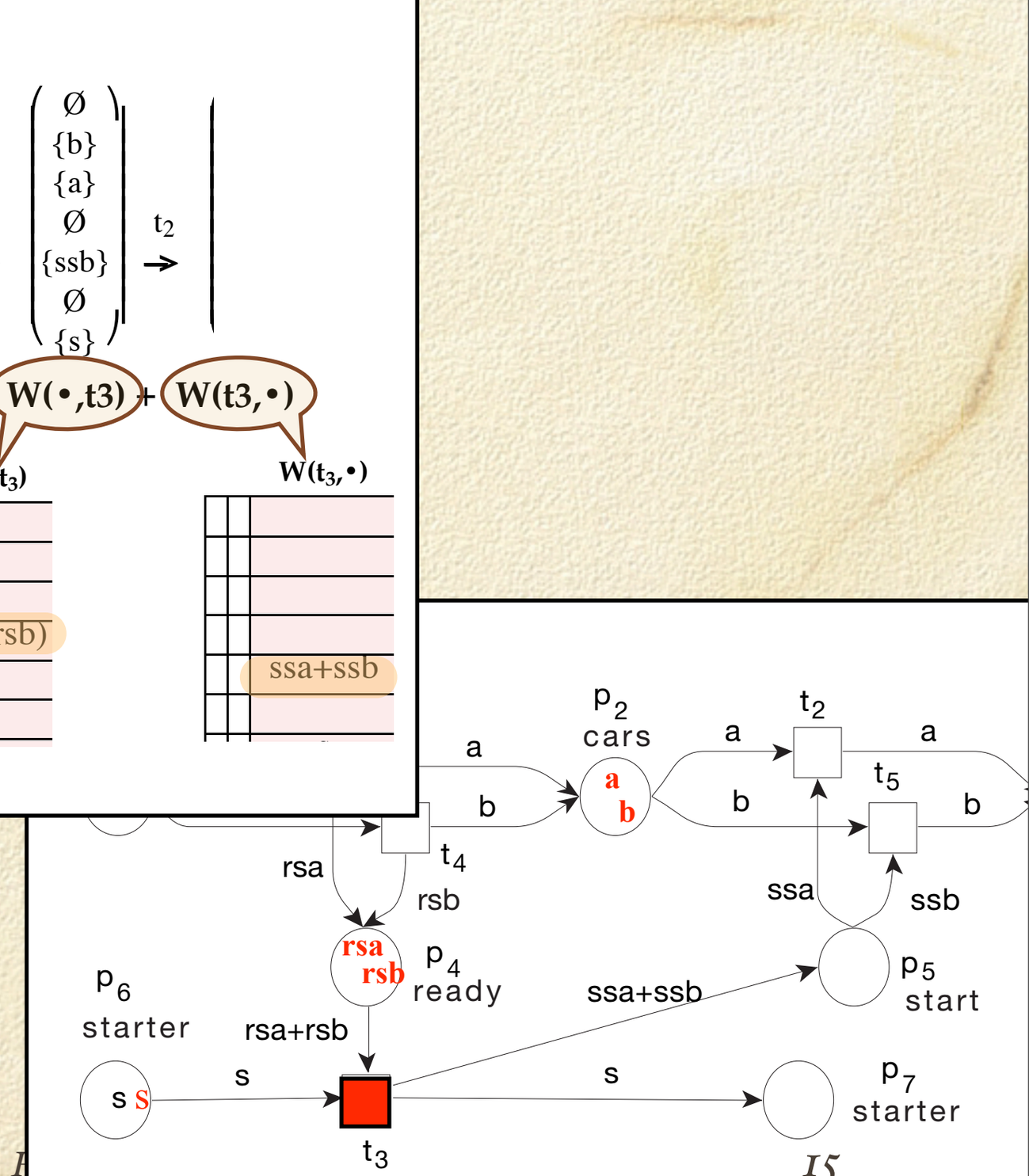
$\mathbf{m' = m - W(\bullet,t3) + W(t3,\bullet)}$

$\mathbf{W(\bullet,t_3)}$

(rsa+rsb)
s

$\mathbf{W(t_3,\bullet)}$

ssa+ssb



Definition **(I)** *Ein kantenkonstantes Petrinetz (Def. 3.4)* $\mathcal{N} = \langle P, T, F, \mathcal{C}, cd, W, \mathbf{m}_0 \rangle$ *heißt Platz/Transitions-Netz (P/T-Netz) oder Stellen/Transitions-Netz (S/T-Netz), falls* $\mathcal{C} = \{token\} = \{\{\bullet\}\}$.

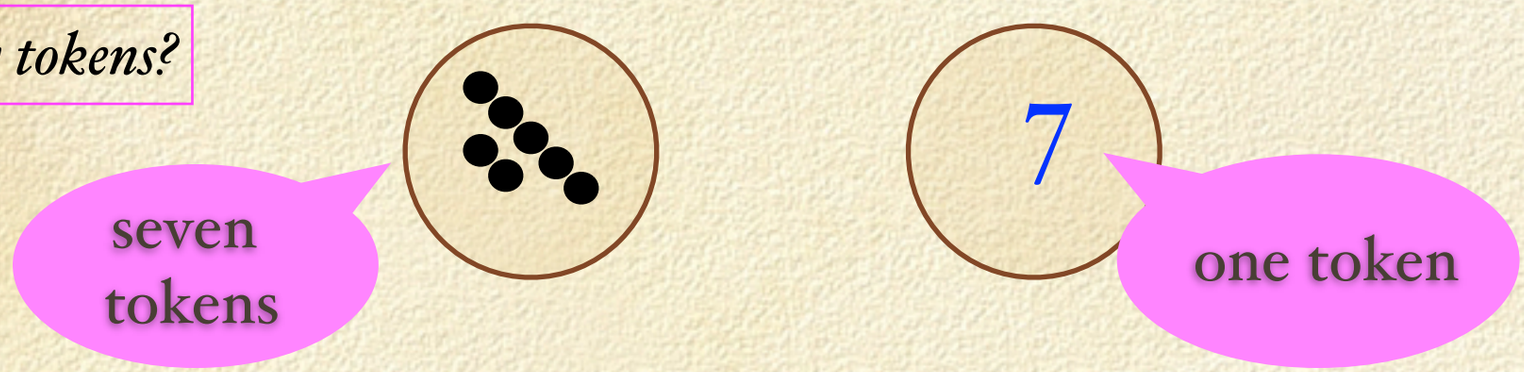
7'$\bullet$ $\quad\cong\quad$ $7 \in \mathbb{N}$

$7' \bullet \overset{?}{\cong} 7 \in \mathbb{N}$

P/T - net

arc-constant net

how many tokens?



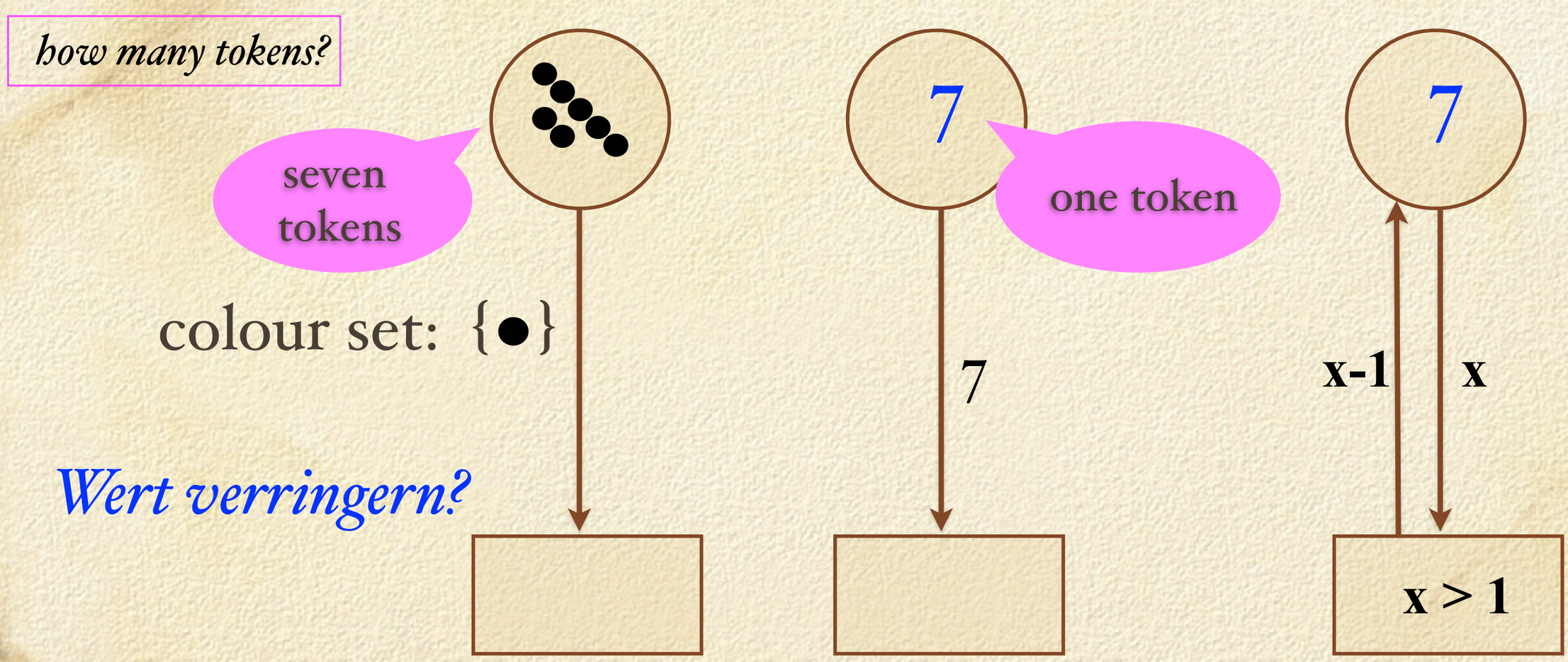
colour set: {●}

$7' \bullet \overset{?}{\cong} 7 \in \mathbb{N}$

P/T - net

arc-constant net

how many tokens?



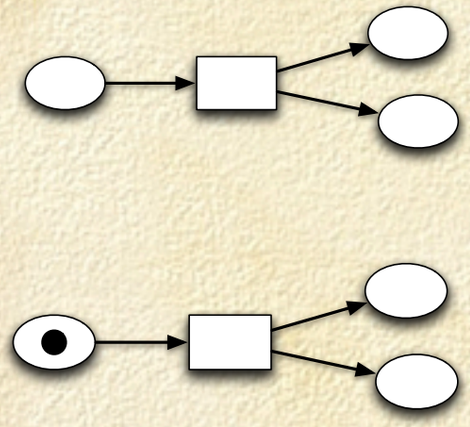
Wert verringern?

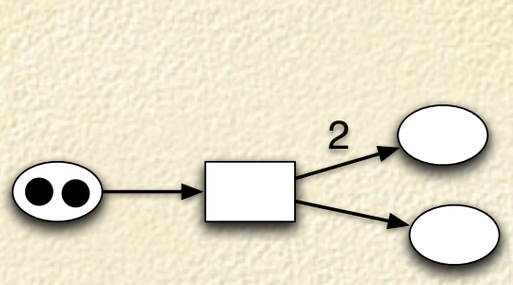
nur die Zahl 7

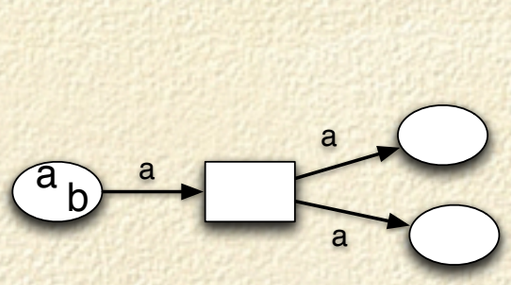
gleiches Verhalten?

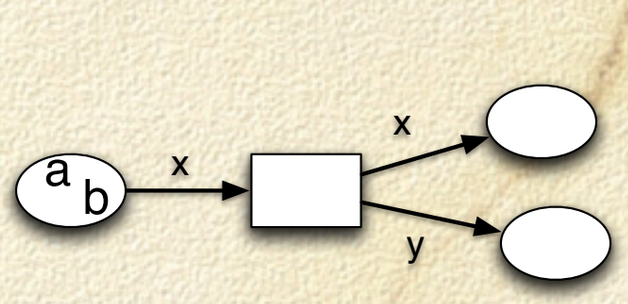
Netzklassen

- *„einfache“ Netze*
- *P/T-Netze*
- *kantenkonstante Netze*
- *gefärbte Netze*

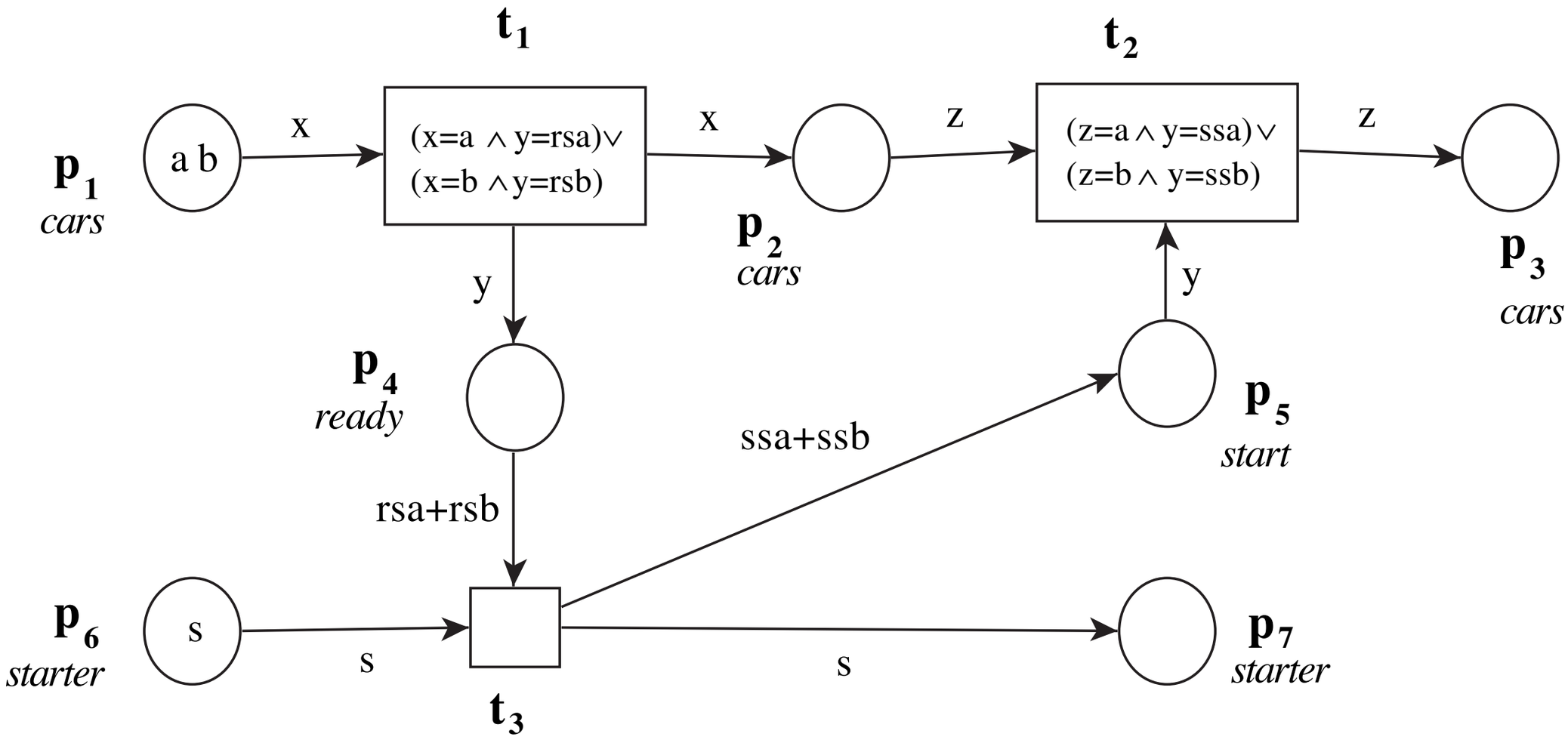








✓ ✓ ✓

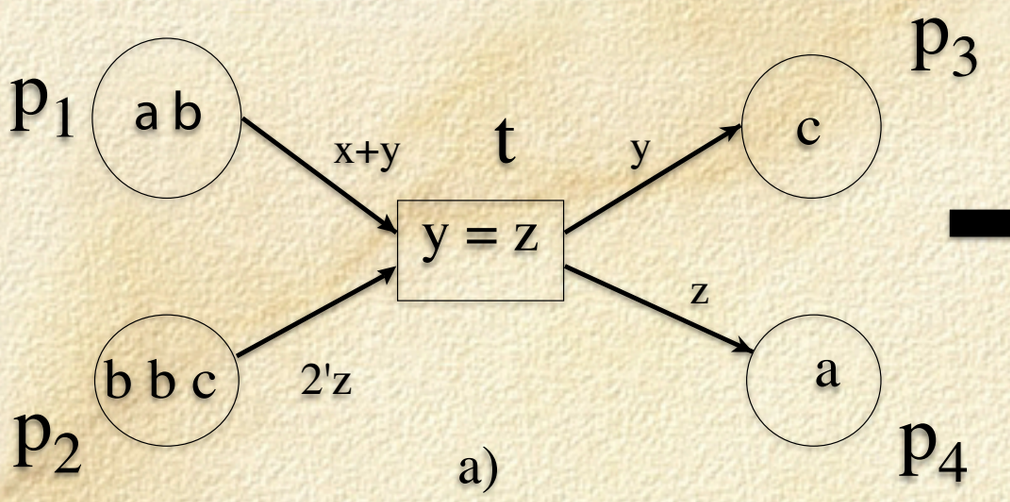


colour sets:
cars = {a,b}
starter = {s}
ready = {rsa, rsb} "*ready signs*"
start = {ssa, ssb} "*start signs*"

variables: x,y,z,s

constants: rsa,rsb,ssa,ssb

Abbildung 7.8 Gefärbtes Netz $\mathcal{N}_5$ mit Guards



Variablen-Bindung

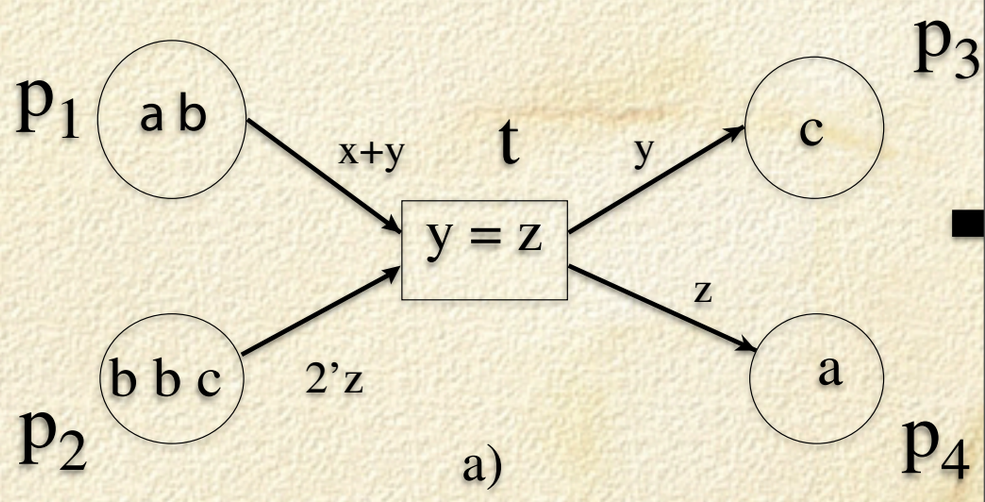
kantenkonstantes Netz

Schalten des kantenkonstanten Netzes

Abbildung 7.6 Schaltregel für gefärbte Netze

Definition 7.7 *Ein* gefärbtes Petrinetz *(coloured Petri net, CPN) wird als Tupel* $\mathcal{N} = \langle P, T, F, \mathcal{C}, cd, Var, Guards, \widehat{W}, \mathbf{m}_0 \rangle$ *definiert, wobei gilt:*

- (P, T, F) *ist ein endliches Netz (Def. 4.1),*
- $\mathcal{C}$ *ist eine Menge von* Farbenmengen,
- $cd\colon P \to \mathcal{C}$ *ist die* Farbzuweisungsabbildung *(colour domain mapping). Sie wird durch* $cd\colon F \to \mathcal{C}$, $cd(x,y) :=$ **if** $x \in P$ **then** $cd(x)$ **else** $cd(y)$ **fi** *auf* F *erweitert.*
- Var *ist eine Menge von Variablen mit Wertebereichen* $dom(x)$ *für jedes* $x \in Var$.
- $Guards = \{guard_t \mid t \in T\}$ *ordnet jeder Transition* $t \in T$ *ein Prädikat* $guard_t$ *mit Variablen aus* Var *zu.*
- $\widehat{W} = \{W_\beta \mid \beta$ *ist Belegung von* $Var\}$ *ist eine Menge von Kantengewichtungen der Form* $W_\beta : F \to Bag(\bigcup \mathcal{C})$, *wobei* $W_\beta(x,y) \in Bag(cd(x,y))$ *für alle* $(x,y) \in F$ *gilt.*
- $\mathbf{m}_0 : P \to Bag(\bigcup \mathcal{C})$ *mit* $\mathbf{m}_0(p) \in Bag(cd(p))$ *für alle* $p \in P$ *ist die Anfangsmarkierung.*



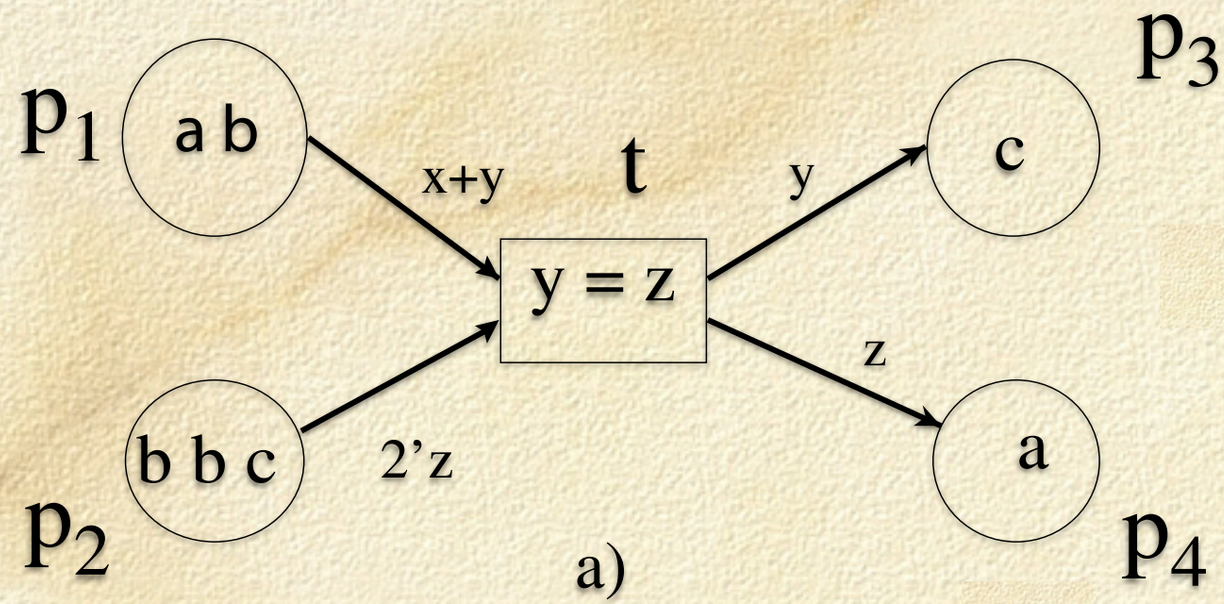
a)

wähle eine Belegung, die $guard_t$ erfüllt,

z.B.: x = a, y = b, z = b — β — a b c — β_1

b)

eine andere Belegung



a)

wähle eine Belegung, die $guard_t$ erfüllt,

z.B.: x = a, y = b, z = b — a, b, c — β_1

Definition 7.8 *a) Die Markierung eines gefärbten Netzes (CPN)*

$\mathcal{N} = \langle P, T, F, \mathcal{C}, cd, Var, Guards, \widehat{W}, \mathbf{m}_0 \rangle$ *ist ein Vektor* $\mathbf{m}$ *mit* $\mathbf{m}(p) \in Bag(cd(p))$ *für jedes* $p \in P$ *(auch als Abbildung* $\mathbf{m} : P \to Bag(\bigcup \mathcal{C})$ *mit* $\mathbf{m}(p) \in Bag(cd(p))$ *für jedes* $p \in P$ *aufzufassen).*

b) Sei β *eine Belegung für* Var*. Die Transition* $t \in T$ *heißt* β-aktiviert *in einer Markierung* $\mathbf{m}$ *falls* $guard_t(\beta) = true$ *und* $\forall p \in {}^\bullet t.\ \mathbf{m}(p) \geq W_\beta(p,t)$ *(als Relation:* $\mathbf{m} \xrightarrow{t,\beta}$*).*

c) Es sei

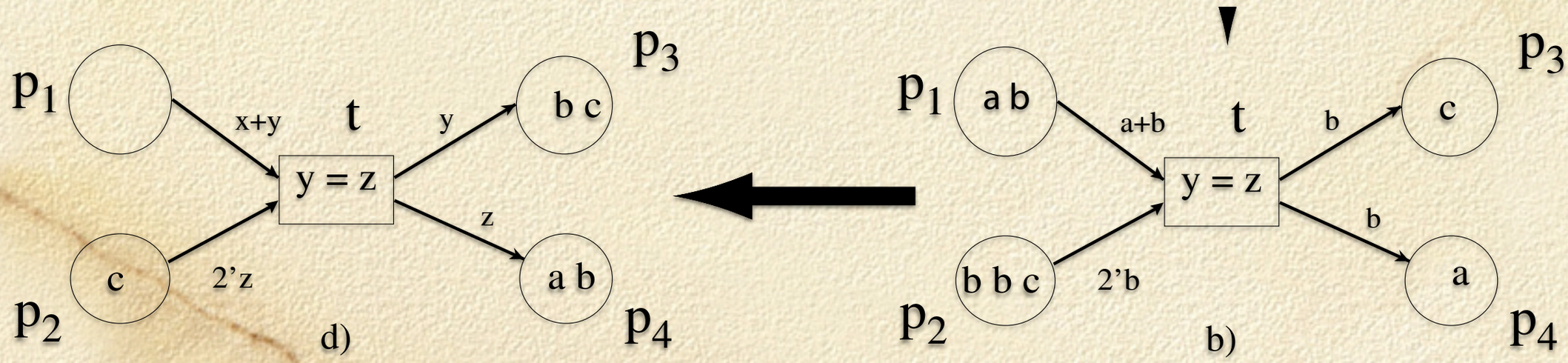
$$\widetilde{W}_\beta(p,t) := \begin{cases} W_\beta(p,t) & \textit{falls } (p,t) \in F \\ \emptyset & \textit{sonst} \end{cases} \quad \textit{und entsprechend}$$

$$\widetilde{W}_\beta(t,p) := \begin{cases} W_\beta(t,p) & \textit{falls } (t,p) \in F \\ \emptyset & \textit{sonst} \end{cases} .$$

Ist t in $\mathbf{m}$ $\beta-$aktiviert, dann ist die Nachfolgemarkierung *definiert durch:*

$\mathbf{m} \xrightarrow{t,\beta} \mathbf{m'} \Leftrightarrow \forall p \in P.\ (\mathbf{m}(p) \geq \widetilde{W}_\beta(p,t) \wedge \mathbf{m'}(p) = \mathbf{m}(p) - \widetilde{W}_\beta(p,t) + \widetilde{W}_\beta(t,p))$.

(Beachte, dass es sich um Multimengenoperationen handelt!).

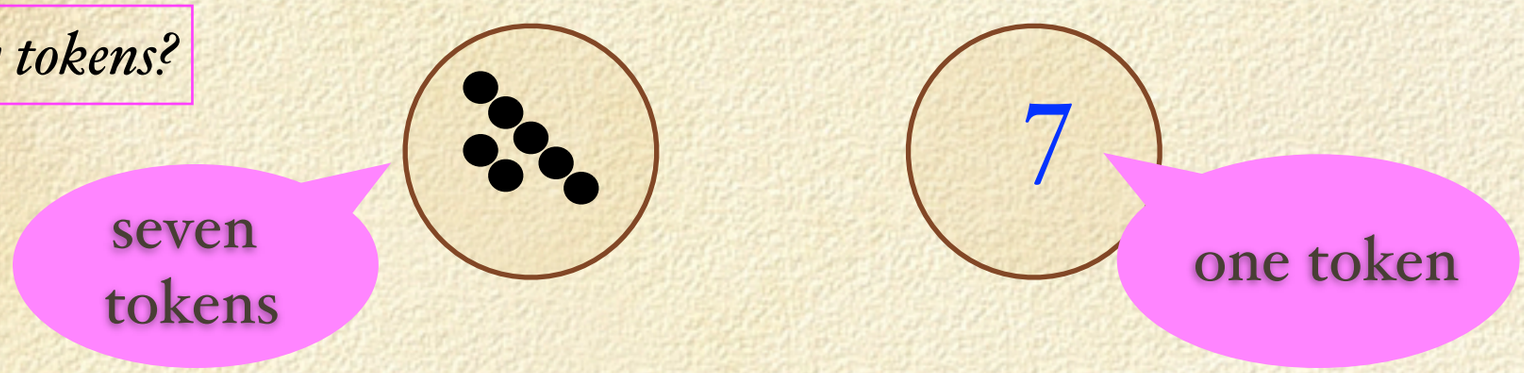


$7' \bullet \quad \overset{?}{\cong} \quad 7 \in \mathbb{N}$

P/T - net

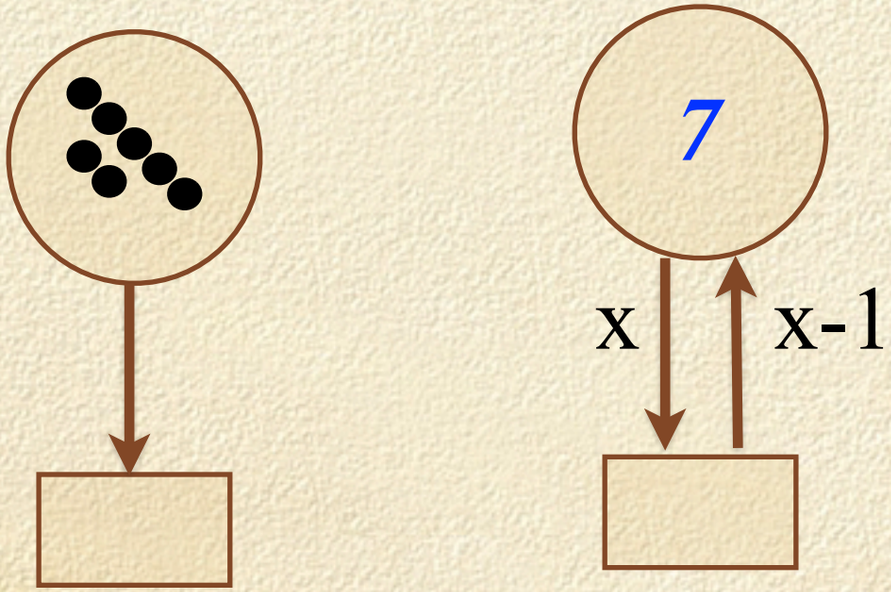
arc-constant net

how many tokens?



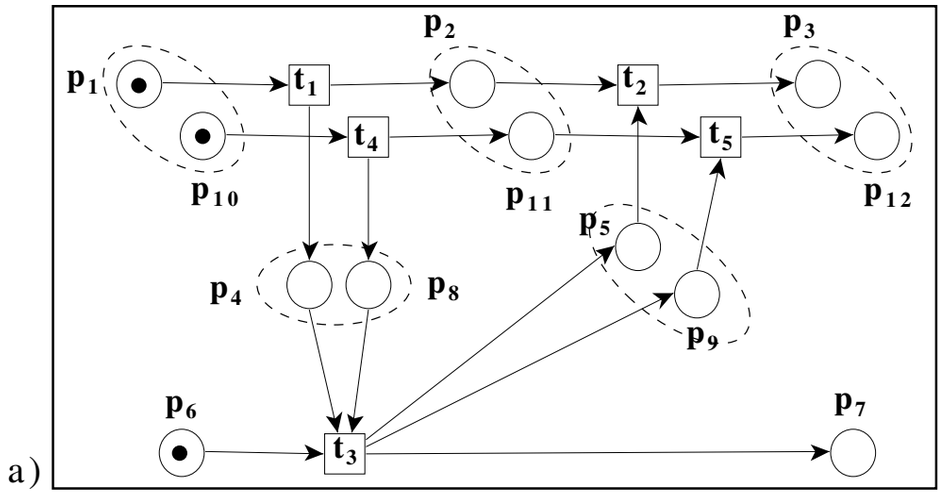
colour set: {●}

counting down

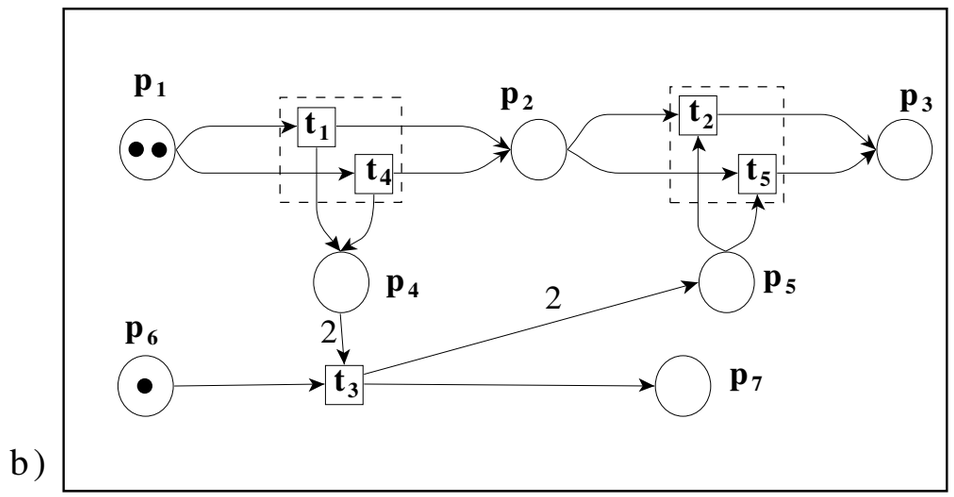


coloured net

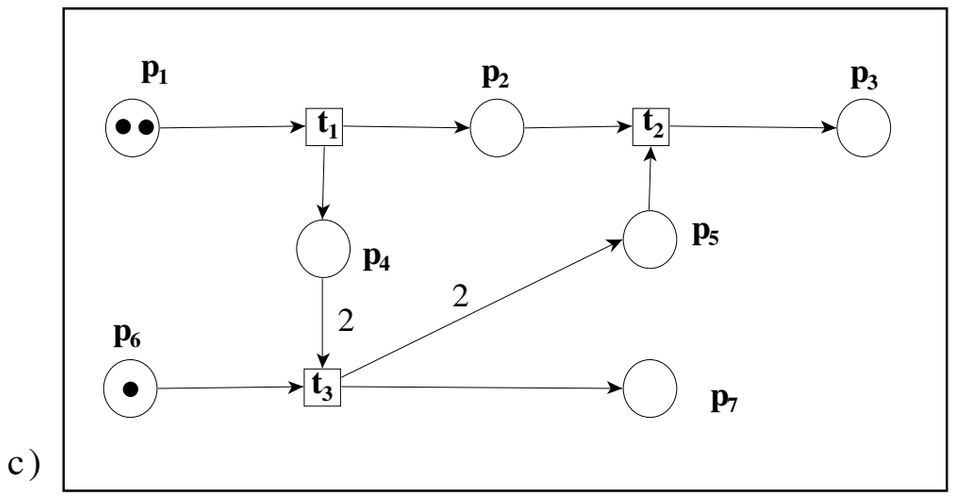
(Seite 243)



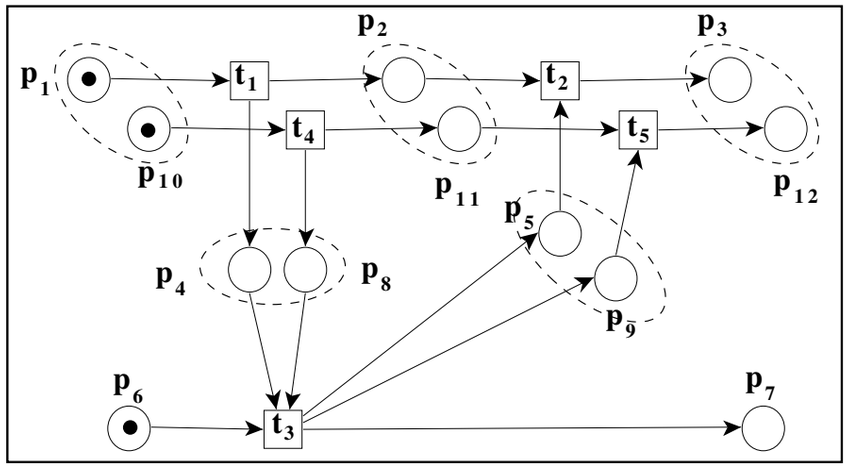
φ_1



φ_2

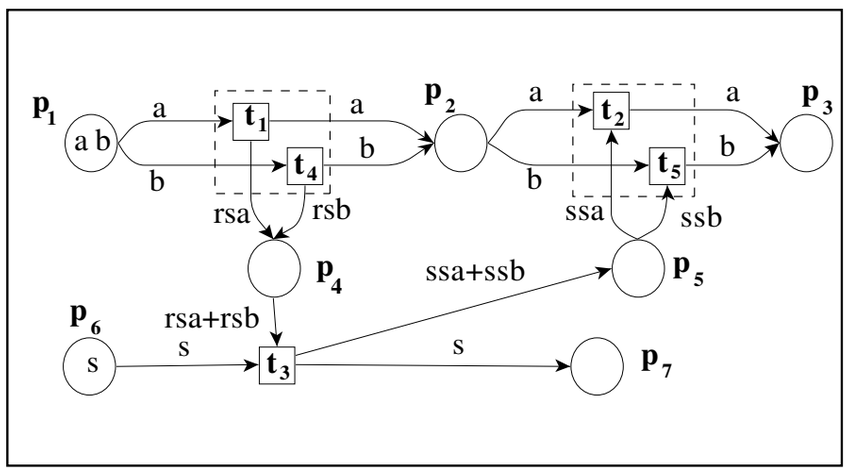


a)



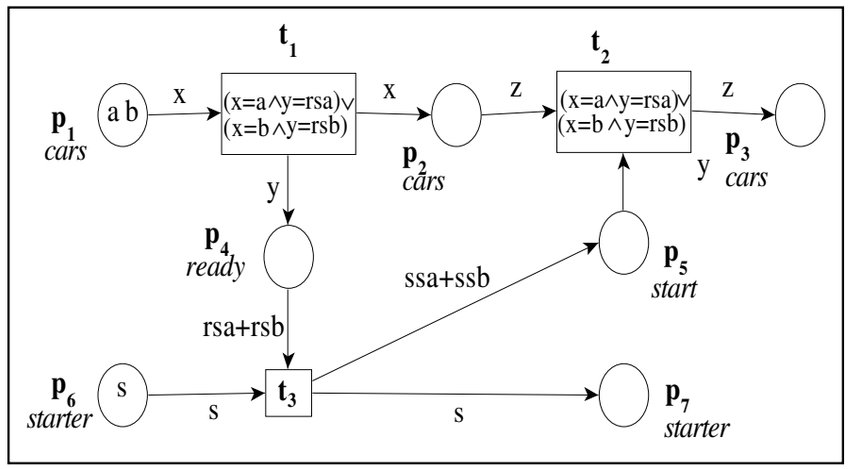
φ_3

b)



φ_4

c)



7.3 Das RENEW-Werkzeug

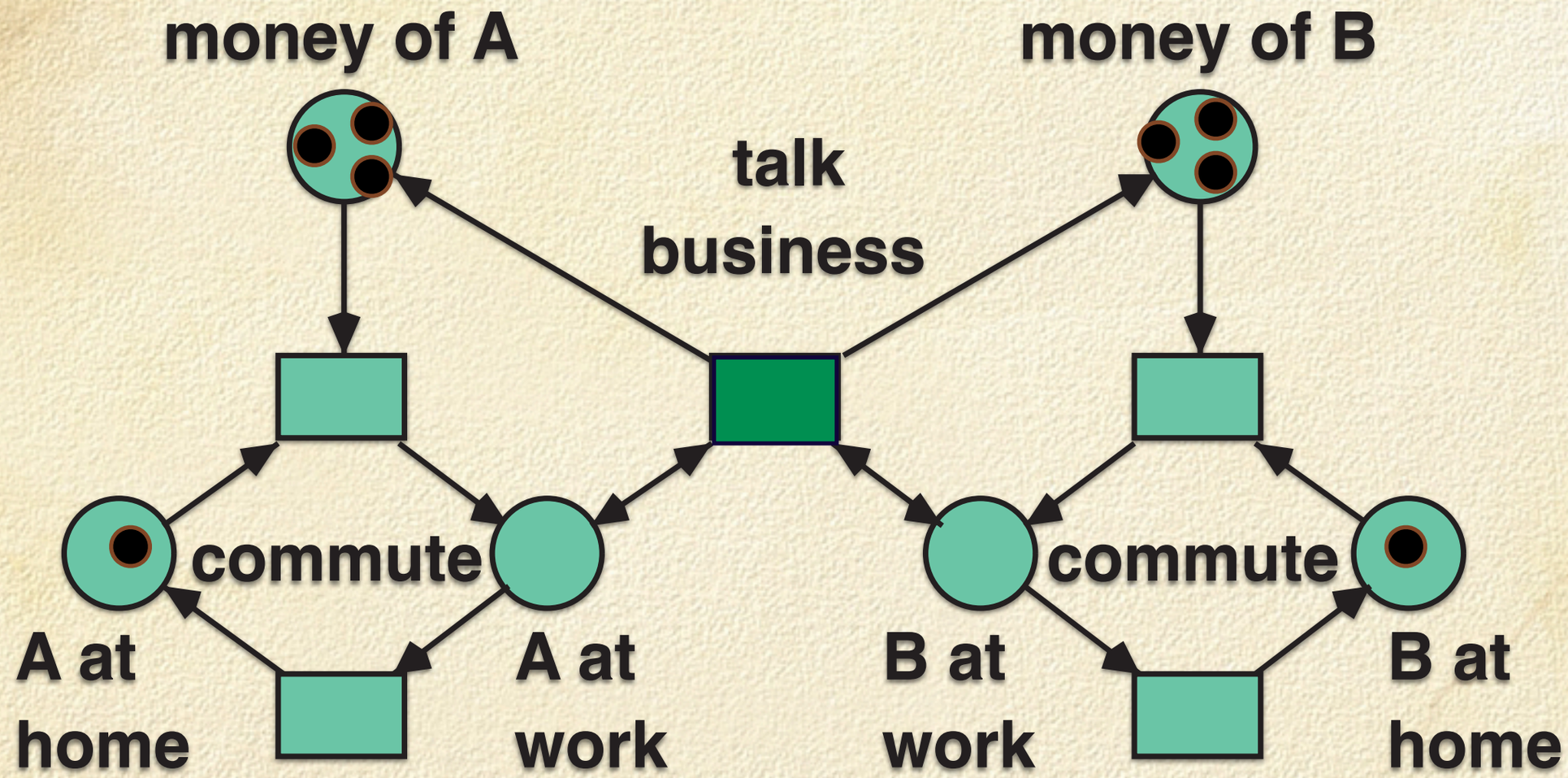


Abbildung 7.11

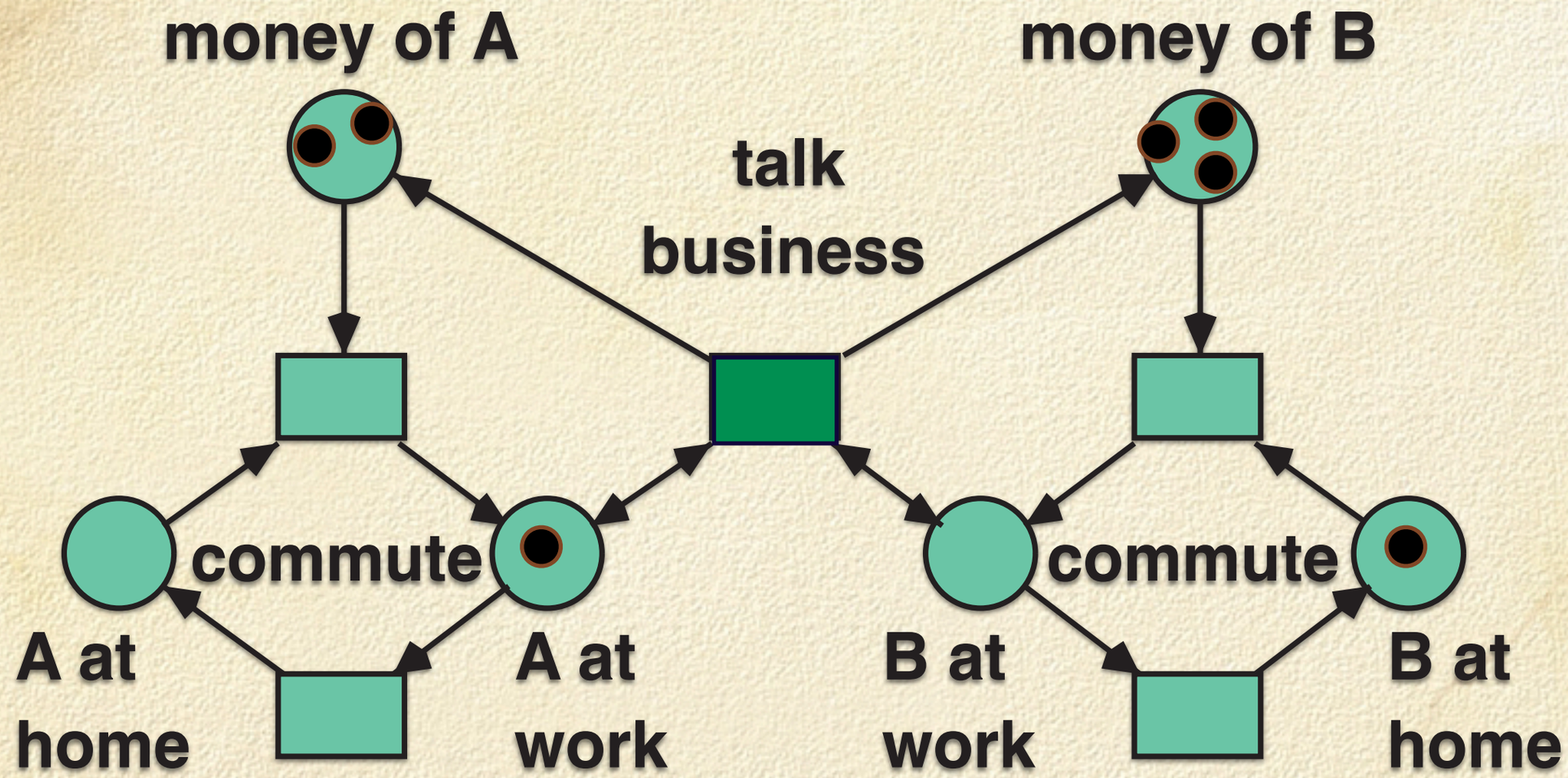


Abbildung 7.11

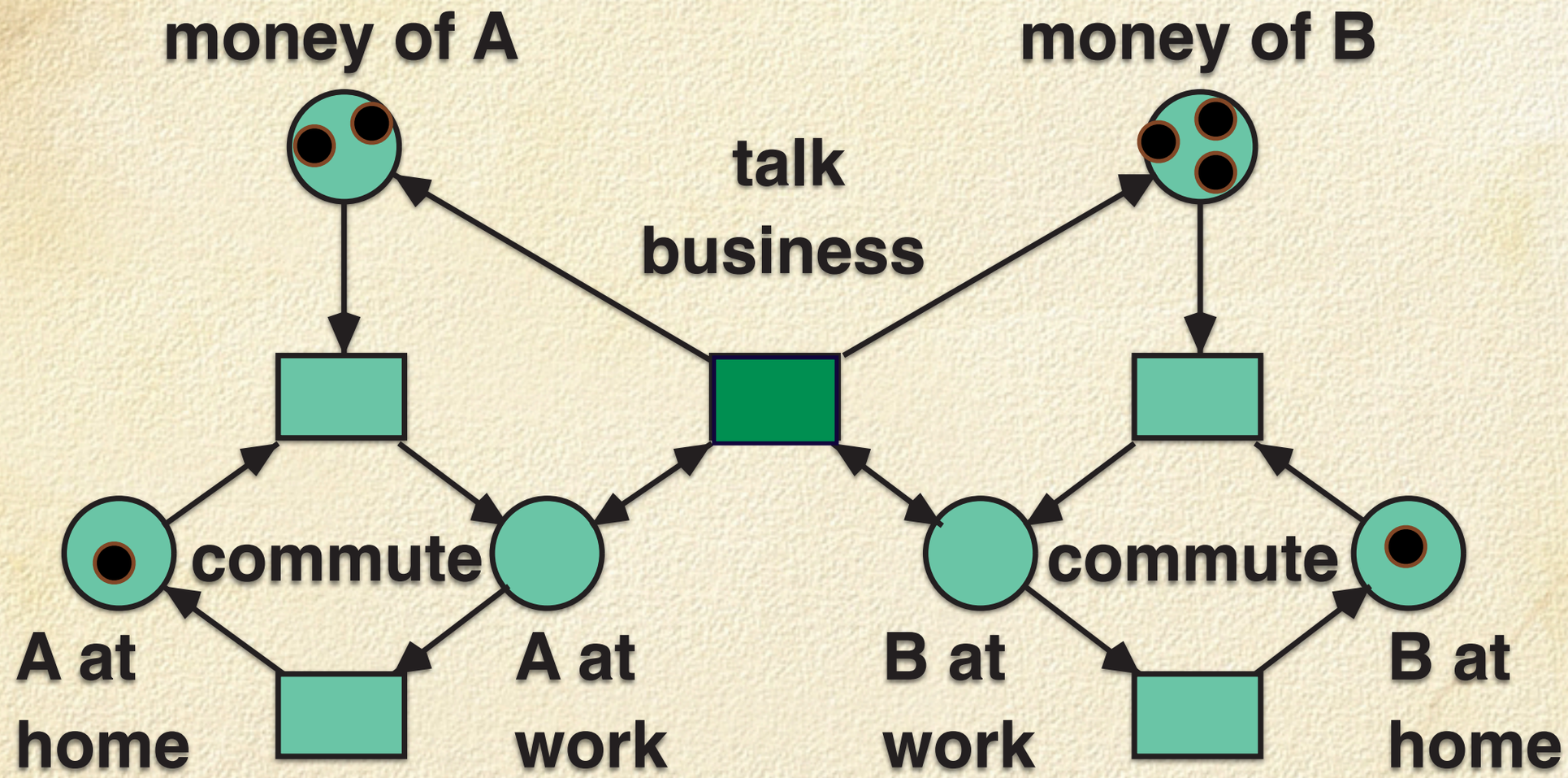


Abbildung 7.11

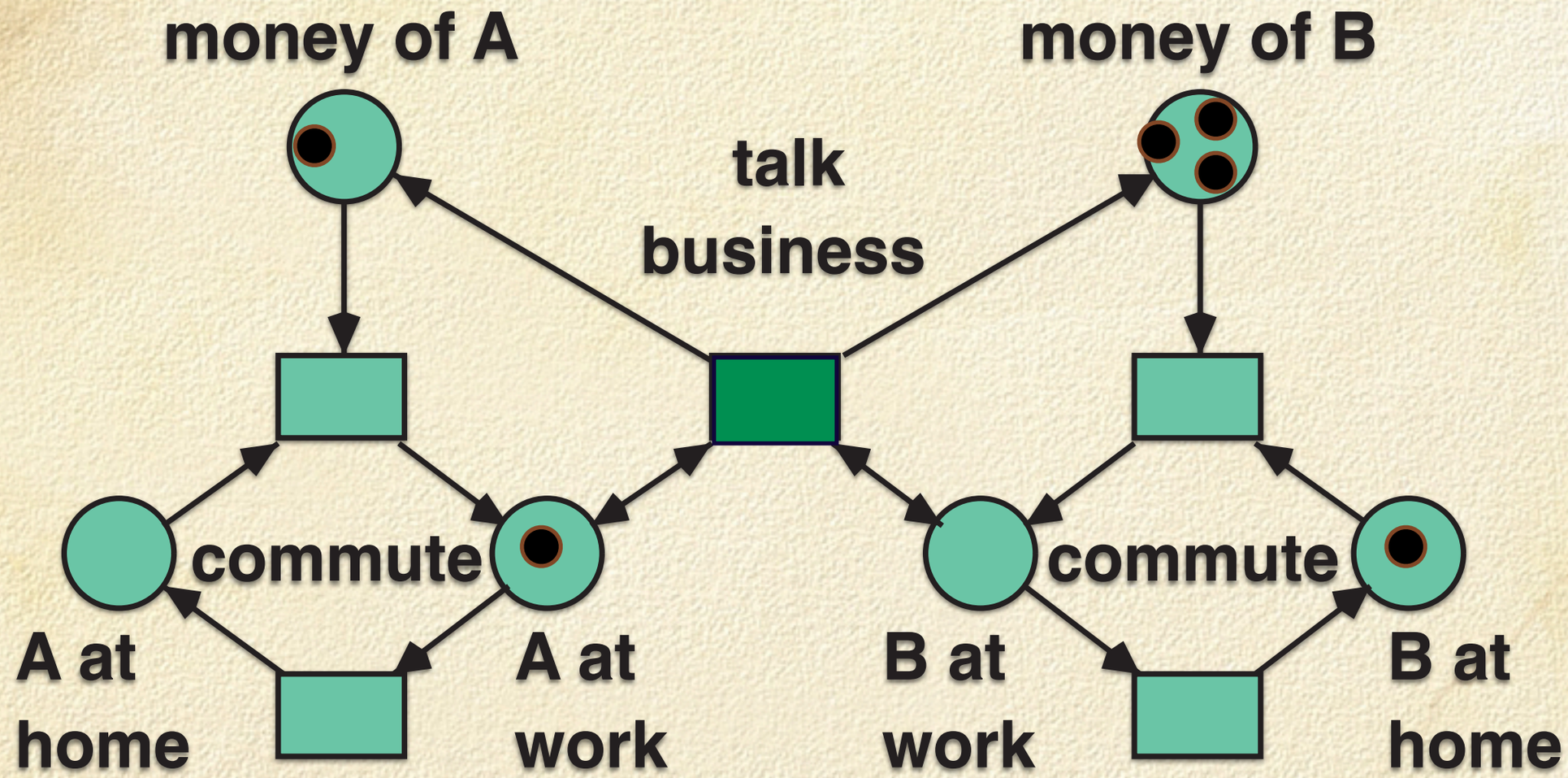


Abbildung 7.11

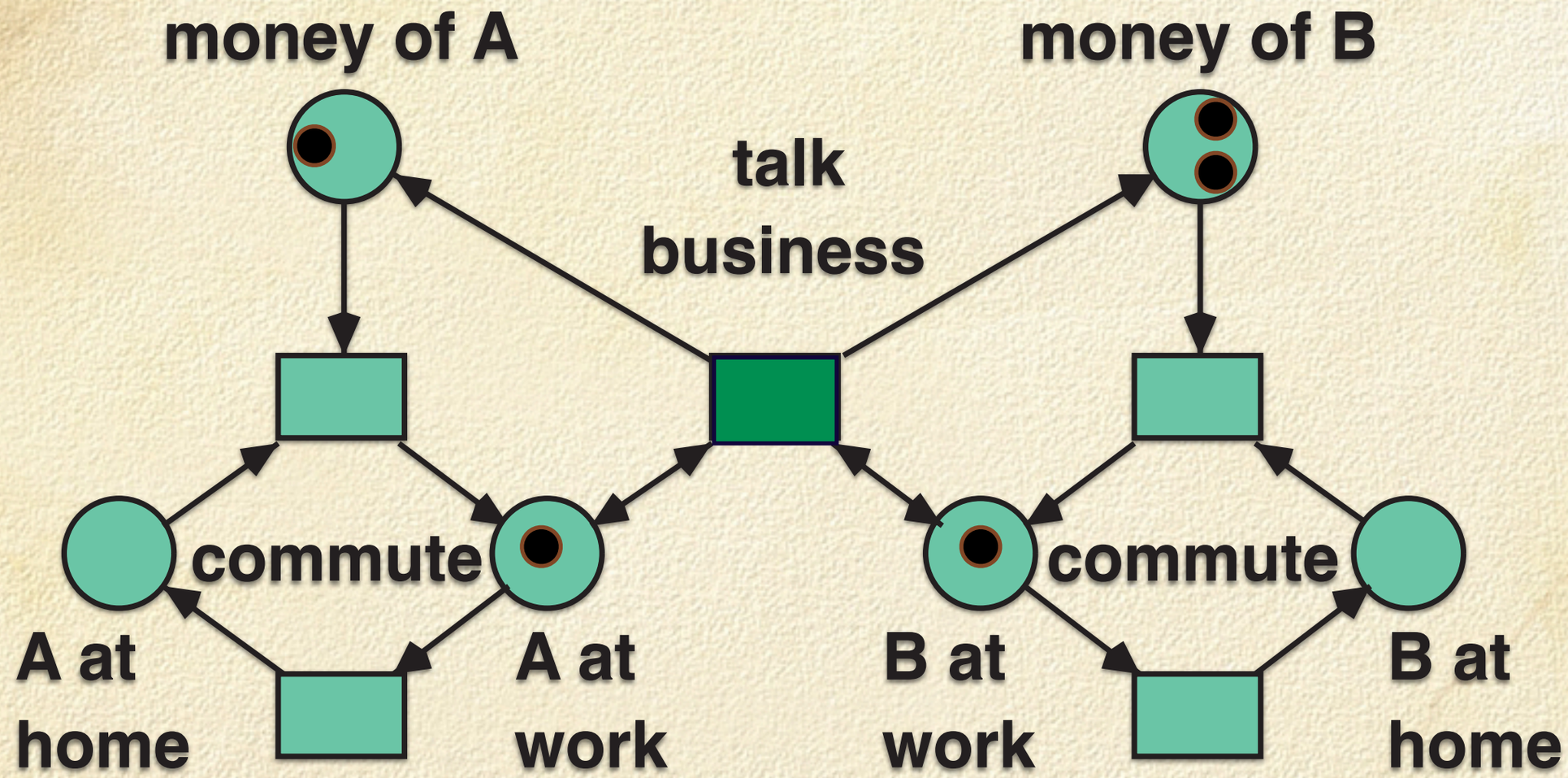


Abbildung 7.11

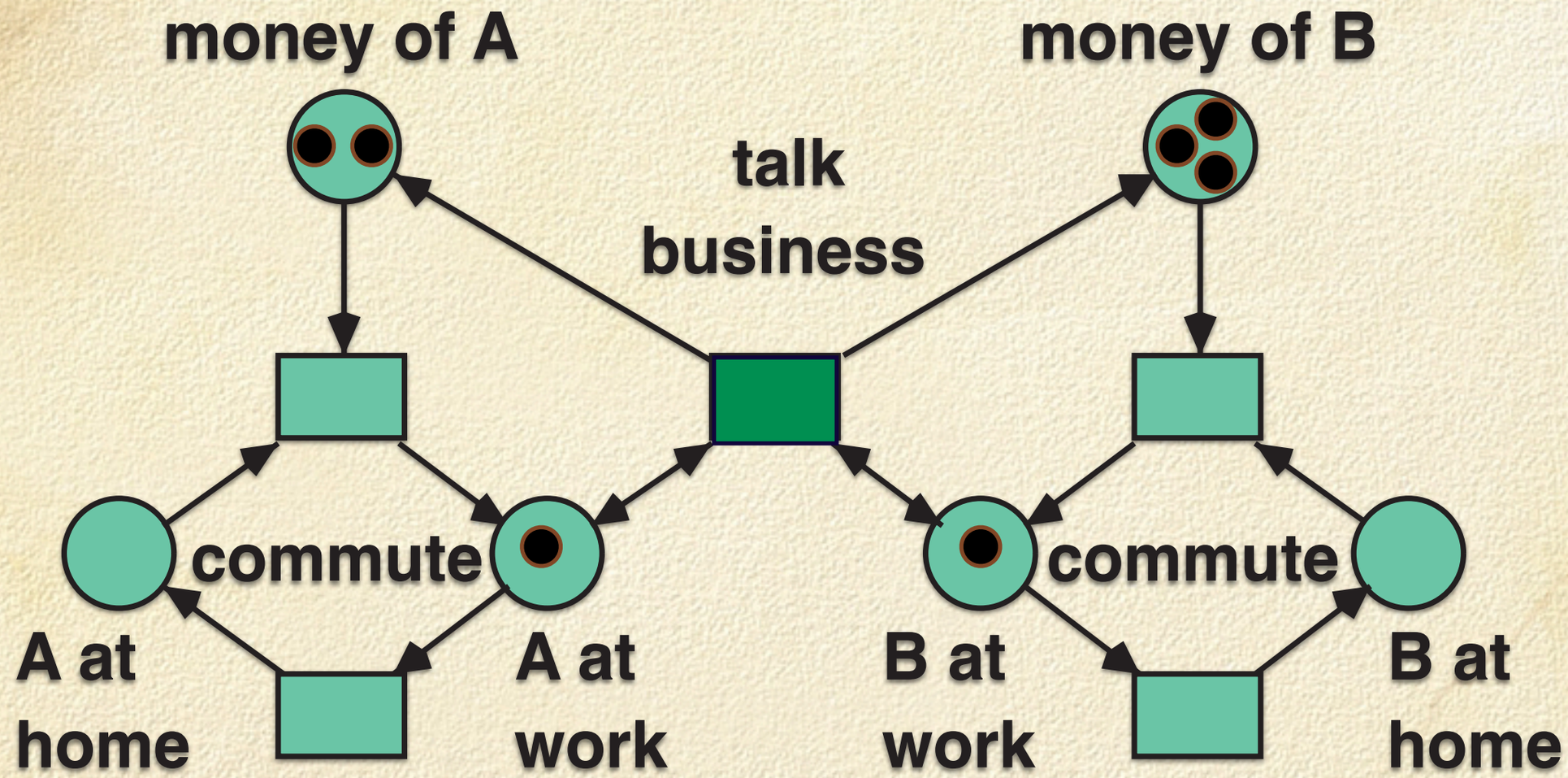


Abbildung 7.11

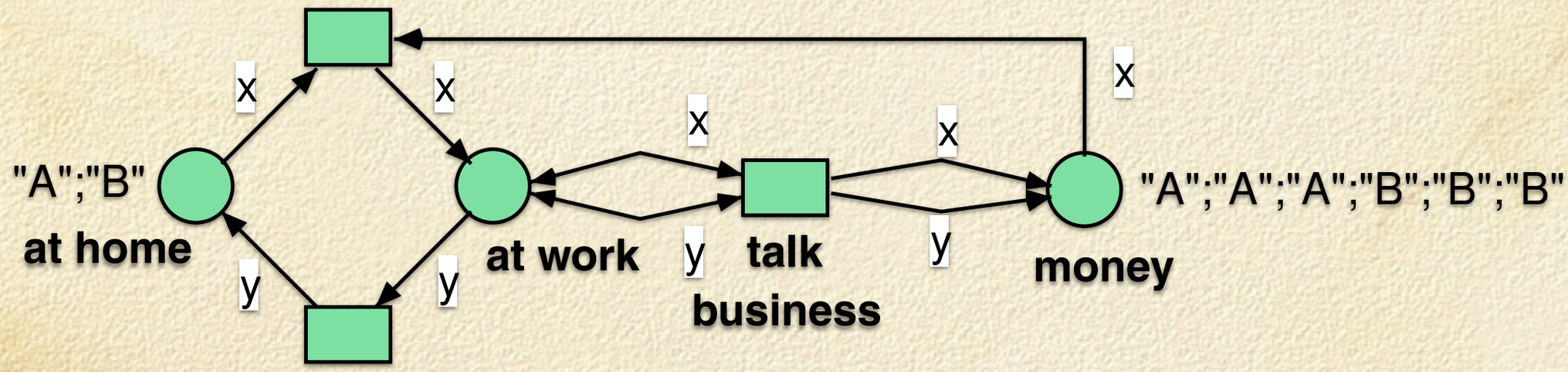


Abbildung 7.12

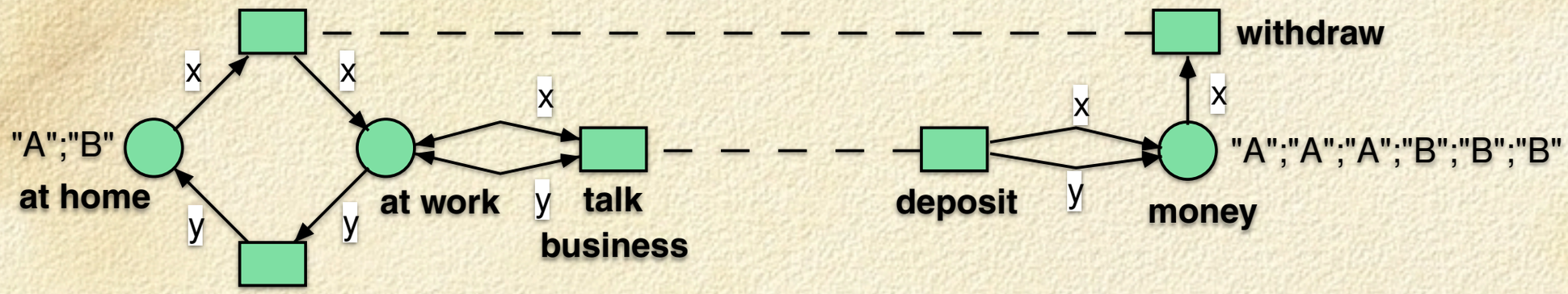


Abbildung **7.15** Personen und Konten werden geteilt

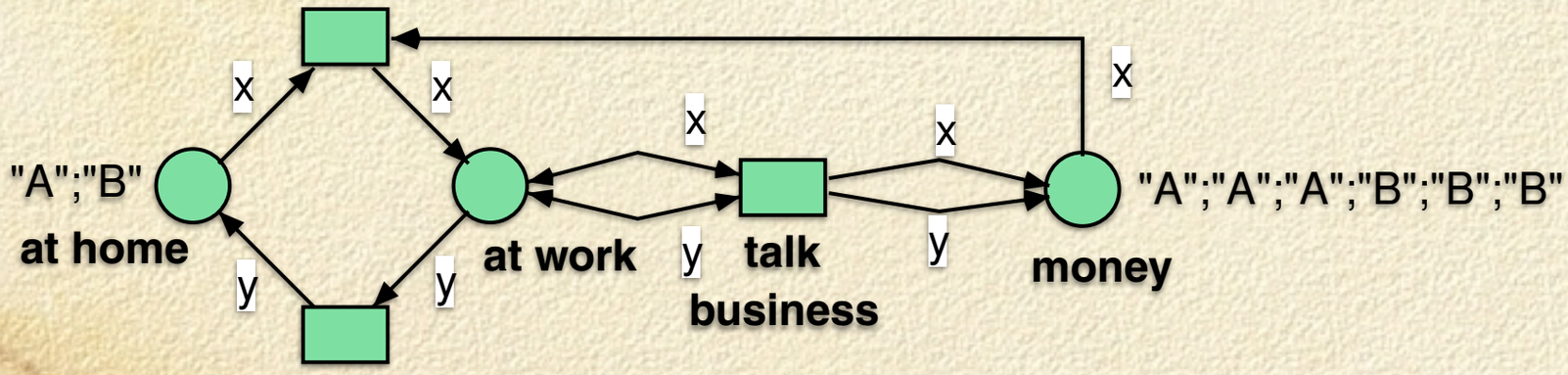


Abbildung 7.12

Netz-Kanäle

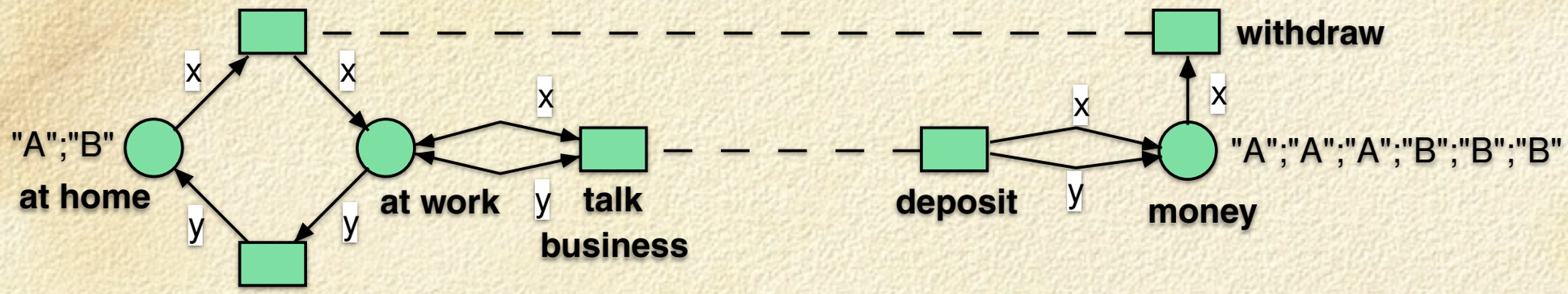


Abbildung **7.15** Personen und Konten werden geteilt

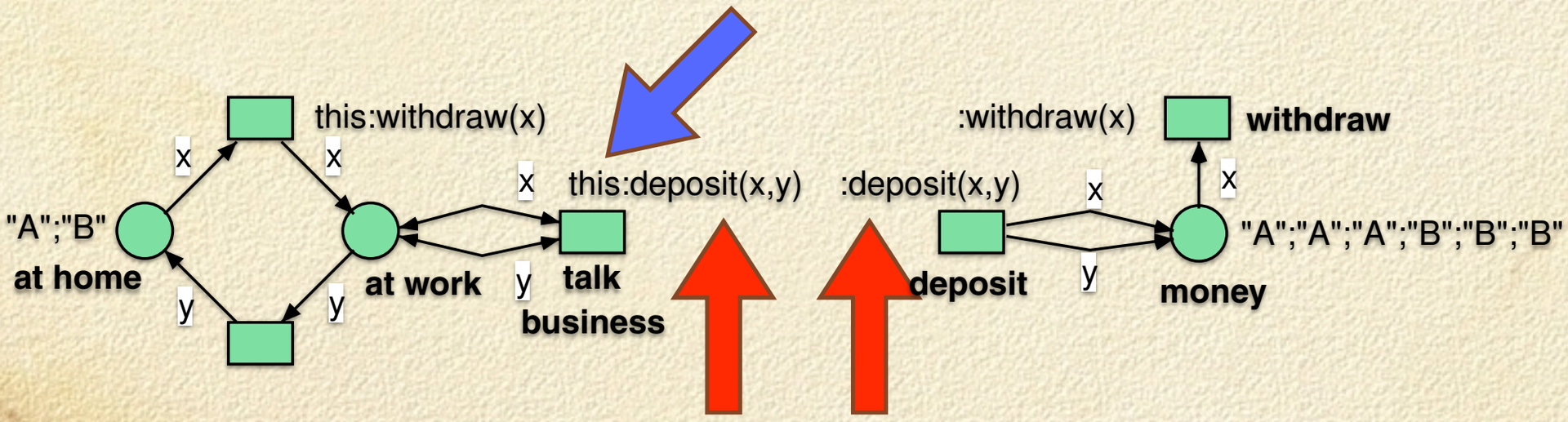


Abbildung **7.16** Textuelle Anschriften kennzeichnen Synchronisation

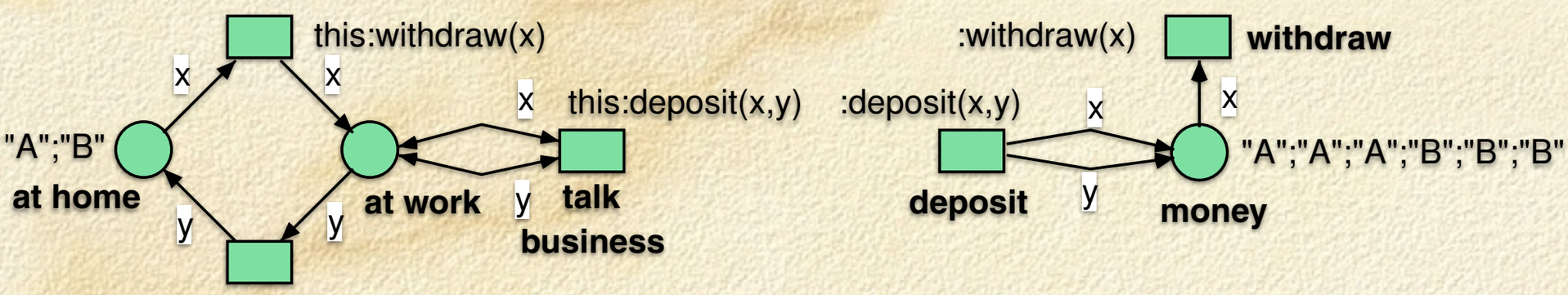


Abbildung 7.16 Textuelle Anschriften kennzeichnen Synchronisation

Bezeichner eines Netzes

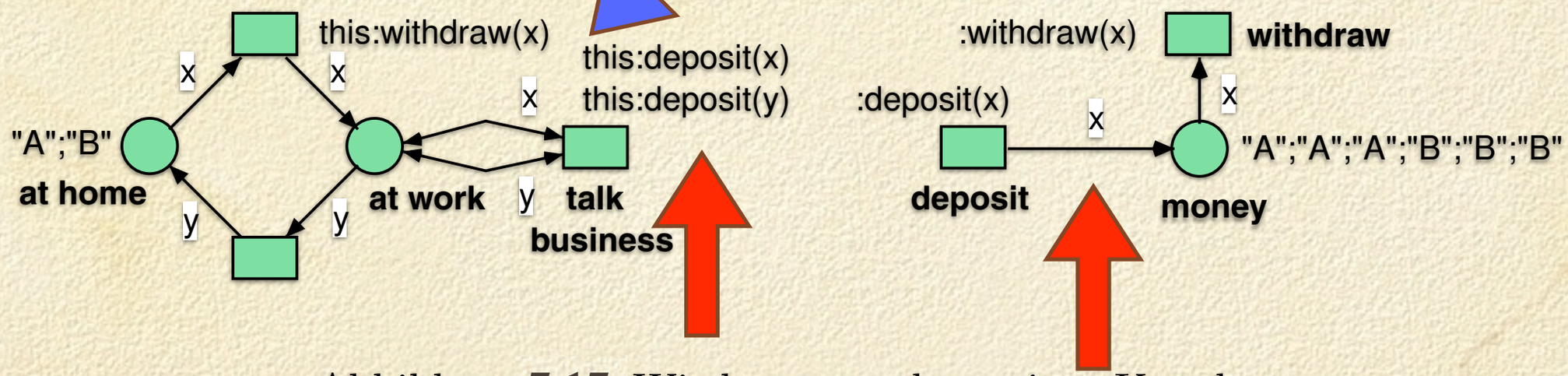


Abbildung 7.17 Wiederverwendung eines Kanals

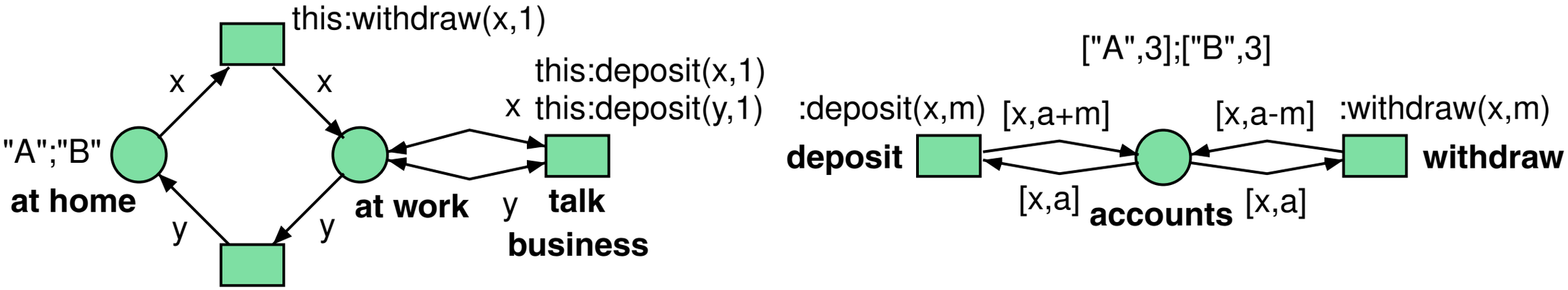


Abbildung **7.18** Buchführung mit Algebra

Netzinstanzen

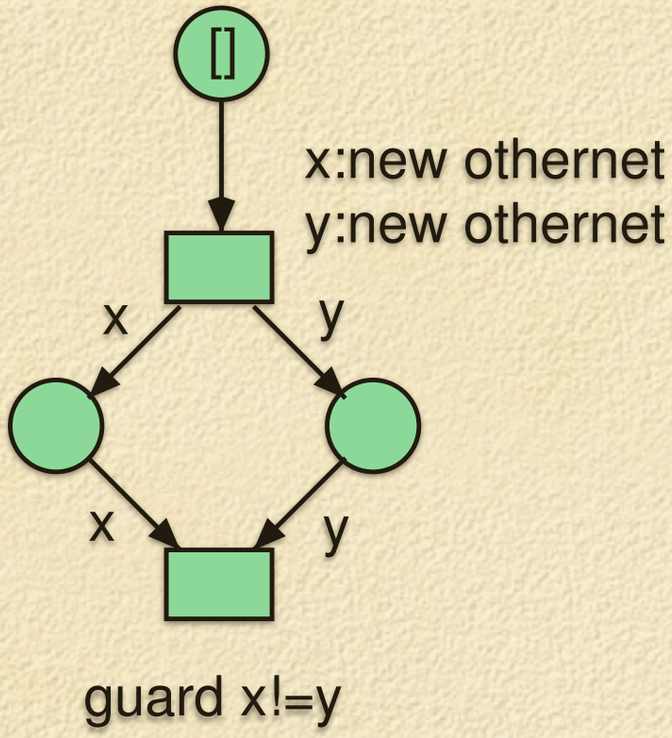


Abbildung Das Haupt-netz `creator`

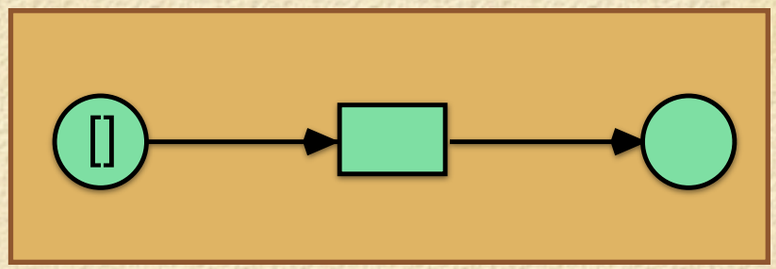


Abbildung Das Netz `othernet`

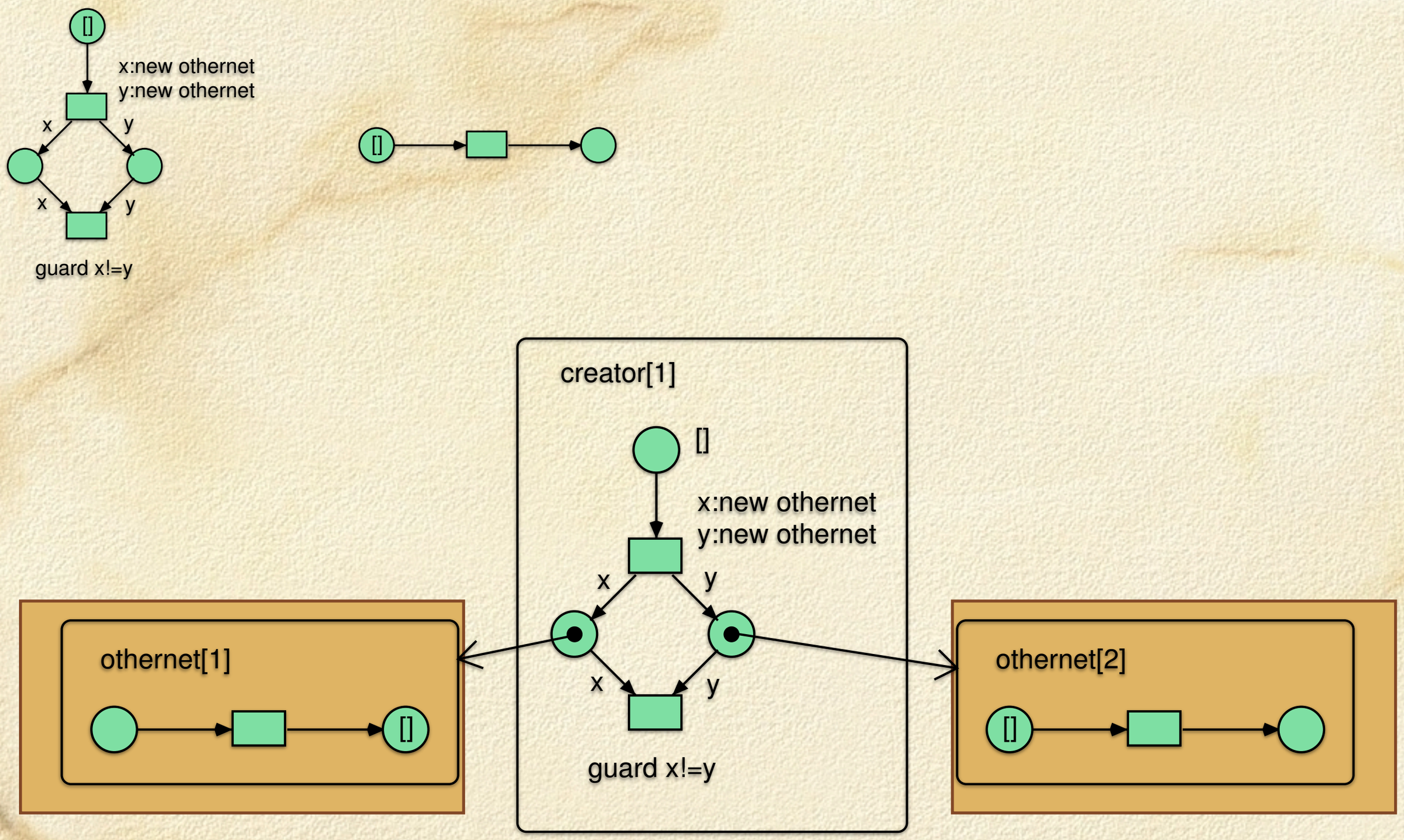


Abbildung Netzexemplare mit Netzreferenzen

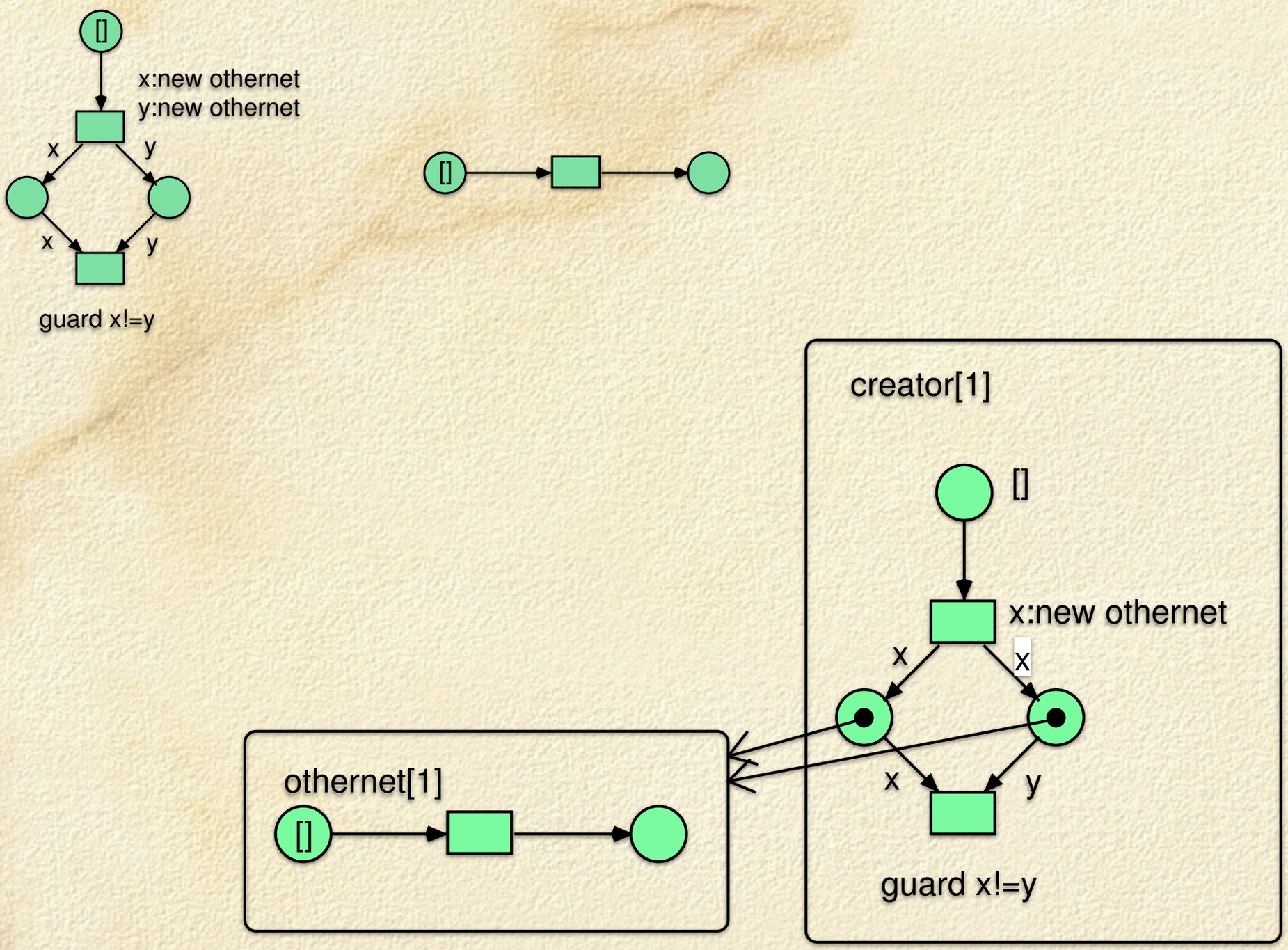
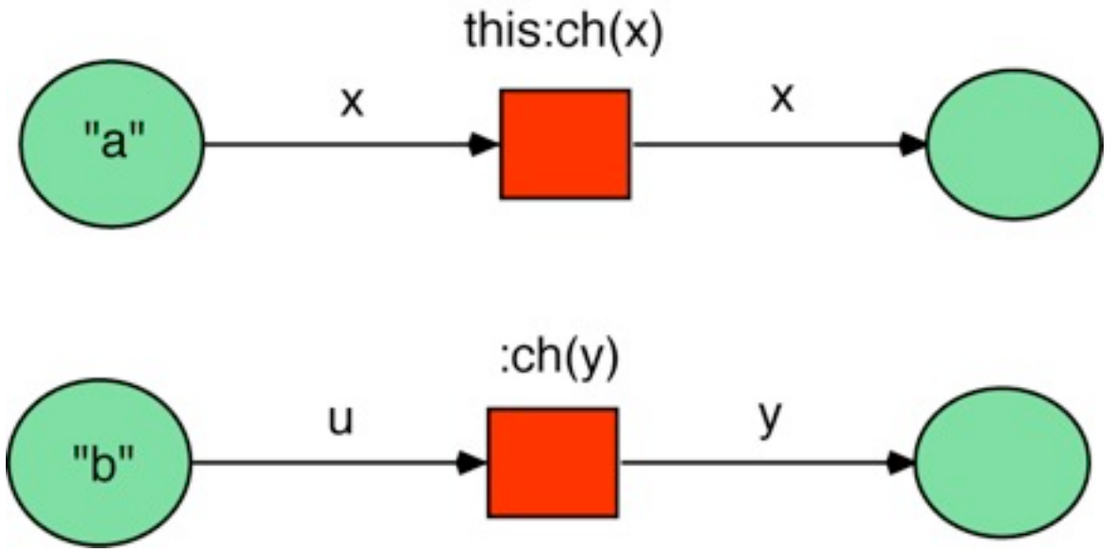
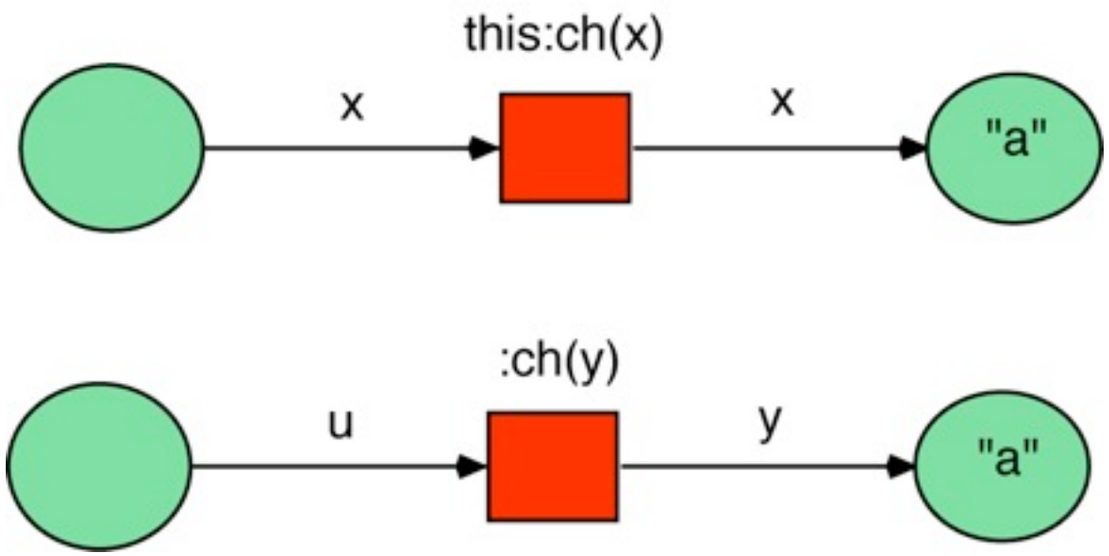
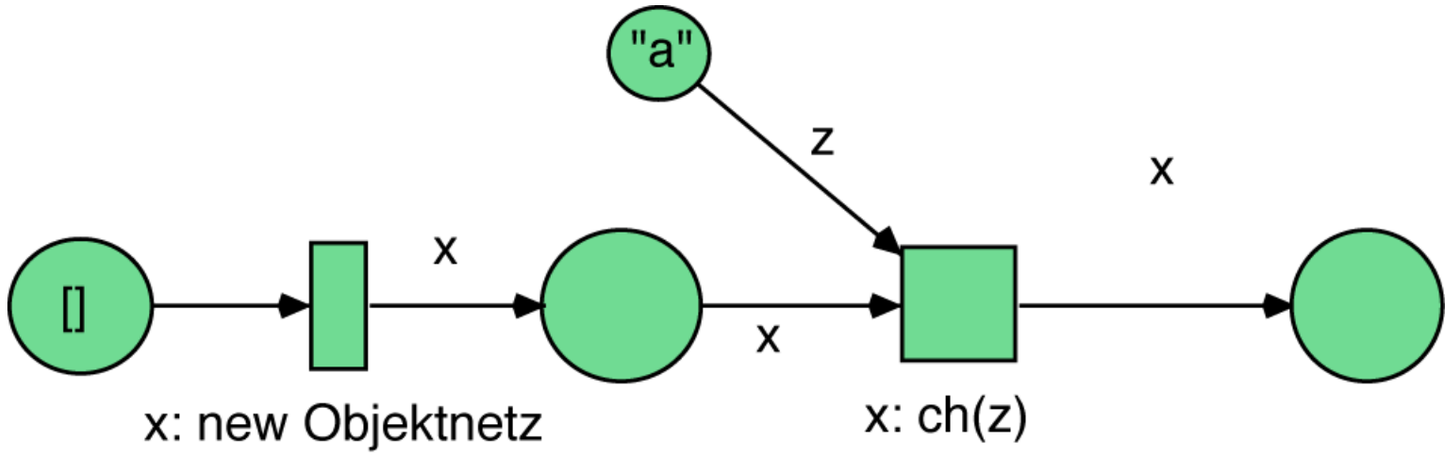


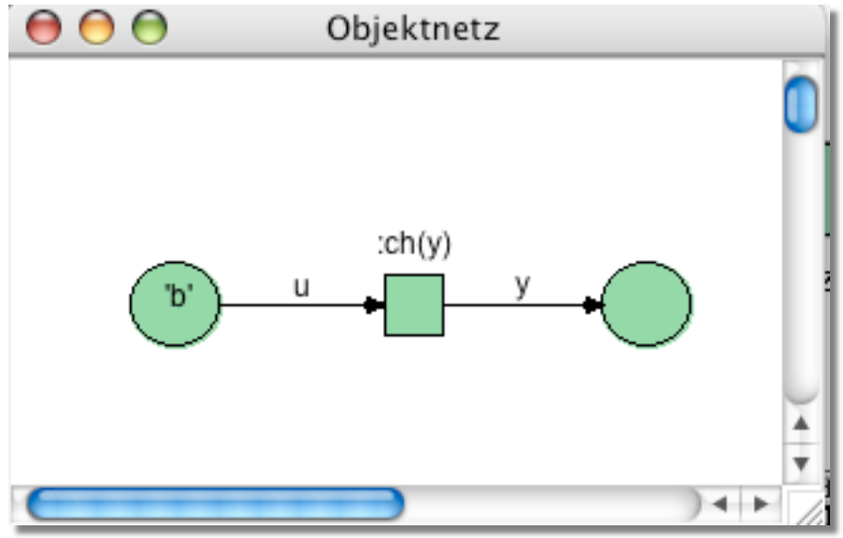
Abbildung · Netzexemplare mit Netzreferenzen: Varian-
te

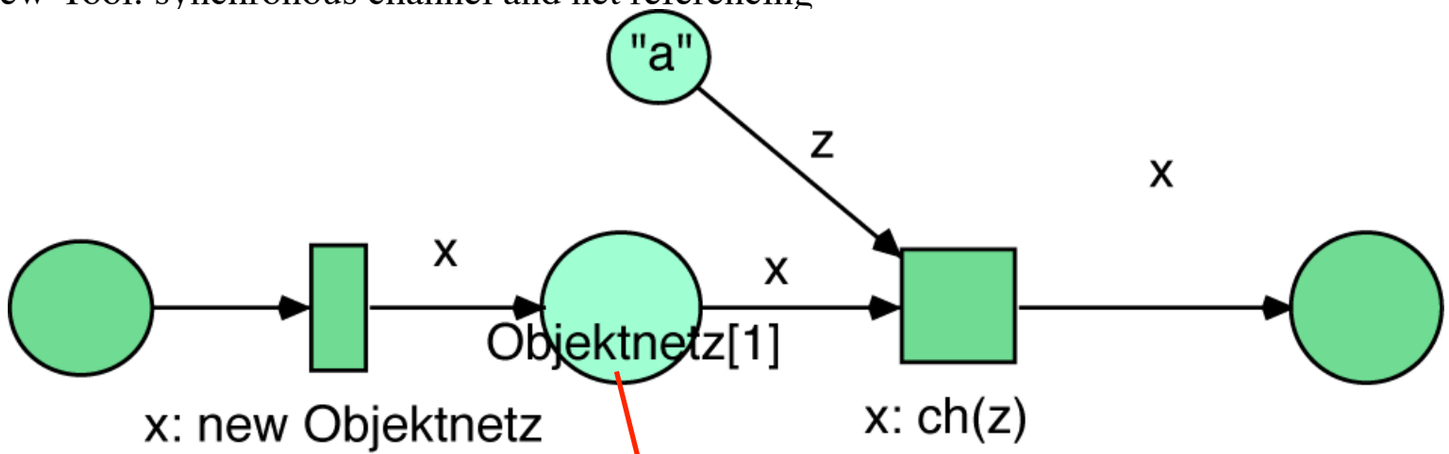
Synchrone Kanäle mit Parameterübergabe

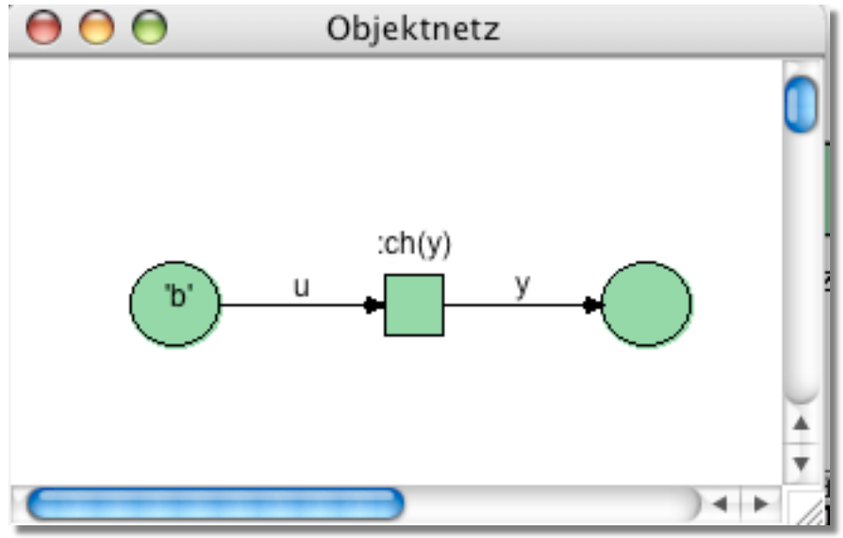


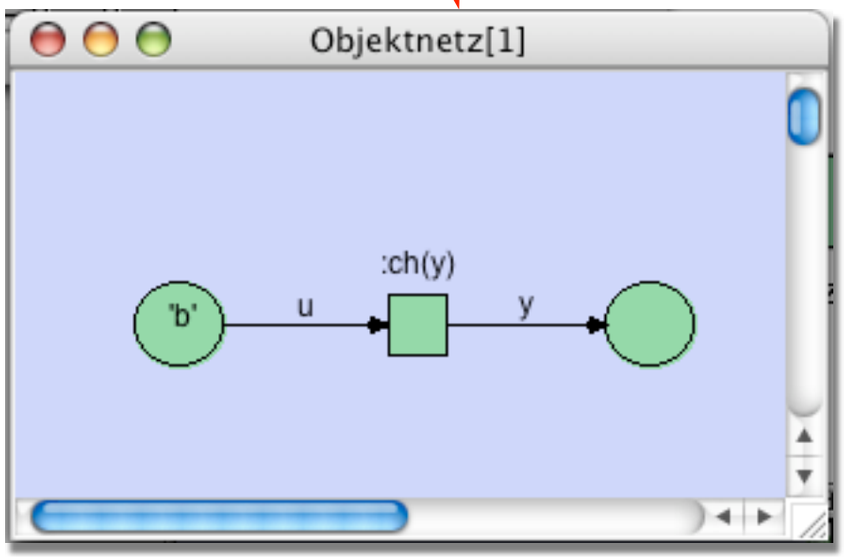


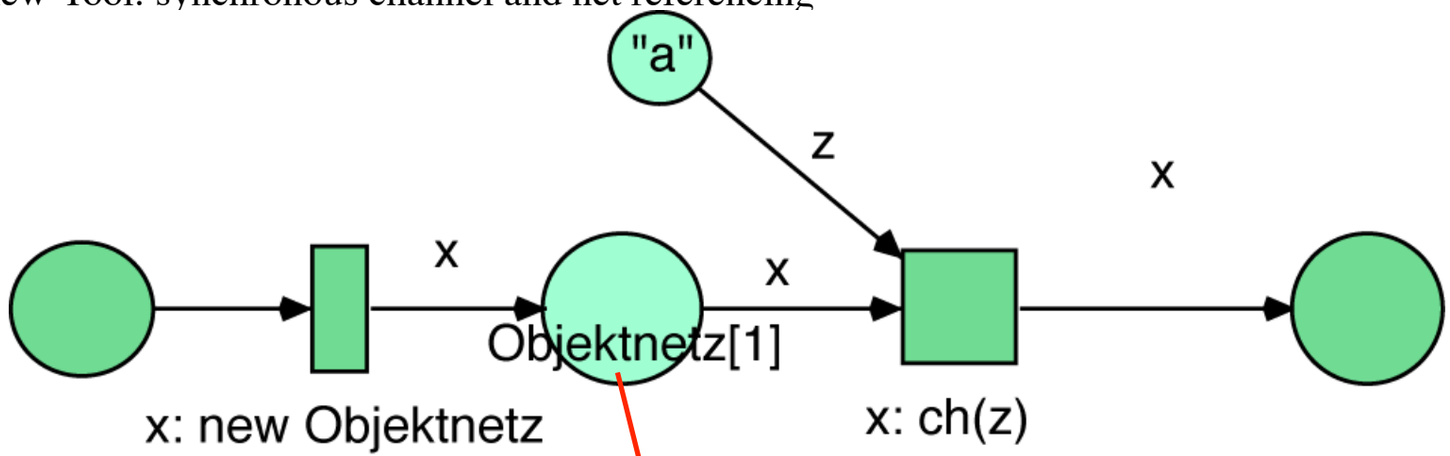

"a"
z
x
[]
x
x
x: new Objektnetz
x: ch(z)

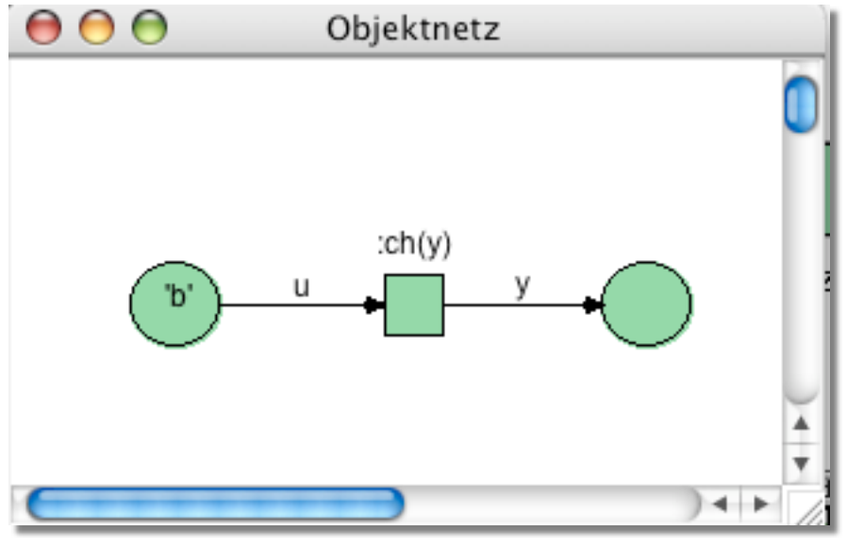

Objektnetz
:ch(y)
'b'
u
y

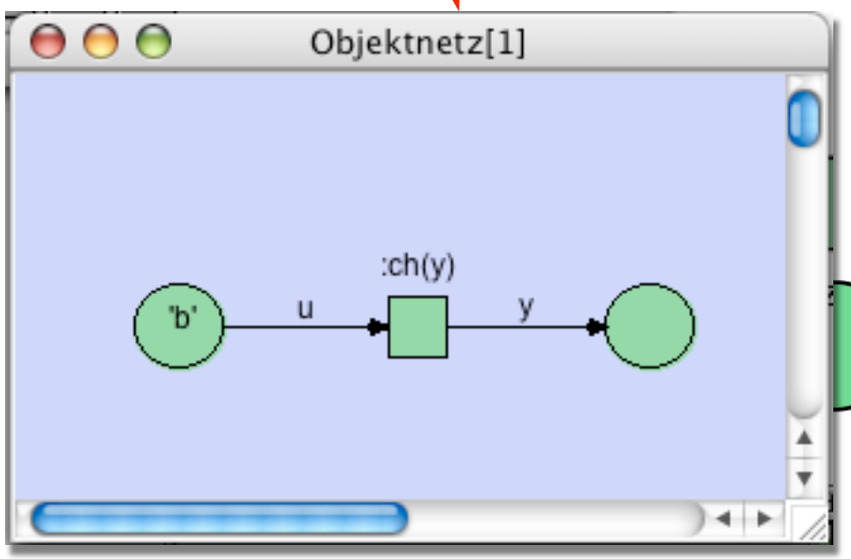

"a"
z
x
x
x
Objektnetz[1]
x: new Objektnetz
x: ch(z)

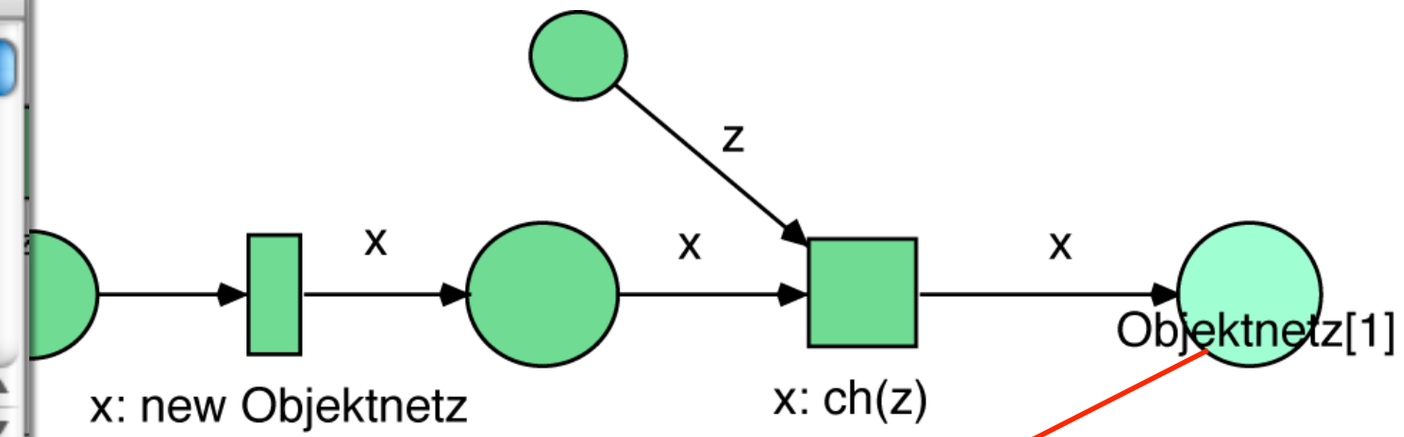

Objektnetz
:ch(y)
'b'
u
y

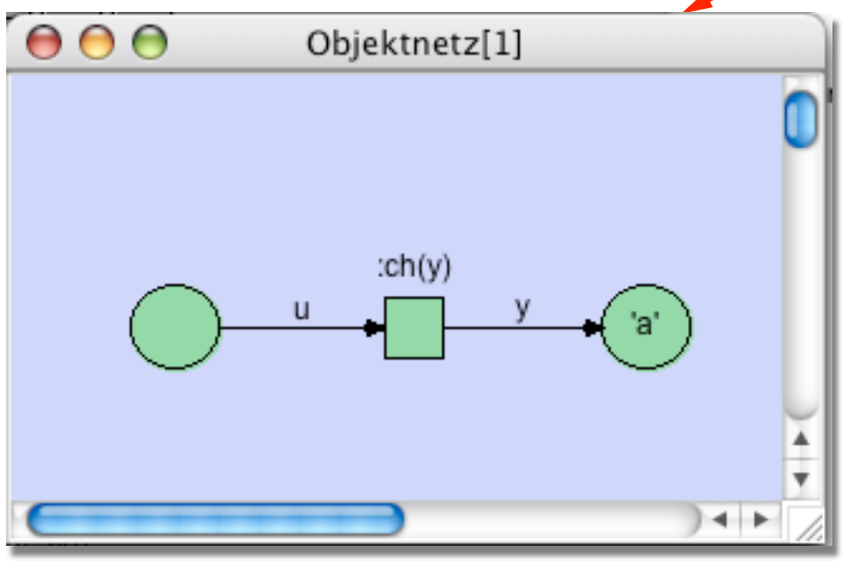

Objektnetz[1]
:ch(y)
'b'
u
y


"a"
z
x
x
x
Objektnetz[1]
x: new Objektnetz
x: ch(z)


Objektnetz
:ch(y)
'b'
u
y


Objektnetz[1]
:ch(y)
'b'
u
y


z
x
x
x
Objektnetz[1]
x: new Objektnetz
x: ch(z)


Objektnetz[1]
:ch(y)
u
y
'a'

Klausurtermine (ohne Gewähr)

- 1. Klausur: 01.03.2011, 9.30 - 11.30, Audi 1 (Achtung! Zeit geändert!)
- 2. Klausur: 01.04.2011, 9.30 - 11.30, ESA B (Achtung! Zeit geändert!)

Zur Vorbereitung auf die Klausuren werden Repetitorien angeboten.

- Alternierbit - Protokoll

  gefärbtes Netz in RENEW

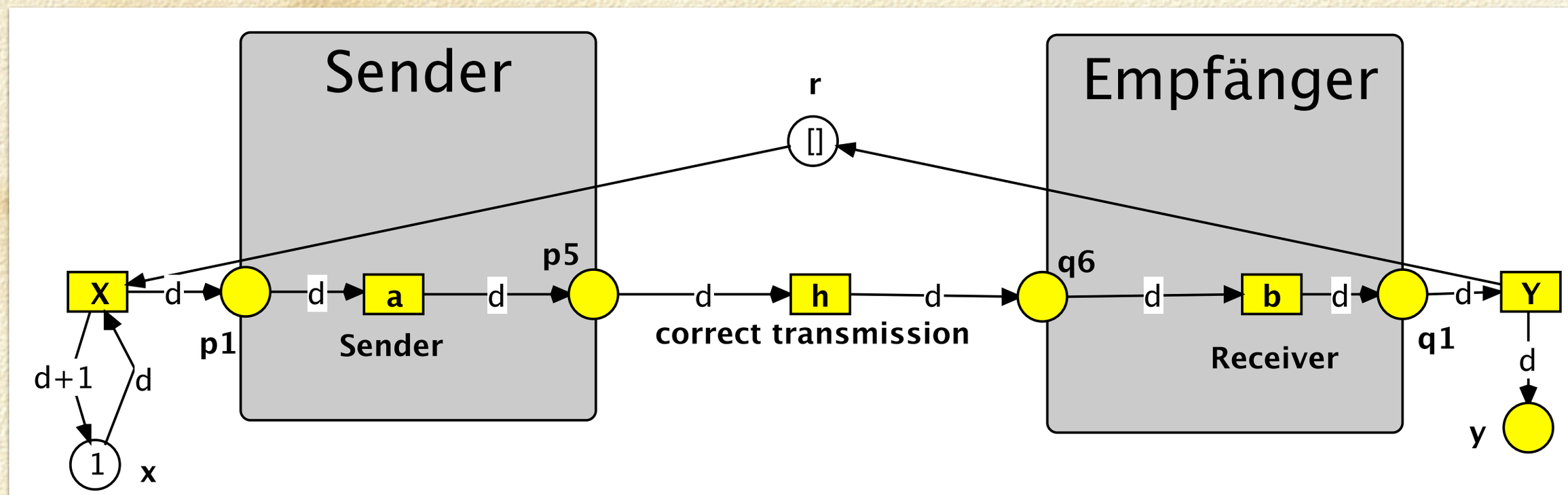


Abbildung 7.21 Spezifikation abp-1 des Alternierbitprotokolls

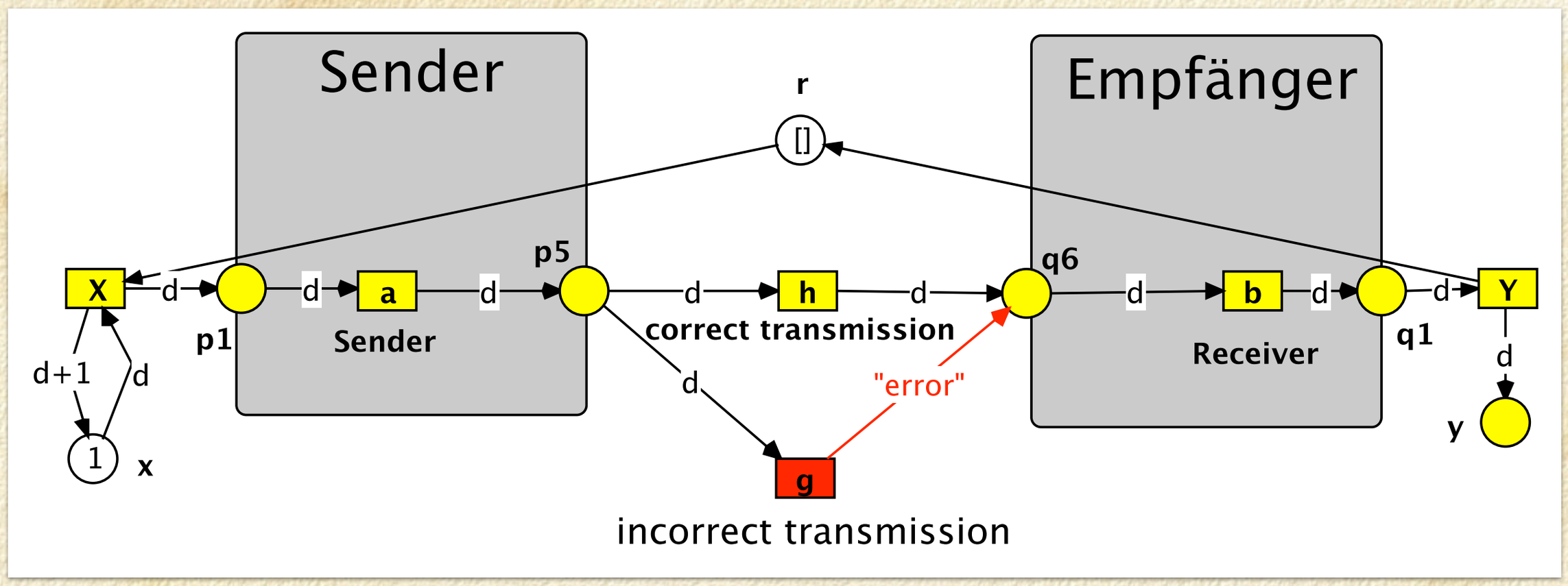


Abbildung 7.22 Übermittlung mit Fehlern: Netz abp-2

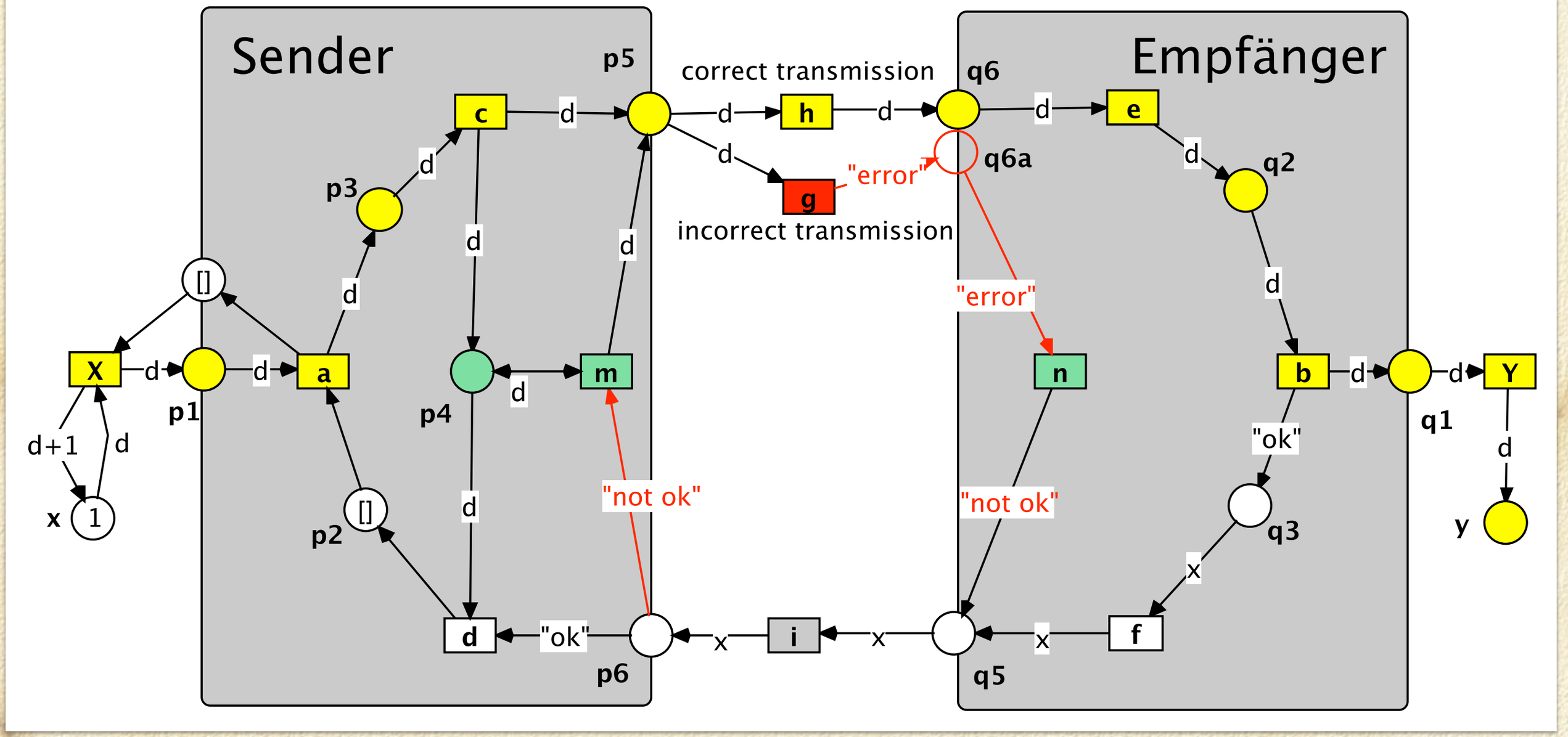


Abbildung 7.23 Übermittlung mit Quittung: Netz abp-3

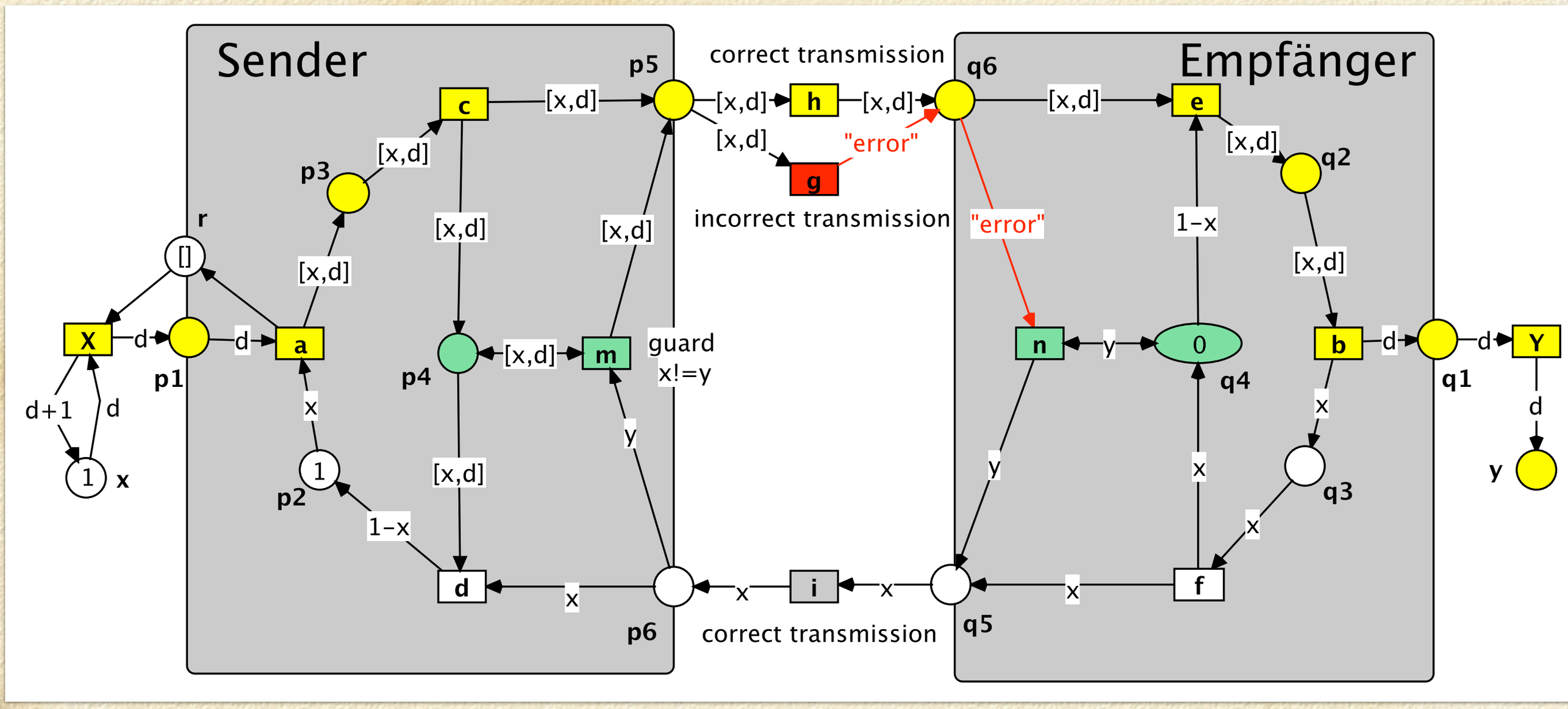


Abbildung 7.24 Übermittlung mit Quittung durch Alternierbit: Netz abp-4

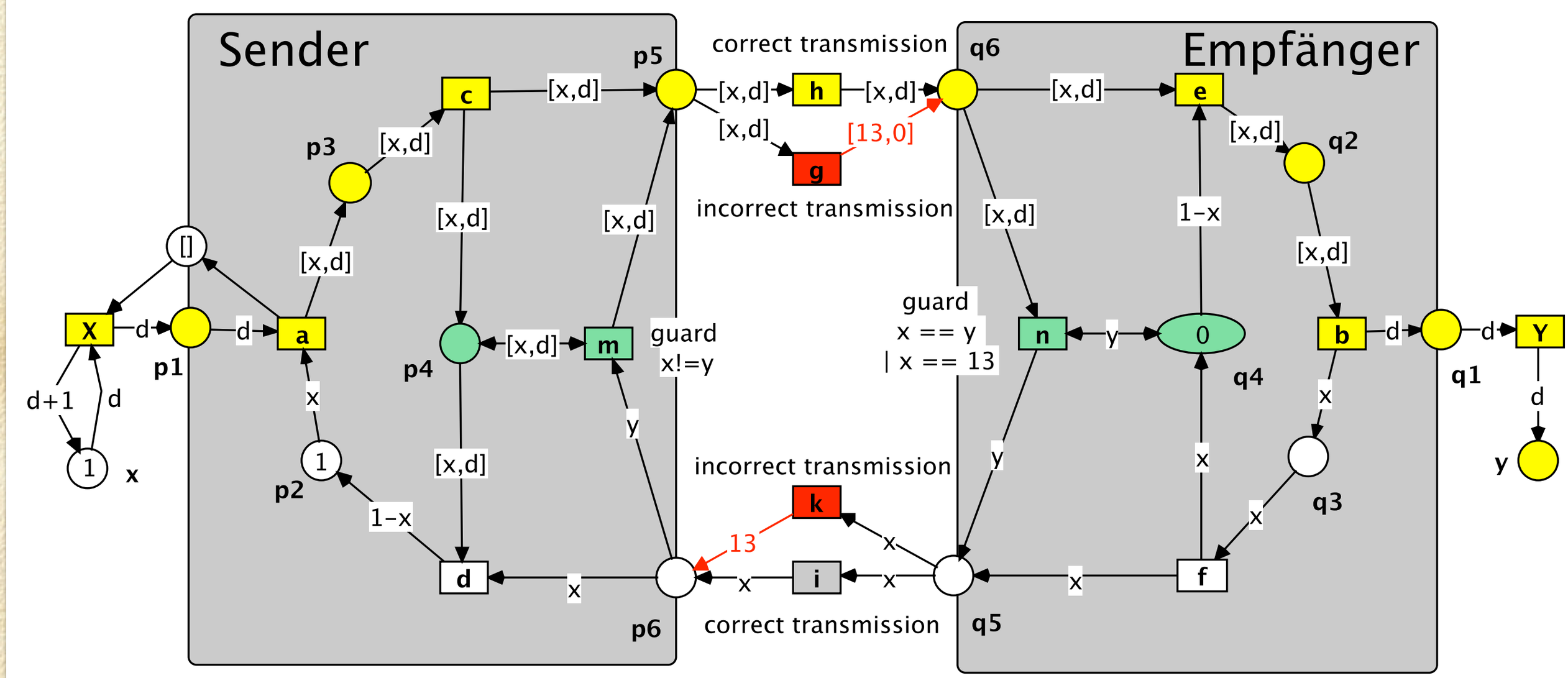
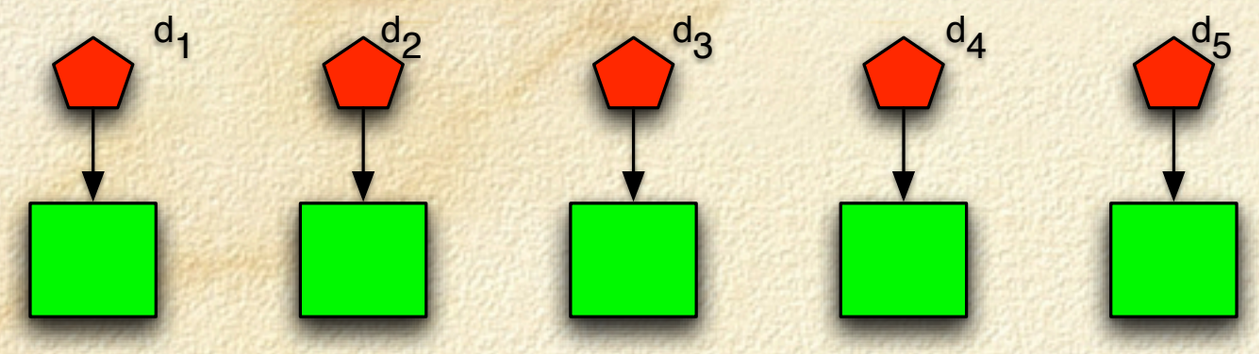
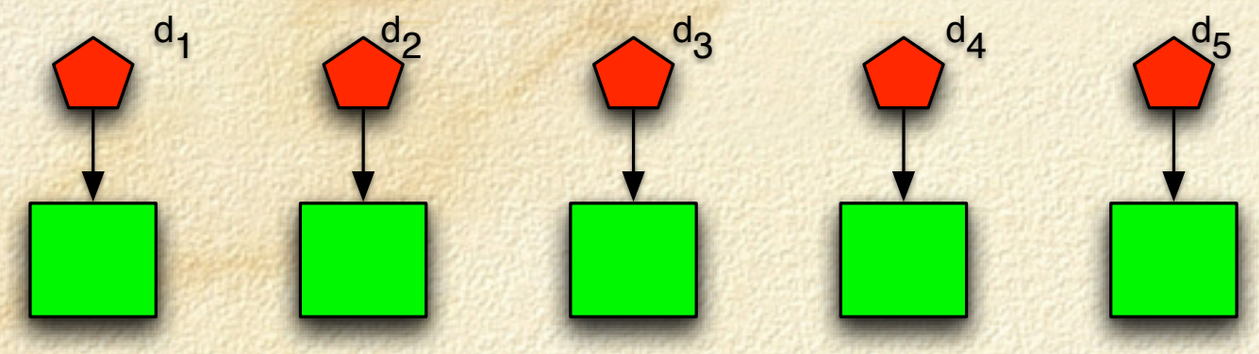


Abbildung 7.25 Das Alternierbitprotokoll abp-5



Das Beispiel der Datenbank-Manager[5]

Eine Menge von $n > 0$ Datenbanken soll von n Prozessen, genannt *Datenbank-Manager*, $DBM = \{d_1, d_2, \ldots, d_n\}$ so verwaltet werden, dass sie immer den gleichen Inhalt haben. Um dies zu erreichen, sollen die Manager miteinander kommunizieren. Jeder Manager kann seine eigene Datenbank aktualisieren. Dabei muß er an jeden anderen Manager eine Nachricht senden, die diesen über die Aktualisierung informiert. Der Manager muß warten, bis alle anderen diese Nachricht erhalten, die Aktualisierung durchgeführt und eine entsprechende Rückmeldung zurückgesandt haben. Erst dann kehrt der Manager in den Zustand *inactive* zurück.



Dabei sollen weder die Datenbanken noch ihre Aktualisierung dargestellt werden, sondern nur der Nachrichtenaustausch.

Zustände der Manager : $inactive$, $waiting$, $performing$

Nachrichten : $MS = \{(s,r) | s,r \in DBM \wedge s \neq r\}$

Zustände der Nachrichtenpuffer : $unused$, $sent$, $received$, $acknowledged$

wechselseitiger Ausschluß : $exclusion$

Spezifikation für das gefärbte Netz von Abb. 4.22:

Farben : $DBM = \{d_1, d_2, \ldots, d_n\}$

$MS = \{(s,r) | s,r \in DBM \wedge s \neq r\}$

$E = \{e\}$

$$Variablen: \; Var = \{e, r, s\}$$

$$dom(e) = E, \quad dom(r) = dom(s) = DBM$$

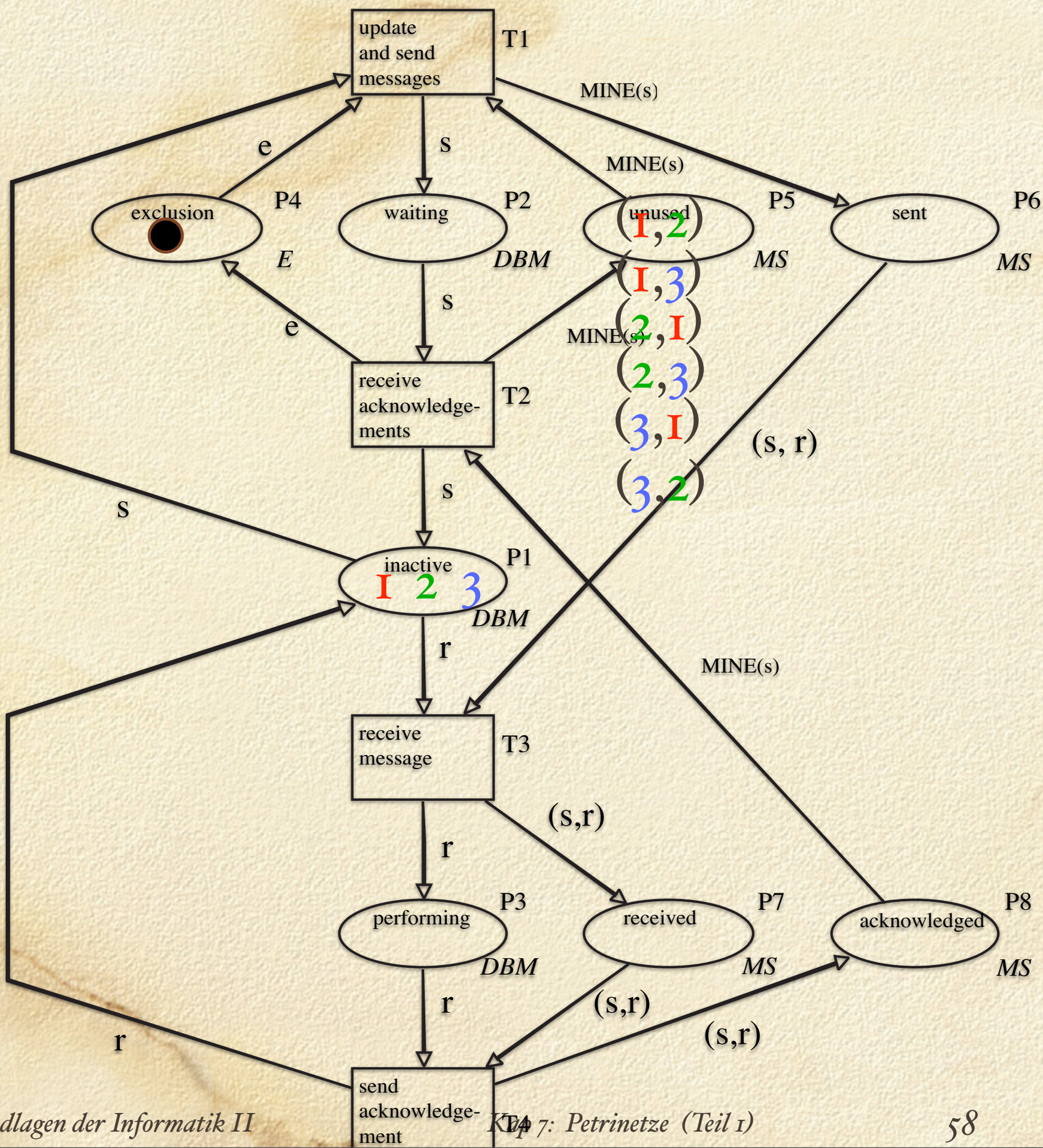
$$Funktionen:$$

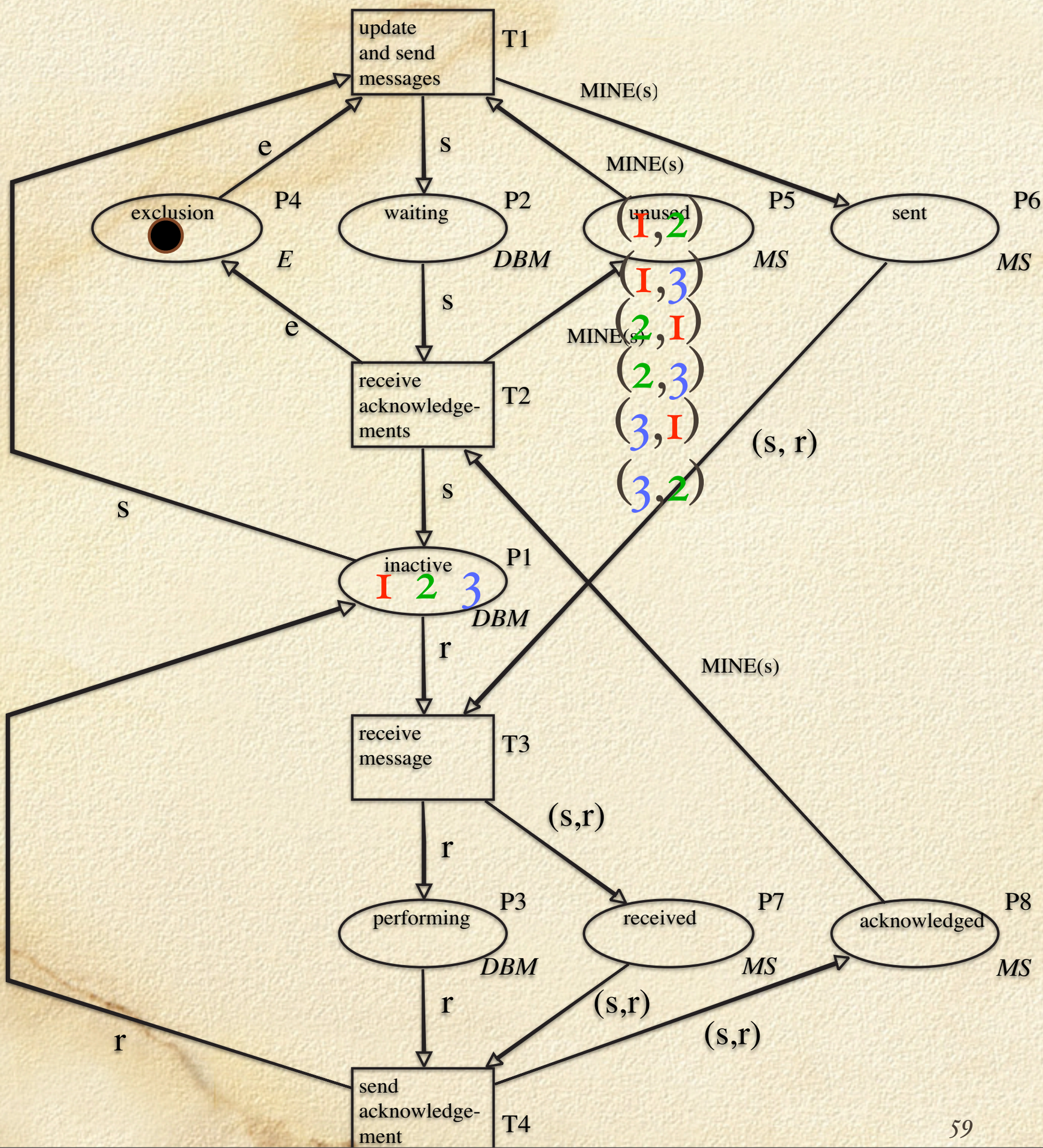
$$MINE : DBM \to Bag(MS) \qquad MINE(s) := \sum_{r \neq s} (s, r)$$

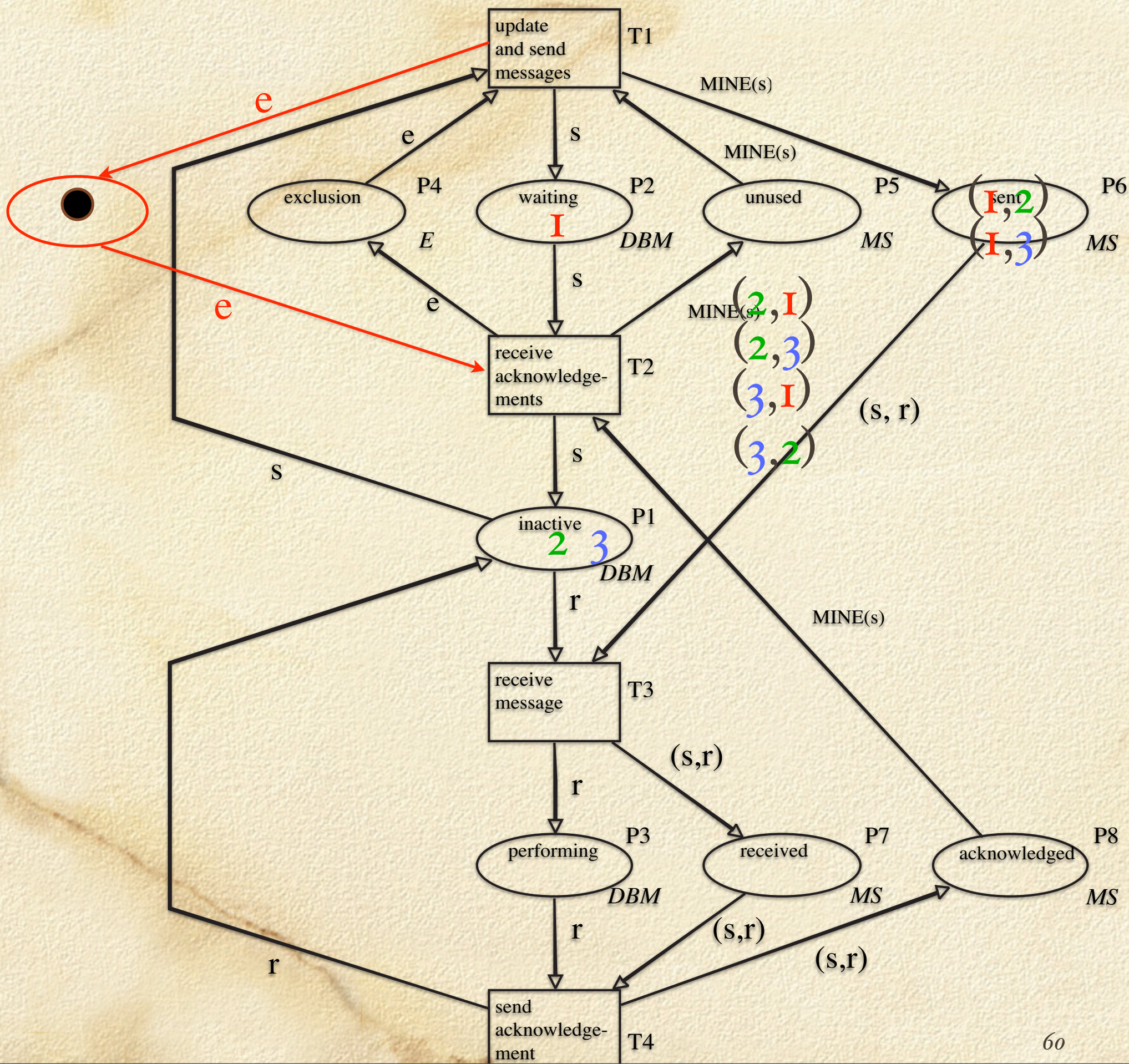
$$REC : MS \to DBM \qquad \boxed{REC((s, r)) := r}$$

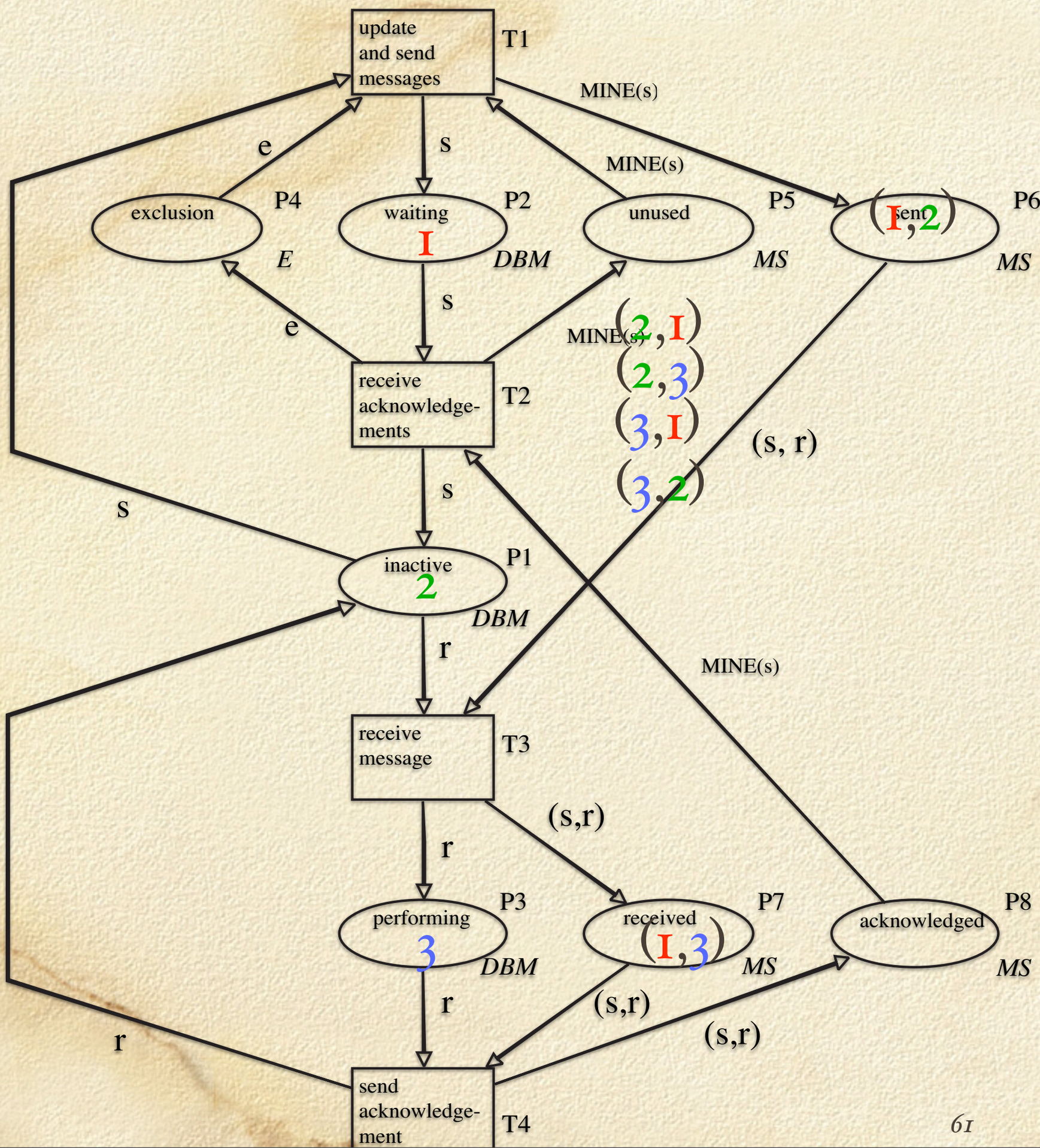
$$ABS : DBM \to E \qquad ABS(s) := e$$

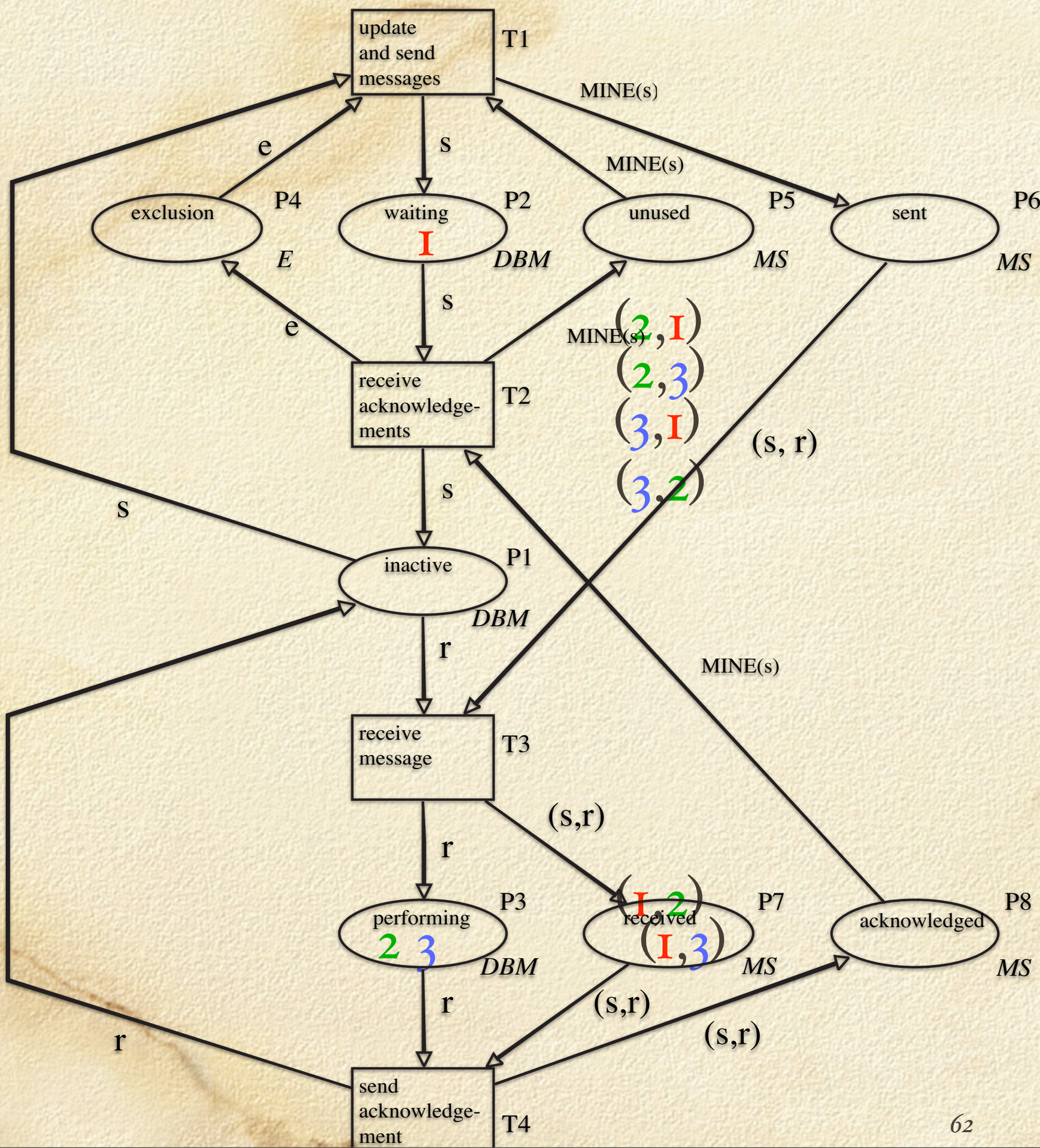
$$Anfangsmarkierung: \mathbf{m}_0(p) := \begin{cases} DBM & \text{falls } p = inactive \\ MS & \text{falls } p = unused \\ \{e\} & \text{falls } p = exclusion \\ \emptyset & \text{sonst} \end{cases}$$

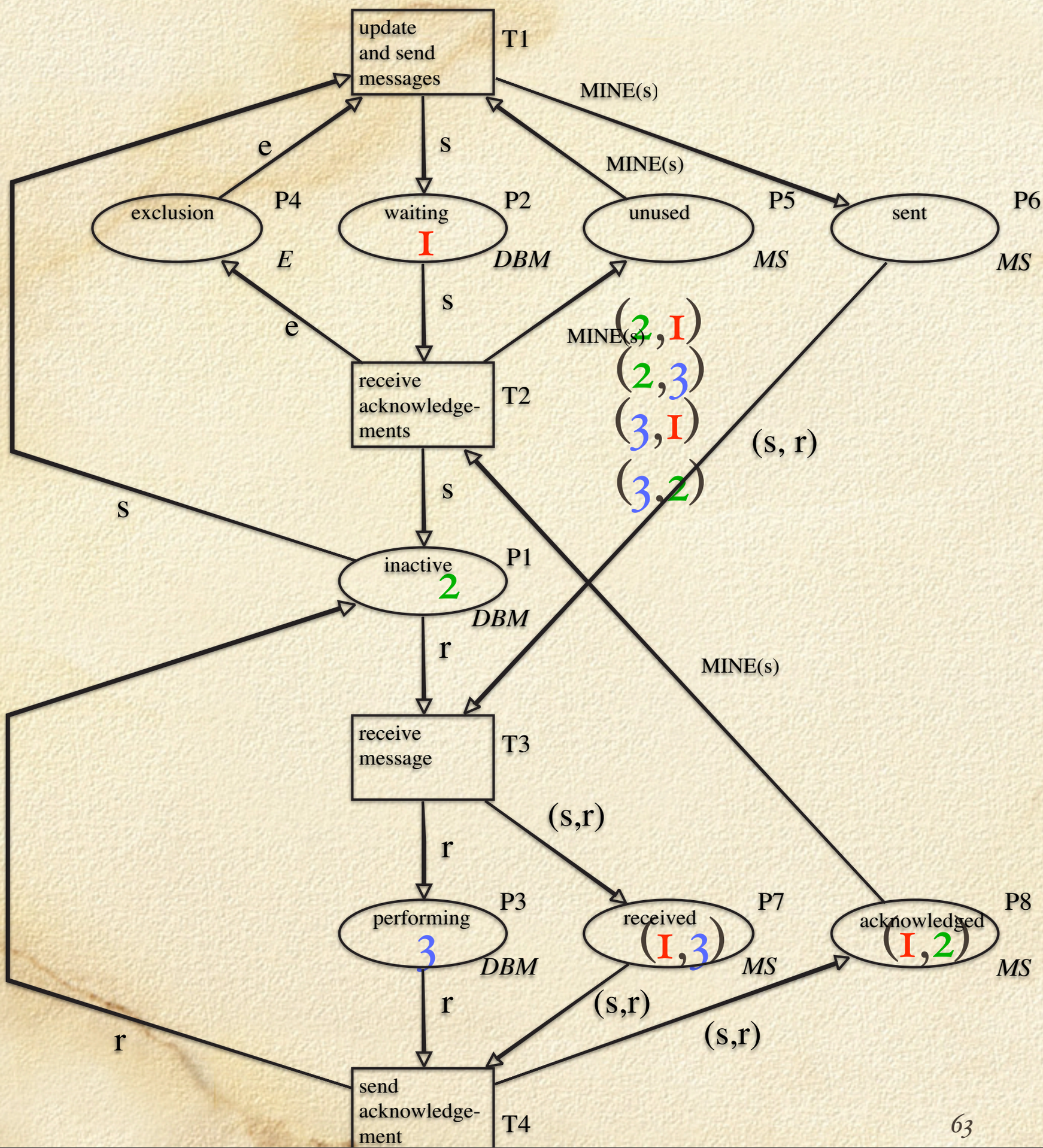

update
and send
messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
E
DBM
MS
MS
s
e
MINE(s)
receive
acknowledge-
ments
T2
(s, r)
s
s
inactive
P1
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
P3
performing
DBM
P7
received
MS
P8
acknowledged
MS
r
(s,r)
(s,r)
r
send
acknowledge-
ment
T4

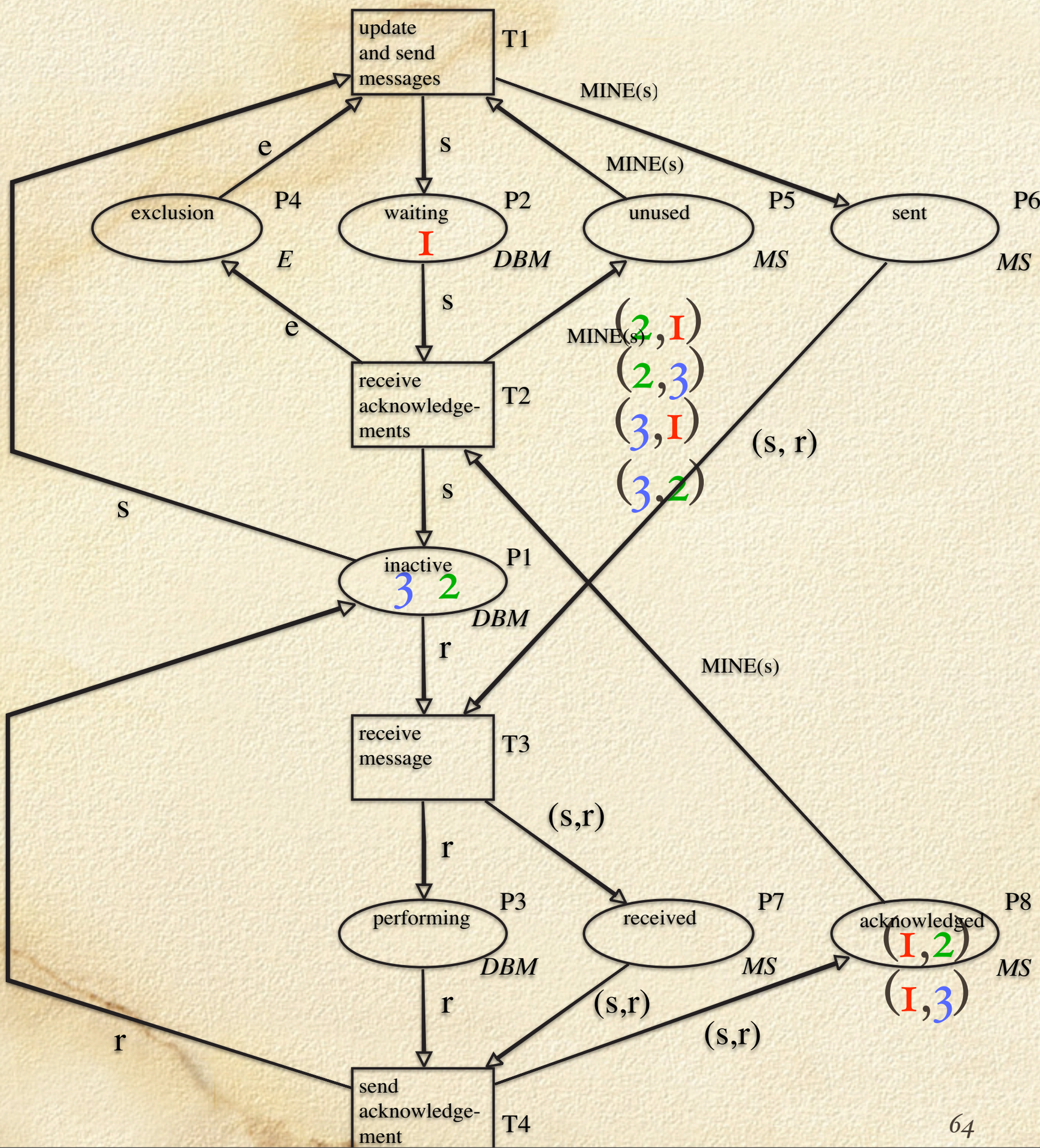

update
and send
messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
E
DBM
MS
MS
s
e
MINE(s)
receive
acknowledge-
ments
T2
(s, r)
s
s
inactive
P1
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
P3
P7
P8
performing
received
acknowledged
DBM
MS
MS
r
(s,r)
(s,r)
r
send
acknowledge-
ment
T4


update
and send
messages
T1
e
e
s
MINE(s)
MINE(s)
exclusion
P4
E
waiting
I
P2
DBM
unused
P5
MS
sent
I,2
I,3
P6
MS
e
e
s
MINE(s)
2,I
2,3
3,I
3,2
receive
acknowledge-
ments
T2
s
(s, r)
s
inactive
2 3
P1
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
performing
P3
DBM
received
P7
MS
acknowledged
P8
MS
r
(s,r)
(s,r)
send
acknowledge-
ment
T4

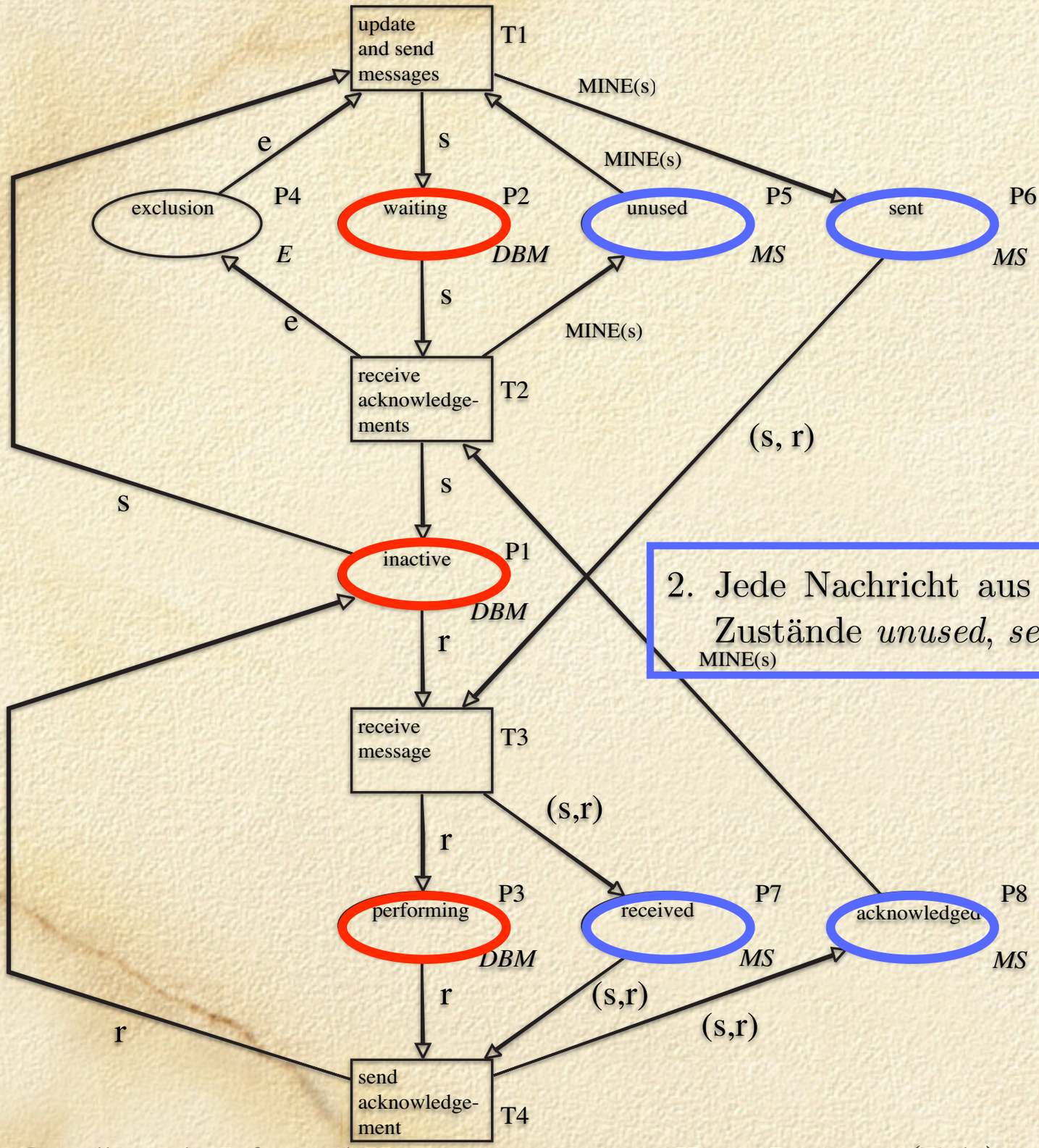

update and send messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
I
2
E
DBM
MS
MS
s
e
MINE(s)
receive acknowledge-ments
T2
(2,I)
(2,3)
(3,I)
(3,2)
(s, r)
s
s
inactive
P1
2
DBM
r
MINE(s)
receive message
T3
(s,r)
r
performing
P3
received
P7
acknowledged
P8
3
(I,3)
DBM
MS
MS
r
(s,r)
(s,r)
r
send acknowledge-ment
T4


update
and send
messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
I
E
DBM
MS
MS
s
e
MINE(s)
(2,I)
(2,3)
(3,I)
(3,2)
receive
acknowledge-
ments
T2
s
(s, r)
s
inactive
P1
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
P3
performing
2 3
DBM
(I,2)
(I,3)
P7
received
MS
P8
acknowledged
MS
r
(s,r)
(s,r)
send
acknowledge-
ment
T4


update
and send
messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
I
E
DBM
MS
MS
s
e
MINE(s)
(2,I)
(2,3)
(3,I)
(3,2)
receive
acknowledge-
ments
T2
(s, r)
s
s
inactive
P1
2
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
P3
P7
P8
performing
3
DBM
received
(I,3)
MS
acknowledged
(I,2)
MS
r
(s,r)
(s,r)
send
acknowledge-
ment
T4

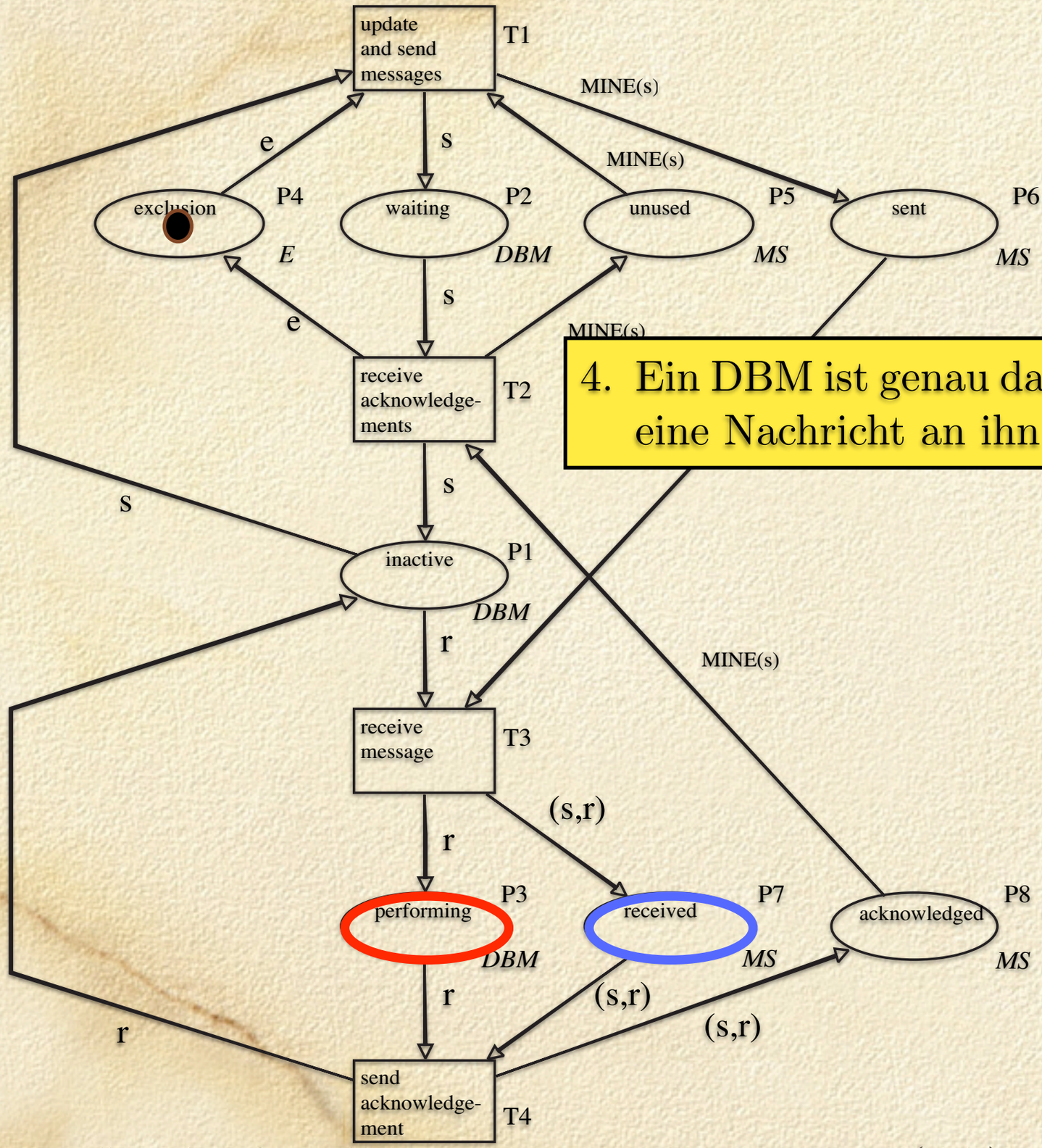

update and send messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
I
E
DBM
MS
MS
s
e
MINE(s)
(2,I)
(2,3)
(3,I)
(3,2)
receive acknowledge-ments
T2
(s, r)
s
s
inactive
P1
3 2
DBM
r
MINE(s)
receive message
T3
(s,r)
r
P3
performing
DBM
P7
received
MS
P8
acknowledged
(I,2)
(I,3)
MS
r
(s,r)
(s,r)
send acknowledge-ment
T4

1. Jeder DBM ist in genau einem der drei Zustände *inactive*, *waiting* oder *performing*.

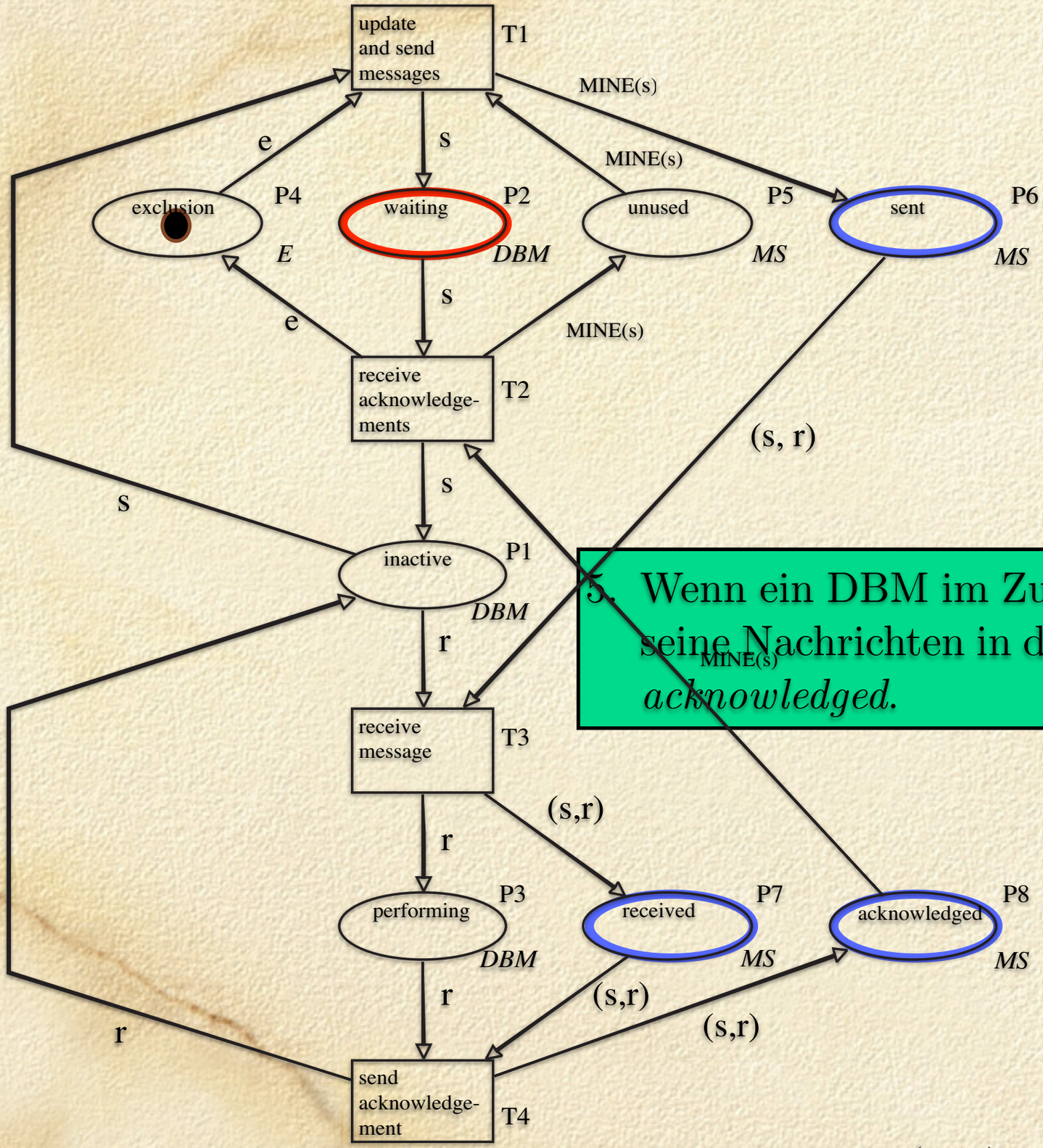


2. Jede Nachricht aus MS ist in genau einem der vier Zustände *unused*, *sent*, *received* oder *acknowledged*.

3. Höchstens ein DBM wartet.



4. Ein DBM ist genau dann im Zustand *performing*, wenn eine Nachricht an ihn im Zustand *received* ist.



5. Wenn ein DBM im Zustand *waiting* ist, dann sind alle seine Nachrichten in den Zuständen *sent*, *received* oder *acknowledged*.

Satz 4.18 *Für alle erreichbaren Markierungen* $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ *des DBM-Netzes* $\mathcal{N}$ *von Abbildung 4.21 gilt:*

1. Jeder DBM ist in genau einem der drei Zustände *inactive*, *waiting* oder *performing*.

$$1.\ \mathbf{m}(inactive)+\mathbf{m}(waiting)+\mathbf{m}(performing) = DBM$$

Multimengen-vereinigung

2. Jede Nachricht aus MS ist in genau einem der vier Zustände *unused*, *sent*, *received* oder *acknowledged*.

$$2.\ \mathbf{m}(unused) + \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged) = MS$$

3. Höchstens ein DBM wartet.

$$3.\ ABS(\mathbf{m}(waiting)) + \mathbf{m}(exclusion) = \{e\}$$

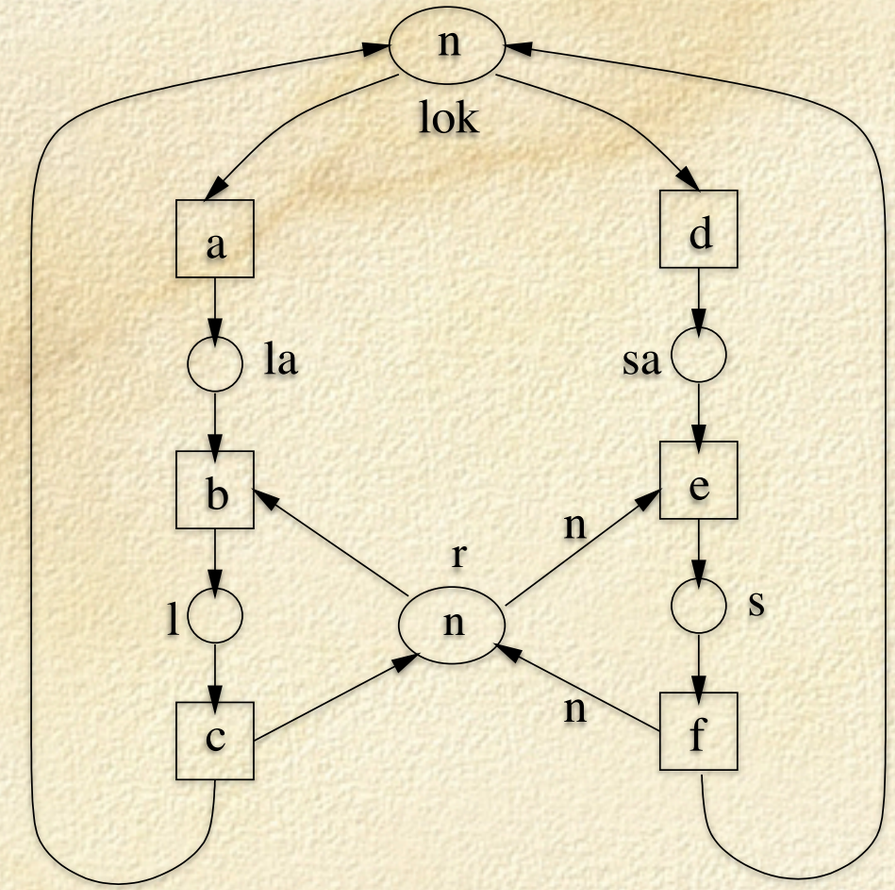
Satz 4.18 *Für alle erreichbaren Markierungen* $\mathbf{m} \in \mathbf{R}(\mathcal{N})$ *des DBM-Netzes* $\mathcal{N}$ *von Abbildung 4.21 gilt:*

4. Ein DBM ist genau dann im Zustand *performing*, wenn eine Nachricht an ihn im Zustand *received* ist.

$$4.\ \mathbf{m}(performing) = REC(\mathbf{m}(received))$$

5. Wenn ein DBM im Zustand *waiting* ist, dann sind alle seine Nachrichten in den Zuständen *sent*, *received* oder *acknowledged*.

$$5.\ MINE(\mathbf{m}(waiting)) = \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged)$$

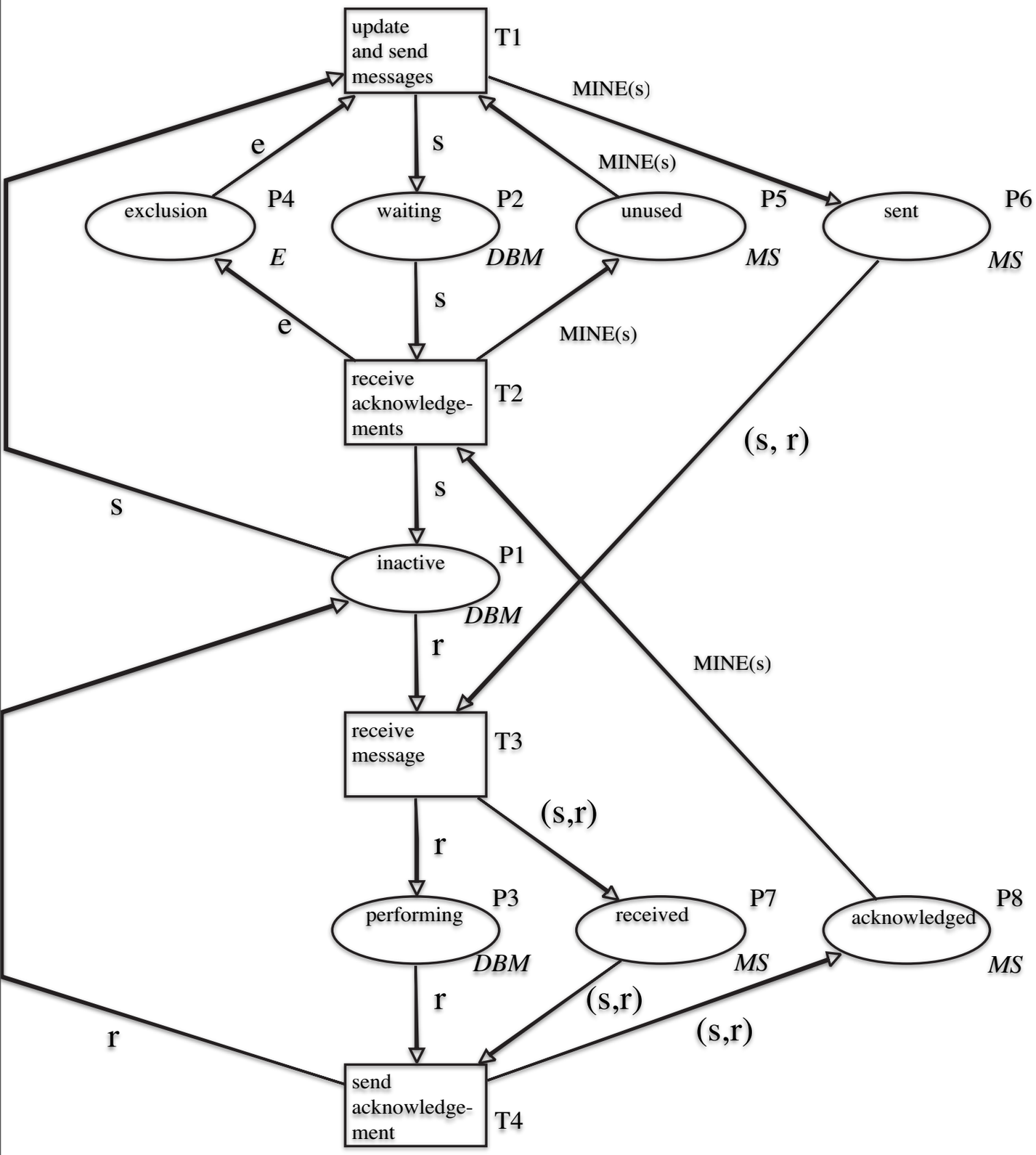


- $i_1: \quad lok + la + sa + l + s = n$
- $i_2: \quad l + r + n \cdot s = n$

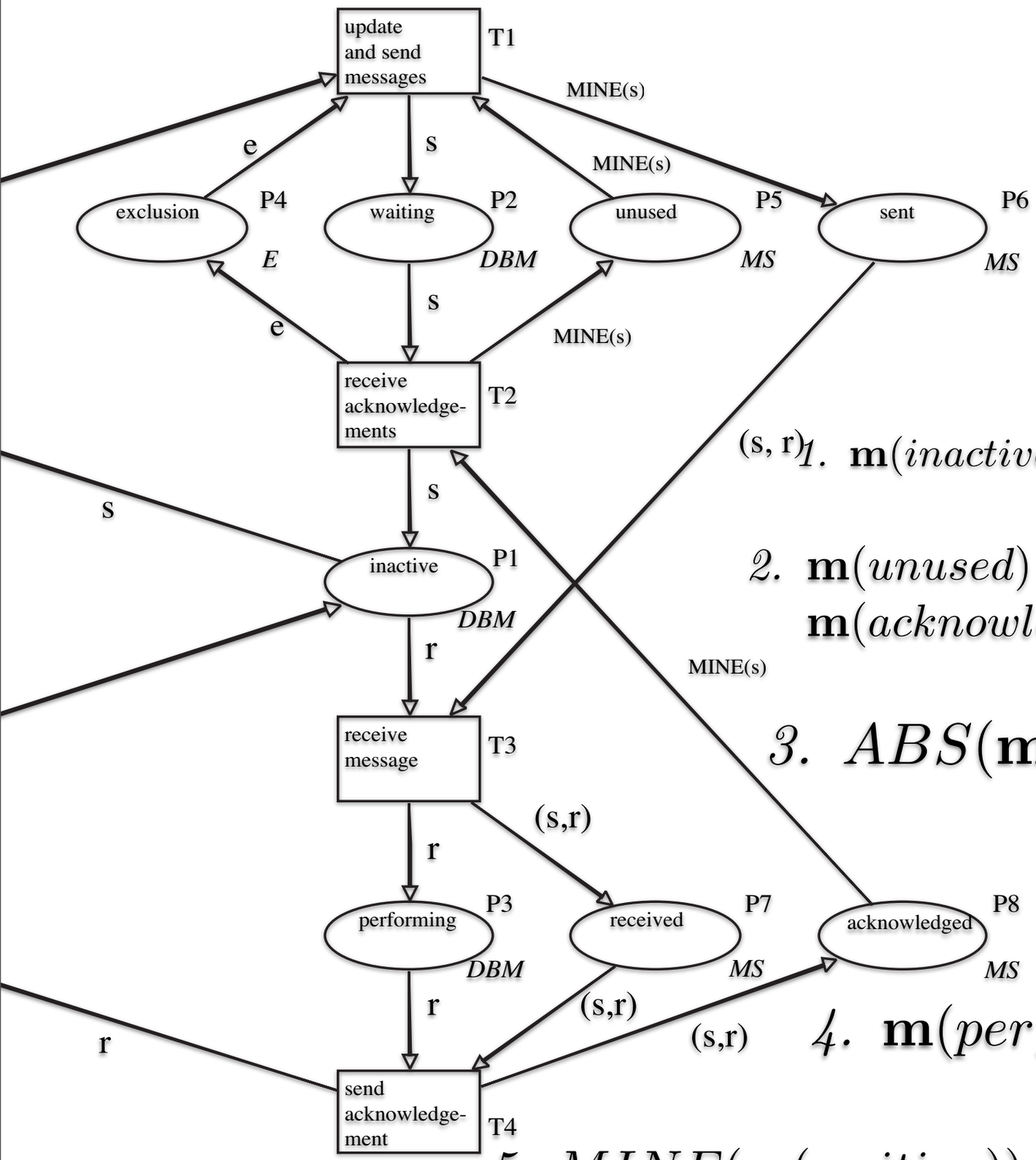
  ($lok, la, \ldots$ stehen hier abkürzend für $\mathbf{m}(lok), \mathbf{m}(la), \ldots$)

Abbildung 4.1: Leser/Schreiber-Problem für n Aufträge

$\mathcal{N}$	a	b	c	d	e	f		i_1		i_2
lok	-1	0	1	-1	0	1		1		0
la	1	-1	0	0	0	0		1		0
sa	0	0	0	1	-1	0		1		0
l	0	1	-1	0	0	0		1		1
s	0	0	0	0	1	-1		1		n
r	0	-1	1	0	$-n$	n		0		1



DATABASE SYSTEM		T1	T2	T3	T4	m_0
		DBM	DBM	MS	MS	
inactive	DBM	-ID	ID	-REC	REC	Σ DBM
waiting	DBM	ID	-ID			
performing	DBM			REC	-REC	
exclusion	E	-ABS	ABS			E
unused	MS	-MINE	MINE			ΣMS
sent	MS	MINE		-ID		
received	MS			ID	-ID	
acknowledged	MS		-MINE		ID	



1. $\mathbf{m}(inactive)+\mathbf{m}(waiting)+\mathbf{m}(performing) = DBM$

2. $\mathbf{m}(unused) + \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged) = MB$

3. $ABS(\mathbf{m}(waiting)) + \mathbf{m}(exclusion) = \{e$

4. $\mathbf{m}(performing) = REC(\mathbf{m}(received))$

5. $MINE(\mathbf{m}(waiting)) = \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged)$

$$1.\ \mathbf{m}(inactive)+\mathbf{m}(waiting)+\mathbf{m}(performing)=DBM$$

DATABASE SYSTEM		T1	T2	T3	T4	$\mathbf{m}_0$	w1
		DBM	DBM	MB	MB		DBM
inactive	DBM	-ID	ID	-REC	REC	Σ DBM	ID
waiting	DBM	ID	-ID				ID
performing	DBM			REC	-REC		ID
exclusion	E	-ABS	ABS			e	
unused	MB	-MINE	MINE			ΣMB	
sent	MB	MINE		-ID			
received	MB			ID	-ID		
acknowledged	MB		-MINE		ID		

DATABASE SYSTEM						*Invariants*					
		T1	T2	T3	T4	$\mathbf{m}_0$	w1	w2	w3	w4	w5
		DBM	DBM	MB	MB		DBM	MB	E	DBM	MB
inactive	DBM	-ID	ID	-REC	REC	Σ DBM	ID				
waiting	DBM	ID	-ID				ID		ABS		MINE
performing	DBM			REC	-REC		ID			ID	
exclusion	E	-ABS	ABS			e			ID		
unused	MB	-MINE	MINE			ΣMB		ID			
sent	MB	MINE		-ID				ID			-ID
received	MB			ID	-ID			ID		-REC	-ID
acknowledged	MB		-MINE		ID			ID			-ID

2. $\mathbf{m}(unused) + \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged) = MB$

3. $ABS(\mathbf{m}(waiting)) + \mathbf{m}(exclusion) = \{e\}$

4. $\mathbf{m}(performing) = REC(\mathbf{m}(received))$

5. $MINE(\mathbf{m}(waiting)) = \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged)$

DATABASE SYSTEM		T1	T2	T3	T4	$\mathbf{m}_0$	*Invariants* w1	w2	w3	w4	w5
		DBM	DBM	MB	MB		DBM	MB	E	DBM	MB
inactive	DBM	-ID	ID	-REC	REC	Σ DBM	ID				
waiting	DBM	ID	-ID				ID		ABS		MINE
performing	DBM			REC	-REC		ID			ID	
exclusion	E	-ABS	ABS			e			ID		
unused	MB	-MINE	MINE			ΣMB		ID			
sent	MB	MINE		-ID				ID			-ID
received	MB			ID	-ID			ID		-REC	-ID
acknowledged	MB		-MINE		ID			ID			-ID

2. $\mathbf{m}(unused) + \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged) = MB$

3. $ABS(\mathbf{m}(waiting)) + \mathbf{m}(exclusion) = \{e\}$

4. $\mathbf{m}(performing) = REC(\mathbf{m}(received))$

5. $MINE(\mathbf{m}(waiting)) = \mathbf{m}(sent) + \mathbf{m}(received) + \mathbf{m}(acknowledged)$

Garbage Can

"Sozionik" =

Informatik/Multiagentensysteme

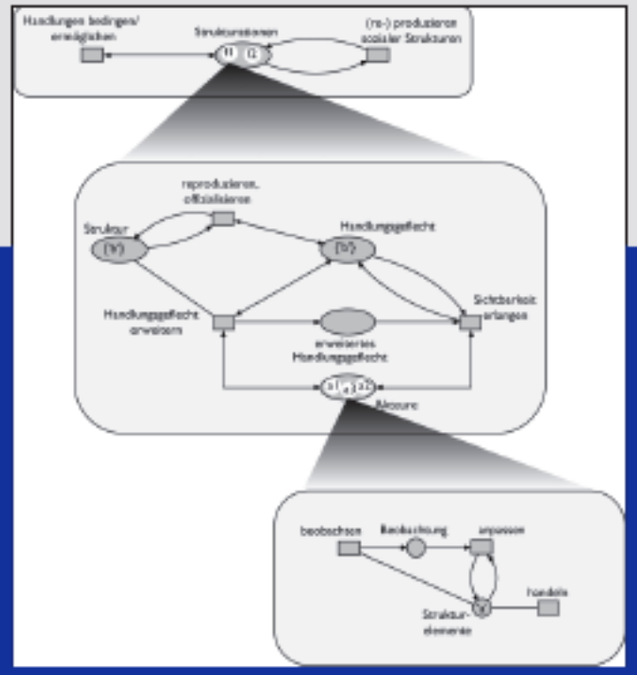
+

Soziologie

Rolf v. Lüde, Daniel Moldt, Rüdiger Valk

Sozionik – Modellierung soziologischer Theorie

Unter Mitarbeit von Michael Köhler, Roman Langer, Heiko Rölke, Daniela Spresny



WIRTSCHAFT – ARBEIT – TECHNIK

PERSPEKTIVEN GESELLSCHAFTLICHEN WANDELS
HG. ROLF v. LÜDE

LIT

M.D. Cohen J.G. March and J.P. Olsen.
A garbage can model of organizational choice.
Administrative Science Quarterly 17:1--25, 1972.

Consider organized anarchies. These are organizations - or decision situations - characterized by three general properties.

unpräzise Prioritäten

The first is **problematic preferences**. In the organization it is difficult to impute a set of preferences to the decision situation that satisfies the standard consistency requirements for a theory of choice.

The organization operates on the basis of a variety of inconsistent and ill defined preferences. It can be described better by a loose collection of ideas than as a coherent structure; it discovers preferences through action more than it acts on the basis of preferences,

unpräzise Methoden

The second property is **unclear technology**. Although the organization manages to survive and even produce, its own processes are not understood by its members. It operates on the basis of simple trial-and-error procedures, the residue of learning from accidents of past experience, and pragmatic inventions of necessity.

wechselnde Partner

The third property is **fluid participation**. Participants vary in the amount of time and effort they devote to different domains; involvement varies from one time to another. As a result, the boundaries of the organization are uncertain and changing; the audiences and decision makers for any particular kind of choice change capriciously.

Probleme

Lösungen

Teilnehmer

Verhaltens-Möglichkeiten

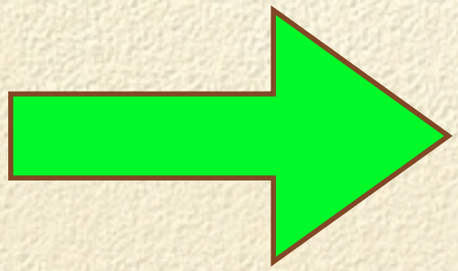
Problems. Problems are the concern of people inside and outside the organization ....

Solutions. A solution is somebody's product. .... Despite the dictum that you cannot find the answer until you have formulated the question well, you often do not know what the question is in organizational problem solving until you know the answer.

Participants. Participants come and go. ..... Substantial variation in participation stems from other demands on the participants' time (rather than from features of the decision under study) .

Choice opportunities. These are occasions when an organization is expected to produce behavior that can be called a decision. Opportunities arise regularly and any organization has ways of declaring an occasion for choice. Contracts must be signed; people hired, promoted, or fired; money spent; and responsibilities allocated.

A stream of choices
A stream of problems
A rate of flow of solutions
A stream of energy from participants



garbage can

M.Masuch and P. LaPotin.
Beyond Garbage Cans:
An AI Model of Organizational Choice.
Administrtive Science Quarterly 36: 38--67, 1989.

To get an approximate impression of the model, reconsider the finale of the James Bond movie, "A View to a Kill." Agent 007 balances on the main cable of the Golden Gate Bridge, a woman in distress clinging to his arm, a blimp approaching for rescue.
In terms of the garbage can model,
the blimp is a solution,
007 a choice opportunity, and
the woman a problem.

In the picture's happy ending, the hero is finally picked up, together with the woman, and a solution by resolution takes place; the problem is solved

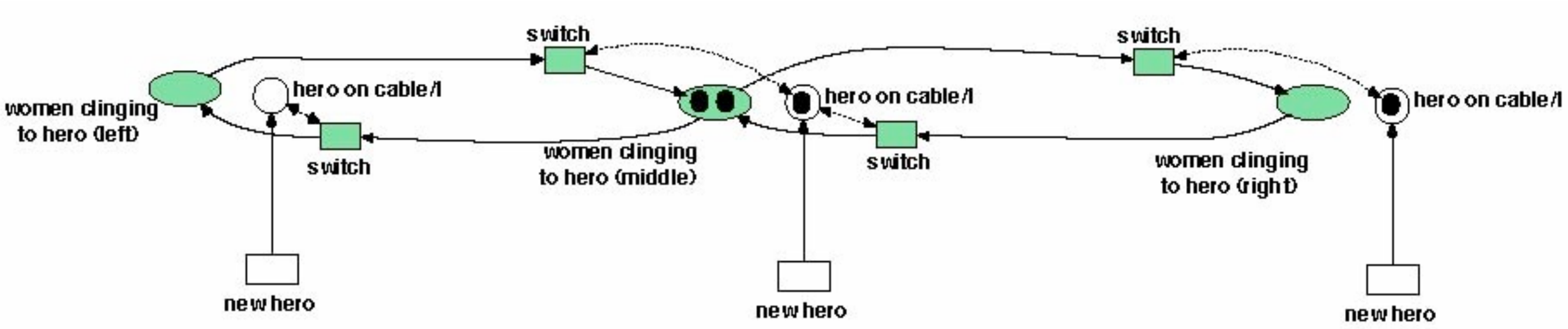
As women, as well as blimps, make their choices simultaneously, but independently of each other, a light hero, on the verge of rescue, may suddenly find himself overburdened. Heavy heroes, in turn, may become rescuable all of a sudden as their women desert them. This mechanism, called "**fluid participation**", creates the possibility of nonsensical solutions or nonsolutions.

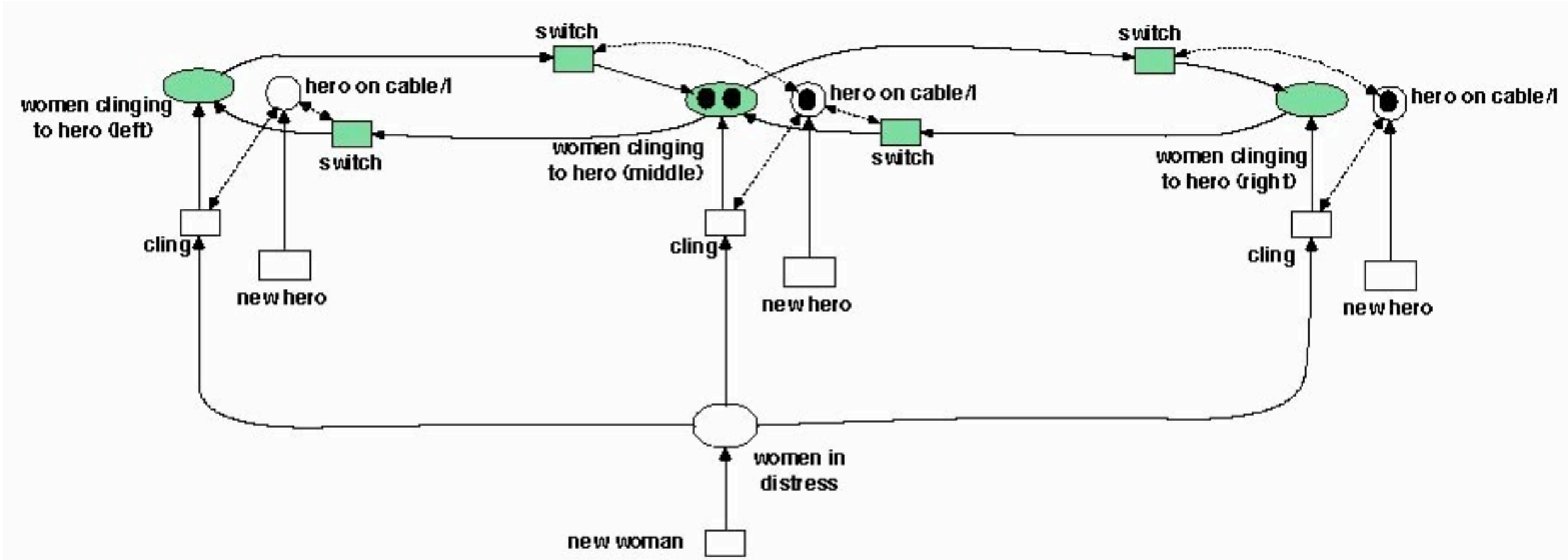
Women may switch heroes too swiftly and end up with an overburdened hero each time; then, problems are not solved. Or heroes are rescued just as all women have left; then, a "**decision by flight**" is made.

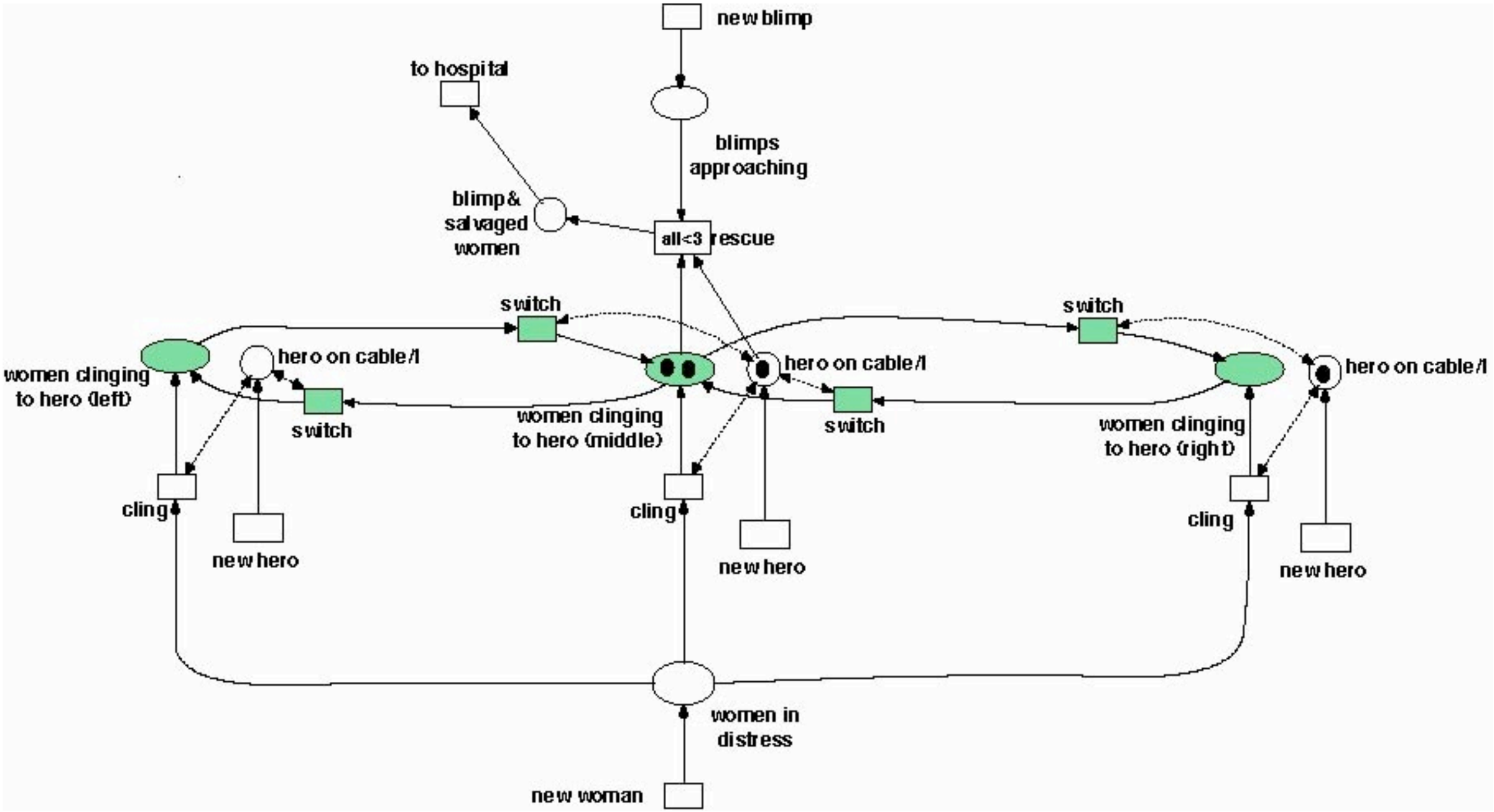
Finally, heroes may be salvaged upon arrival at the scene before any woman in distress has a chance to grab their arm; then "**decisions by oversight**" are said to be made.

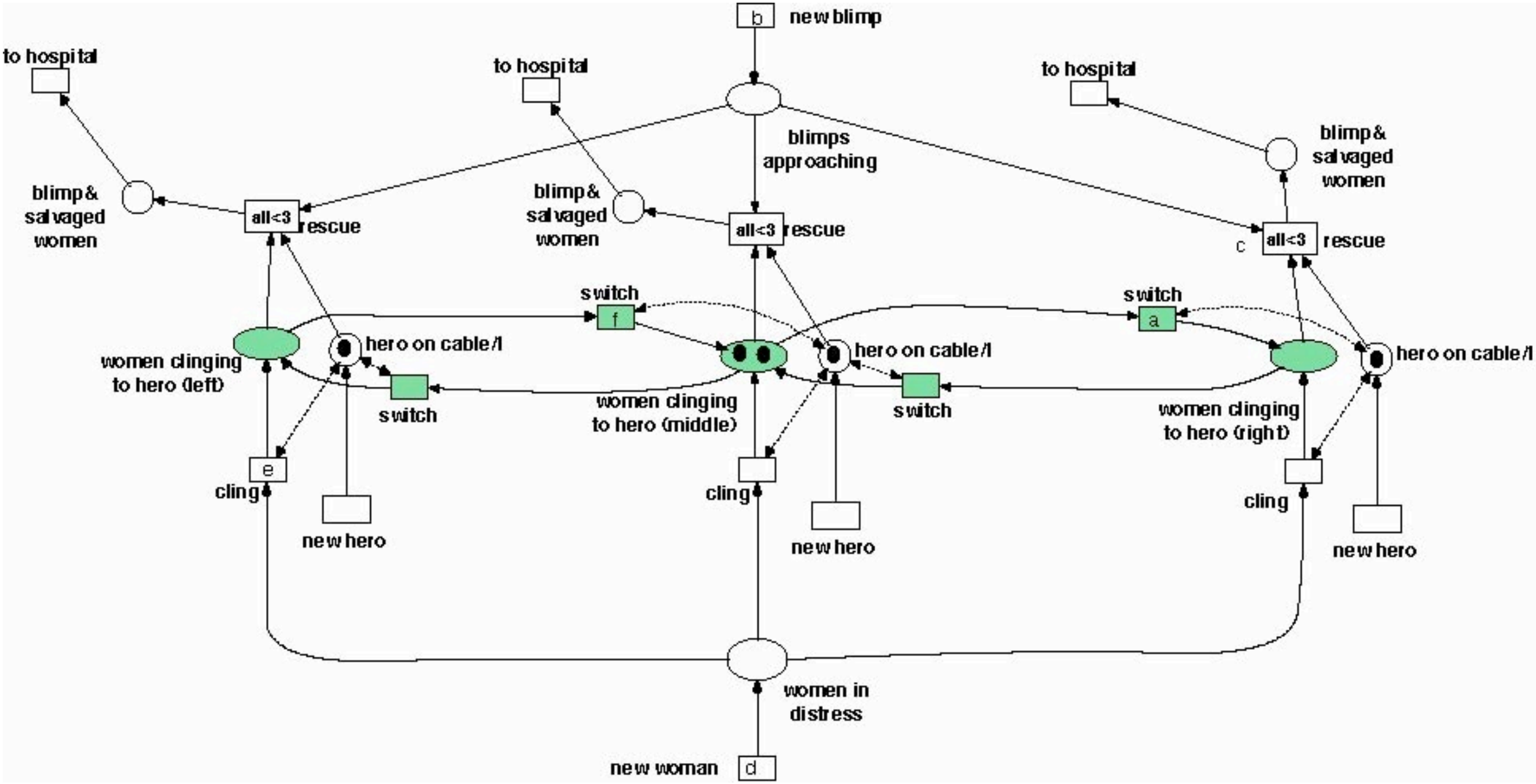
Nevertheless, decisions by resolution do occur.

Entscheidung aus Versehen


switch
switch
hero on cable/1
hero on cable/1
hero on cable/1
women clinging
to hero (left)
switch
switch
women clinging
to hero (middle)
women clinging
to hero (right)
new hero
new hero
new hero


switch
switch
switch
women clinging
to hero (left)
hero on cable/1
switch
women clinging
to hero (middle)
hero on cable/1
switch
women clinging
to hero (right)
hero on cable/1
cling
new hero
cling
new hero
cling
new hero
women in
distress
new woman


new blimp
to hospital
blimps approaching
blimp & salvaged women
all<3
rescue
switch
switch
hero on cable/l
hero on cable/l
hero on cable/l
women clinging to hero (left)
switch
women clinging to hero (middle)
switch
women clinging to hero (right)
cling
cling
cling
new hero
new hero
new hero
women in distress
new woman

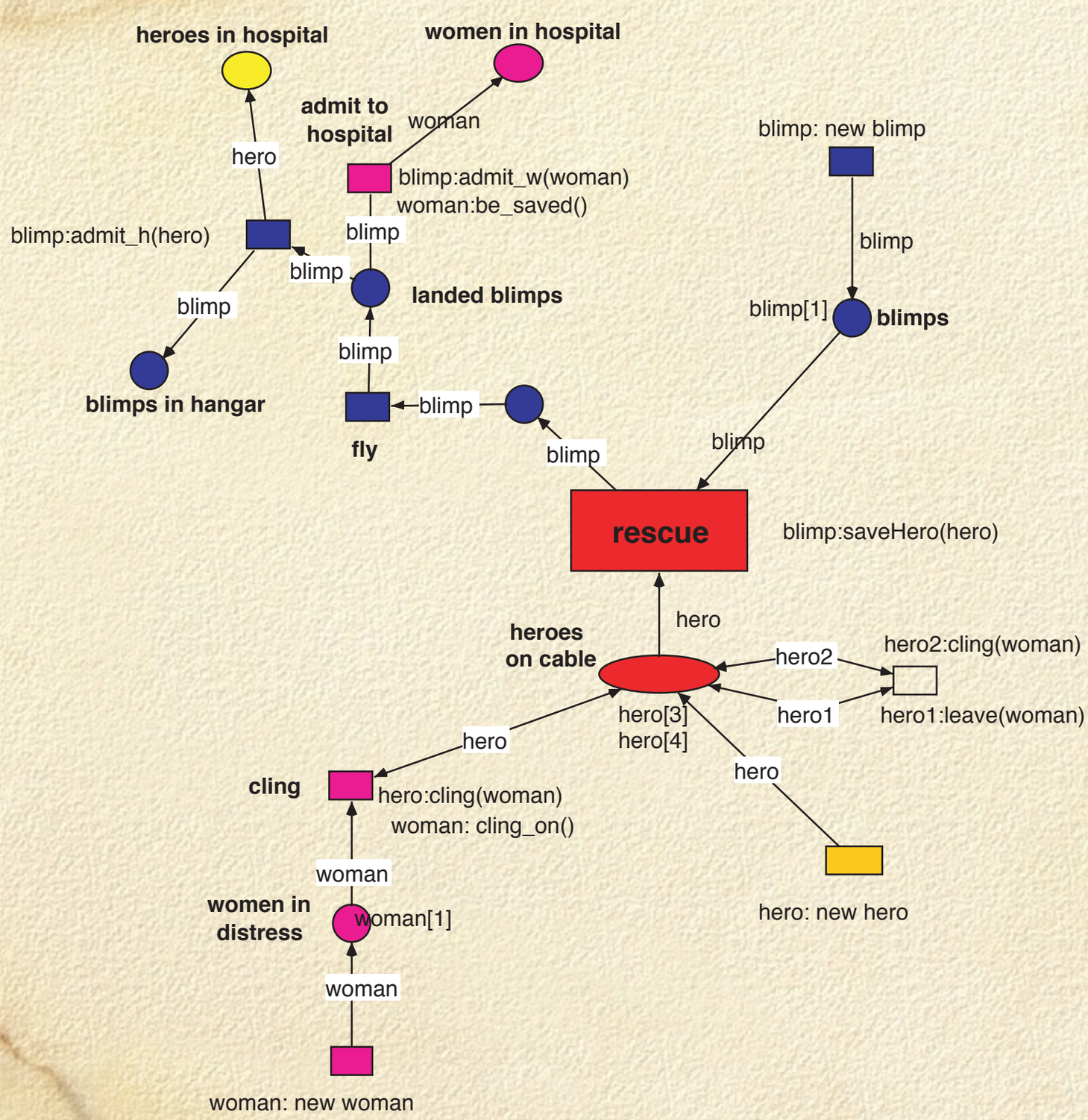

b
new blimp
to hospital
to hospital
to hospital
blimps approaching
blimp & salvaged women
blimp & salvaged women
blimp & salvaged women
all<3
rescue
all<3
rescue
c
all<3
rescue
switch
f
switch
a
hero on cable/1
hero on cable/1
hero on cable/1
women clinging to hero (left)
switch
switch
women clinging to hero (middle)
women clinging to hero (right)
e
cling
cling
cling
new hero
new hero
new hero
women in distress
new woman
d

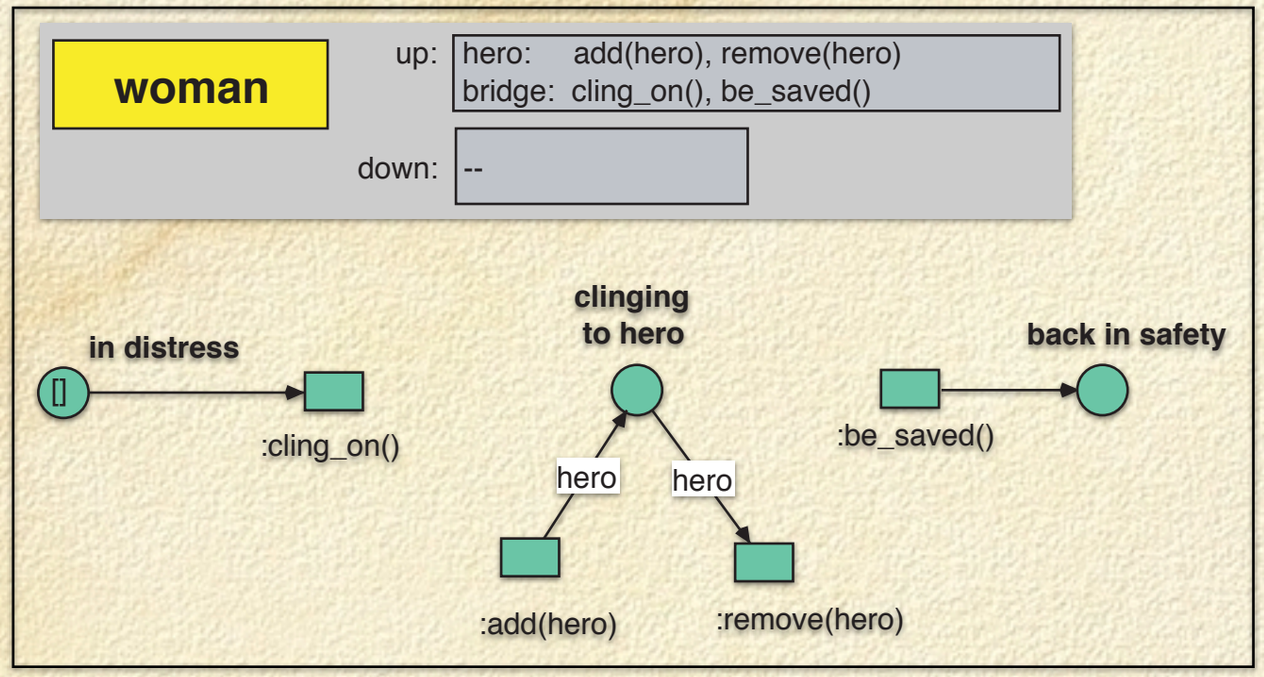
bridge

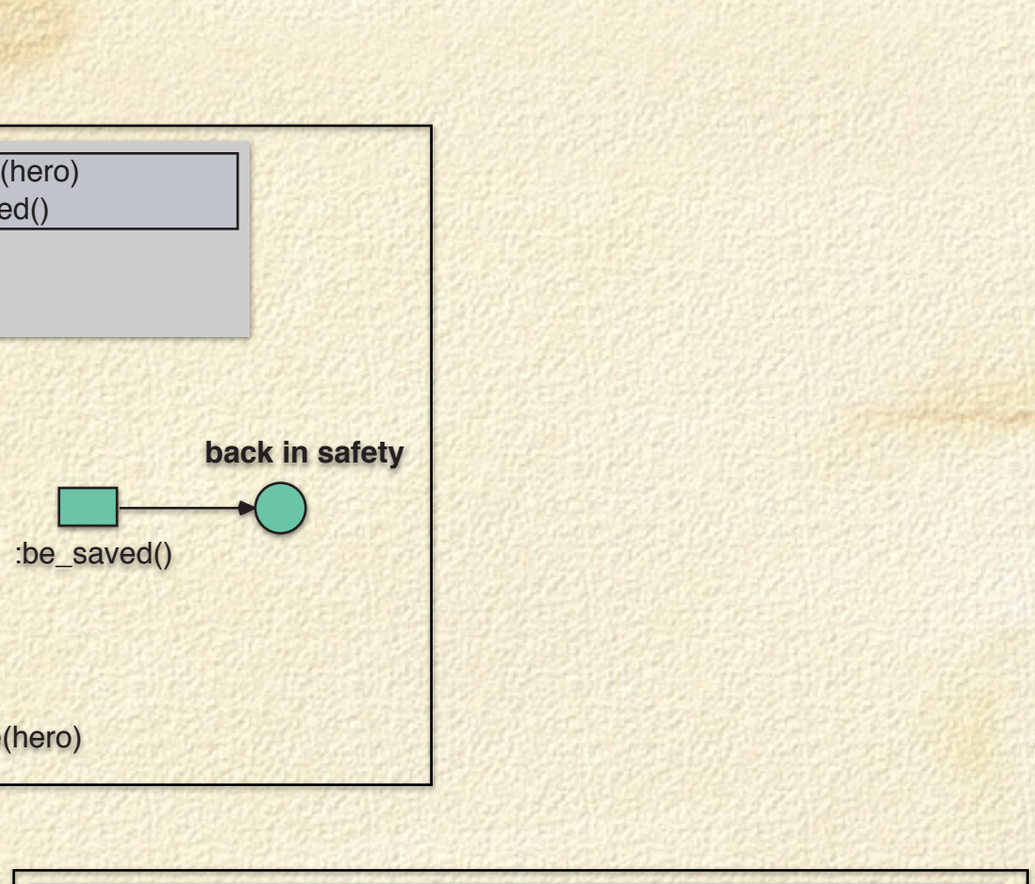
up: --

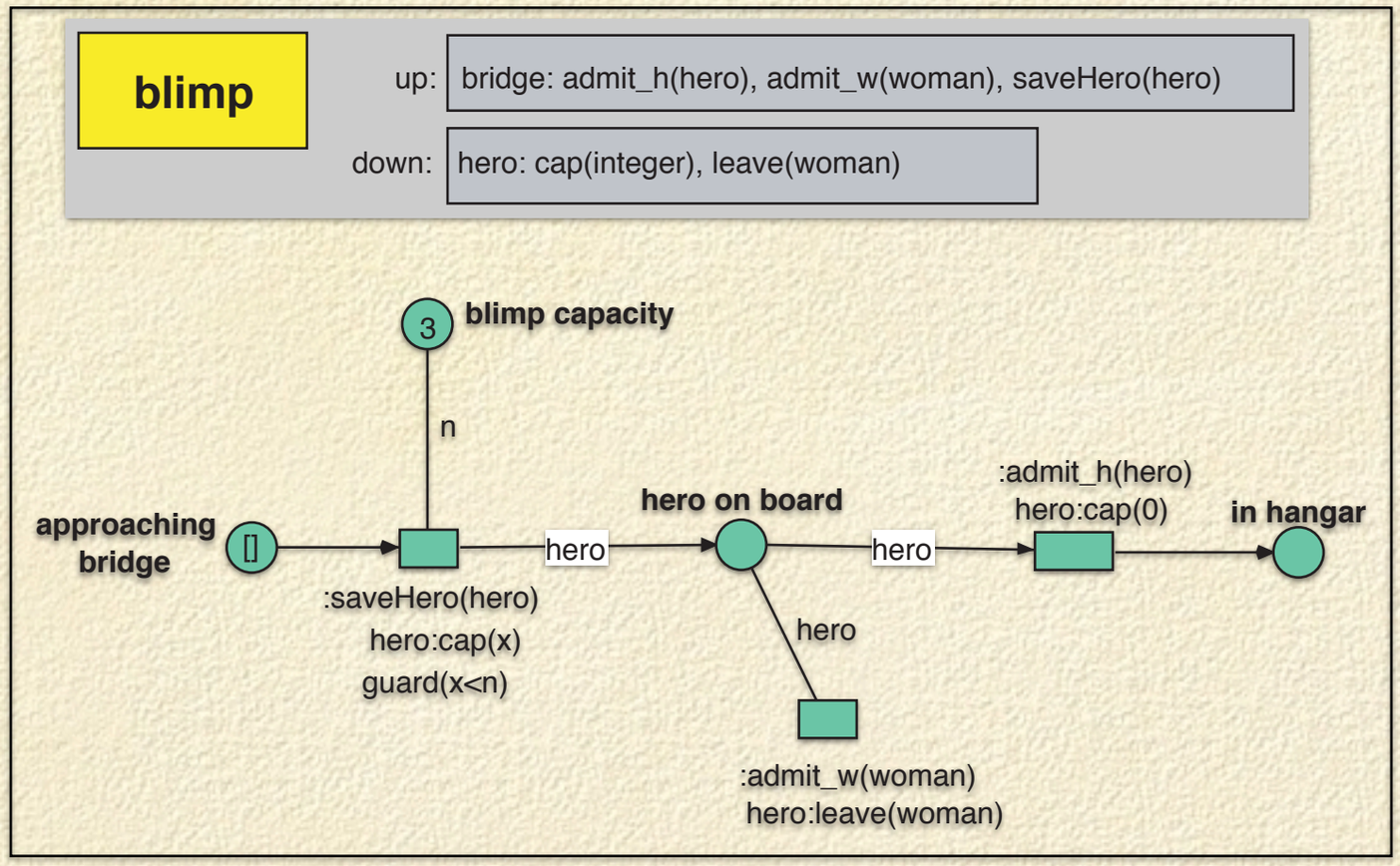
down:

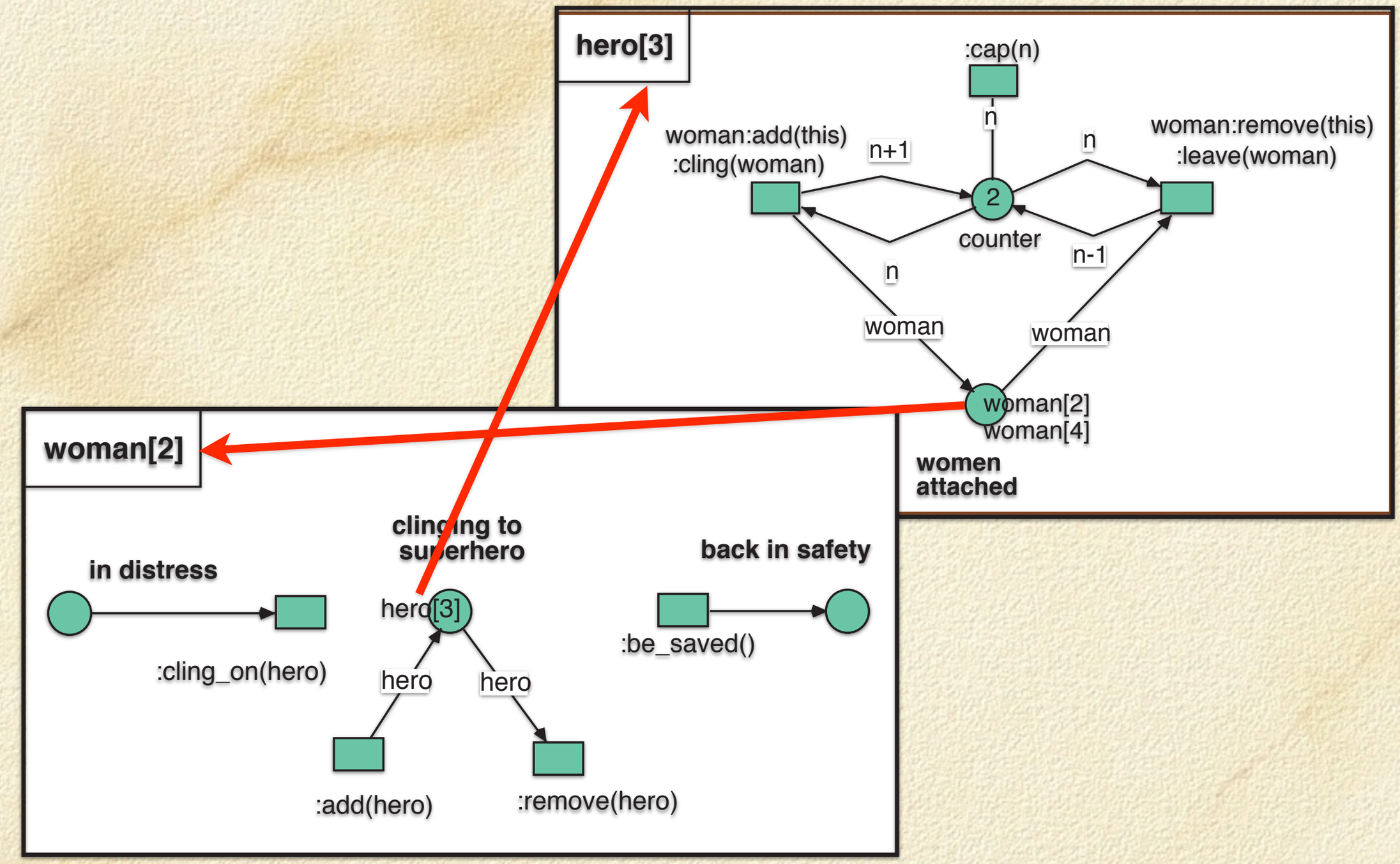
blimp:	admit_h(hero), admit_w(woman),saveHero(hero)
hero:	leave(woman), cling(woman)
woman:	cling_on(), be_saved()

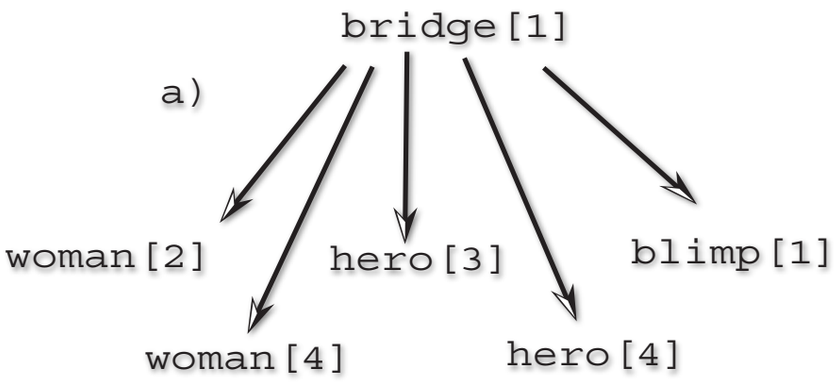


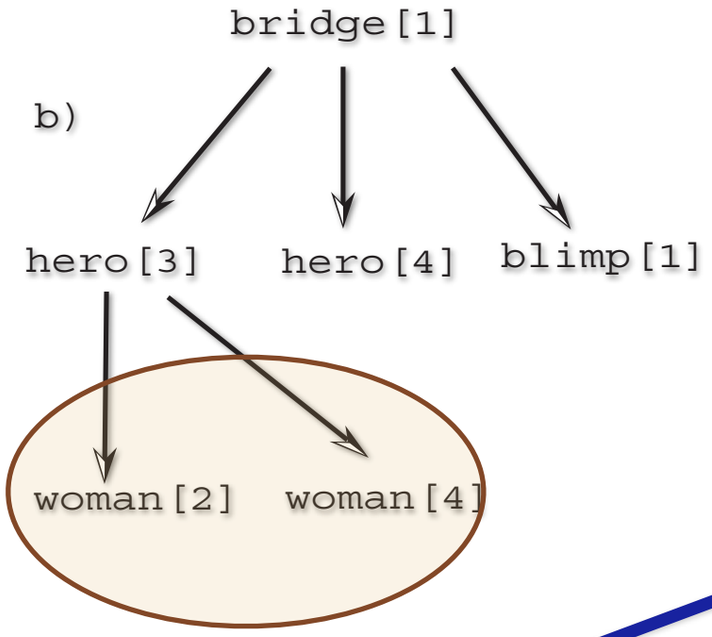

woman
up:
hero: add(hero), remove(hero)
bridge: cling_on(), be_saved()
down:
--
in distress
[]
:cling_on()
clinging
to hero
hero
hero
:add(hero)
:remove(hero)
back in safety
:be_saved()

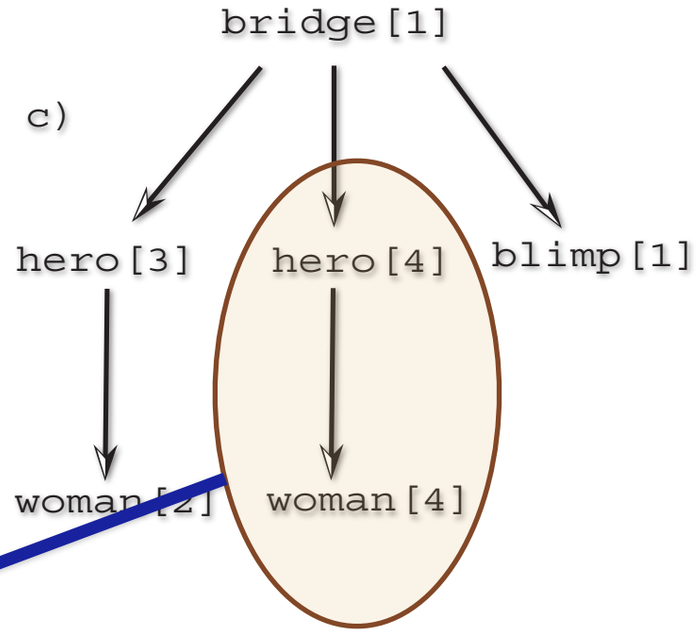

hero
up:
bridge: cling(woman), leave(woman)
blimp: cap(integer)
down:
woman: add(hero), remove(hero)
:cap(n)
n
woman:add(this)
:cling(woman)
n+1
0
n
woman:remove(this)
:leave(woman)
n-1
n
woman
woman
women
attached

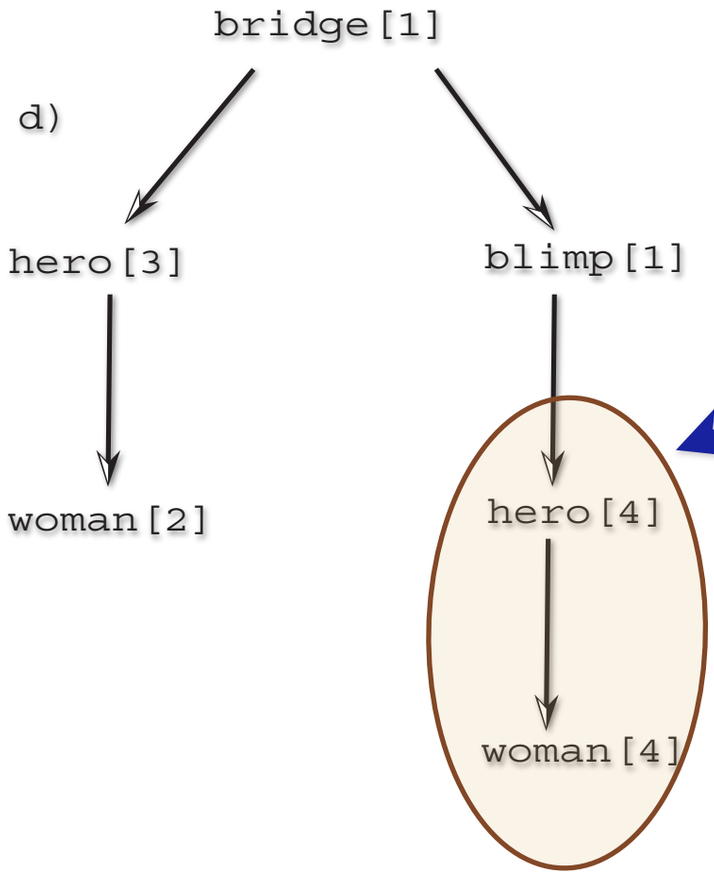

blimp
up:
bridge: admit_h(hero), admit_w(woman), saveHero(hero)
down:
hero: cap(integer), leave(woman)
3
blimp capacity
n
approaching bridge
[]
:saveHero(hero)
hero:cap(x)
guard(x<n)
hero
hero on board
hero
hero
:admit_h(hero)
hero:cap(0)
in hangar
:admit_w(woman)
hero:leave(woman)

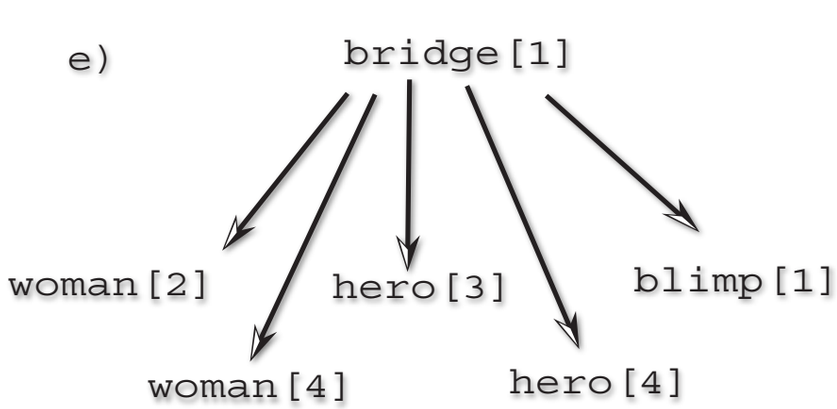

hero[3]
:cap(n)
n
woman:add(this)
:cling(woman)
n+1
n
woman:remove(this)
:leave(woman)
2
counter
n-1
n
woman
woman
woman[2]
woman[4]
women
attached
woman[2]
clinging to
superhero
in distress
back in safety
hero[3]
:cling_on(hero)
hero
hero
:be_saved()
:add(hero)
:remove(hero)


a)
bridge[1]
woman[2]
hero[3]
blimp[1]
woman[4]
hero[4]


b)
bridge[1]
hero[3]
hero[4]
blimp[1]
woman[2]
woman[4]


c)
bridge[1]
hero[3]
hero[4]
blimp[1]
woman[2]
woman[4]


d)
bridge[1]
hero[3]
blimp[1]
woman[2]
hero[4]
woman[4]


e)
bridge[1]
woman[2]
hero[3]
blimp[1]
woman[4]
hero[4]