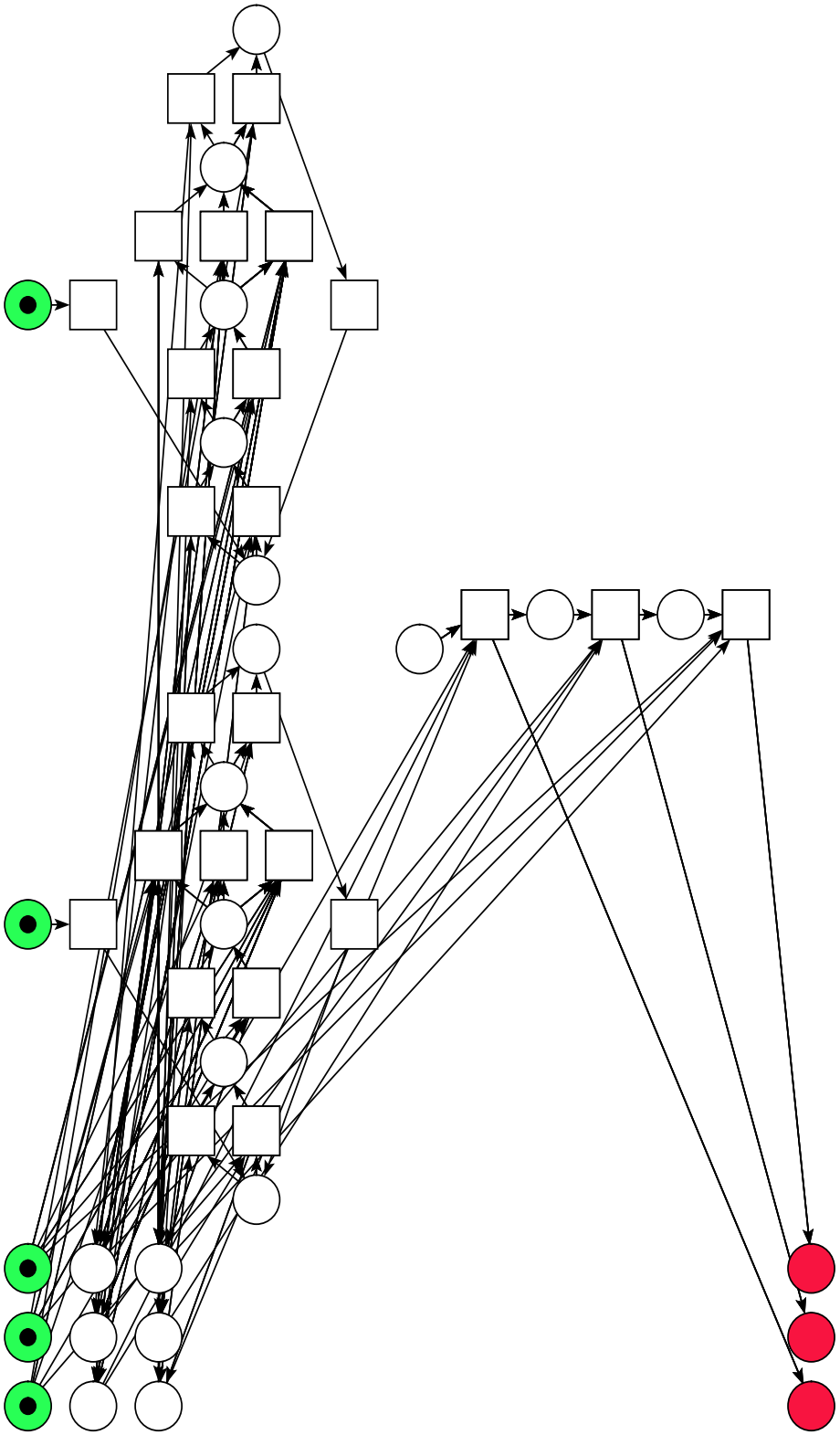
1 High Level Nets

- P/T nets are not applicable with large systems:

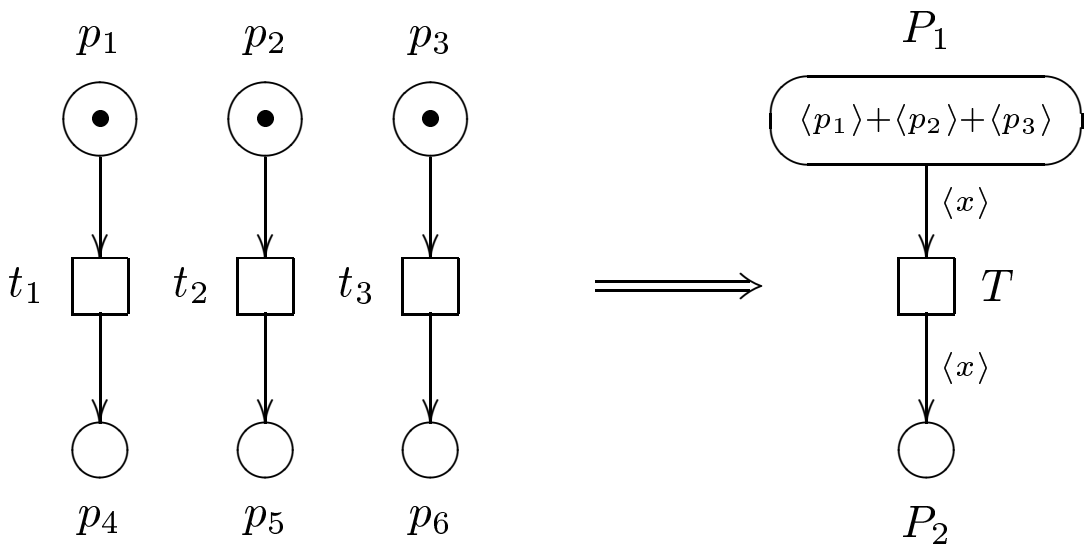


- A more compact notation is needed:

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Introduction

- P/T nets are problematic with something else, too: Identity was not distinguishable in the multi-processor example.
- How about replacing indentityless tokens in places with information?



- t_1, t_2, t_3 are 'instances' of the functionality T for different processes.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

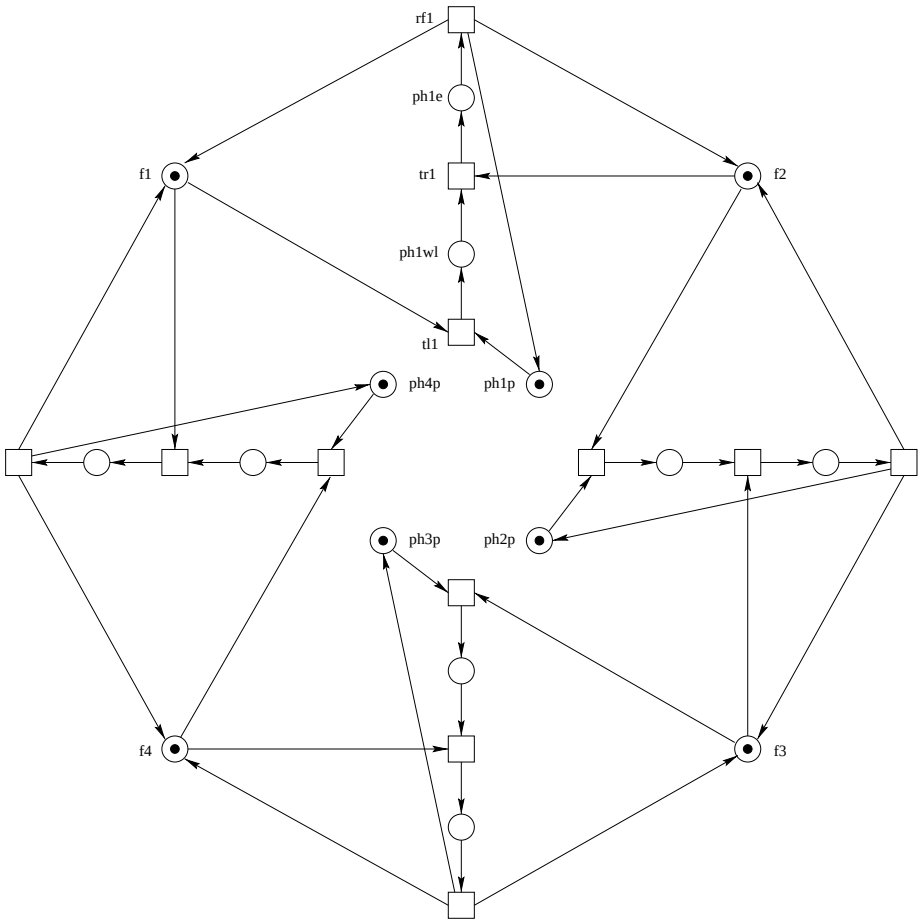
Introduction

- Dining philosophers:
  This is a classical example about distributed resources. Five philosophers are dining at a table, having two modes: thinking or eating. They are eating spaghetti which requires two forks. However, there is only one shared fork between two plates. An added difficulty is that the philosophers cannot grab both of the forks within their reach at the same time but only one fork first, then the second.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Introduction

- A solution depicted as a 1-bounded P/T net:

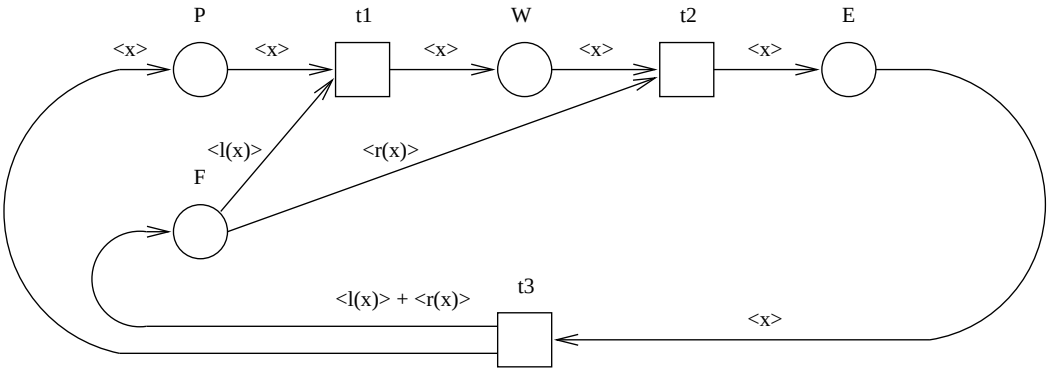


- There is a deadlock in the net but it is irrelevant in this consideration.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Introduction

- The high level net is constructed as follows:
  - The philosophers and forks are modelled as tokens, and
  - Arc expressions are used to depict which philosophers eat with which forks.

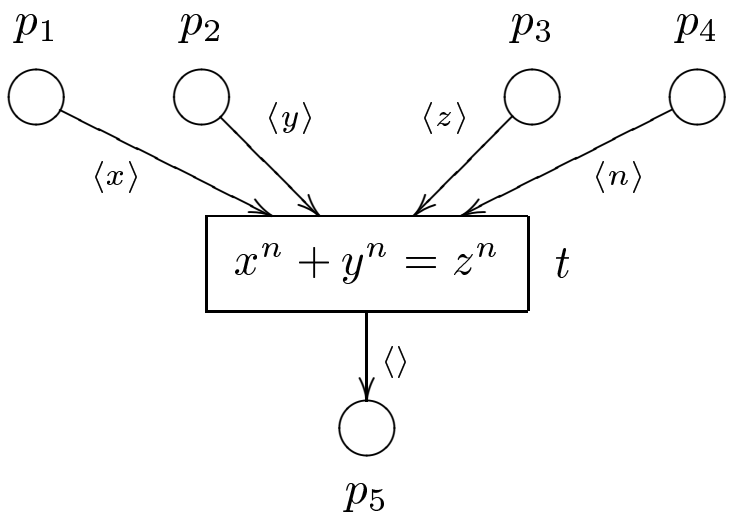


- Initial marking:
  $P \mapsto \{\langle ph_1 \rangle + \langle ph_2 \rangle + \langle ph_3 \rangle + \langle ph_4 \rangle + \langle ph_5 \rangle\}$,
  $F \mapsto \{\langle f_1 \rangle + \langle f_2 \rangle + \langle f_3 \rangle + \langle f_4 \rangle + \langle f_5 \rangle\}$
- Functions: $l(ph_i) = f_i$, $r(ph_i) = f_{(i \bmod 5)+1}$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Introduction

- Now the *annotations* written on the pre-arcs indicate what tuples need to be found in a place so as to enable concession of a transition.
- Moreover, it is possible to attribute the transition with conditions which give further requirements for the tuples:



- In order to be able to interpret the annotation, it has to be *fixed* with a structure.
- Within the scope of this course it is assumed the the structure in question is that of integers:

$$Z = \langle \mathbb{Z}; \{+, \times, \dots\}, \{=, \neq, \leq, \geq, \dots\} \rangle.$$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

- As annotations we will have:

  L_f the set of well-formed formulas (truth-valued expressions)

  $$\text{Example:} \quad (x < 5 \wedge x > 1), x = y + 3, \exists z : P(x, z), \ldots$$

  L_t the set of well-formed terms (integer-valued expressions)

  $$\text{Esim.:} \quad f(x), x + 5, \ldots$$

- In the end, we need:
  - *tuple*: $\langle x, y \rangle, \langle x + 2, y \rangle, \langle x + y, 3 \rangle, \langle x \rangle, \langle \rangle$
  - *formal sum*: a means to represent a finite bag as an expression with positive integer multipliers:

    $$2\langle x \rangle + 3\langle y \rangle \equiv \{\langle x \rangle + \langle x \rangle + \langle y \rangle + \langle y \rangle + \langle y \rangle\}$$

  - $L_\Sigma(A)$ is the set of all formal sums of the set A.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T Net

Definition 1 Let L be a language of a first order predicate calculus. A *Predicate/Transition net* (Pr/T Net) is the 3-tuple $N_L = \langle N, A, M_0 \rangle$, where

1. $N = \langle S, T; F \rangle$ is the so called *underlying* net,
2. $A = \langle A_N, A_S, A_T, A_F \rangle$ is its *annotation*, where
   (a) A_N is a model of a first order predicate calculus L (here: the set of integers Z), the so called *support* of N_L,
   (b) $A_S : S \to \mathbb{N}$ associates an *arity* with each place, the size (width) of tuples found in the different markings,
   (c) $A_T : T \to L_f$ associates a well-formed formula, the so called *firing condition, transition selector* with each transition; however, the free variables in this expression have to appear in one the formal sums of the arc annotations A_F, and

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

(d) A_F associates in each arc $\langle a, a' \rangle$ a *formal sum* of tuples of terms; all tuples in the expression have to be of the same length which has to match with the arity of the place connected with the arc;

3. $M_0 : S \to L_\Sigma([A_N]^*)$ is the initial marking of the Pr/T net N_L, and it fixes each place s with some amount of the elements of the carrier set of the structure A_N; the size of the tuples will have to match with the arity of the place $A_s(s)$.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

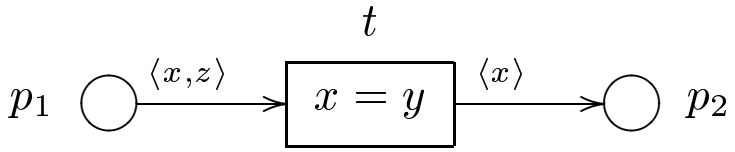
Pr/T net

- $\langle S, T; F \rangle =$
  $\langle \{p_1, p_2, p_3\}, \{t\}, \{\langle p_1, t \rangle, \langle p_2, t \rangle, \langle t, p_3 \rangle\} \rangle$
- $A_N = Z$
- $A_S = \{p_1 \mapsto 1, p_2 \mapsto 1, p_3 \mapsto 1\}$
- $A_T = \{t \mapsto z = x + 1\}$
- $A_F = \{\langle p_1, t \rangle \mapsto \langle x \rangle, \langle p_2, t \rangle \mapsto \langle z \rangle, \langle t, p_3 \rangle \mapsto \langle x \rangle\}$
- $M_0 = \{p_1 \mapsto \langle 1 \rangle, p_2 \mapsto \langle 2 \rangle\}$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T net

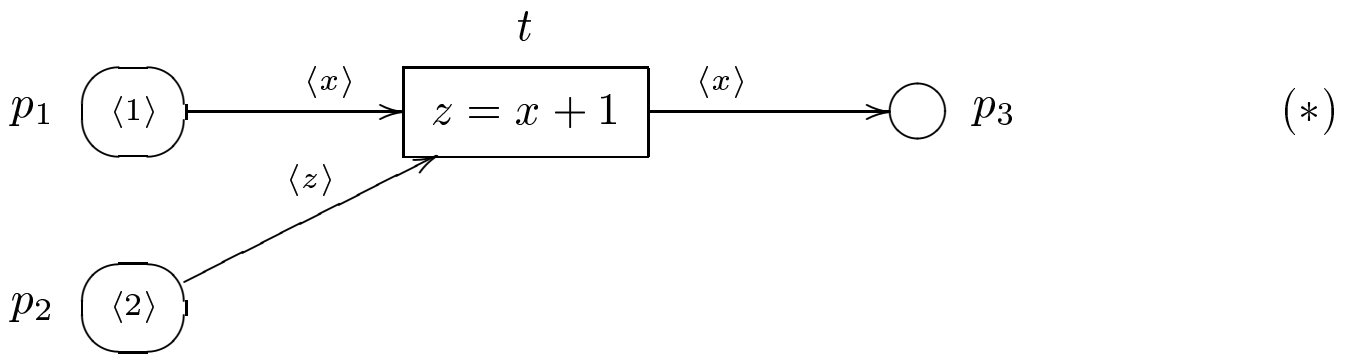
- Please note:
  - Point 2c



    disallowed
  - No capacity restrictions.
  - Arity is often left undefined since it can be deduced from the form of the arc expressions.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Firing rule



- The P/T net firing rule has to be modified so that instead of "unnamed" tokens we refer to the elements of the structure A_N.
- Intuition:
  - Consider all possible valuations (assignments) for variables: $x \mapsto a, z \mapsto b$, where $a, b \in \mathbb{Z}$).
  - if the tuples determined by the assignment are found in the pre-places then the transition is in concession (and can fire).
  - In the example the transition t is in concession since the tuples $(\langle 1 \rangle, \langle 2 \rangle)$ corresponding with the assignment $x \mapsto 1, z \mapsto 2$ are found in places p_1, and p_2.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Firing rule

Definition 2 Let N_L be a Pr/T net and t one of its transitions.

- The *index* of transition t, denoted $I(t)$, is the set of variables found in the arc expressions of the incoming or outgoing arcs.
- Function $\sigma_t : I(t) \rightarrow \mathbb{Z}$ is the *valuation* of the transition t.
- A valuation is *feasible* iff $Z \models (A_T(t)) : \sigma_t$, in other words, the expression obtained by replacing all the variables with the integers in the valuation σ is true.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Firing rule

- In the example $(*)$:

$$\begin{aligned} I(t) &= \{x, z\}, \\ \sigma_t &= \{x \mapsto 2, z \mapsto 3\}, \\ \sigma'_t &= \{x \mapsto 2, z \mapsto 1\}, \end{aligned}$$

- σ_t is feasible because $Z \models (z = x + 1) : \sigma_t$ or $Z \models 3 = 2 + 1$ but
- σ'_t is not feasible because $Z \not\models (z = x + 1) : \sigma'_t$ or $Z \not\models 1 = 2 + 1$.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Firing rule

Definition 3 Let N_L be a Pr/T net, M its marking, and t its transition, and σ the valuation of t. The σ-*instance* of the transition t is in concession (denoted $M[t:\sigma>$) in the marking M iff

1. σ is feasible, and
2. if sFt then $M(s) \geq A_F((s,t)):\sigma$, where $\geq$ denotes the ordering relation defined point-to-point for bags of integers.

- In the example $(*)$:
  - $M(p_1) = \langle 1 \rangle, M(p_2) = \langle 2 \rangle$,
  - and $A_F((p_1,t)):\sigma = \langle 1 \rangle$ and $A_F((p_2,t)):\sigma = \langle 2 \rangle$.
  - Now both $M(p_1) \geq A_F((p_1,t)):\sigma = \langle 1 \rangle$ and $M(p_2) \geq A_F((p_2,t)):\sigma = \langle 2 \rangle$,
  - or, in other words, $M[t:\sigma>$.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Firing rule

Definition 4 Let N_L be a Pr/T net, let M and M' be its markings, t one of its transitions, and σ the valuation of the transition t. The σ-*instance* of the transition t is from the marking M to the marking M' (denoted $M[t:\sigma>M'$), iff

1. $M[t:\sigma>$
2. for all places s of the net N_L the following holds: $M'(s) = M(s) - (A_F((s,t)):\sigma) + (A_F((t,s)):\sigma)$.

- In the example we get:

$$\begin{aligned}
M'(p_1) &= M(p_1) - (A_F((p_1,t)):\sigma) + \\
&\quad (A_F((t,p_1)):\sigma) \\
&= \langle 1\rangle - \langle 1\rangle + \emptyset \\
M'(p_2) &= M(p_2) - (A_F((p_2,t)):\sigma) + \\
&\quad (A_F((t,p_2)):\sigma) \\
&= \langle 2\rangle - \langle 2\rangle + \emptyset \\
M'(p_3) &= M(p_3) - (A_F(<p_3,t>):\sigma) + \\
&\quad (A_F(<t,p_3>):\sigma) \\
&= \emptyset - \emptyset + \langle 1\rangle
\end{aligned}$$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Modelling with Pr/T nets

- In the following we consider different approaches to modelling the AB protocol.
- When modelling with Pr/T nets a choice has to be made:
  1. What part of the system is modelled with a net, and
  2. What part is modelled with annotations.
- Since any P/T net is trivially a Pr/T net, too, our original net was an extreme solution from a Pr/T net point of view: All of the system is modelled with the net.

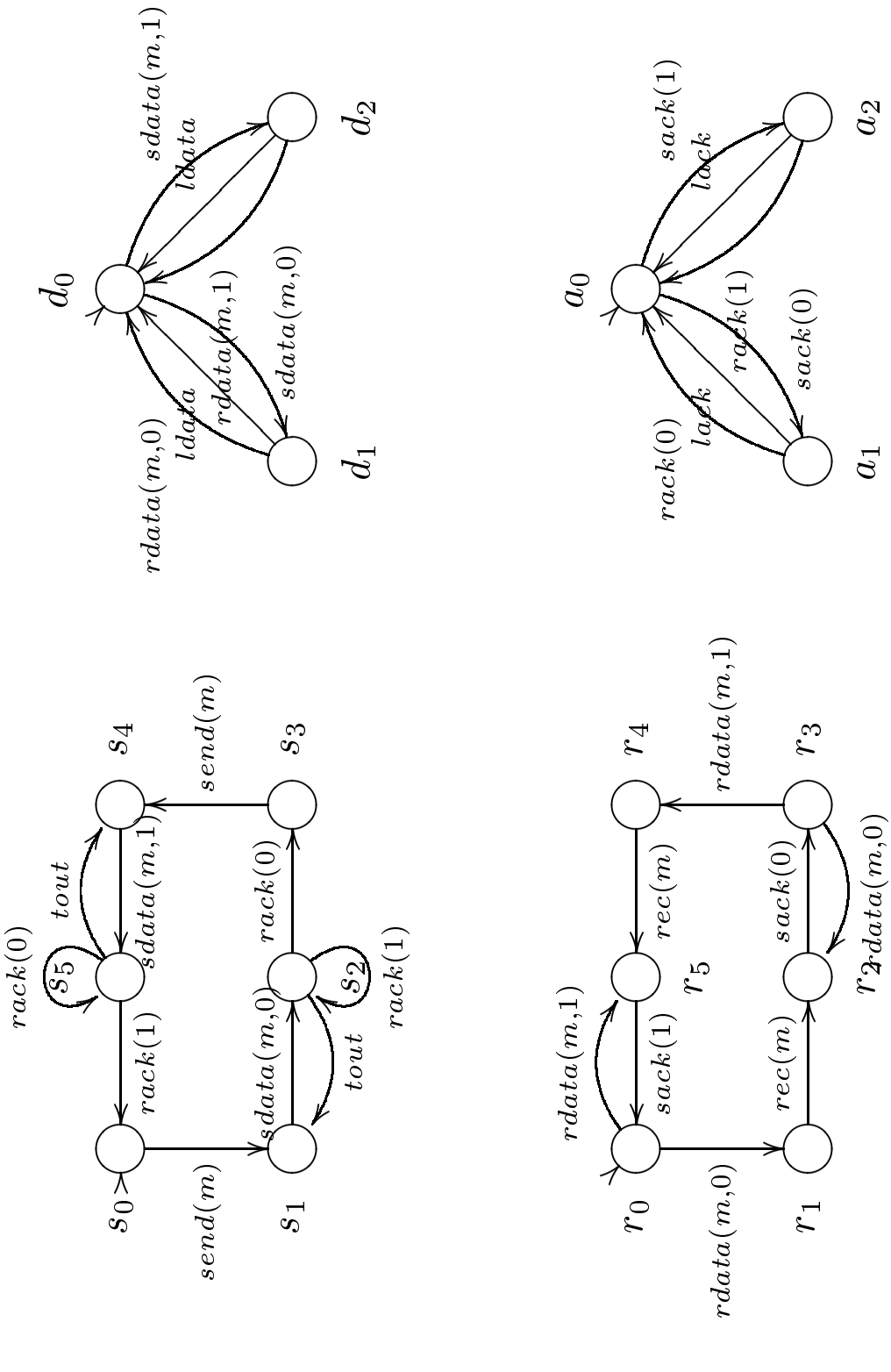
©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

AB protocol (corrected)

- AB is a simple communications protocol.
- Messages are sent tagged with bits 0 or 1.
  - Sender process: Once started up, a message is sent with the bit b.
    - If the message is lost then the process time-outs and resends the message with the bit b.
    - If the process receives the bit b as an acknowledgement then it will send a new message with the bit $\hat{b} = 1 - b$.
    - If the process receives an incorrect bit an acknowledgement it will be forgotten.
  - Receiver process: The process receives the message with the bit b. The bit b is acknowledged.
    - If the process receives a new message with the bit $\hat{b}$ it is acknowledged.
    - If the process receives a new message with the bit b then the handshake is repeated.
  - This means the protocol sends and receives signals alternating the bit with them.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Labelled transition systems (or communicating automata)



©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of Alternating Bit Protocol

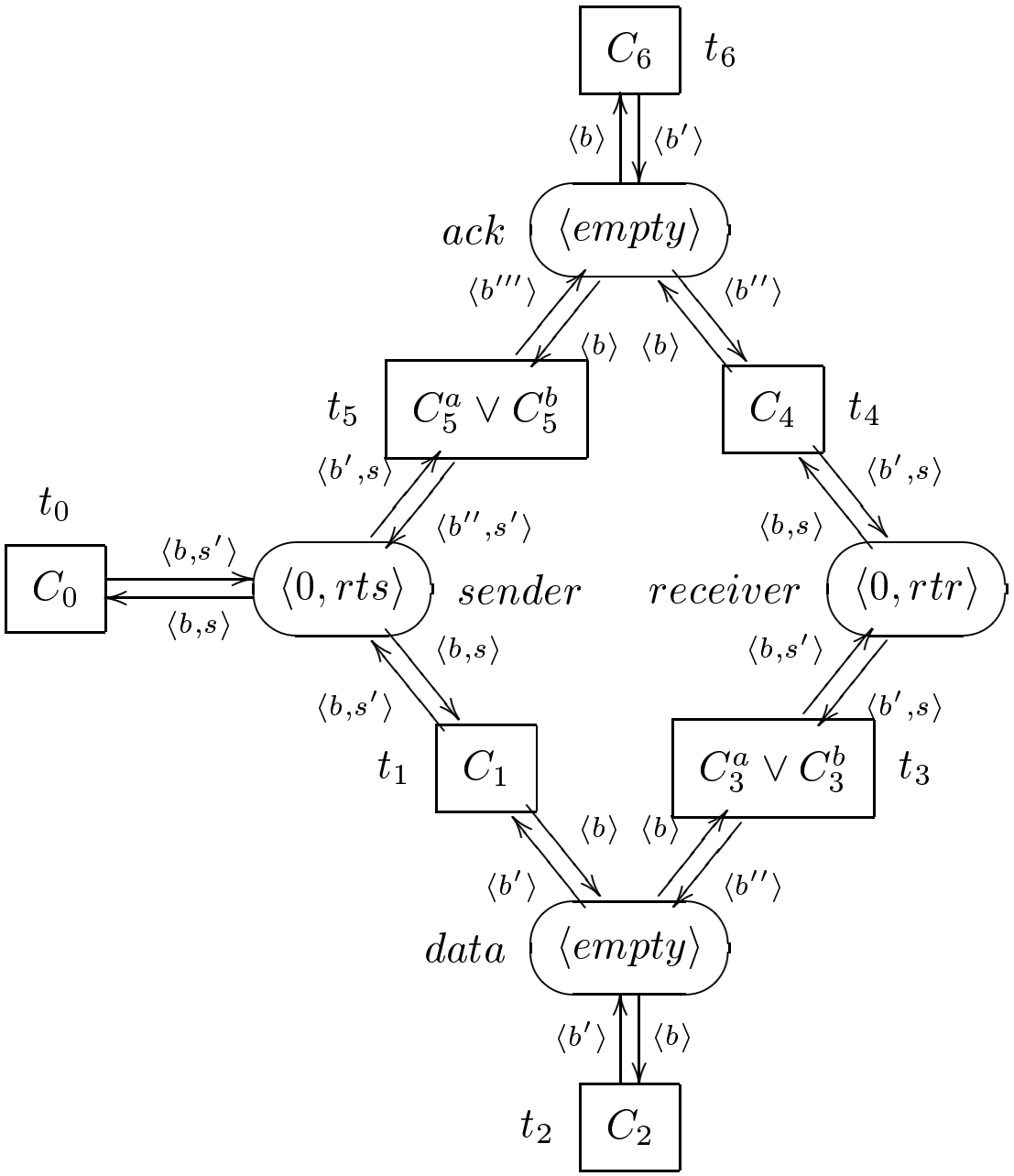
- Some simplifications are made:
  - It is negligible regarding the correctness of the system what data the message actually carries.
  - Equally well, the transitions $send$ and rec can be removed.
  - One place is used for each automaton: the state of the automaton is encoded into a token.

sender			receiver		
$\langle 0, rts \rangle$	$=$	s_1	$\langle 0, rtr \rangle$	$=$	r_0
$\langle 0, wack \rangle$	$=$	s_2	$\langle 0, sack \rangle$	$=$	r_2
$\langle 1, rts \rangle$	$=$	s_4	$\langle 1, rtr \rangle$	$=$	r_3
$\langle 1, wack \rangle$	$=$	s_5	$\langle 1, sack \rangle$	$=$	r_5

  - The channels have three possible states $(0, 1, empty)$. (There are no capacity restrictions for Pr/T nets.)

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP (version 1)



©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP (version 1)

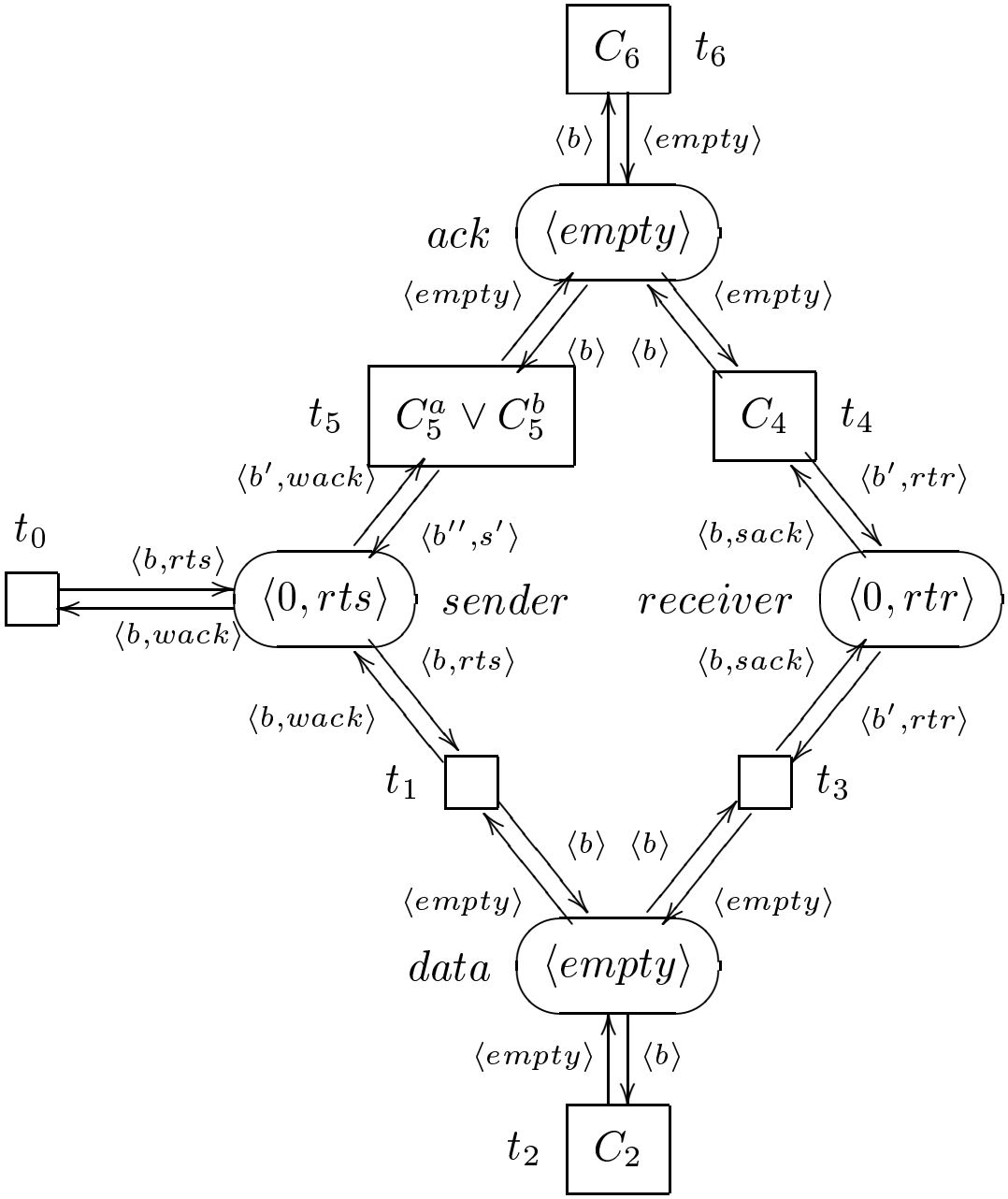
- Where:

$$
\begin{aligned}
\text{tout: } C_0 &\equiv s = wack \wedge s' = rts \\
\text{sdata: } C_1 &\equiv s = rts \wedge b' = empty \wedge s' = wack \\
\text{ldata: } C_2 &\equiv b \neq empty \wedge b' = empty \\
\text{rdata OK: } C_3^a &\equiv s = rtr \wedge b = b' \wedge s' = sack \wedge b'' = empty \\
\text{rdata NOK: } C_3^b &\equiv s = rtr \wedge b = \neg b' \wedge s' = sack \wedge b'' = empty \\
\text{sack: } C_4 &\equiv s = sack \wedge b' = \neg b \wedge s' = rtr \wedge b'' = empty \\
\text{ack OK: } C_5^a &\equiv s = wack \wedge b' = b \wedge s' = rts \wedge b'' = \neg b \wedge b''' = empty \\
\text{ack NOK: } C_5^b &\equiv s = wack \wedge b' = \neg b \wedge s' = wack \wedge b'' = \neg b \wedge b''' = empty \\
\text{lack: } C_6 &\equiv b \neq empty \wedge b' = empty
\end{aligned}
$$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP (version 2)

- The constants are moved to the arcs and the conditions are simplified:



©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP (version 2)

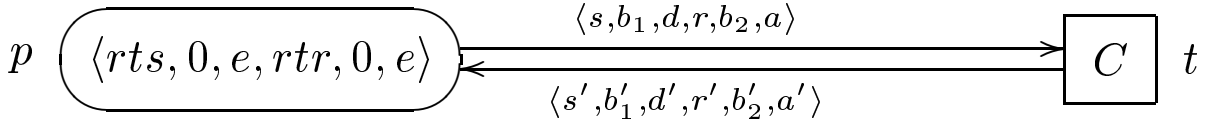
- Where:

$$
\begin{aligned}
C_2 &\equiv b \neq empty \\
C_4 &\equiv b' = \neg b \\
C_5^a &\equiv b' = b \wedge s' = rts \wedge b'' = \neg b \\
C_5^b &\equiv b' = \neg b \wedge s' = wack \wedge b'' = \neg b \\
C_6 &\equiv b \neq empty
\end{aligned}
$$

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP (version 3)

- The transition conditions can be used to encode all of the reachability graph of the ABP:



missä,

	s	b_1	d	r	b_2	a	s'	b_1'	d'	r'	b_2'	a'
$send(m,0)$	rts	0	e	rtr	0	e	$wack$	0	0	rtr	0	e
$ldata$	$wack$	0	0	rtr	0	e	$wack$	0	e	rtr	0	e
$tout$	$wack$	0	0	rtr	0	e	rts	0	e	rtr	0	e
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Any system can be represented as a Pr/T net which has one transition and one place.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

Pr/T model of ABP

- Version 3 of the Pr/T model of ABP is one extreme: anything substantial is modelled with annotations. The net part is only needed to model the dynamics.
- Version 2 of the Pr/T model of ABP is a representative of a balanced approach to modelling: It is easy to grasp the flow of information in the net; information processing is encoded into the firing conditions of the transitions.
- Versions 1 and 2 are in principle equivalent. Version 1 suffers from too much information in the annotations.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

On the relationship between Pr/T nets and P/T nets

- If there is only a finite number of feasible valuations for each transition in a Pr/T net then the Pr/T net can be interpreted as an abbreviation of a P/T net.
- Actually, any such Pr/T net N can be *unfolded* to a P/T net $U(N)$.
- The reachability graphs of N and $U(N)$ are isomorphic.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

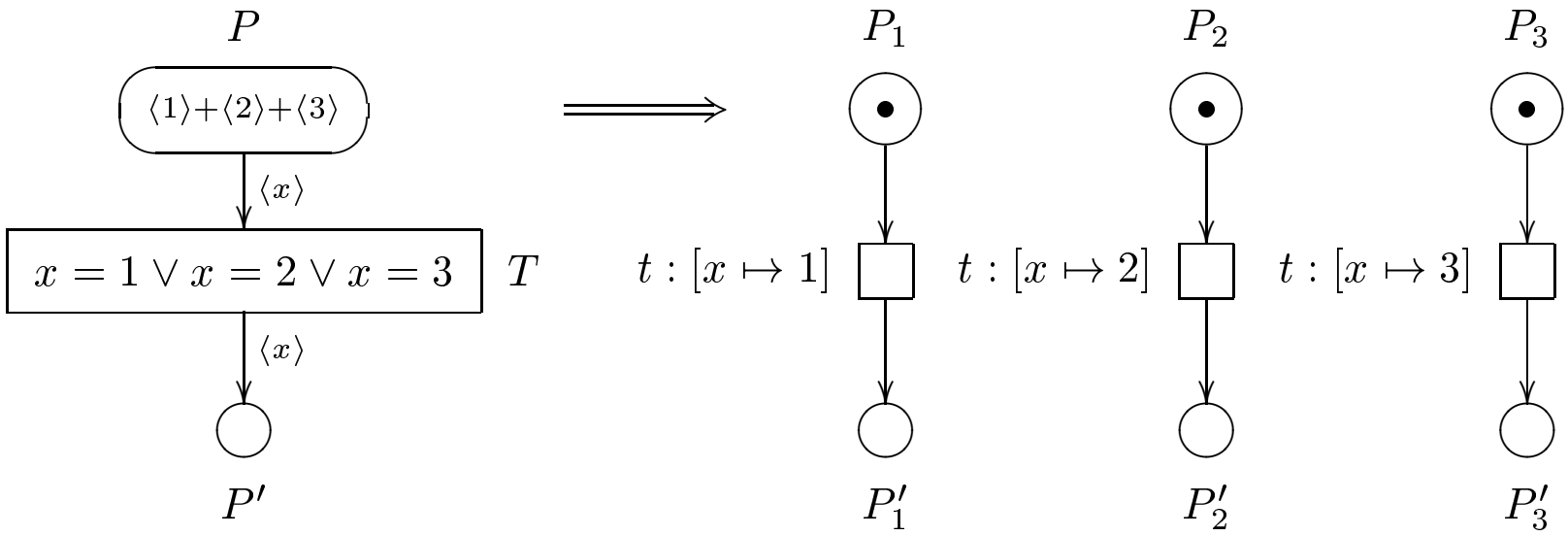
On the relationship between Pr/T nets and P/T nets

- Unfolding goes as follows:
  1. For each possible tuple m in the place p, a P/T net place p_m is created.
  2. For each feasible valuation σ of the Pr/T net transition t, a P/T net transition $t : \sigma$ is created.
  3. The flow relation is unfolded by connecting the place p_m with the transition $t : \sigma$ iff $A_F(p, t) : \sigma = m$ or vice versa.
  4. All the unconnected places are removed.
  5. The initial marking is unfolded,

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

On the relationship between Pr/T nets and P/T nets

- Example:



©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

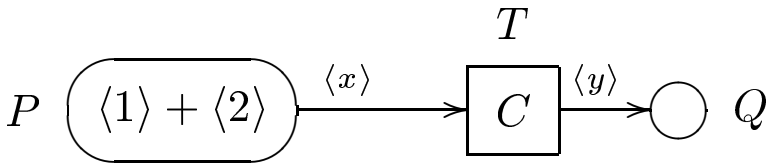
On the relationship between Pr/T nets and P/T nets

- *Dead* transitions may exist in the unfolded net; they are transitions which are never in concession. They can be removed, too, but only after their deadness property has been detected from the reachability graph.
- Size of the P/T net:
  - Let M_p be the set of possible tuples in the place p
  - In the net $U(M)$ the number of places is $\Sigma_{p \in P}|M_p| \approx O(max|M_p| \times |P|)$.
  - Let S_t be the set of feasible valuations of the transition t.
  - In the unfolding, the number of transitions will be $\Sigma_{t \in T}|S_t| \approx O(max|S_t| \times |T|)$.
  - E.g., if there are n variables in the index of the transition t and the valuations of the variables are in the range $\{0, 1\}$ then the number of generated transitions is 2^n.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

On the relationship between Pr/T nets and P/T nets

- The original idea about high level nets was to use the same underlying net to model several disjoint concurrent processes.
- With annotations it is possible to hide the explicit behaviour depicted with the net.

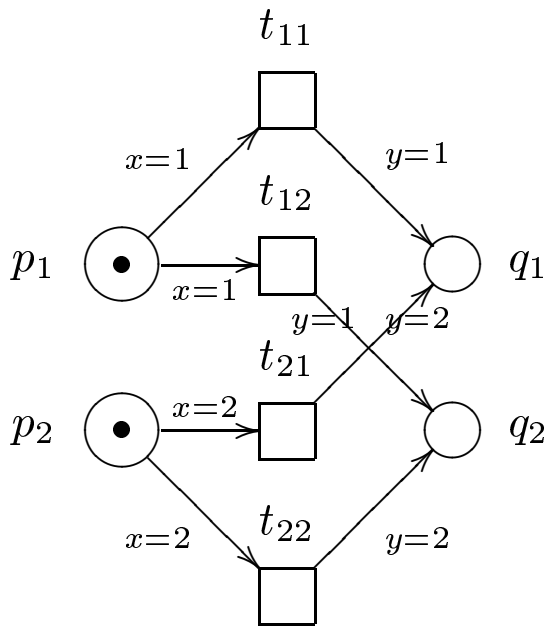


where $C \equiv (x = 1 \vee x = 2) \wedge (y = 1 \vee y = 2)$.

©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000

On the relationship between Pr/T nets and P/T nets

- The unfolded net looks like this:



©Johan Lilius, Keijo Heljanko, and Esa Kettunen 1997-2000