FORTE 2006 Tutorial

Petri Nets

From Formal Specification to Software Design

Rüdiger Valk
University of Hamburg, Germany
valk@informatik.uni-hamburg.de

What I hope to convince you:

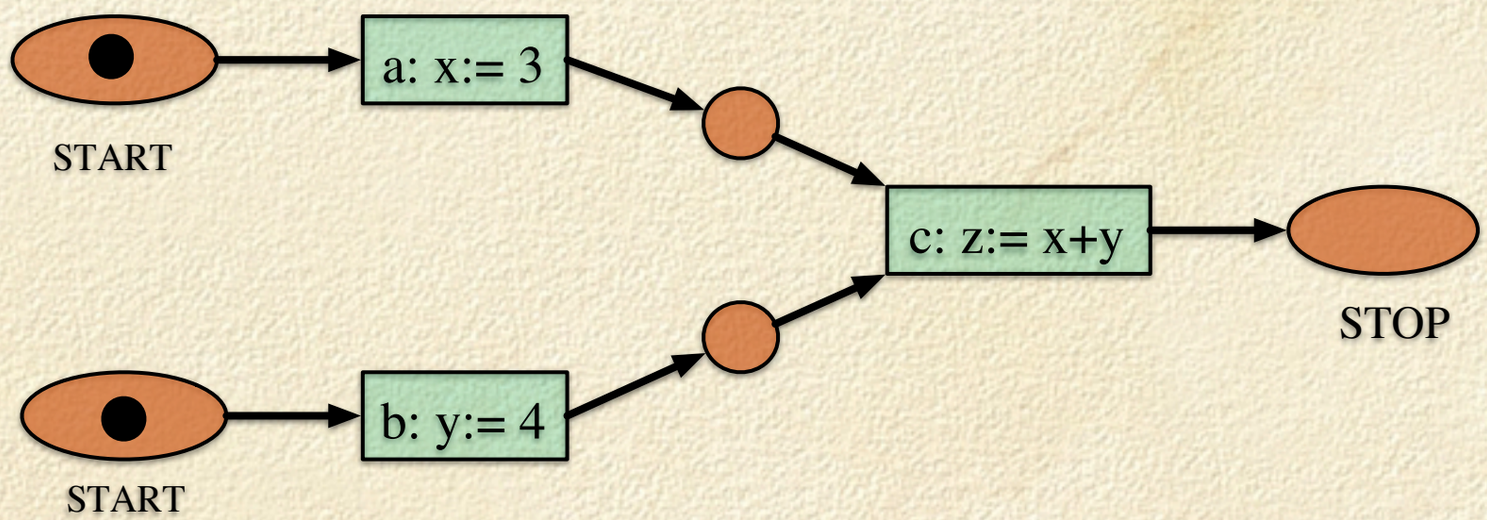
- ☑ that Petri nets nets provide a “natural” modeling method
- ☑ that they are “easy” to learn
- ☑ that they allow to “fix” important features of software systems, in particular distributed systems
- ☑ that they support “clear” structuring in software development
- ☑ that are supported by “nice” tools

What I cannot show you here:

- ☑ the complete mathematical formalism (see references)
- ☑ formal theorems and structural results
- ☑ all application areas

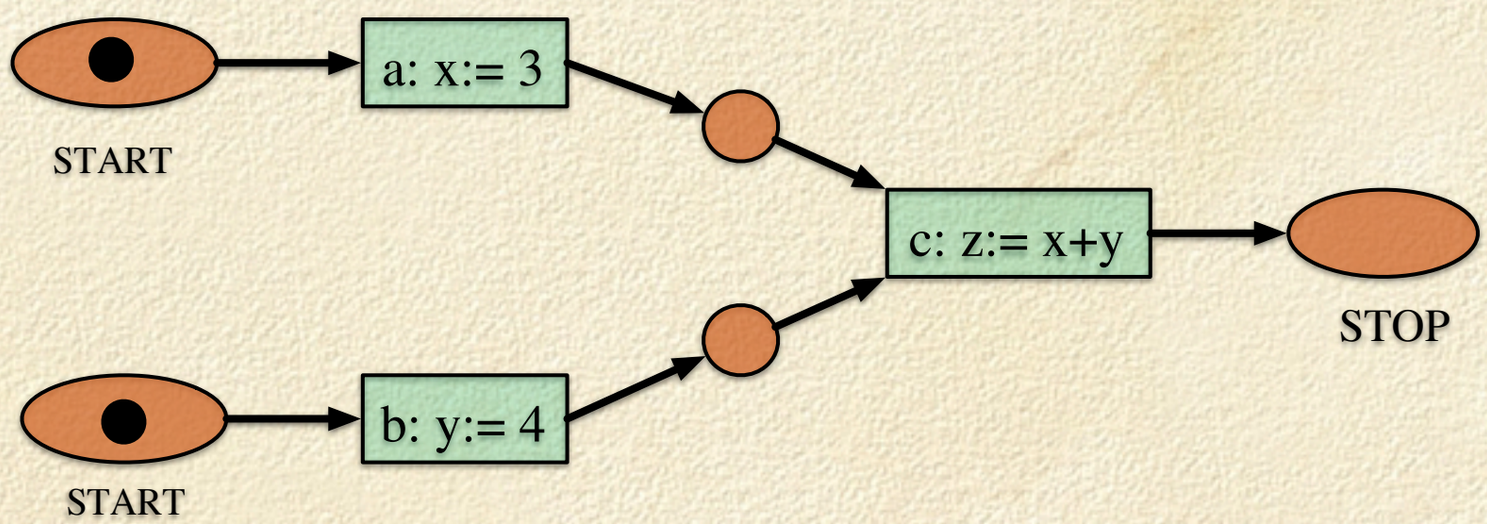
General Properties of Petri Nets

- graphical and equivalent algebraic/textual representation
- (formally verified) analysis algorithms
- abstraction and hierarchical structures
- striking concepts in concurrency



General Properties of Petri Nets

- Tools for eding, execution and analysis
  - RENEW http://www.renew.de/
  - Design/CPN http://wiki.daimi.au.dk/cpntools/cpntools.wiki
  - MARS [MACAO | CPN-AMI] http://www-src.lip6.fr/logiciels/mars/
- universality in applications (nearly all application areas)
- variants of the concept
  - time
  - stochastic
  - high-level
  - object oriented
  - ...



Introducting Petrinets by 3 approaches:

- starting from program code
- communicating automata/transition systems
- from lower to higher level models

starting from program code

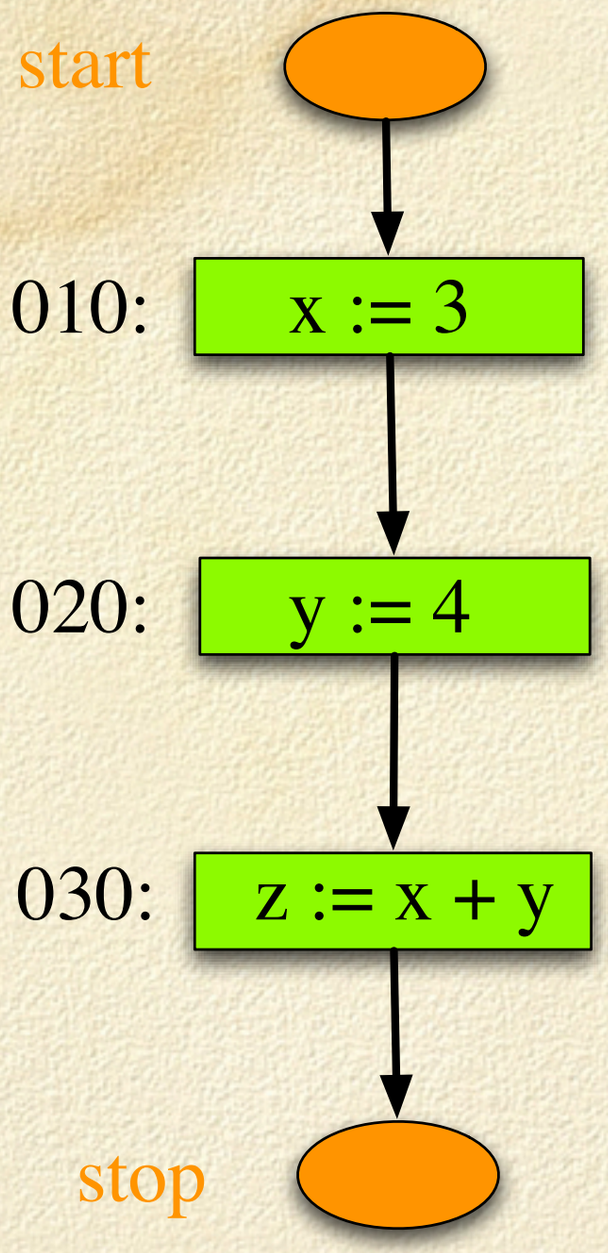
start

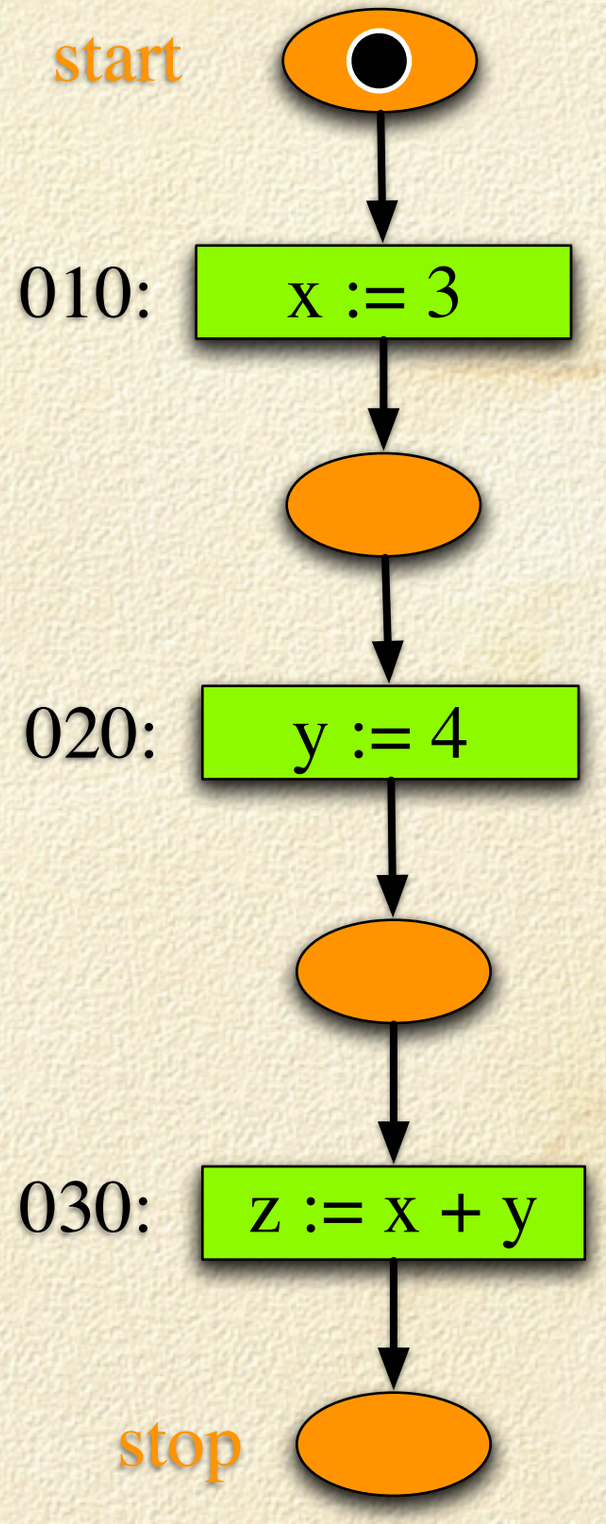
010: x := 3

020: y := 4

030: z := x + y

stop





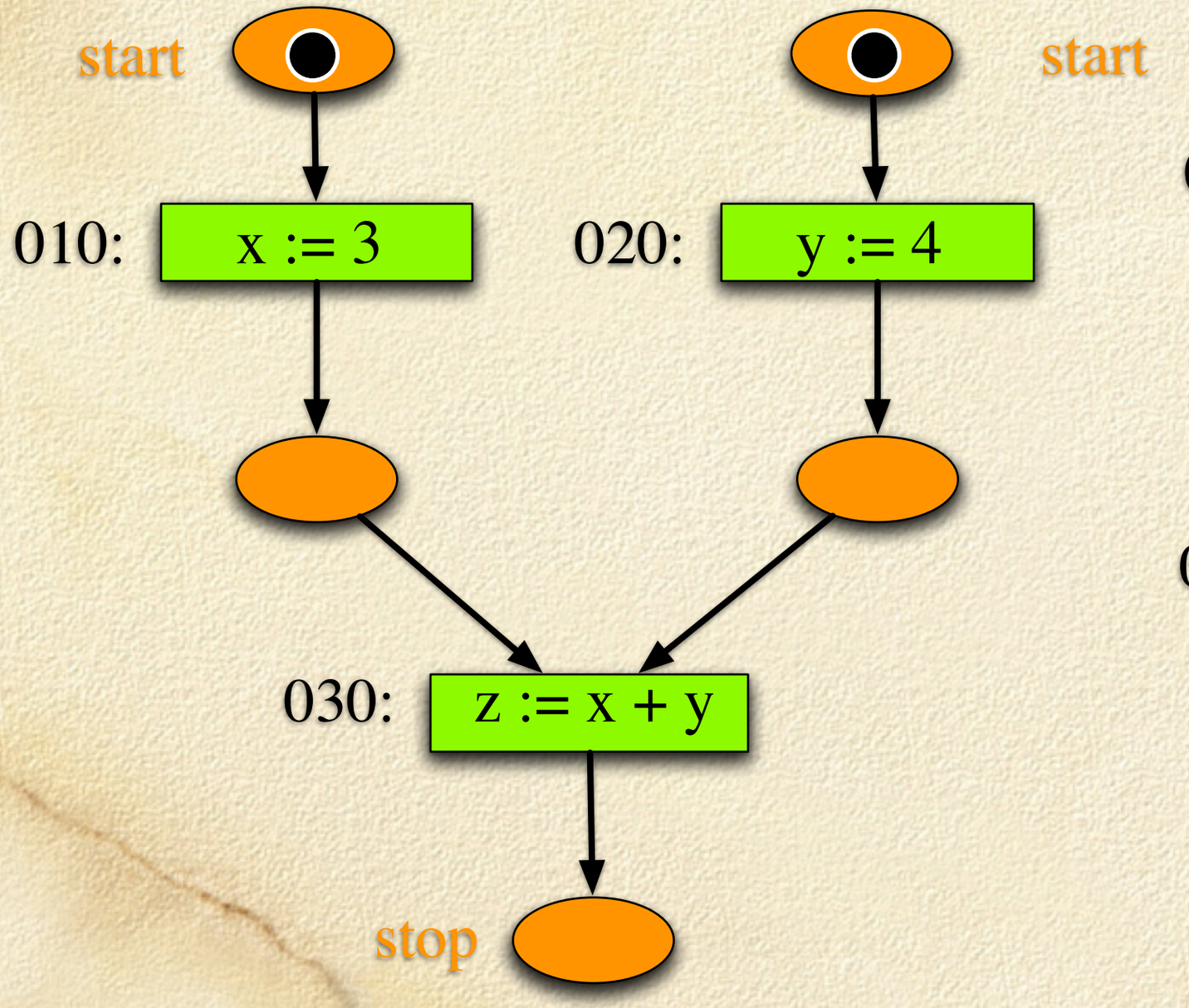
start

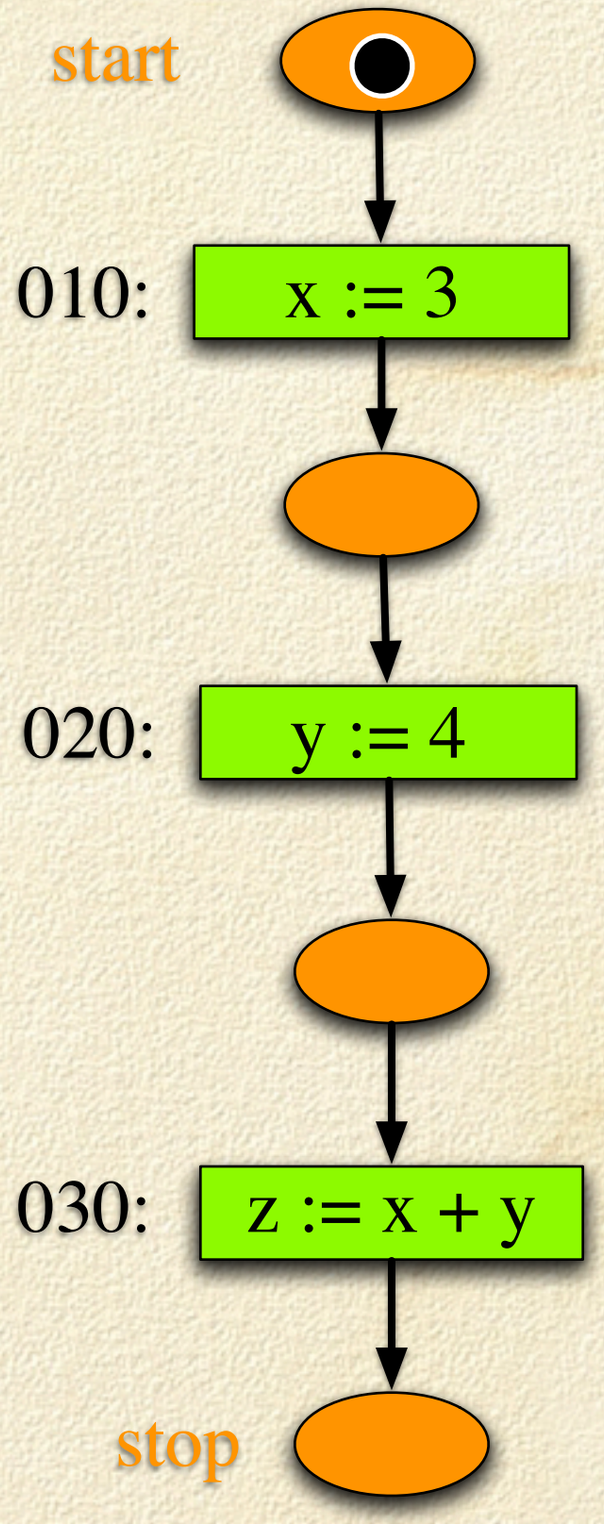
010: x := 3

020: y := 4

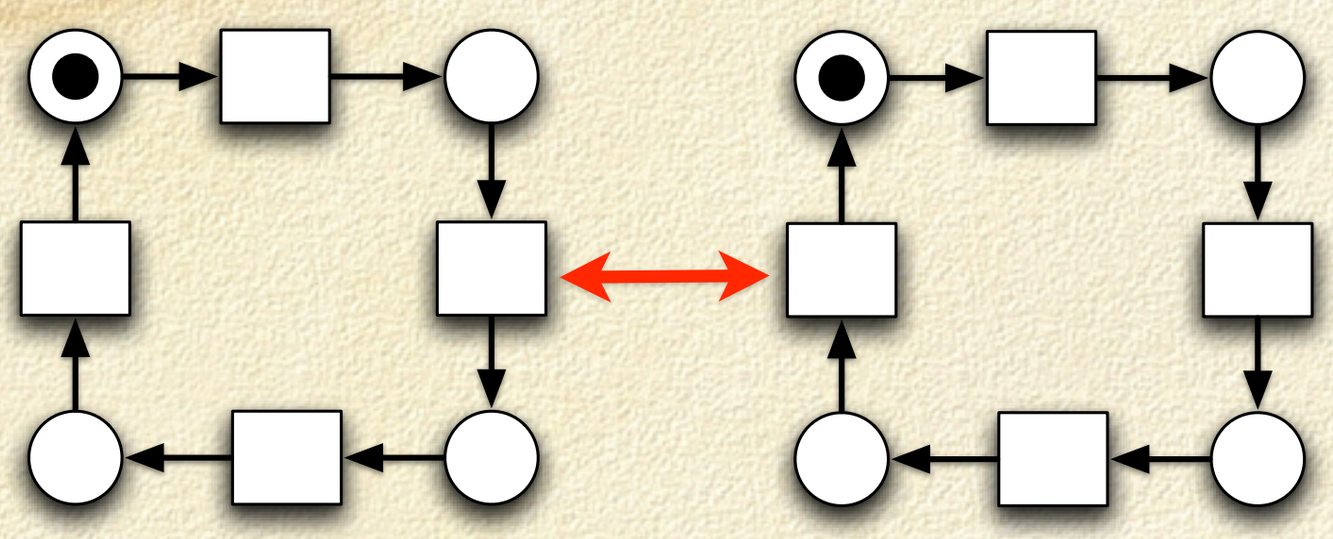
030: z := x + y

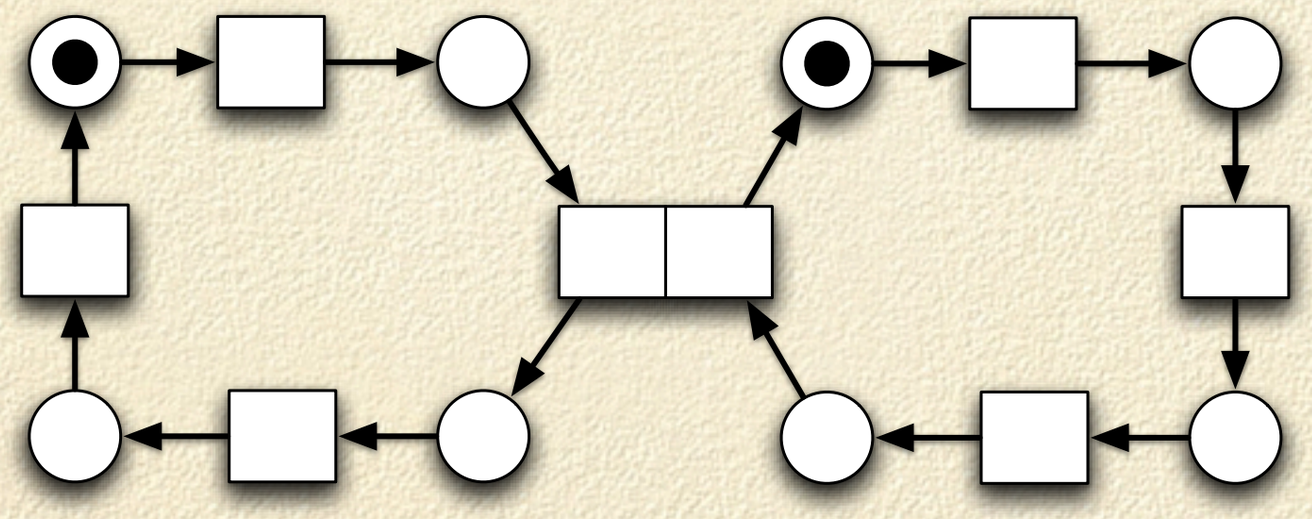
stop





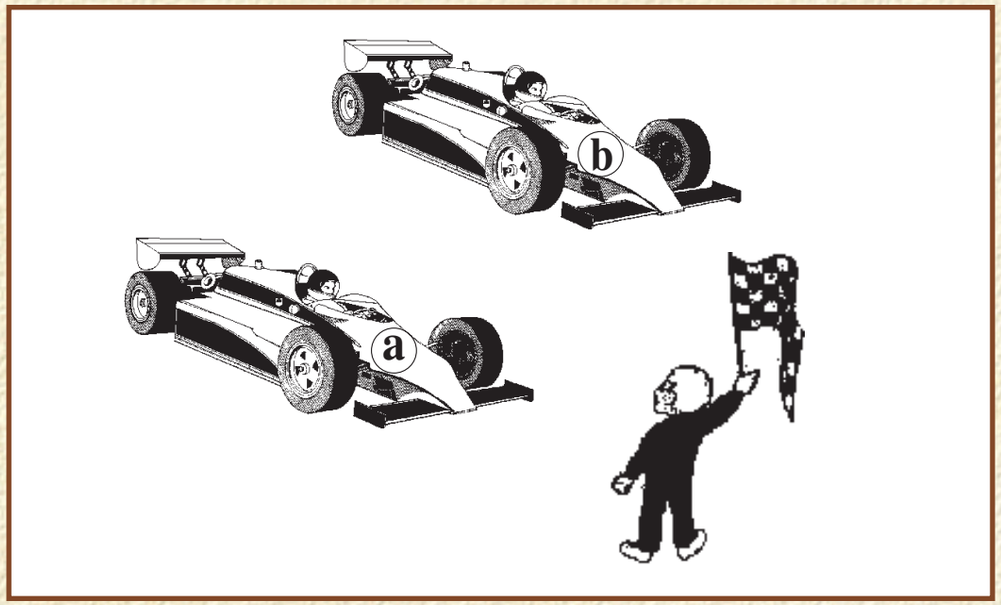
- communicating automata/transition systems





- from lower to higher level models

starting two racing cars



List of conditions

p_1: car a; preparing for start

p_2: car a; waiting for start

p_3: car a; running

p_4: ready sign of car a

p_5: start sign for car a

p_6: starter; waiting for ready signs

p_7: starter; start sign given

p_8: ready sign of car b

p_9: start sign for car b

p_{10}: car b; preparing for start

p_{11}: car b; waiting for start

p_{12}: car b; running

List of actions

t_1: car a; send ready sign

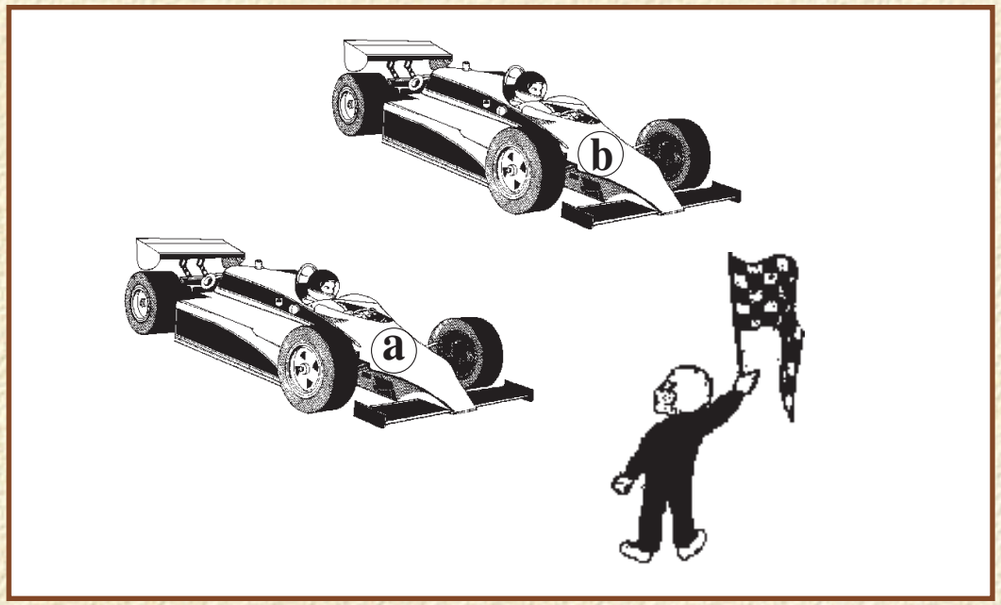
t_2: car a; start race

t_3: starter; give start sign

t_4: car b; send ready sign

t_5: car b; start race

starting two racing cars



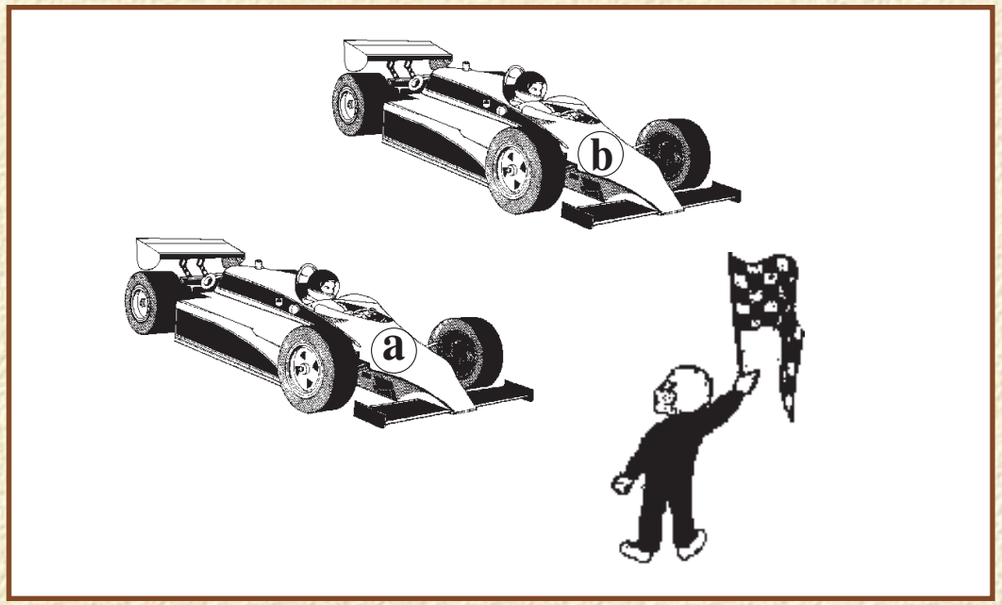
global state transition

car a; preparing for start	p1=true
car a; waiting for start	p2
car a; running	p3
ready sign of car a	p4
start sign for car a	p5
starter; waiting for ready s.	p6=true
starter; start sign given	p7
ready sign of car b	p8
start sign for car b	p9
car b; preparing for start	p10=true
car b; waiting for start	p11
car b; running	p12

→ t1 →

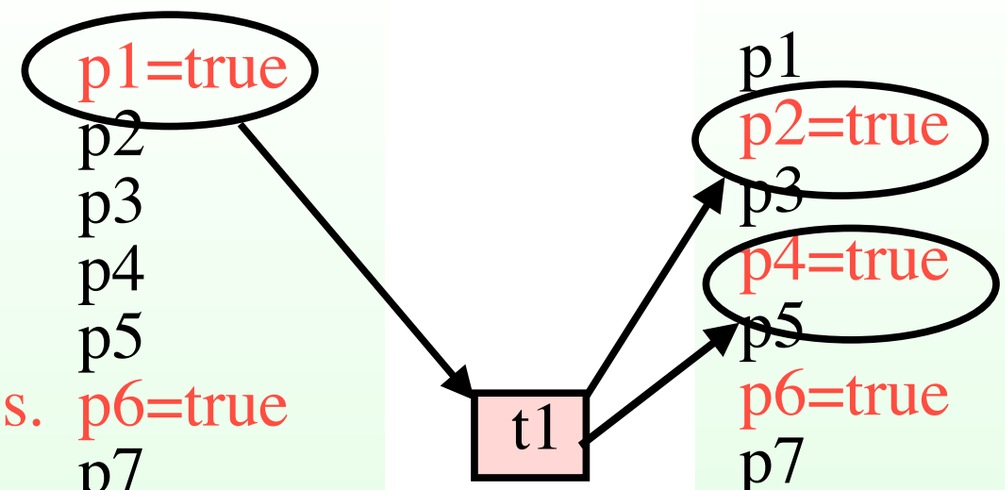
p1	car a; preparing for start
p2=true	car a; waiting for start
p3	car a; running
p4=true	ready sign of car a
p5	start sign for car a
p6=true	starter; waiting for ready s.
p7	starter; start sign given
p8	ready sign of car b
p9	start sign for car b
p10=true	car b; preparing for start
p11	car b; waiting for start
p12	car b; running

starting two racing cars



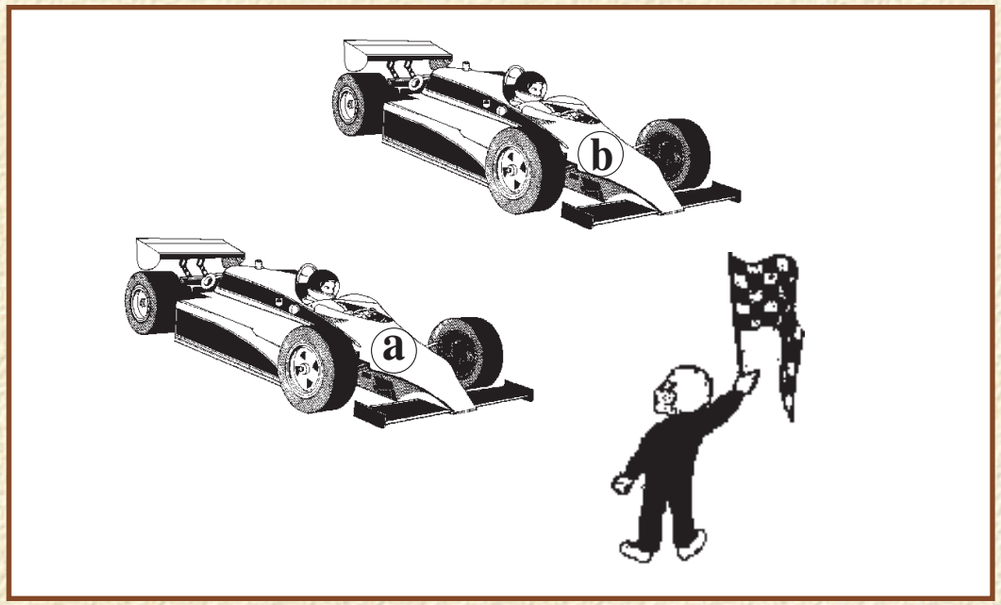
local state transition

car a; preparing for start	p1=true
car a; waiting for start	p2
car a; running	p3
ready sign of car a	p4
start sign for car a	p5
starter; waiting for ready s.	p6=true
starter; start sign given	p7
ready sign of car b	p8
start sign for car b	p9
car b; preparing for start	p10=true
car b; waiting for start	p11
car b; running	p12



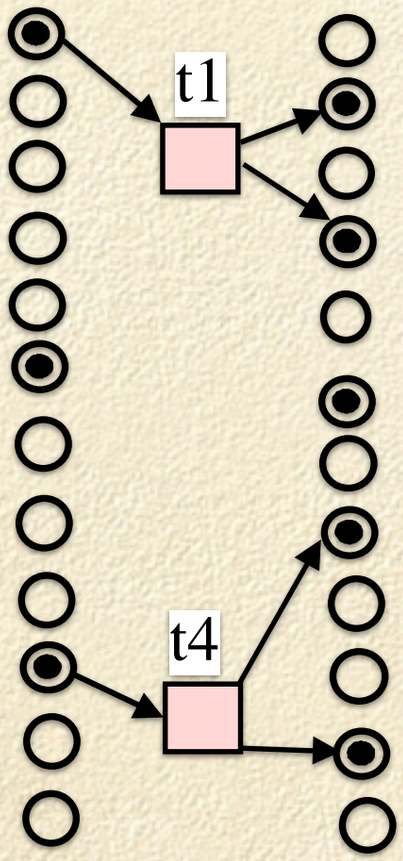
p1	car a; preparing for start
p2=true	car a; waiting for start
p3	car a; running
p4=true	ready sign of car a
p5	start sign for car a
p6=true	starter; waiting for ready s.
p7	starter; start sign given
p8	ready sign of car b
p9	start sign for car b
p10=true	car b; preparing for start
p11	car b; waiting for start
p12	car b; running

starting two racing cars



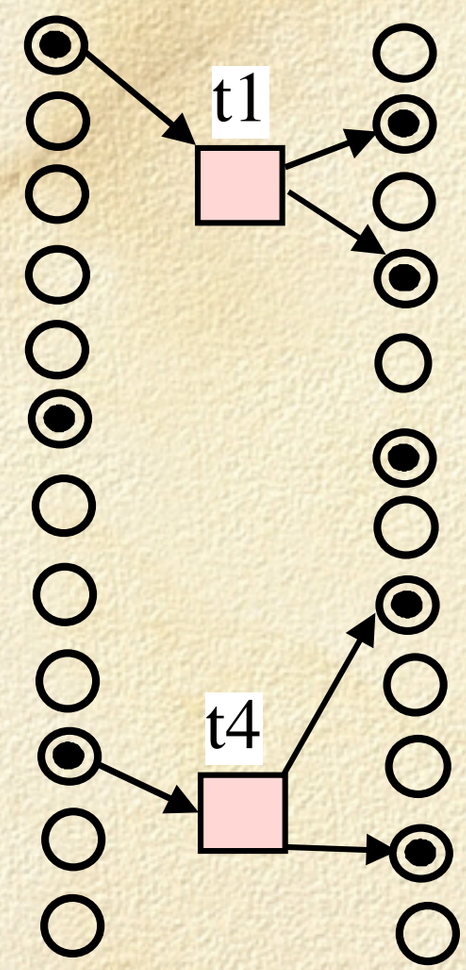
independent state transition
concurrent state transition

car a; preparing for start	p1=true
car a; waiting for start	p2
car a; running	p3
ready sign of car a	p4
start sign for car a	p5
starter; waiting for ready s.	p6=true
starter; start sign given	p7
ready sign of car b	p8
start sign for car b	p9
car b; preparing for start	p10=true
car b; waiting for start	p11
car b; running	p12



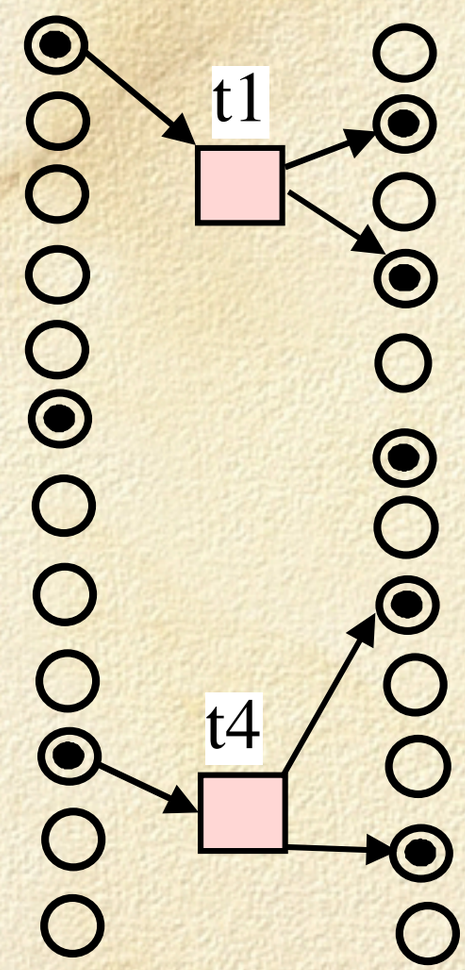
p1	car a; preparing for start
p2=true	car a; waiting for start
p3	car a; running
p4=true	ready sign of car a
p5	start sign for car a
p6=true	starter; waiting for ready s.
p7	starter; start sign given
p8=true	ready sign of car b
p9	start sign for car b
p10	car b; preparing for start
p11=true	car b; waiting for start
p12	car b; running

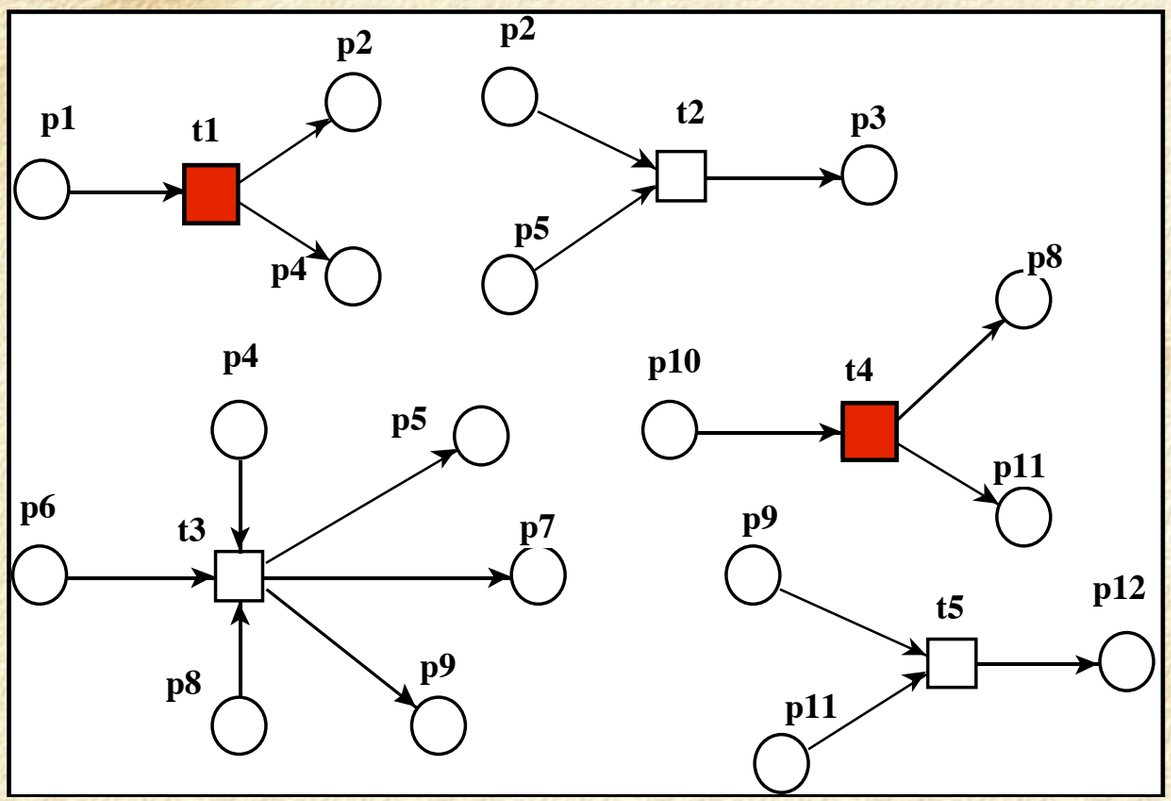
I	The Principle of Duality
	Two kinds of elements: state elements, passive elements: Places P (conditions, places, resources, waiting pools, channels) transition elements, active elements: Transitions T (conditions, places, resources, waiting pools, channels)

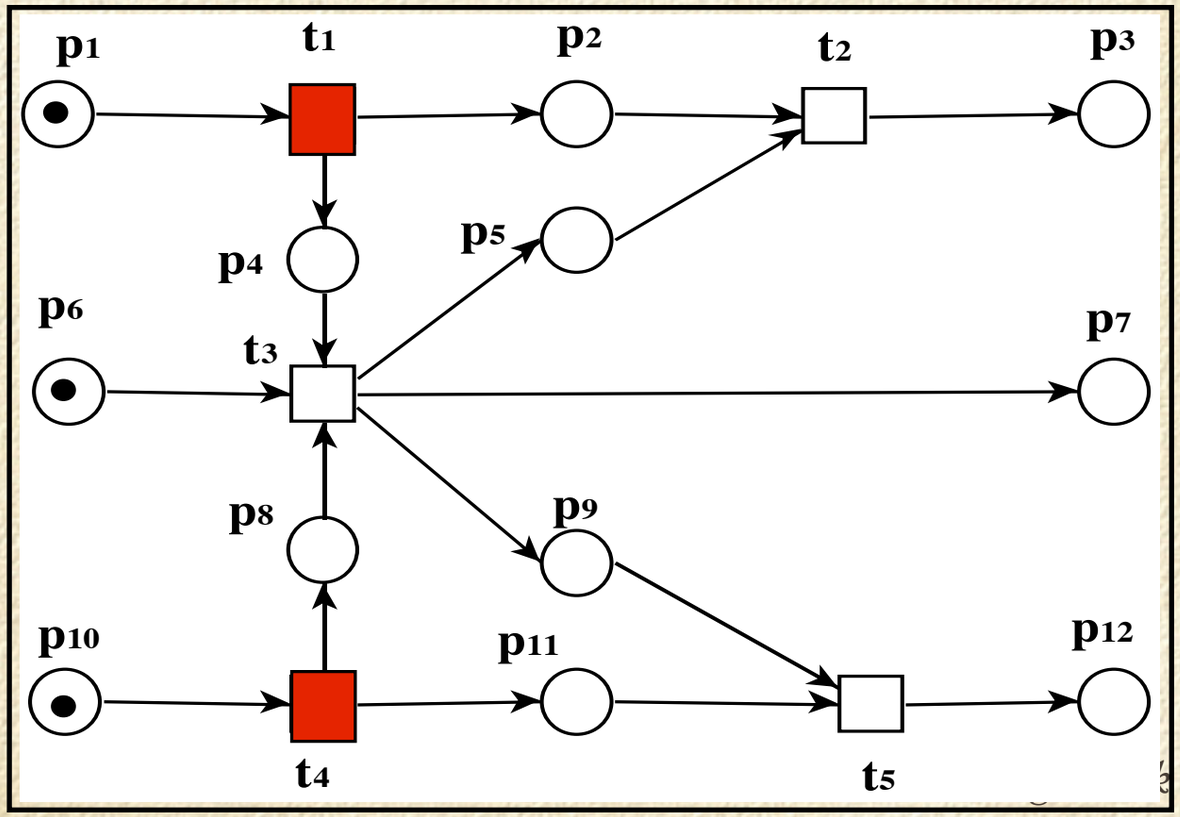


II	The Principle of Locality
	The behaviour of a transition exclusively depends on its locality.

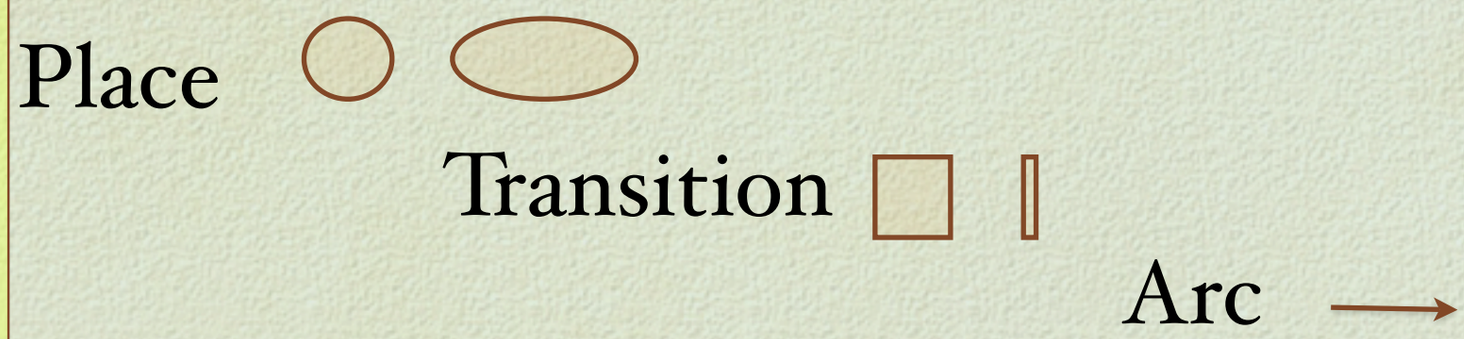
III	The Principle of Concurrency
	Transitions having disjoint locality occur independently (concurrent).


t1
t4


p1
t1
p2
p4
p2
t2
p3
p5
p8
p4
p5
p10
t4
p11
p6
t3
p7
p9
t5
p12
p8
p9
p11

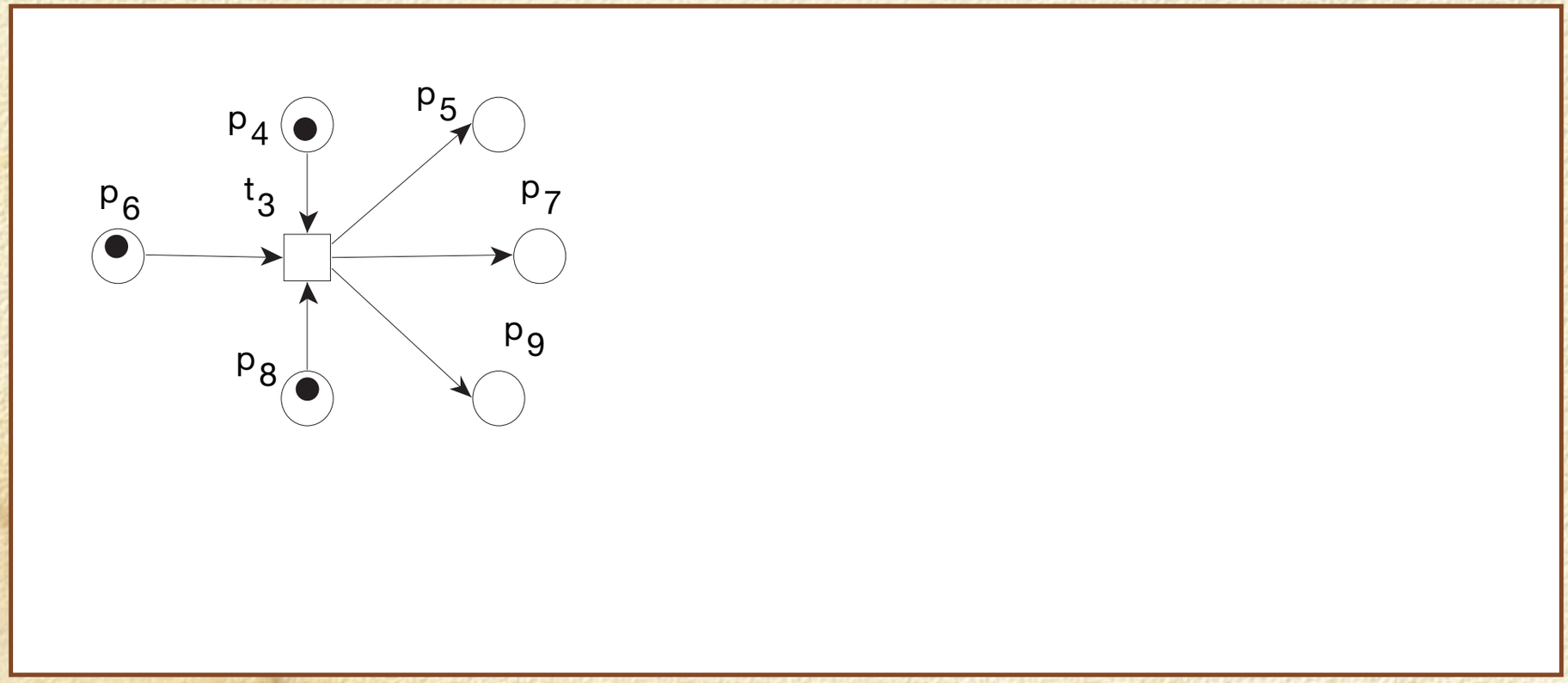

p1
t1
p2
t2
p3
p4
p5
p6
t3
p7
p8
p9
p10
p11
p12
t4
t5

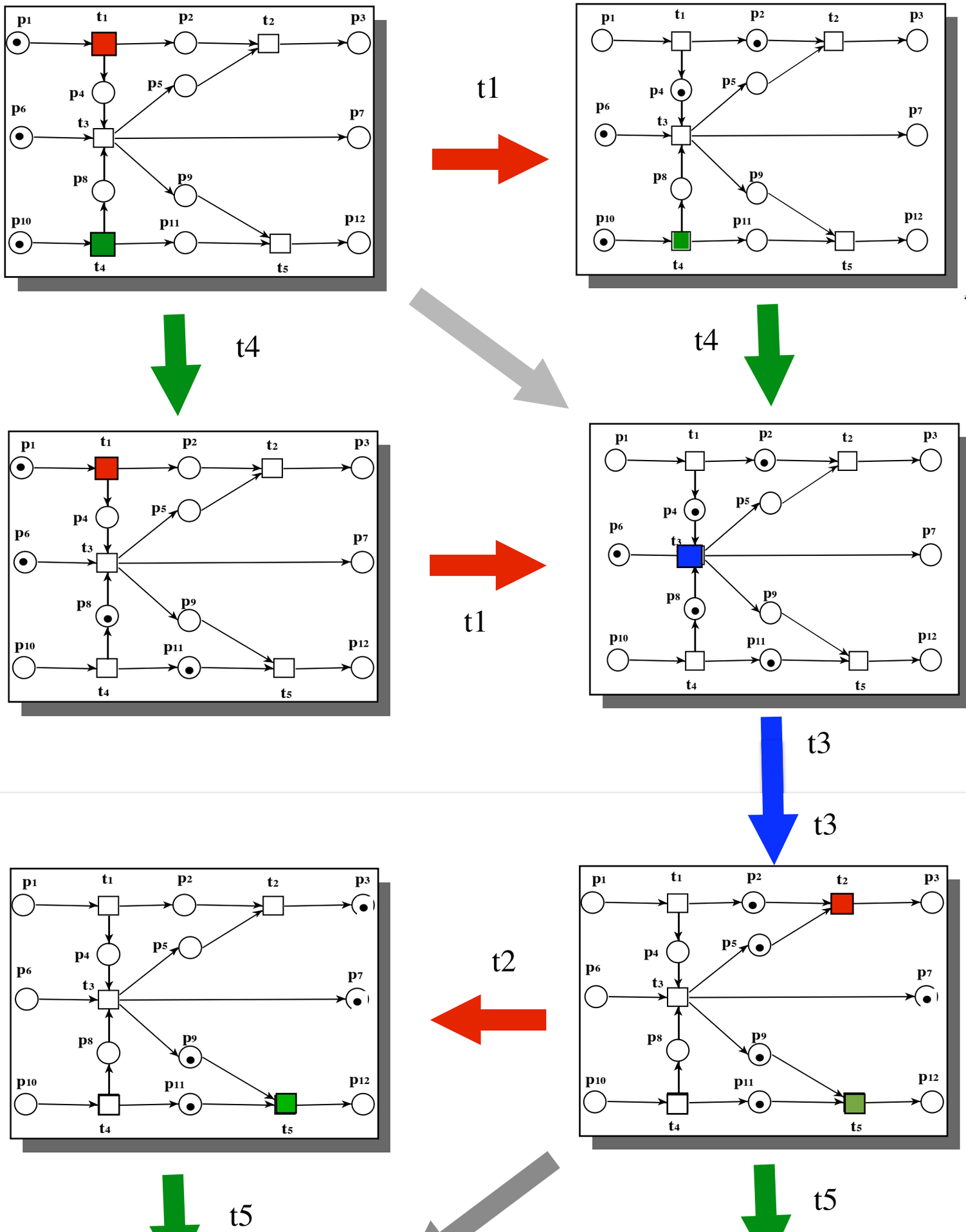
IV The Principle of Graphical Representation



V	The Principle of textual Representation
	$N = (P,T,F)$ $F \subseteq (P \times T) \cup (T \times P)$

$(p_6, t_3) \in F$ $\qquad$ $(t_3, p_5) \in F$

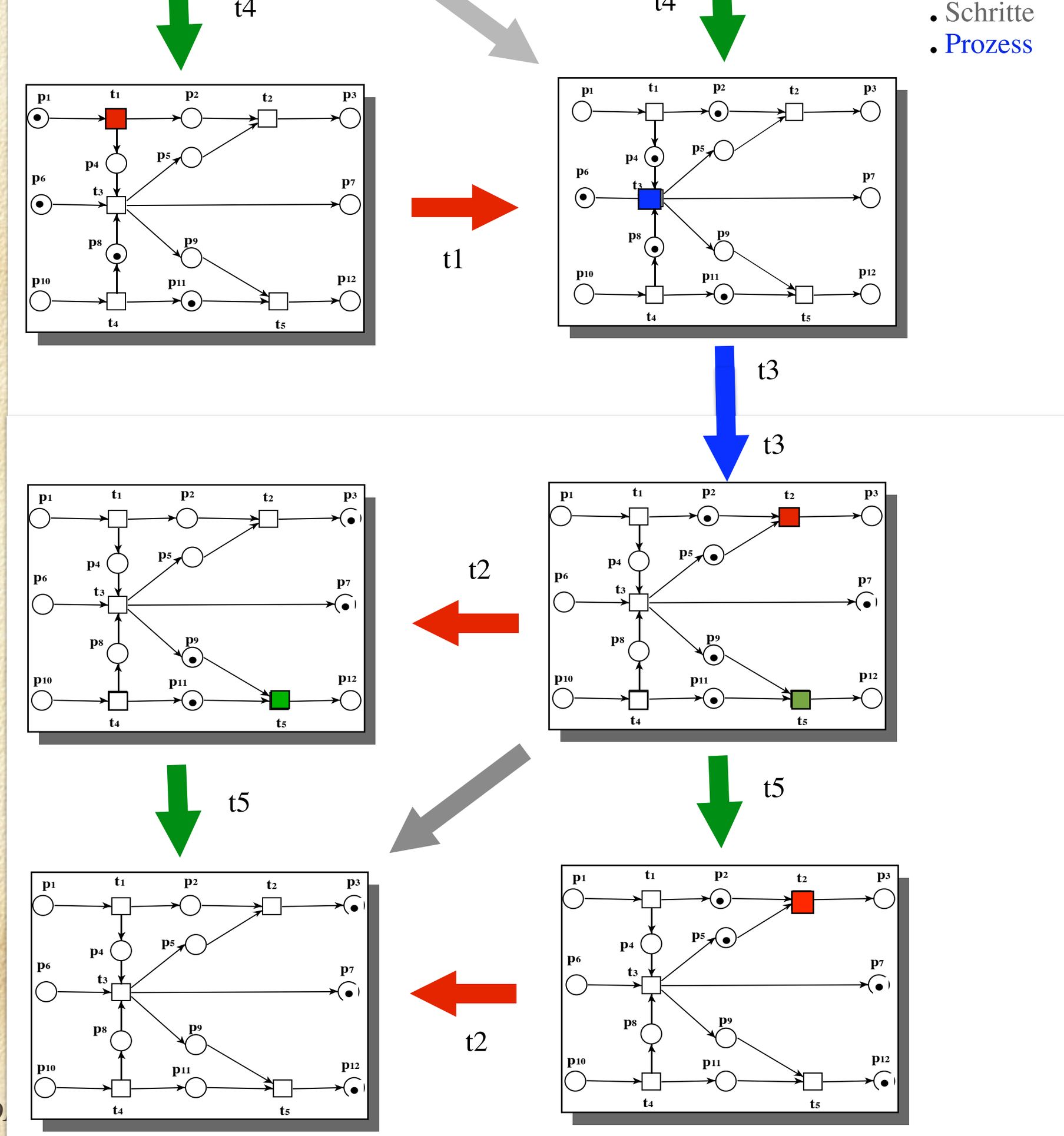


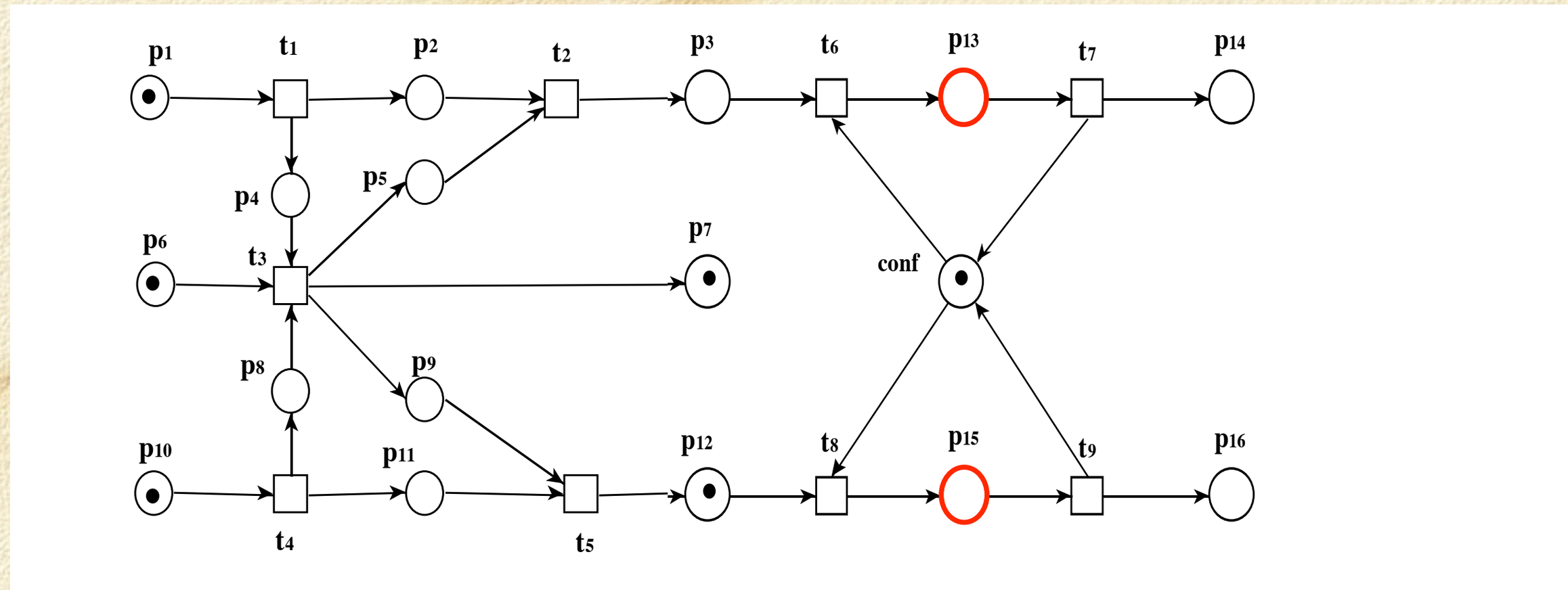


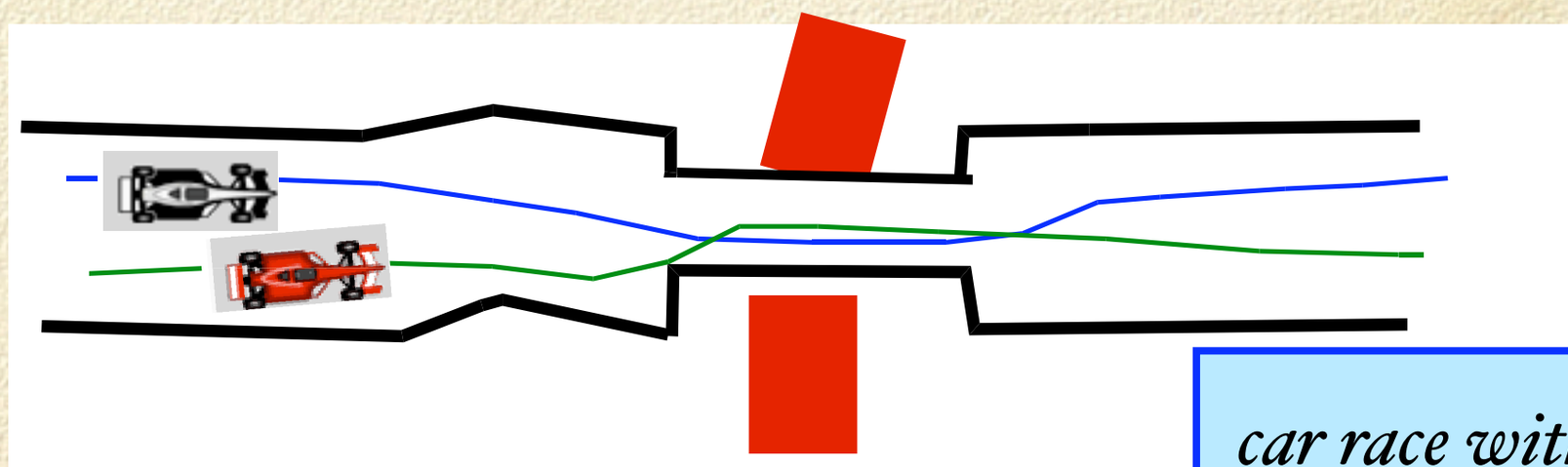
Ablaufsemantik:

- Folgen
- Schritte
- Prozess

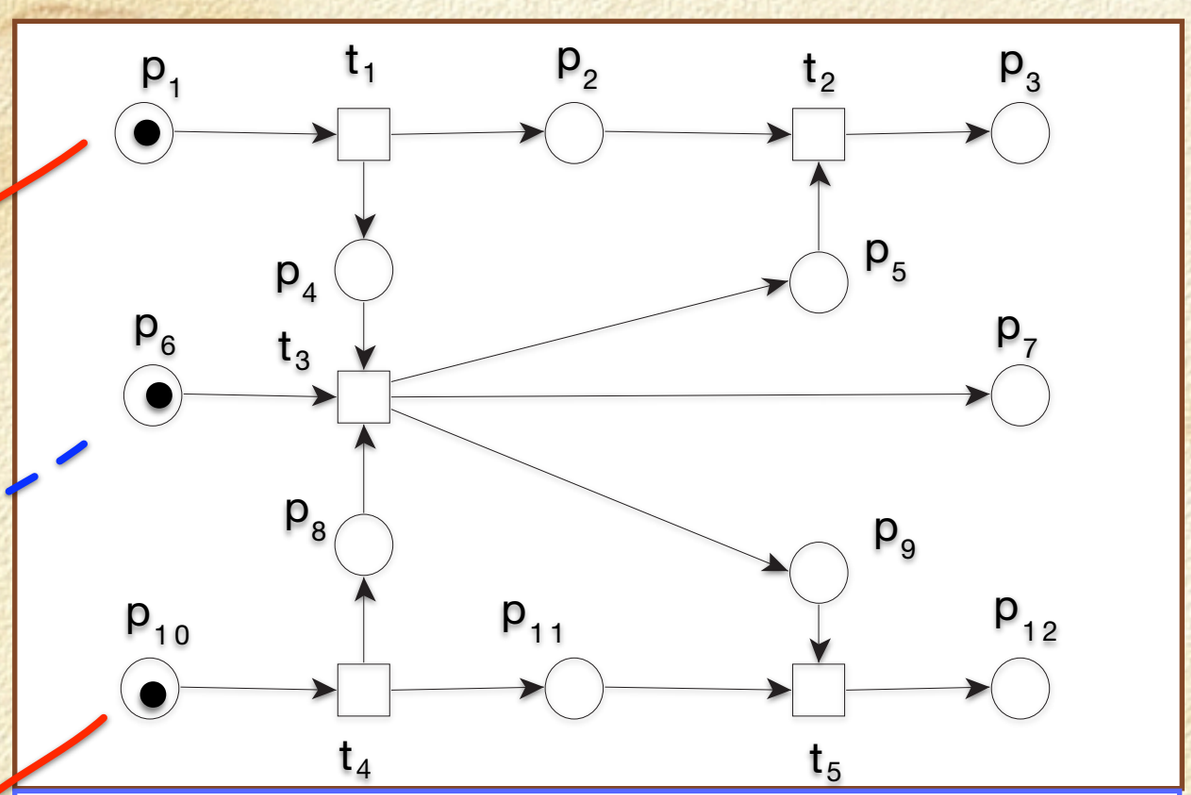
- Schritte
- Prozess







car race with bottleneck

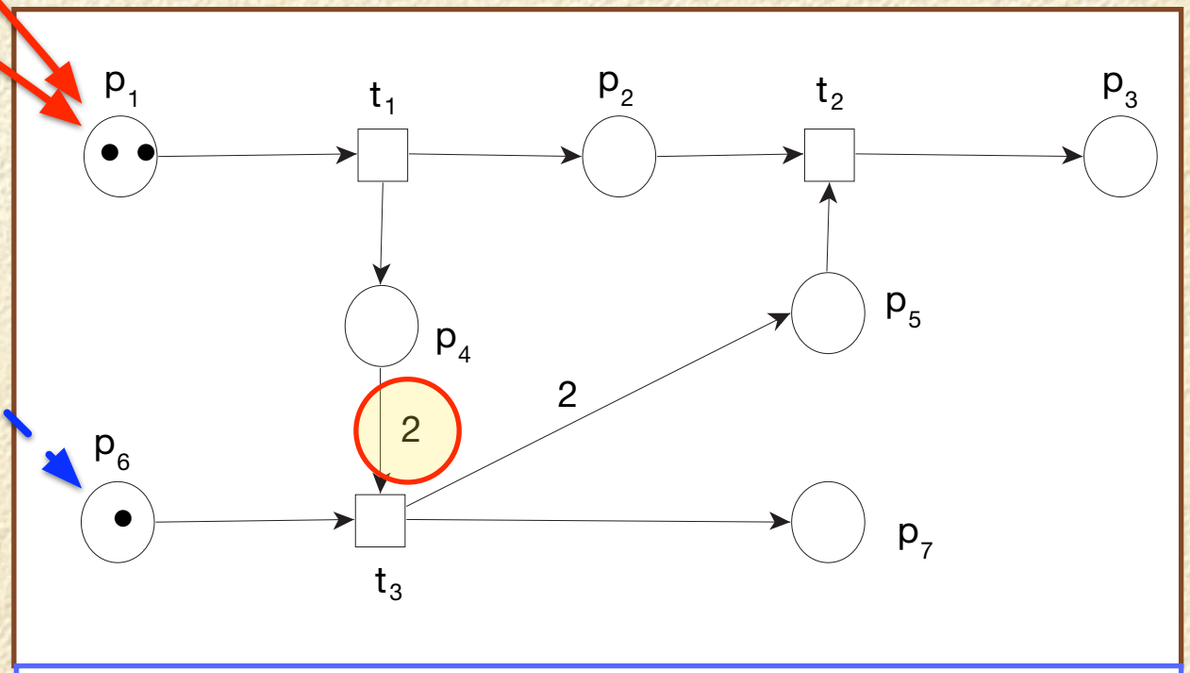


marked net

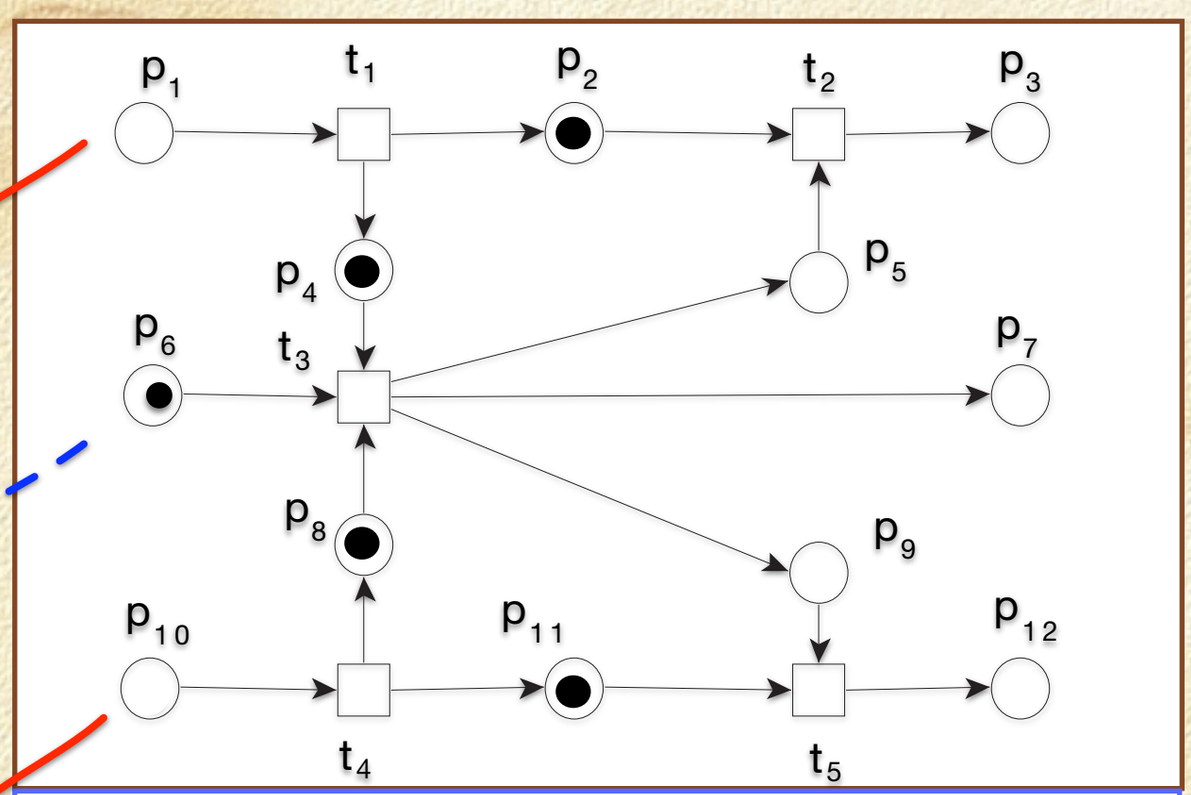
Place/Transition-Net

new:

multiple tokens
arc weights



P/T-net

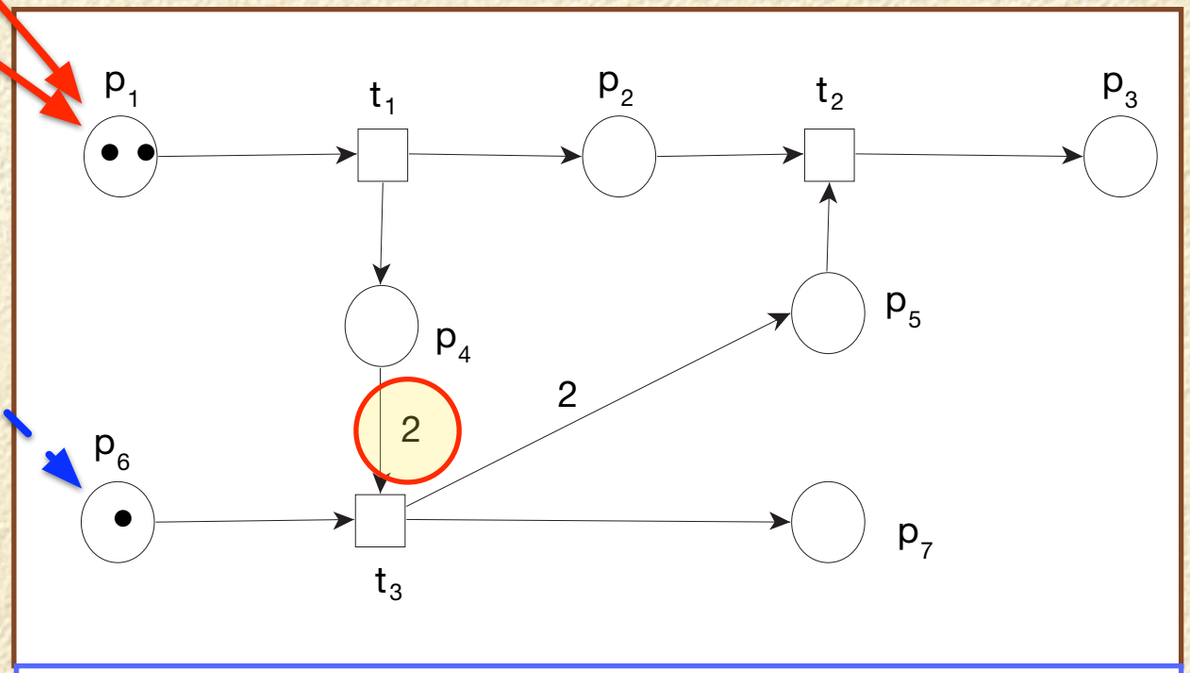


marked net

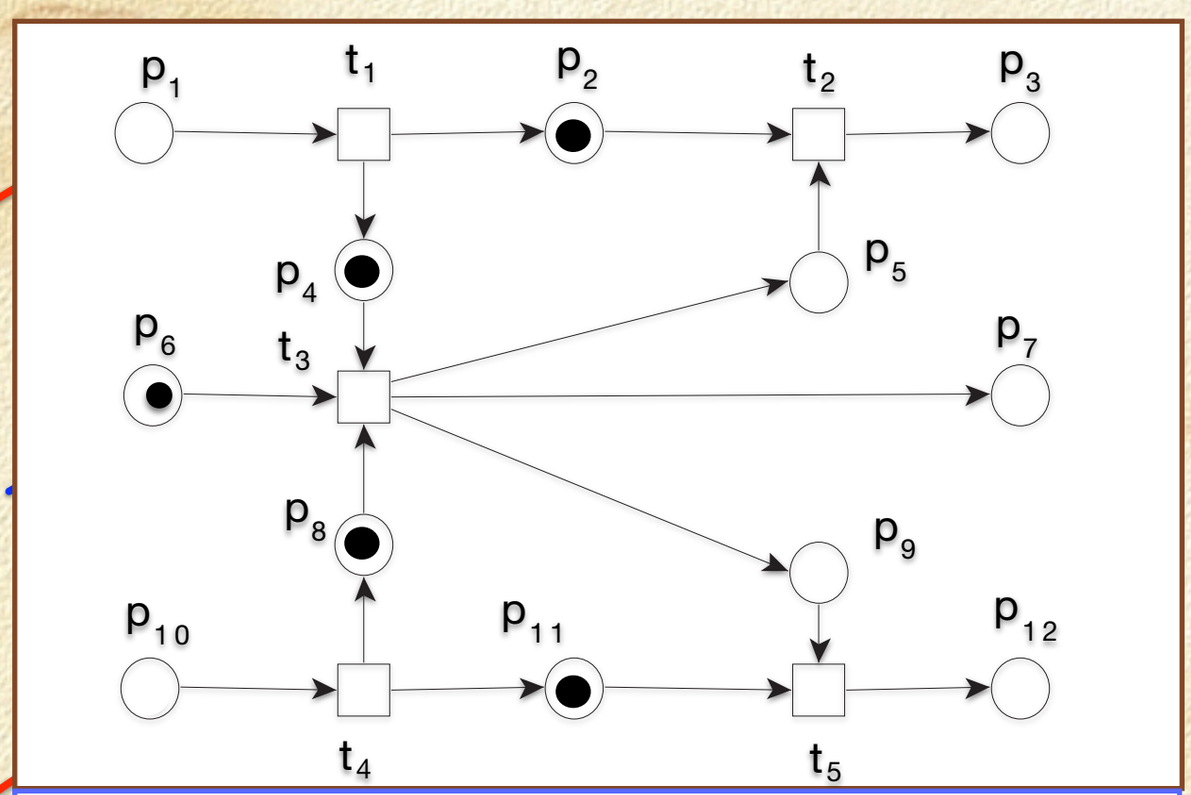
Place/Transition-Net

new:

multiple tokens
arc weights



P/T-net

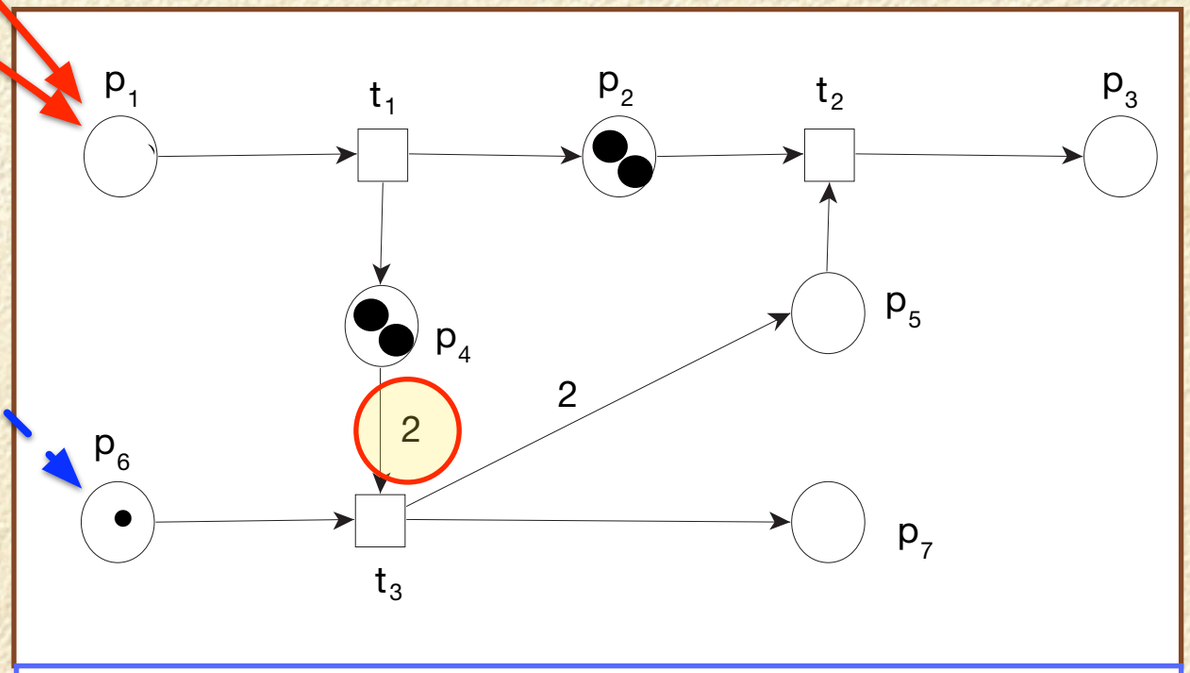


marked net

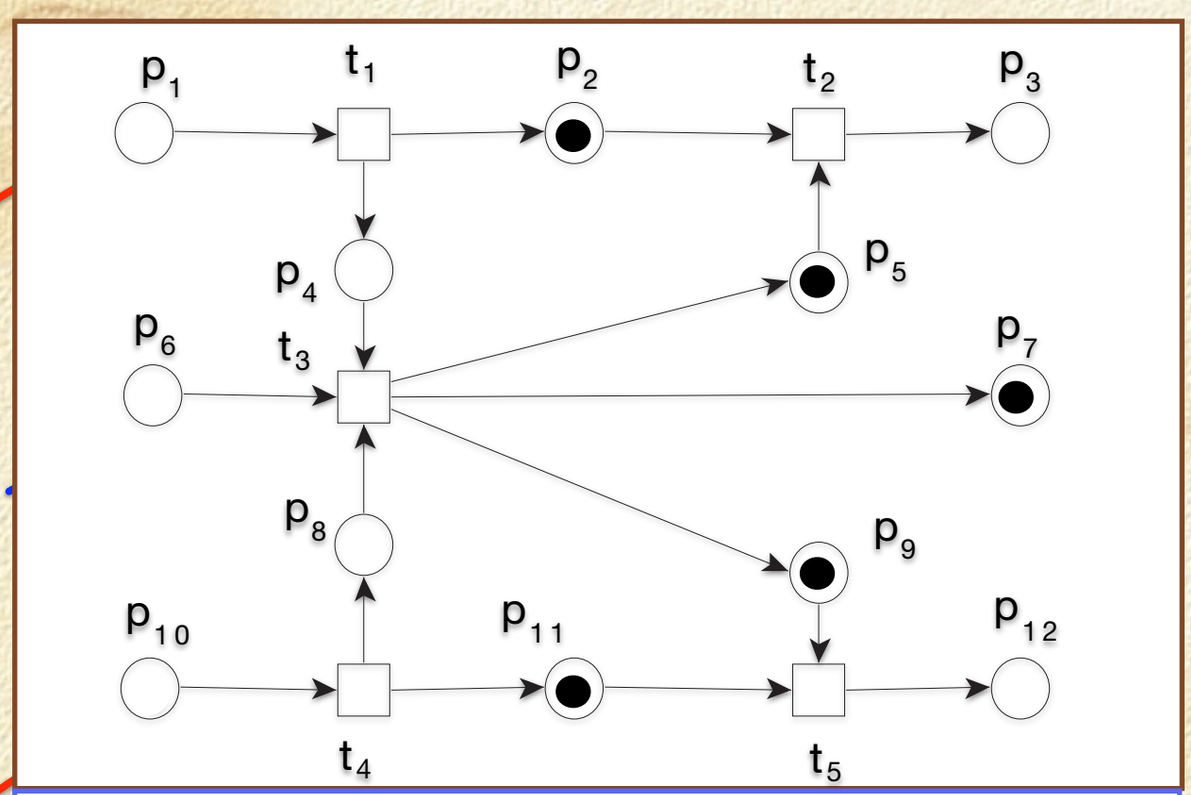
Place/Transition-Net

new:

multiple tokens
arc weights



P/T-net

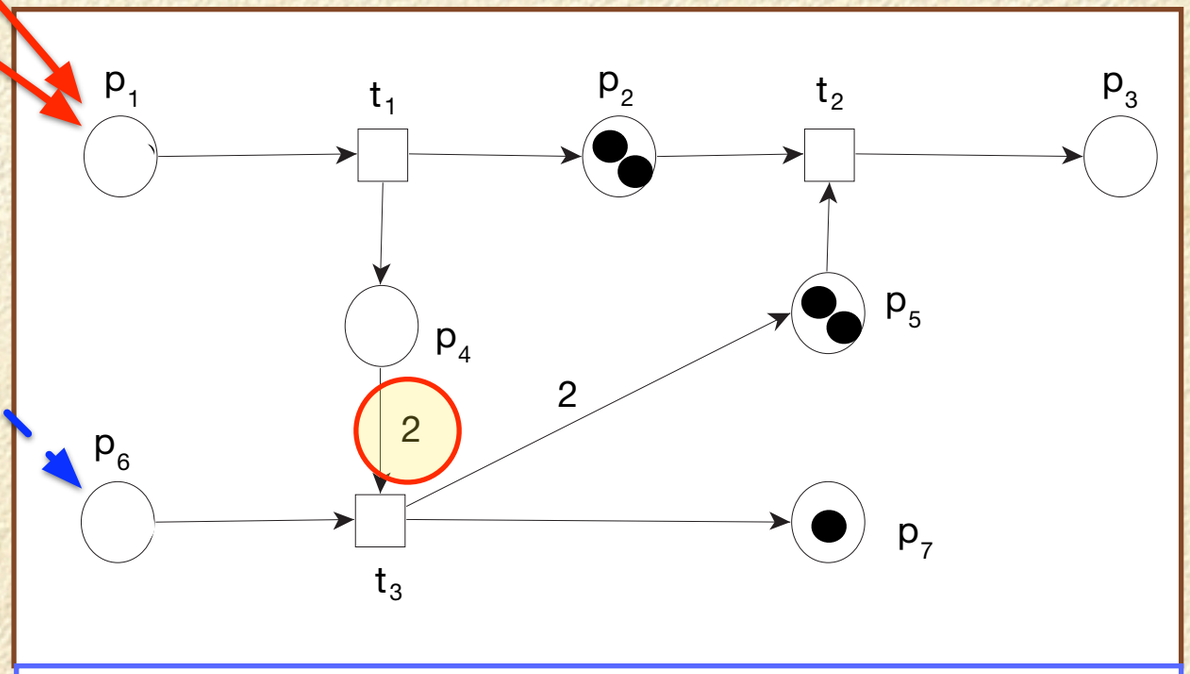


marked net

Place/Transition-Net

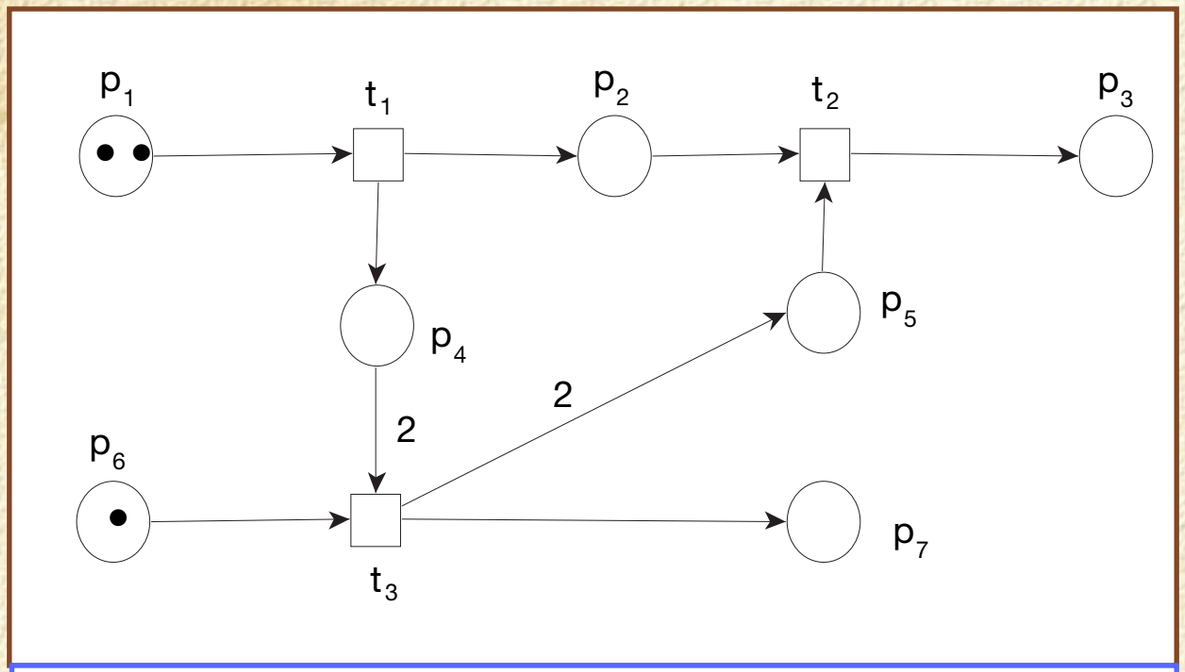
new:

multiple tokens
arc weights



P/T-net

Place/Transition net *P/T net*

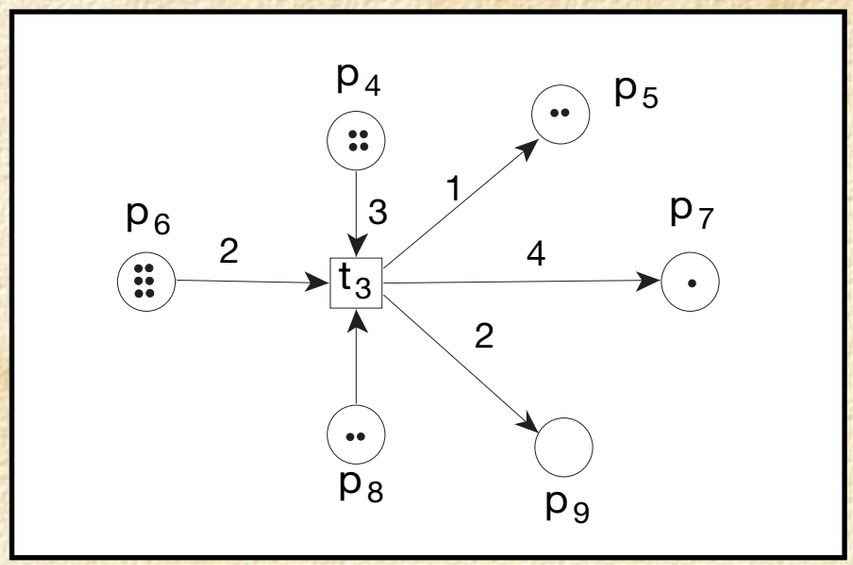


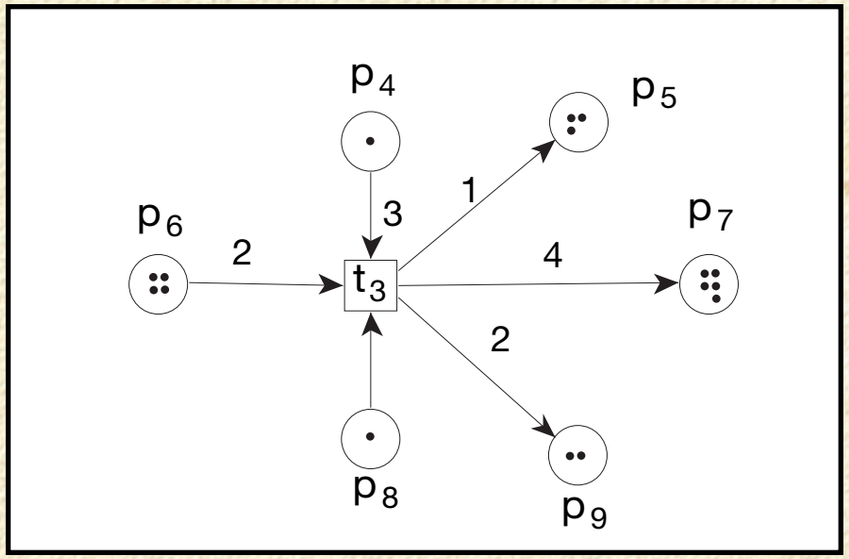
P/T-net

Definition 4.1.2. *A* place/transition net *(P/T net) is defined by a tuple* $\mathcal{N} = \langle P, T, F, W \rangle$, *where*

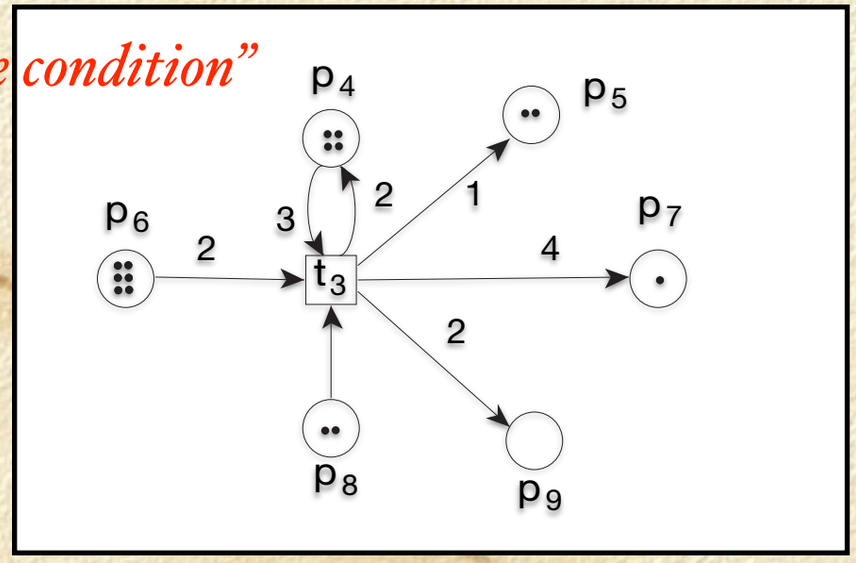
- $\langle P, T, F \rangle$ *is a net (Definition 2.2.1) with finite sets* P *and* T, *and*
- $W : F \to \mathbb{N} \backslash \{0\}$ *is a function (*weight function*).*

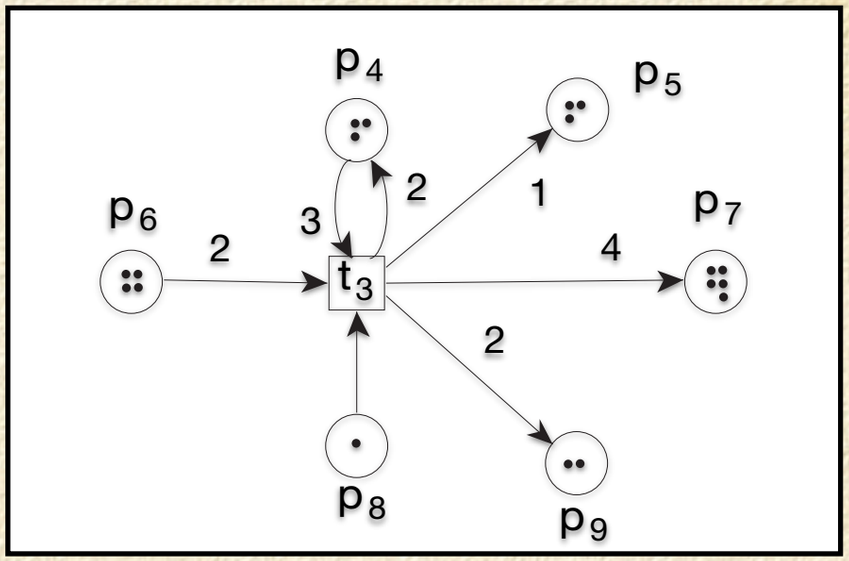
transition rule for *Place/Transition-Nets*





"side condition"





- *the state of a P/T- net: a marking*
  - *represented as a mapping:*

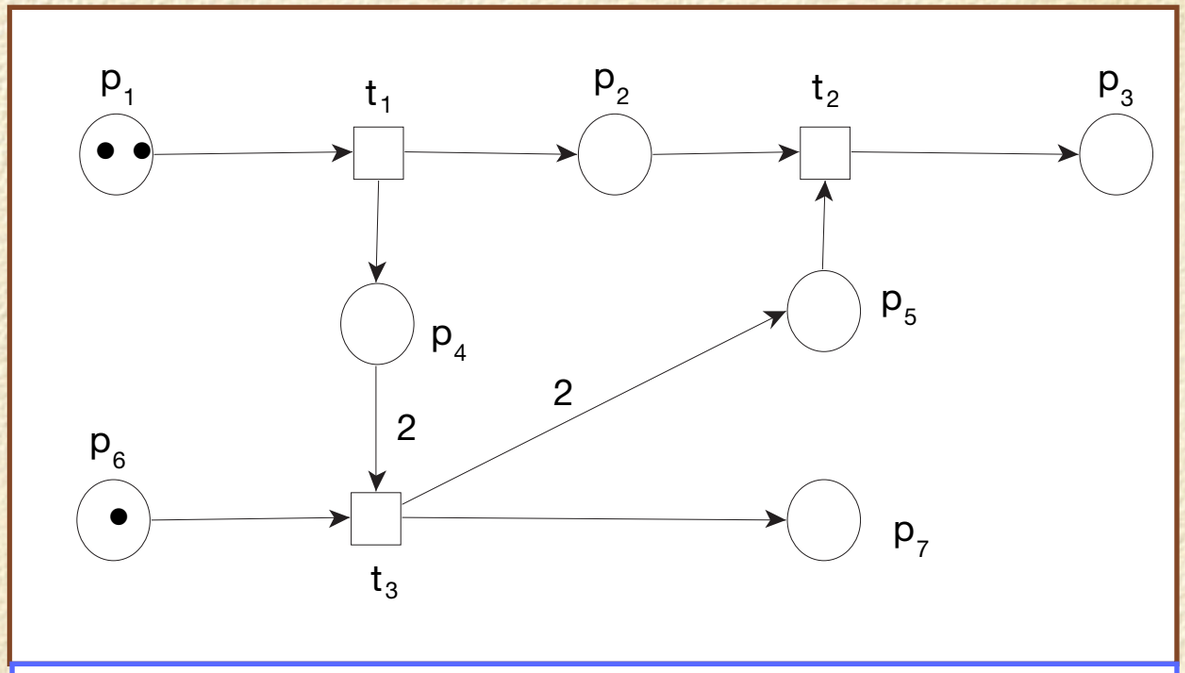
$$\mathbf{m}(p_1) = 2, \mathbf{m}(p_6) = 1, \mathbf{m}(p_i) = 0 \text{ für } i \in \{p_2, p_3, p_4, p_5, p_7\}$$

  - *represented as a vector:*

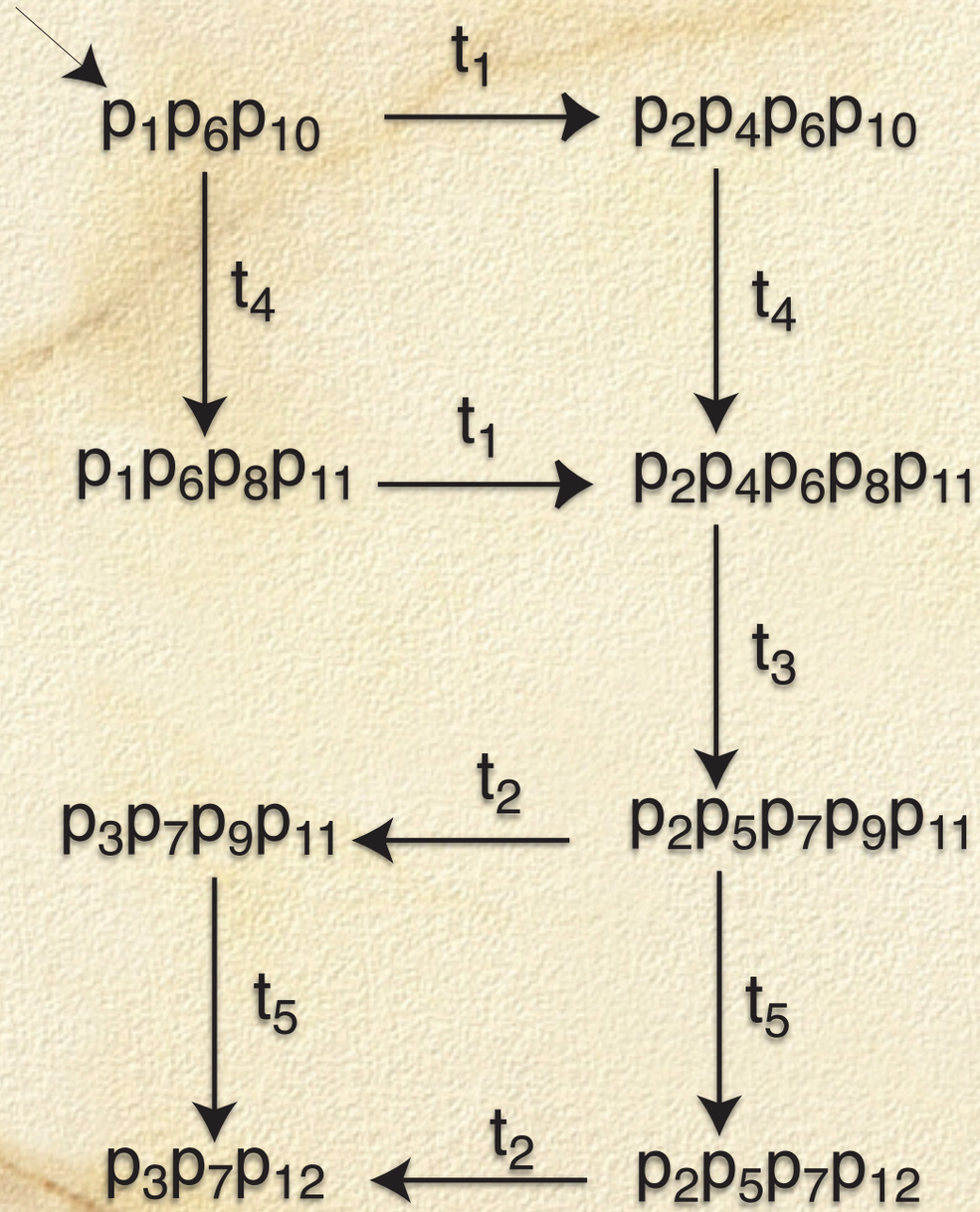
$$\begin{array}{c} p_1 \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ p_7 \end{array} \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

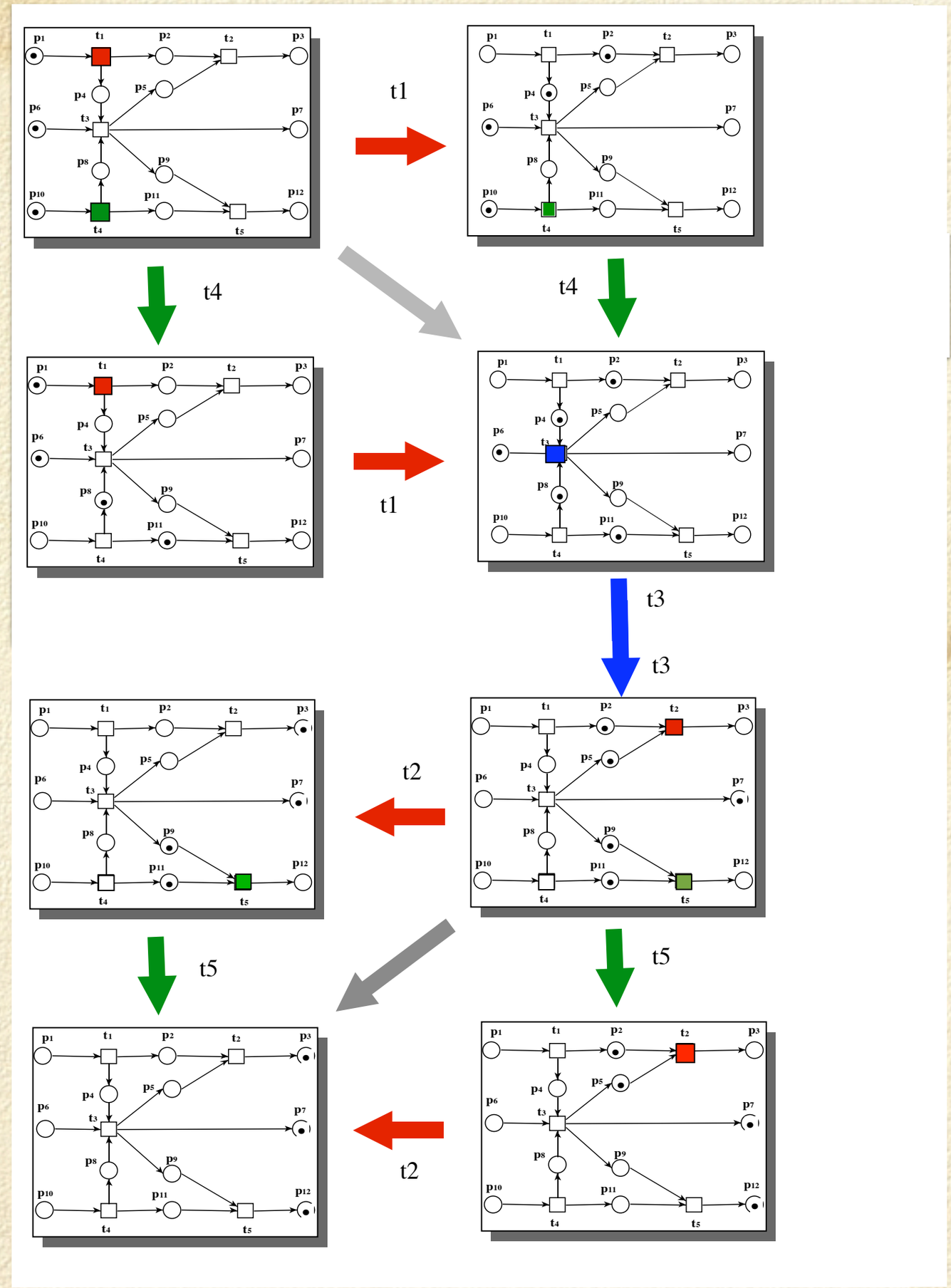
  - *represented as a multiset:*

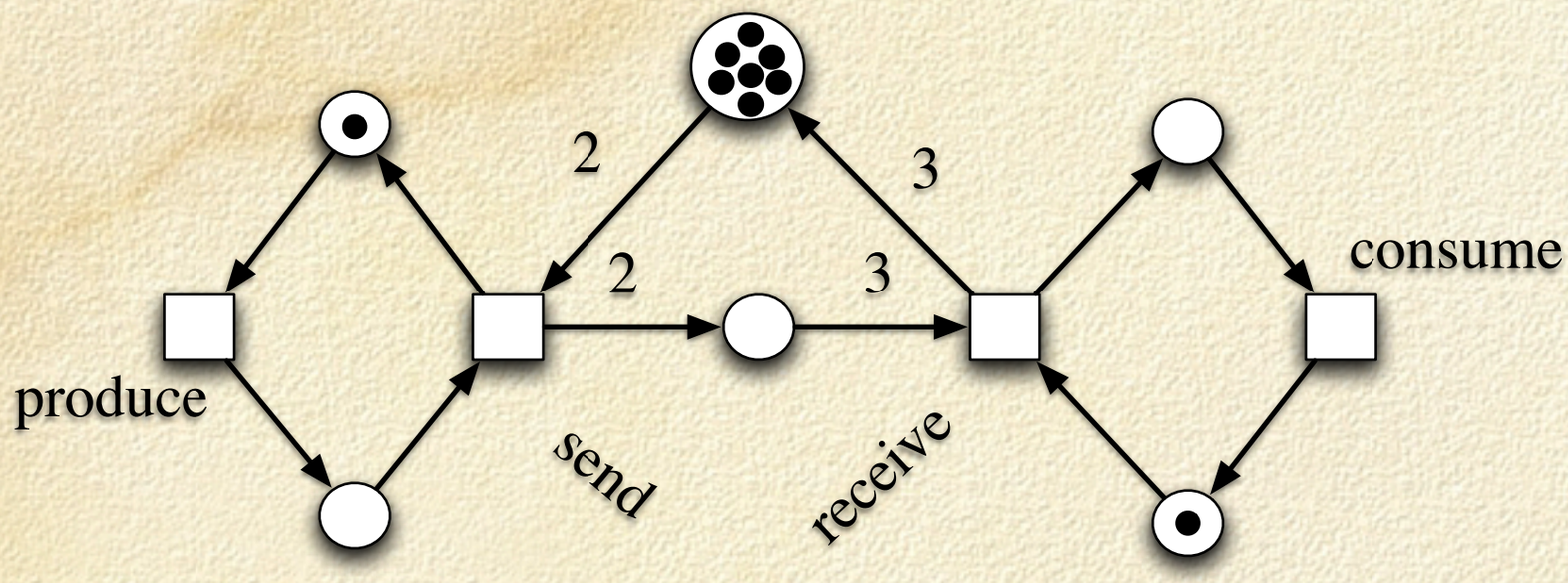
$$2'p_1 + 1'p_6$$



P/T-net


$p_1p_6p_{10}$
t_1
$p_2p_4p_6p_{10}$
t_4
t_4
$p_1p_6p_8p_{11}$
t_1
$p_2p_4p_6p_8p_{11}$
t_3
$p_3p_7p_9p_{11}$
t_2
$p_2p_5p_7p_9p_{11}$
t_5
t_5
$p_3p_7p_{12}$
t_2
$p_2p_5p_7p_{12}$


t1
t4
t4
t1
t3
t3
t2
t5
t5
t2

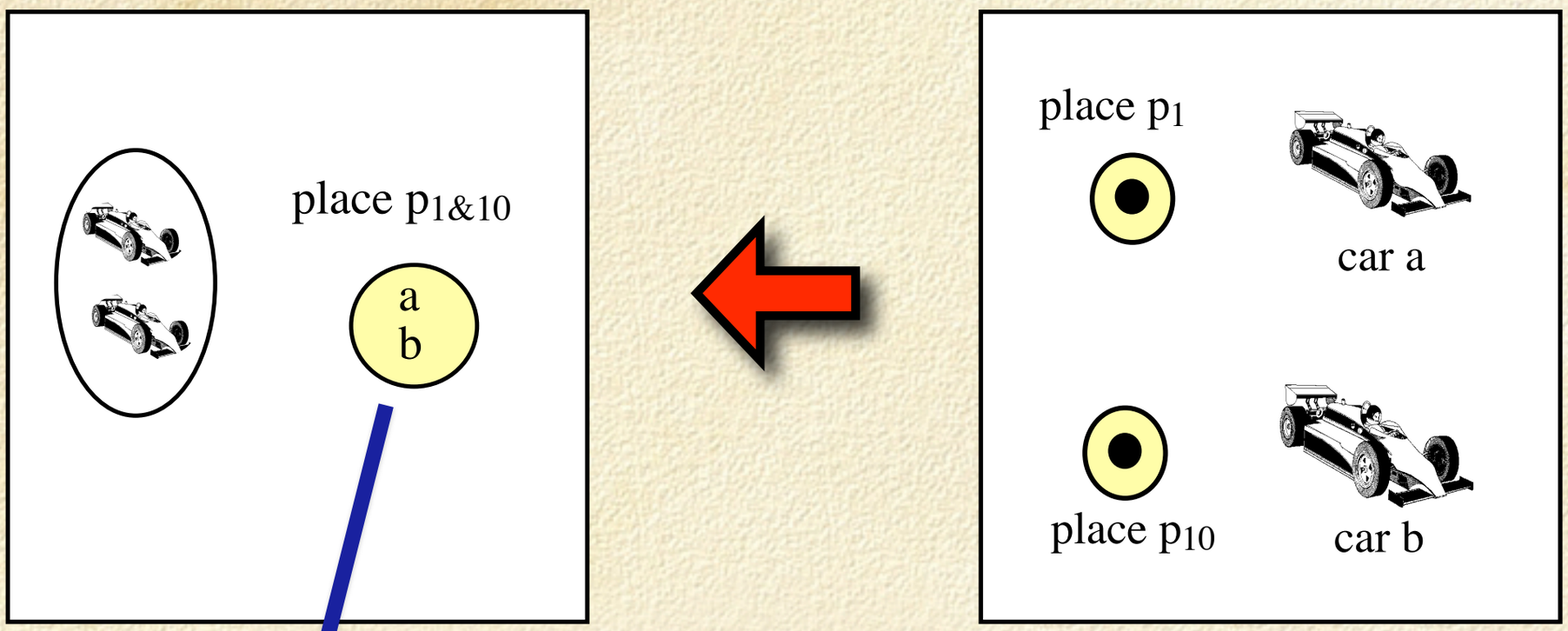


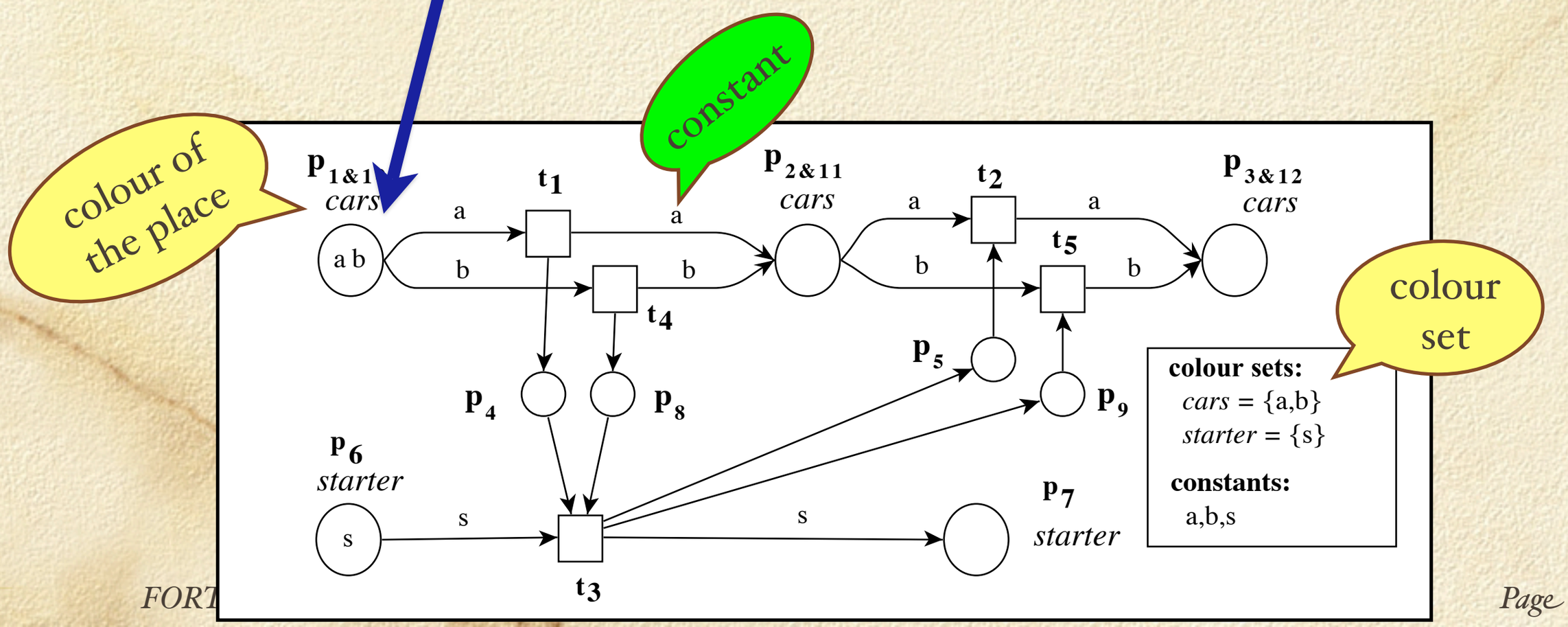
producer consumer system

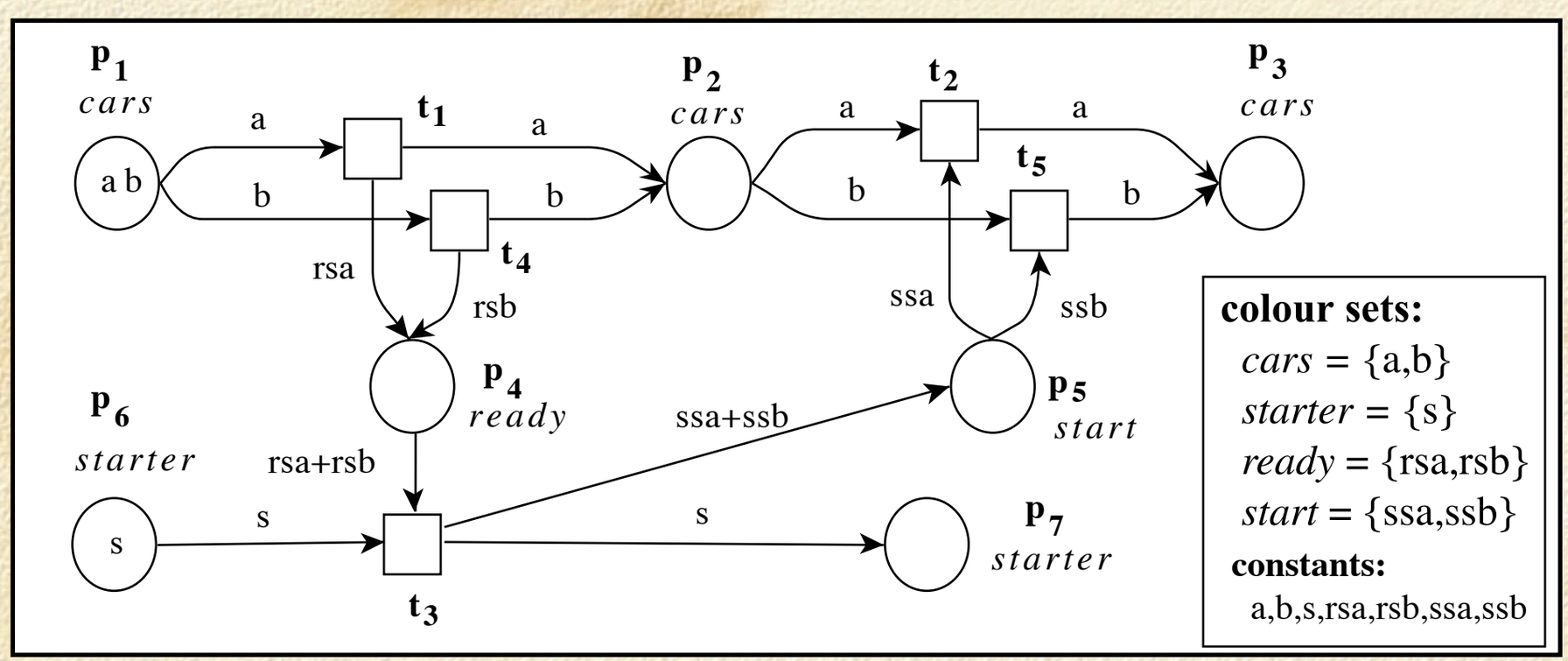
sender receiver system

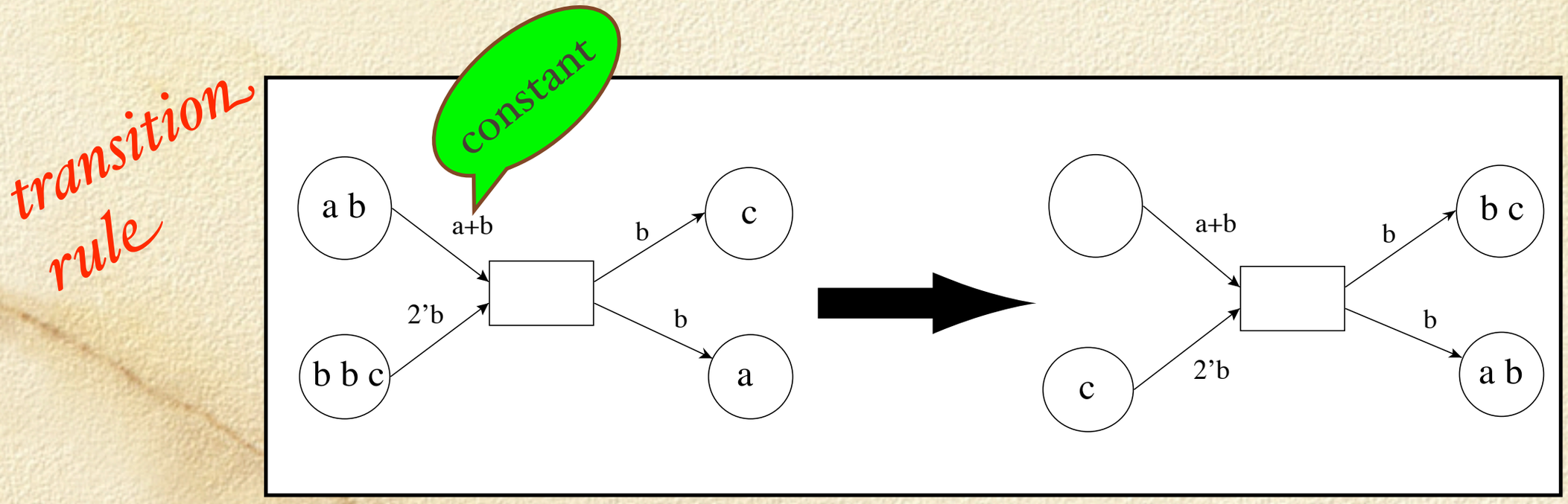
bounded buffer system

arc - constant nets





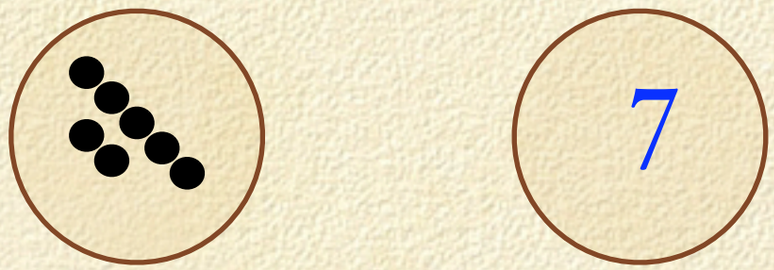

p1
cars
a b
a
b
t1
t4
a
b
p2
cars
a
b
t2
t5
a
b
p3
cars
rsa
rsb
ssa
ssb
p4
ready
p5
start
p6
starter
s
rsa+rsb
ssa+ssb
s
s
t3
p7
starter
colour sets:
cars = {a,b}
starter = {s}
ready = {rsa,rsb}
start = {ssa,ssb}
constants:
a,b,s,rsa,rsb,ssa,ssb


transition rule
constant
a b
a+b
2'b
b b c
b
b
c
a
a+b
2'b
c
b
b
b c
a b

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

P/T - net

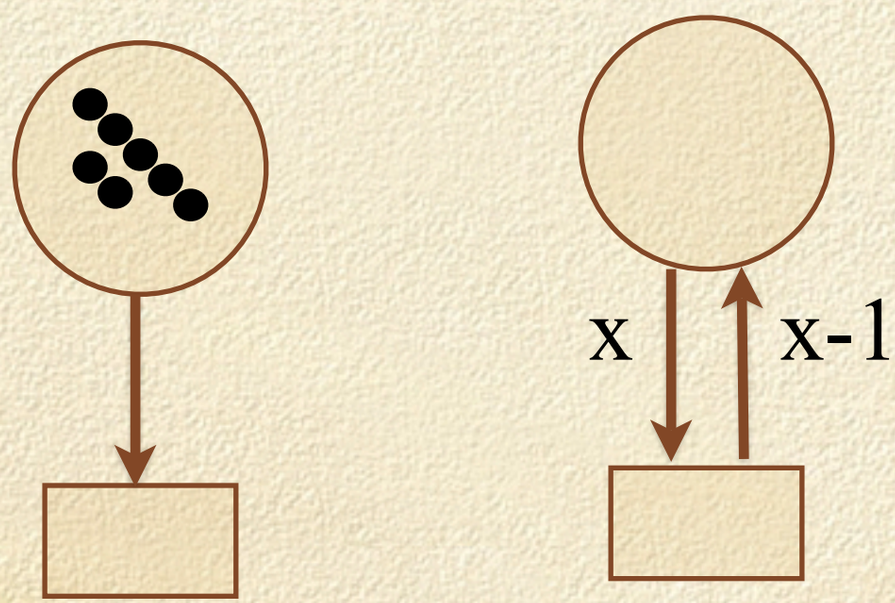
arc-constant net



colour set: {●}

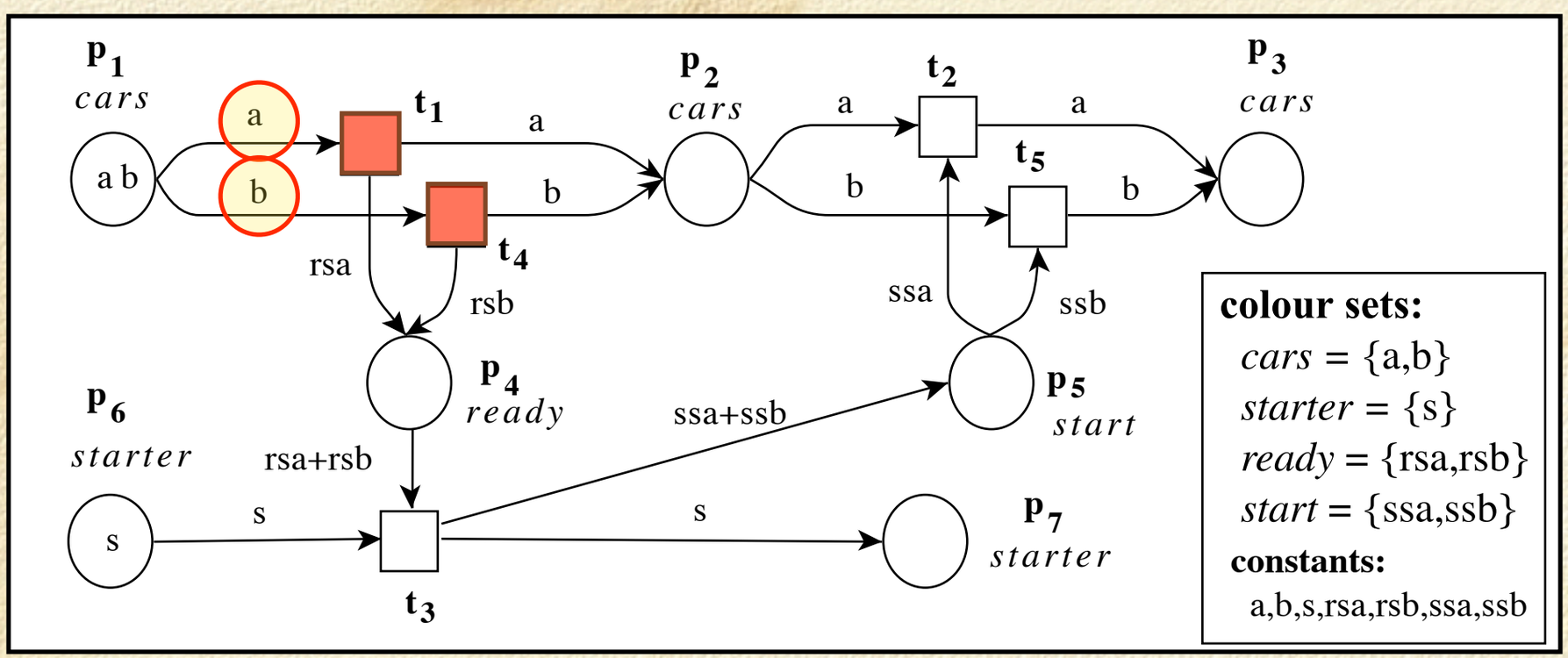
integer

counting down

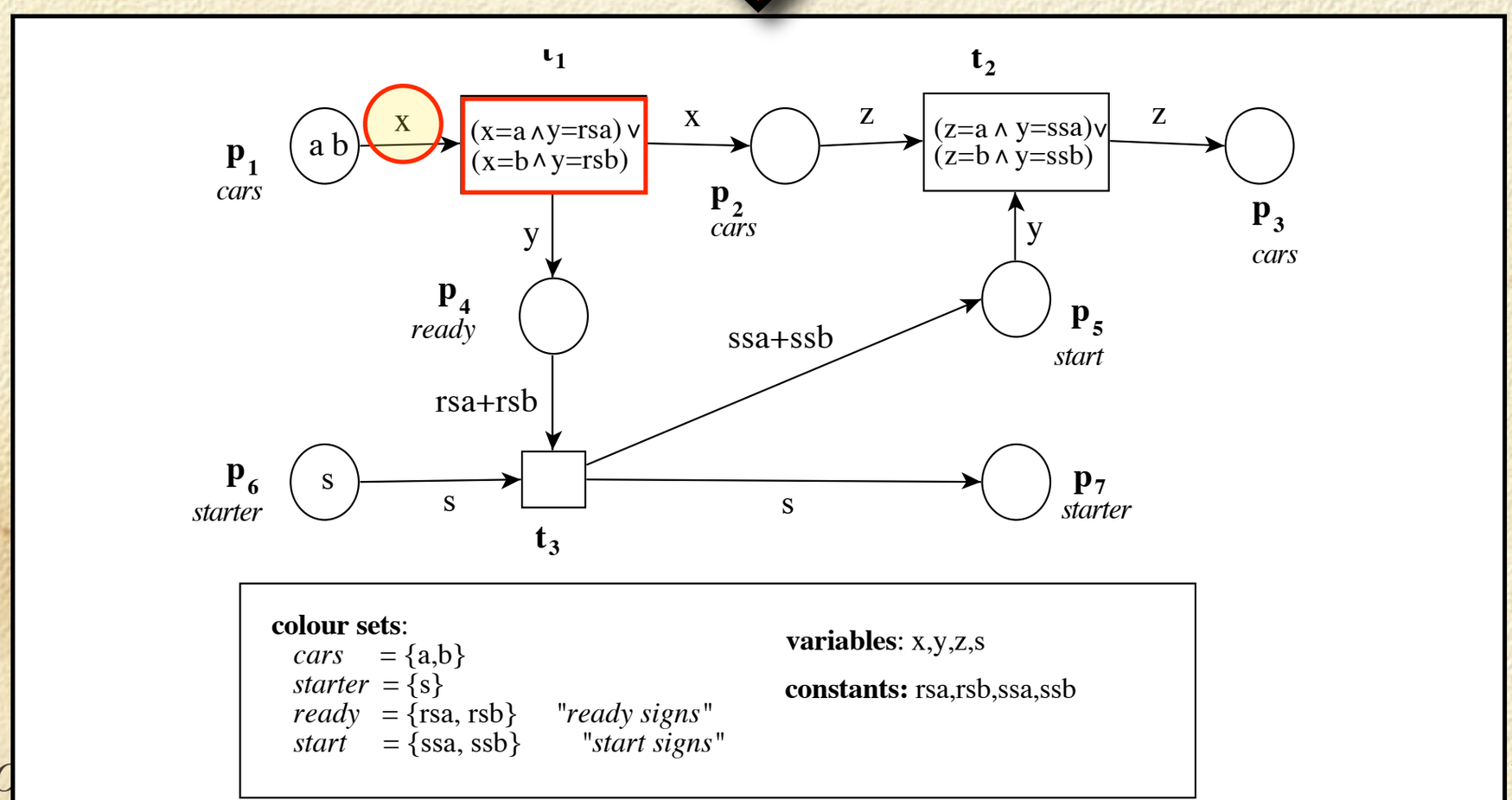


coloured net

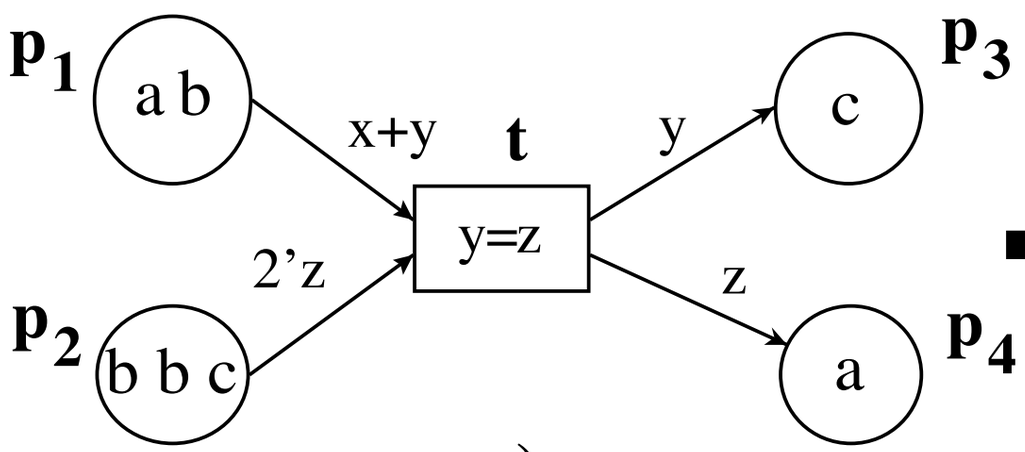
arc - constant net



coloured net



transition rule of coloured nets



a)

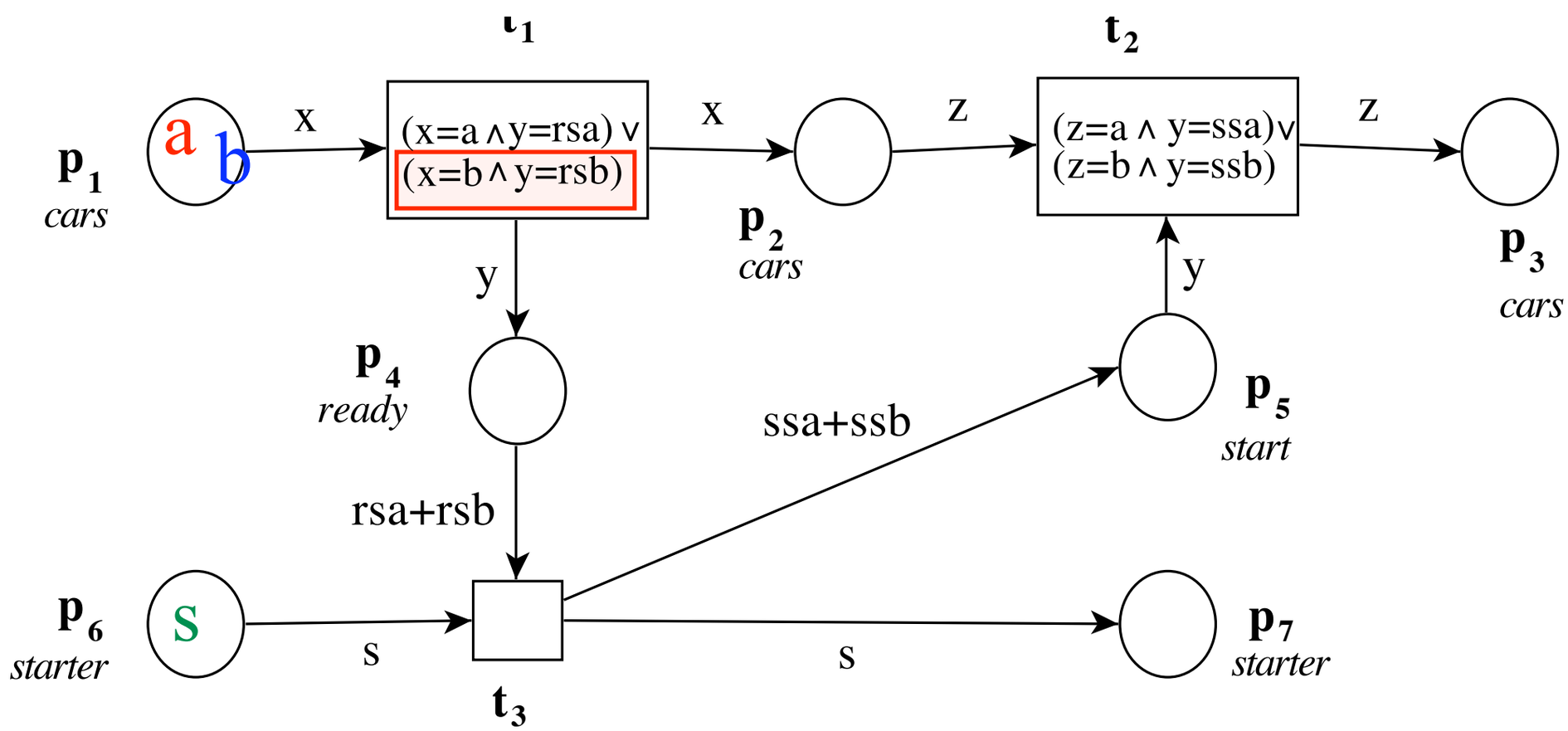
binding of (local) variables

b)

arc-constant net

transition rule of arc-constant net

coloured net



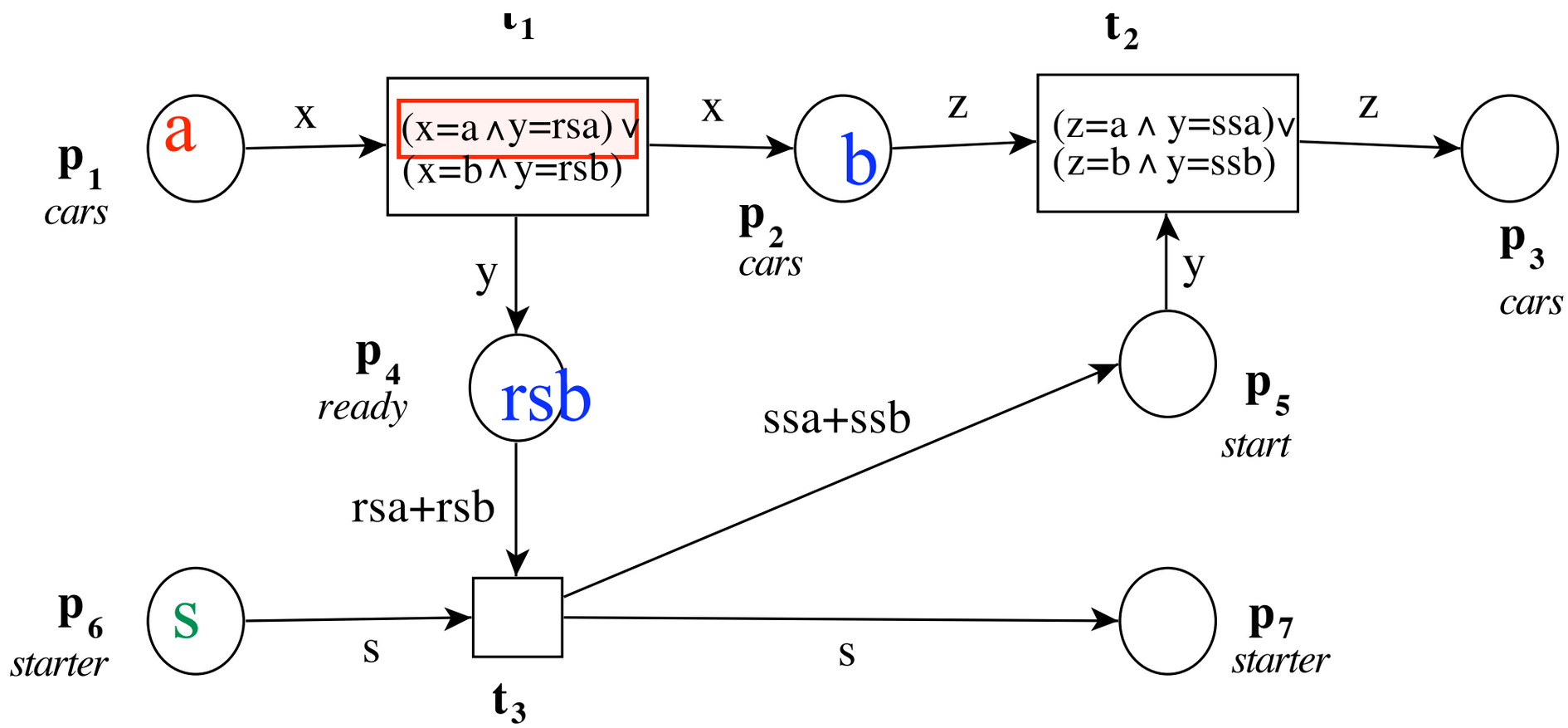
colour sets:

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

coloured net



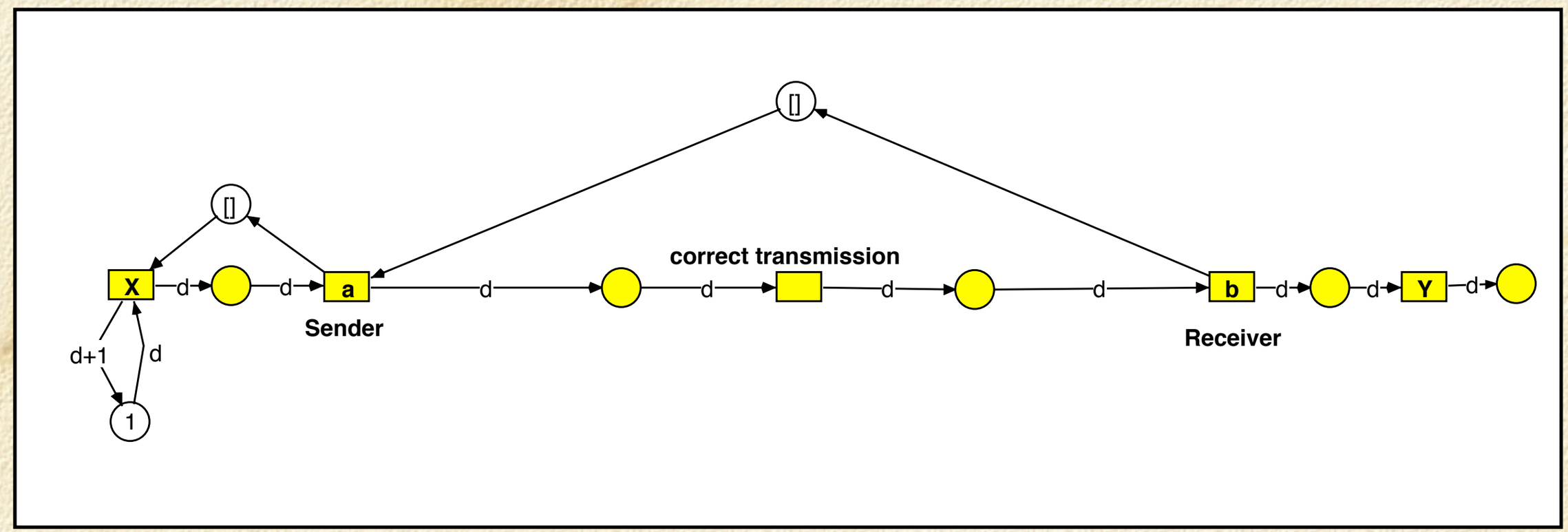
colour sets:

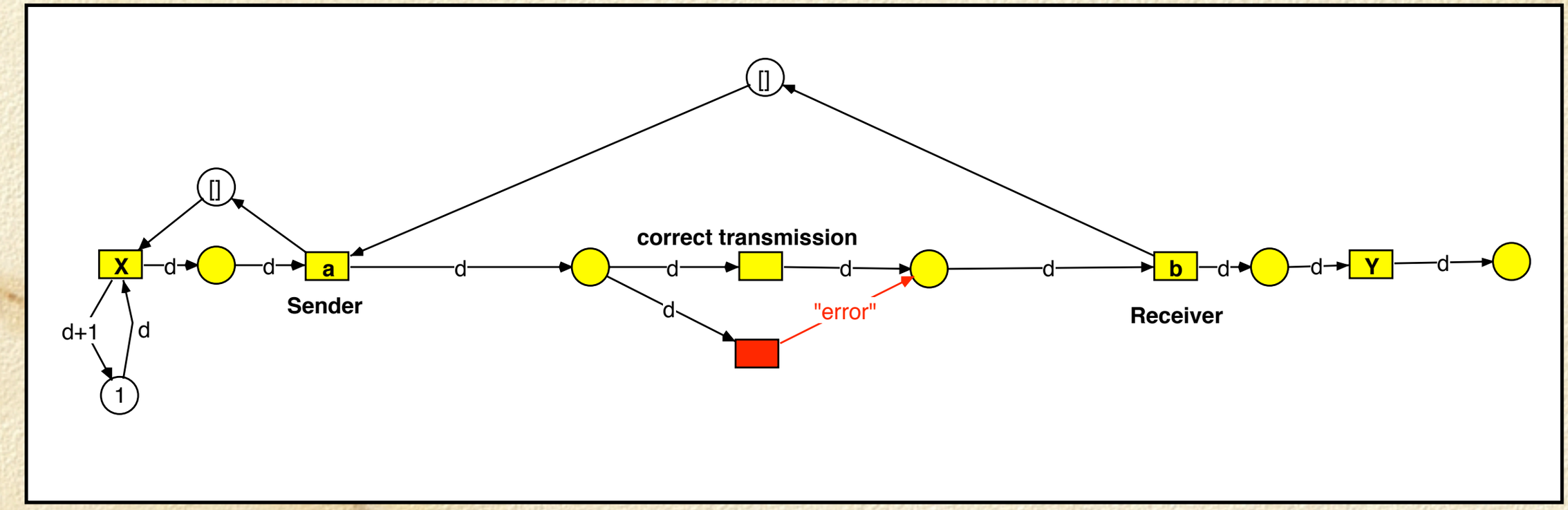
cars = {a,b}
starter = {s}
ready = {rsa, rsb} "*ready signs*"
start = {ssa, ssb} "*start signs*"

variables: x,y,z,s

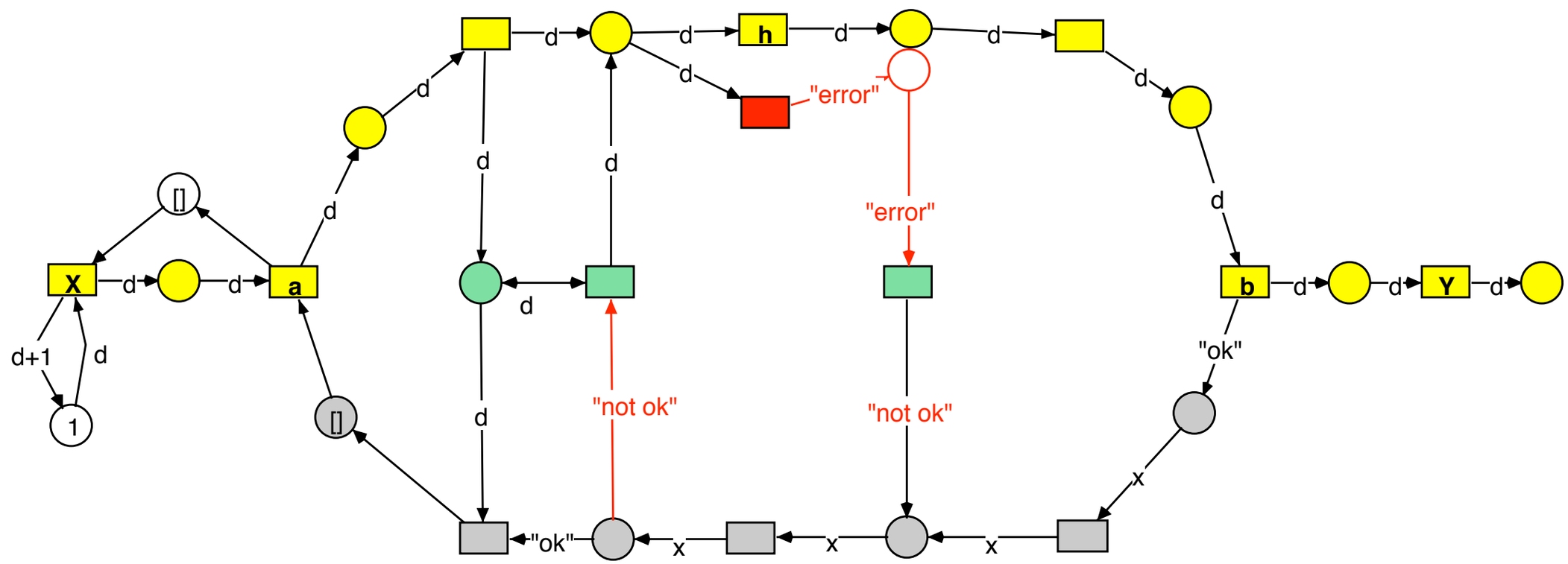
constants: rsa,rsb,ssa,ssb

case: the Alternating Bit Protocol

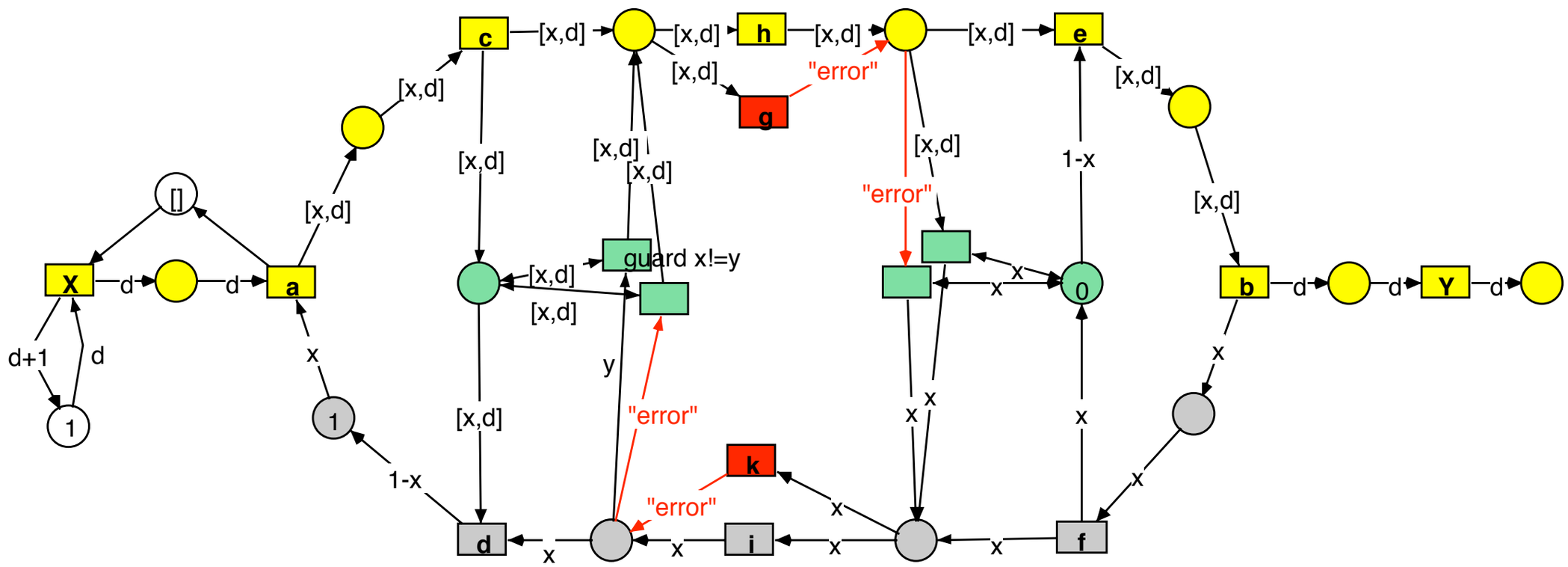


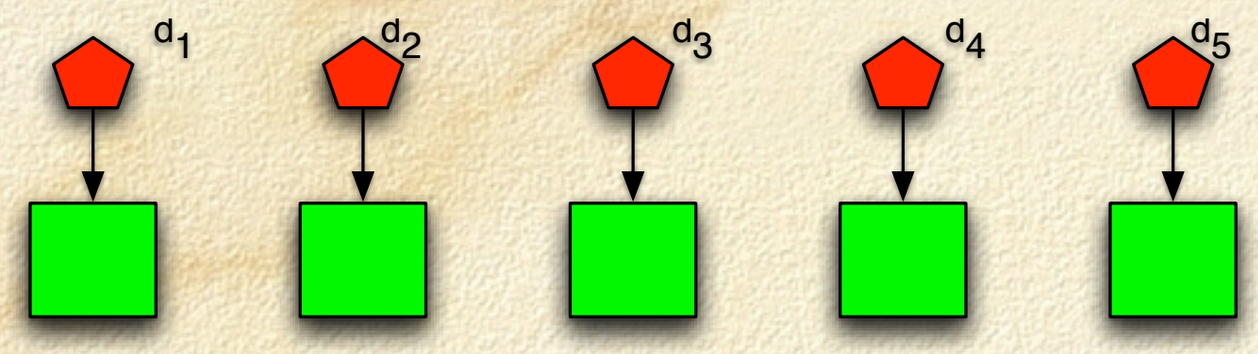


case: the Alternating Bit Protocol



case: the Alternating Bit Protocol





The data base managers

States of managers: *inactive, waiting, performing*

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

States of channels: *unused, sent, received, acknowledged*

mutual exclusion: *exclusion*

colours:

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

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

$$E = \{e\}$$

Variables: $Var = \{e, r, s\}$

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

Functions:

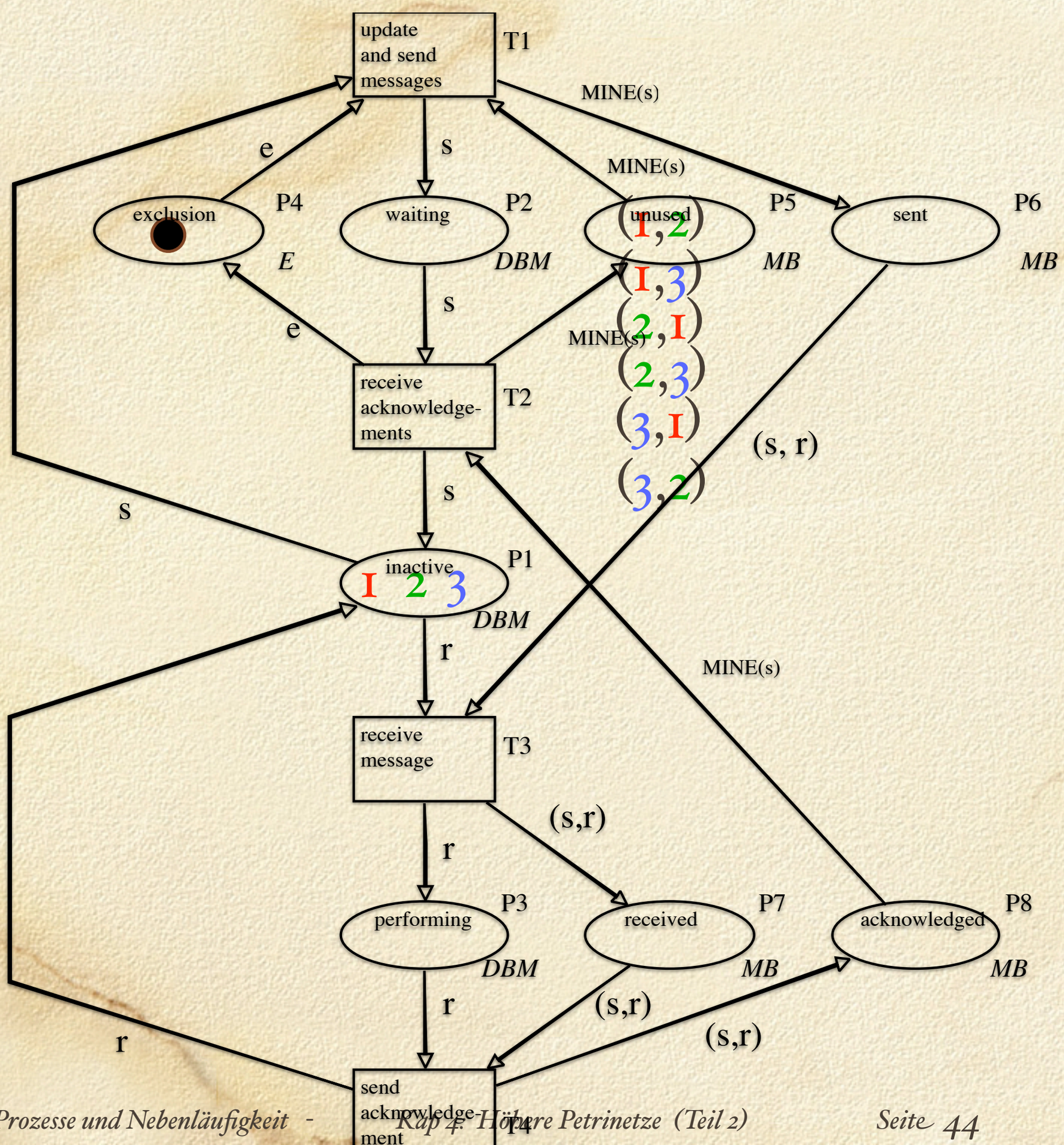
$MINE : DBM \to Bag(MS)$ $\qquad MINE(s) := \sum_{r \neq s} (s, r)$

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

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

Initial marking:

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


update
and send
messages
T1
MINE(s)
e
s
MINE(s)
exclusion
P4
waiting
P2
unused
P5
sent
P6
E
DBM
MB
MB
(1,2)
(1,3)
(2,1)
(2,3)
(3,1)
(3,2)
s
e
MINE(s)
receive
acknowledge-
ments
T2
(s, r)
s
s
inactive
P1
1 2 3
DBM
r
MINE(s)
receive
message
T3
(s,r)
r
performing
P3
received
P7
acknowledged
P8
DBM
MB
MB
r
(s,r)
(s,r)
r
send
acknowledge-
ment
T4

End of Part I

Content of Part II:

Petri nets as objects
Petrinets as mobile agents
Software development