Object Nets for Mobility

Michael Köhler and Berndt Farwer

Department for Informatics
University of Hamburg

Application and Theory of Petri Nets

Object Nets

- Nets within Nets
- Nets as tokens
- Recursion for Petri nets

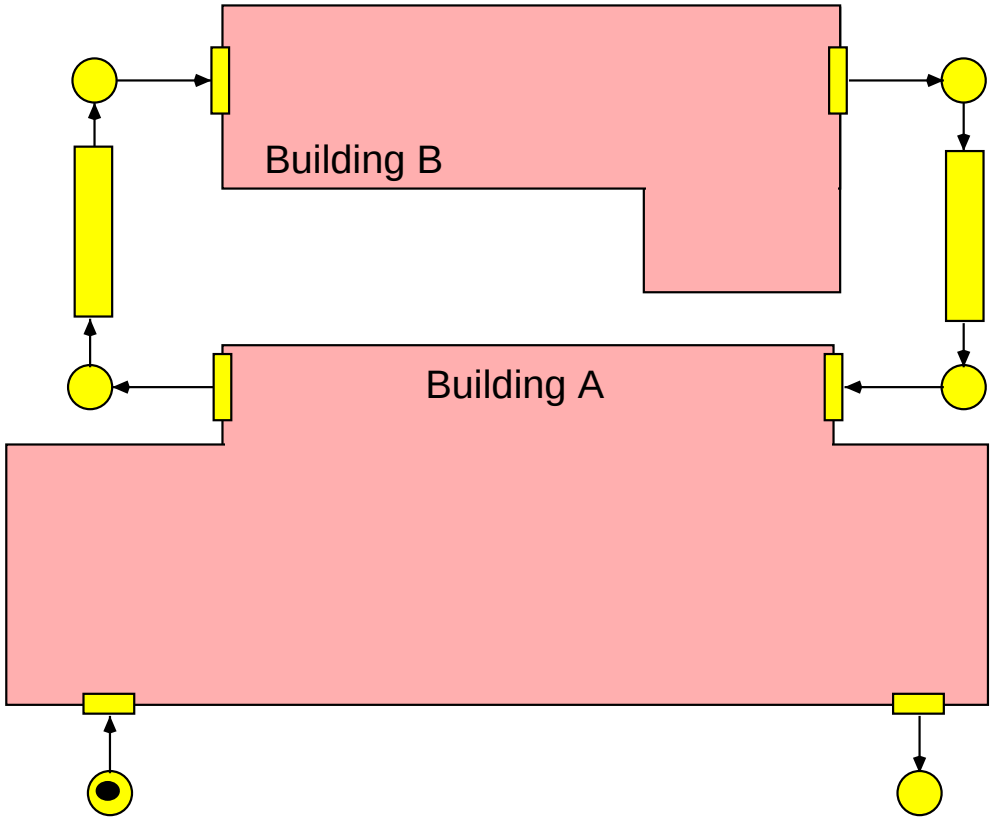
- Object-orientation
- Mobile systems
- Multi-agents systems

Object Nets

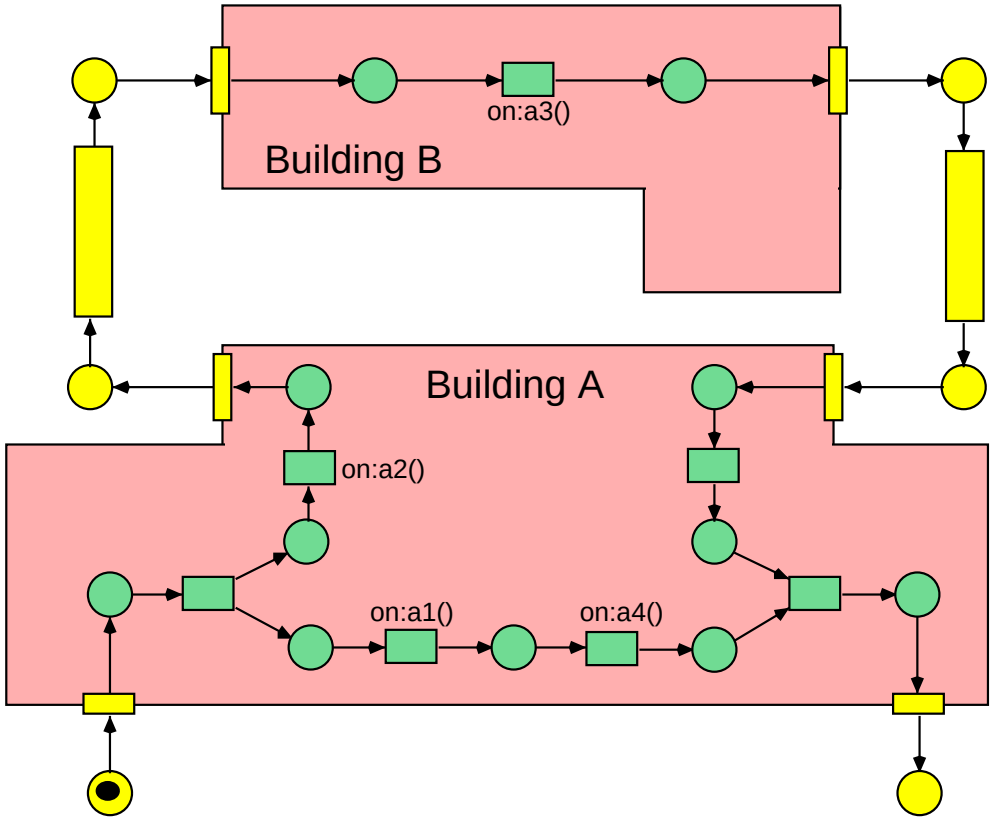
- Nets within Nets
- Nets as tokens
- Recursion for Petri nets

- Object-orientation
- Mobile systems
- Multi-agents systems

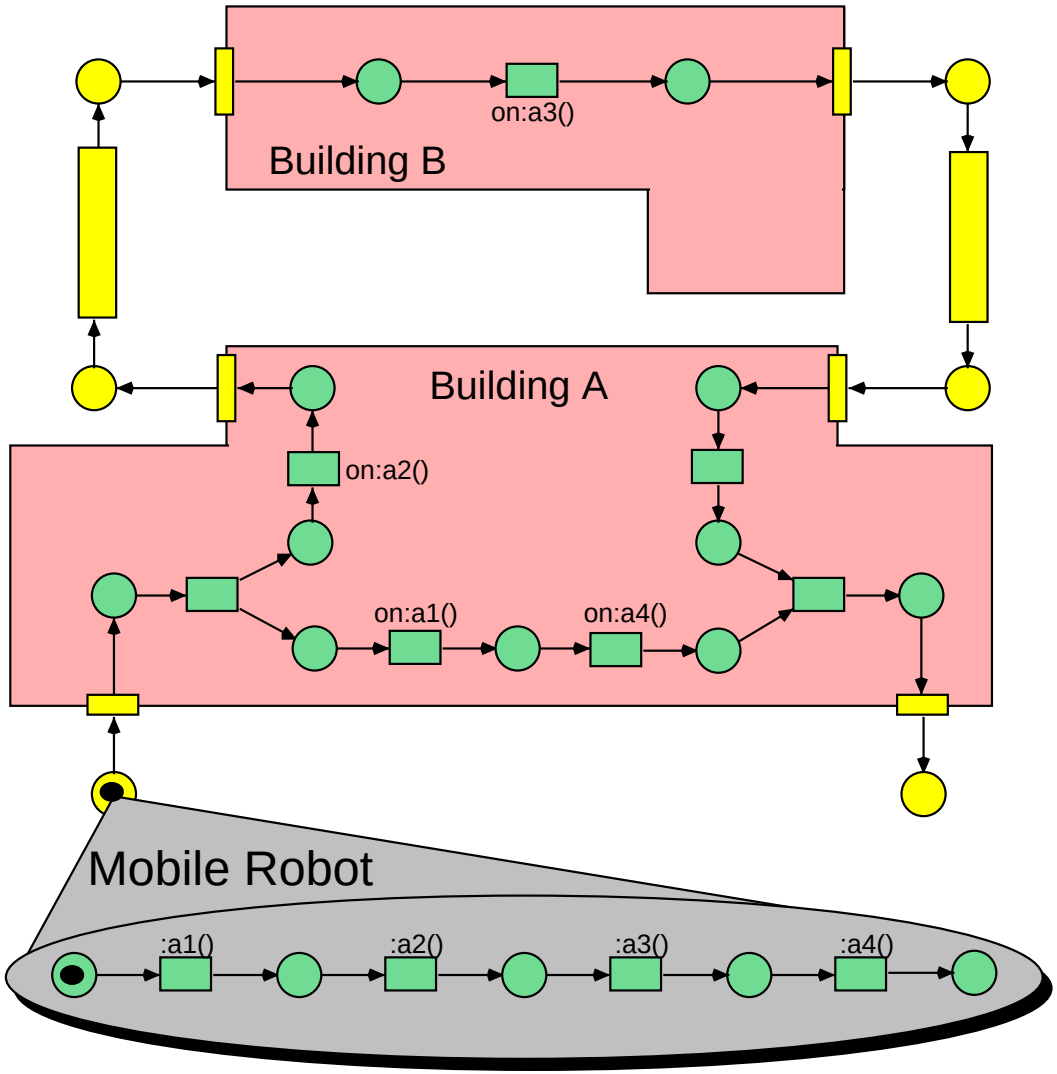
A Scenario



A Scenario



A Scenario

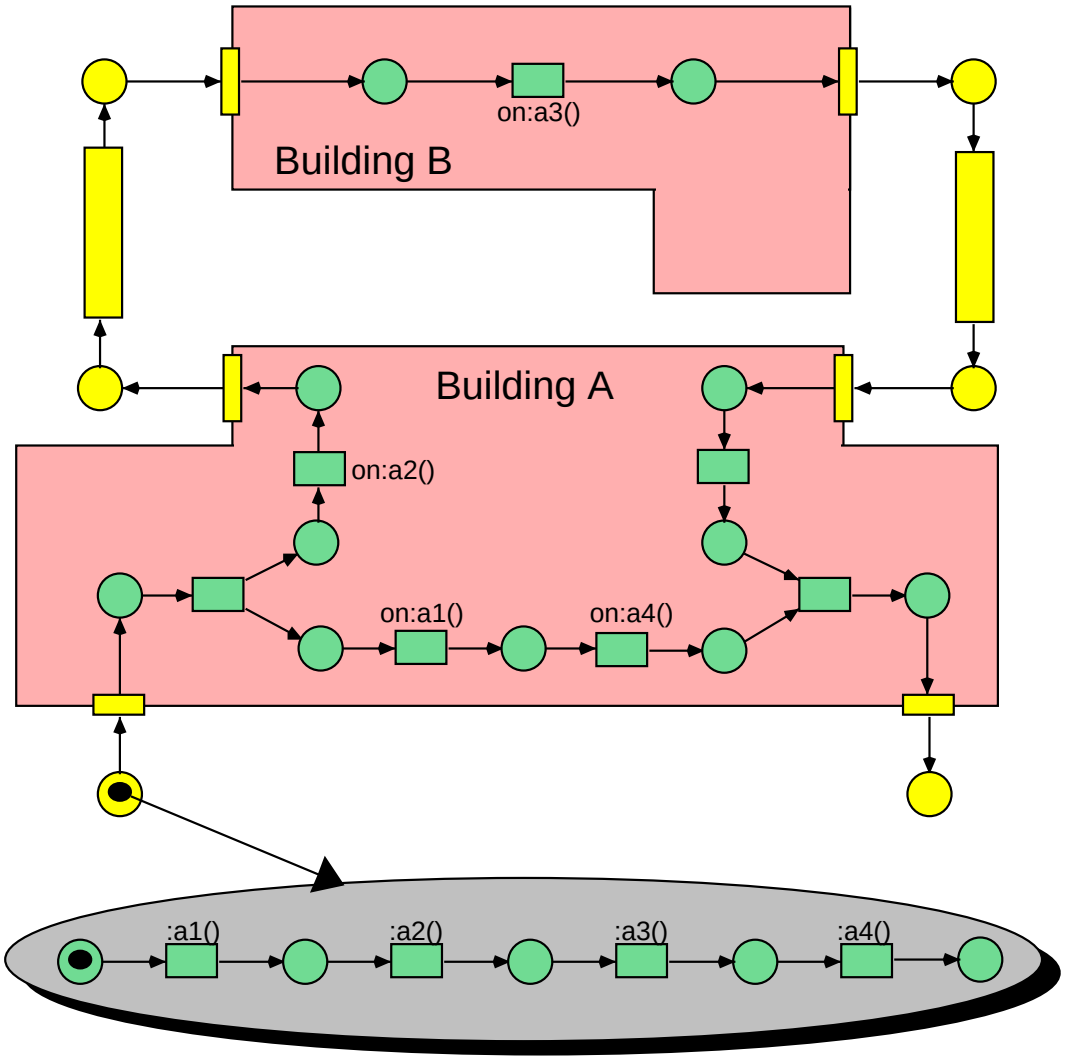

on:a3()
Building B
Building A
on:a2()
on:a1()
on:a4()
Mobile Robot
:a1()
:a2()
:a3()
:a4()

Net Tokens

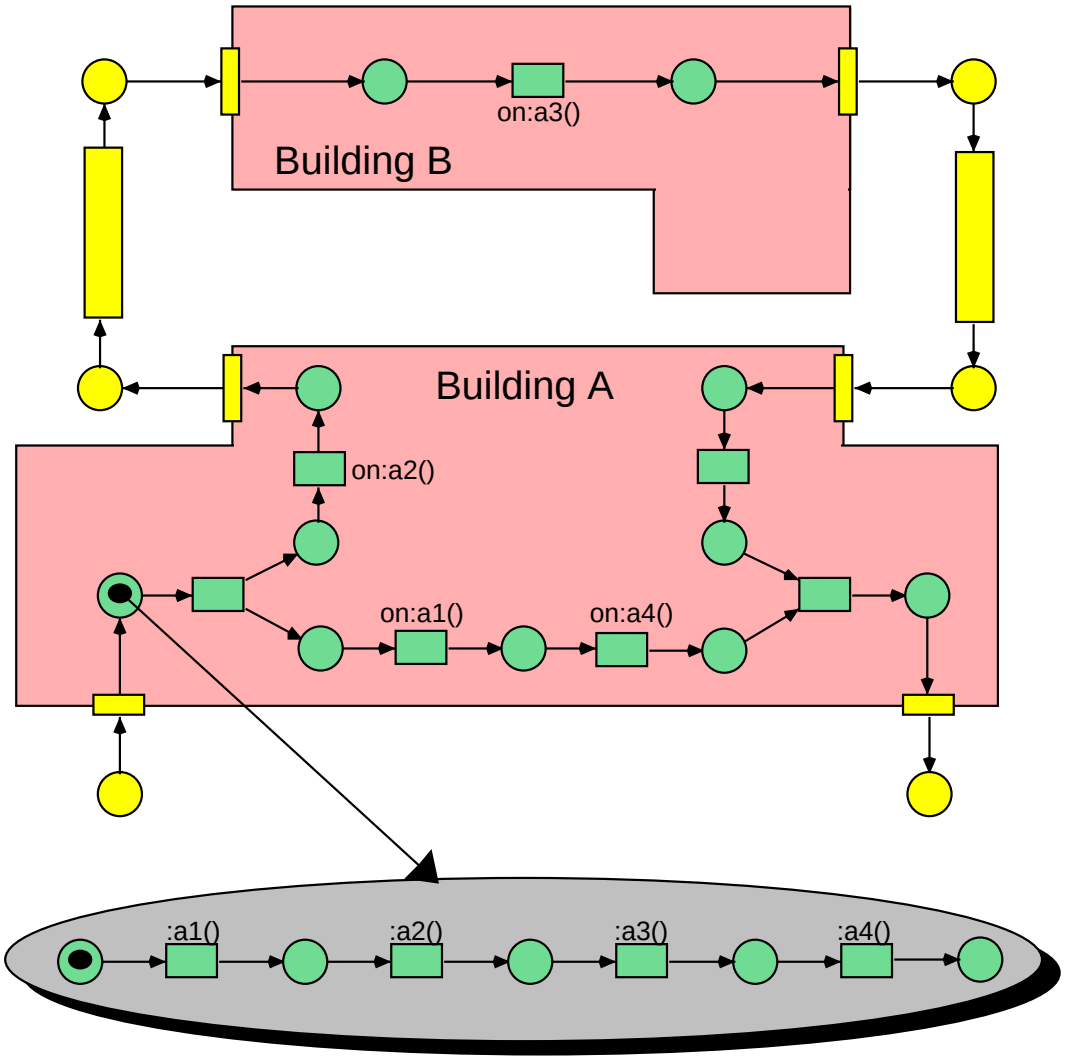
How to define Net Tokens?

1. Net Tokens as **References**:
   - Assumes a global name space ($\rightarrow$ side effects)
   - Easy to implement
   - Closely related to programming languages
2. Net Tokens as **Values**:
   - Assumes a distributed space
   - Easy to understand
   - Adequate for mobility

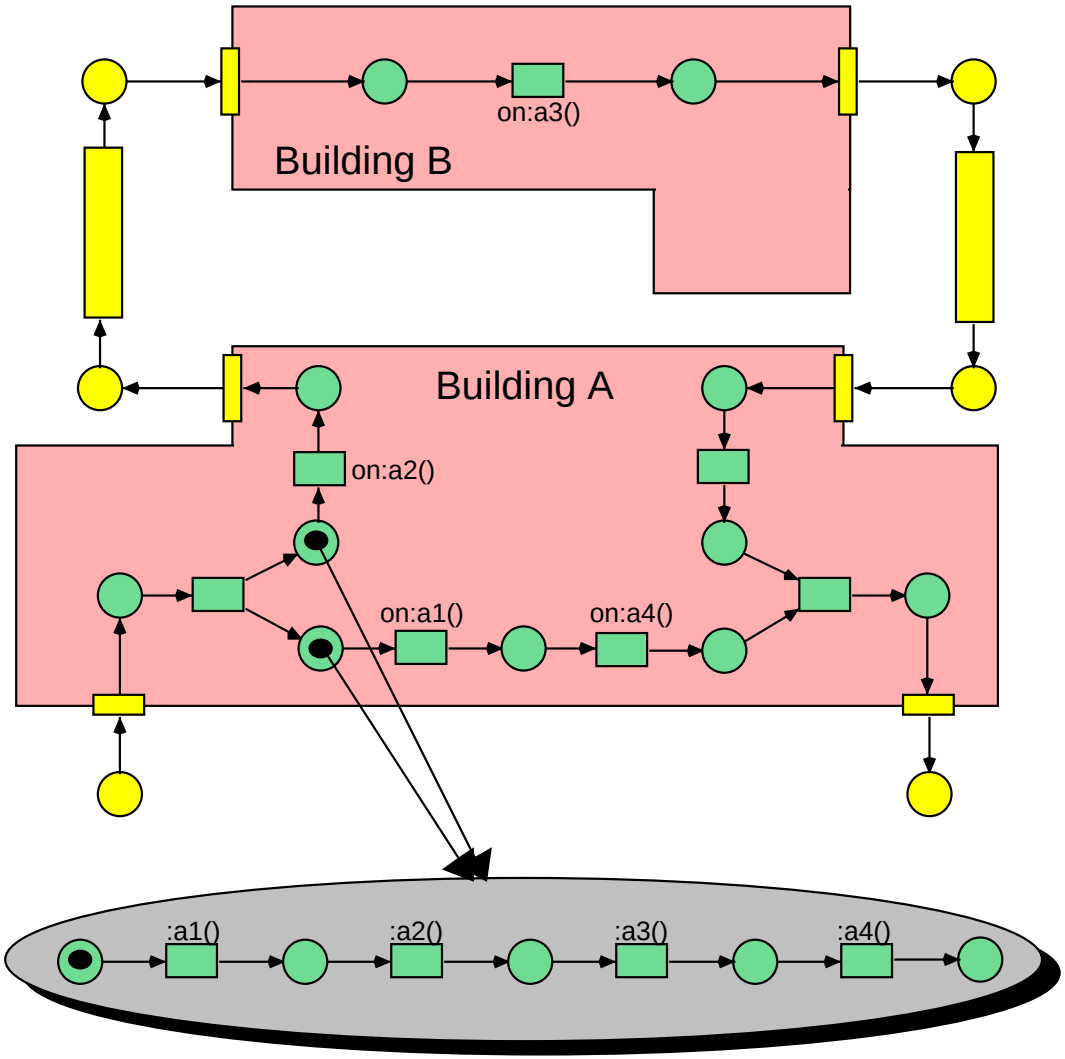
Reference Semantics



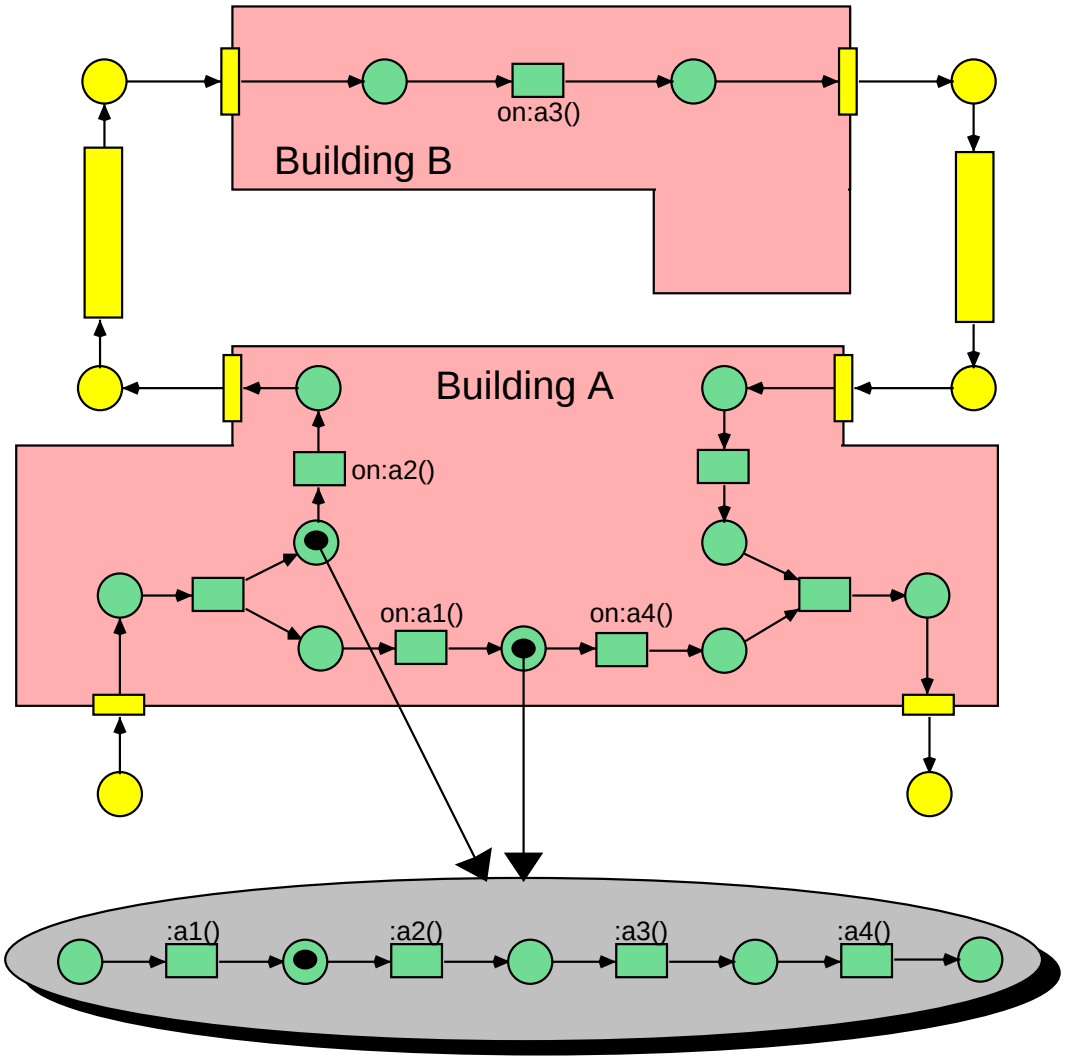
Reference Semantics



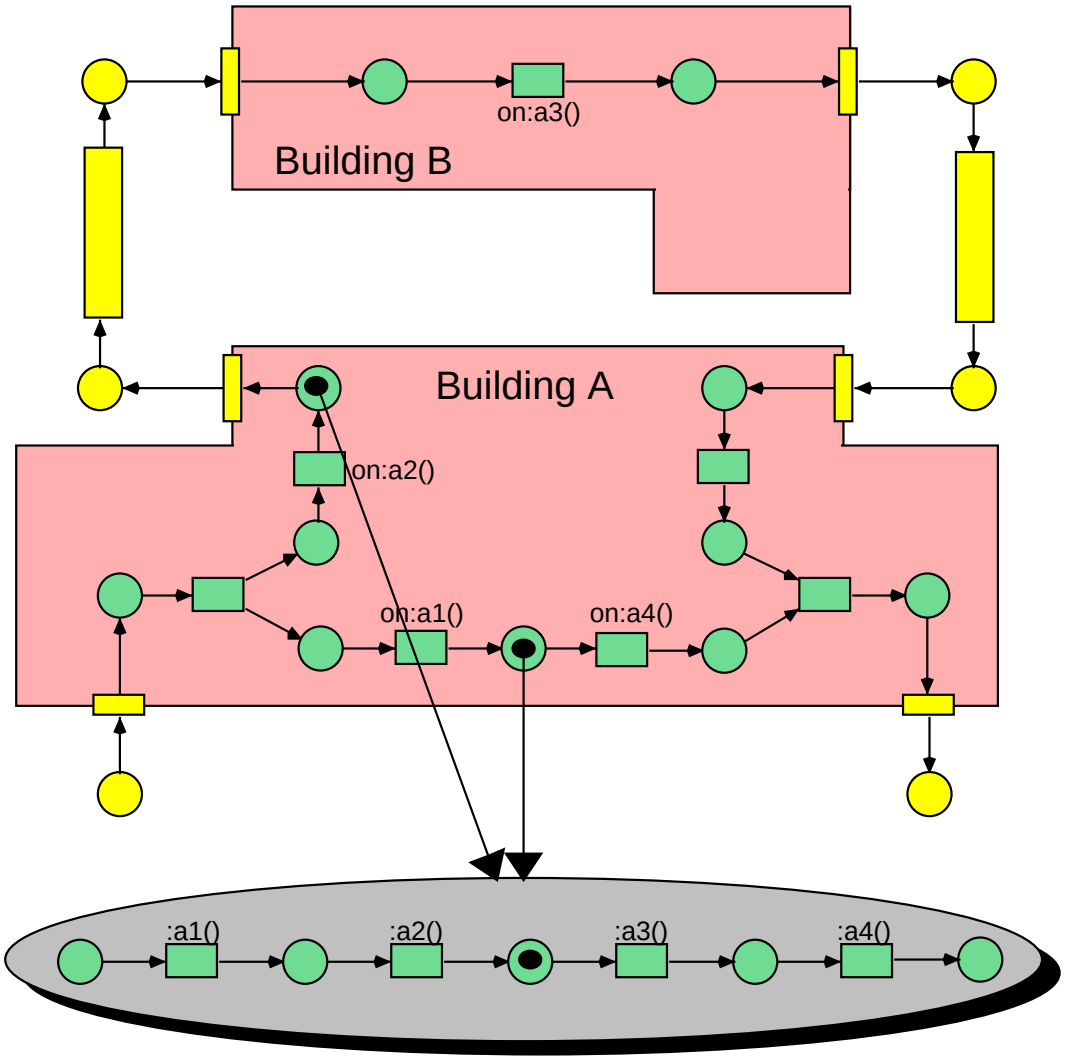
Reference Semantics



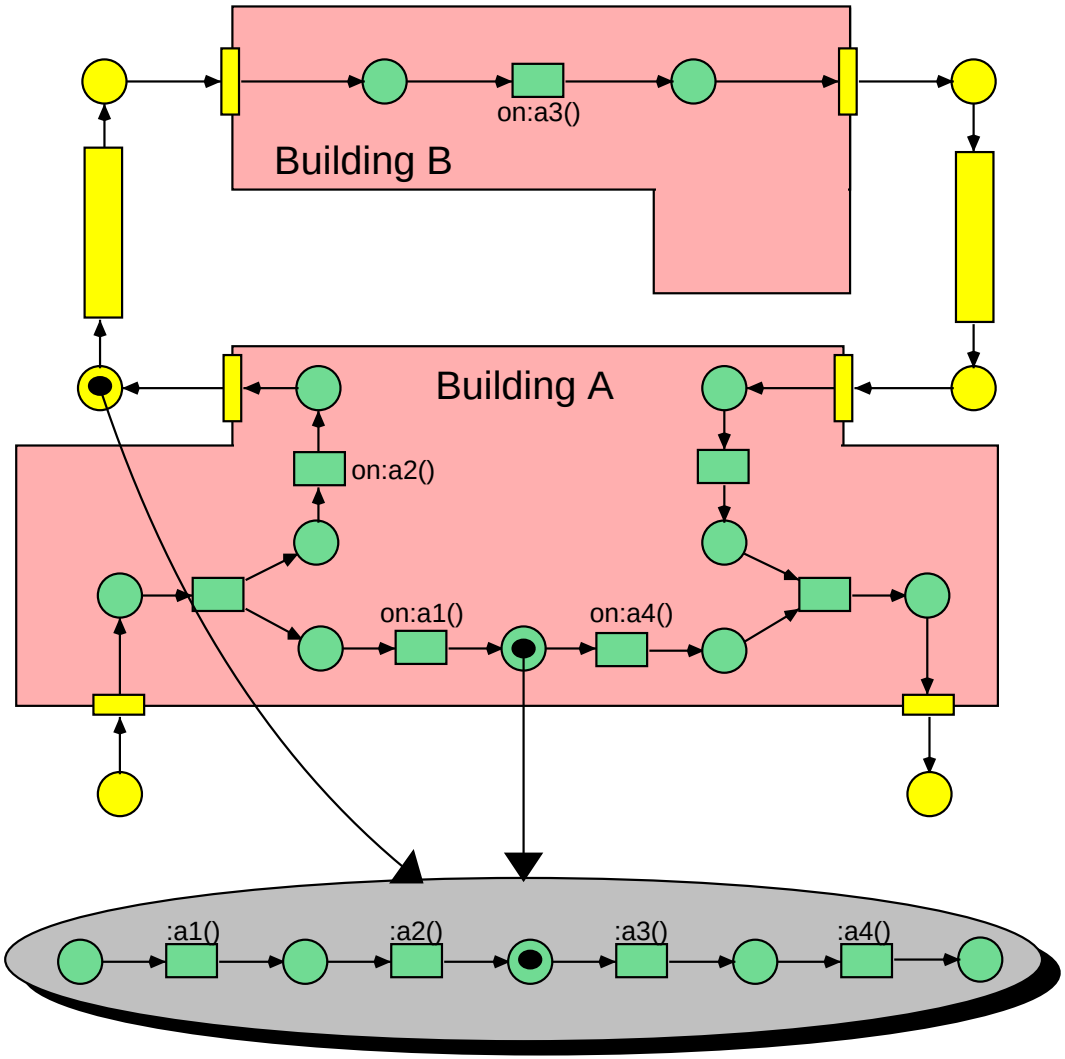
Reference Semantics



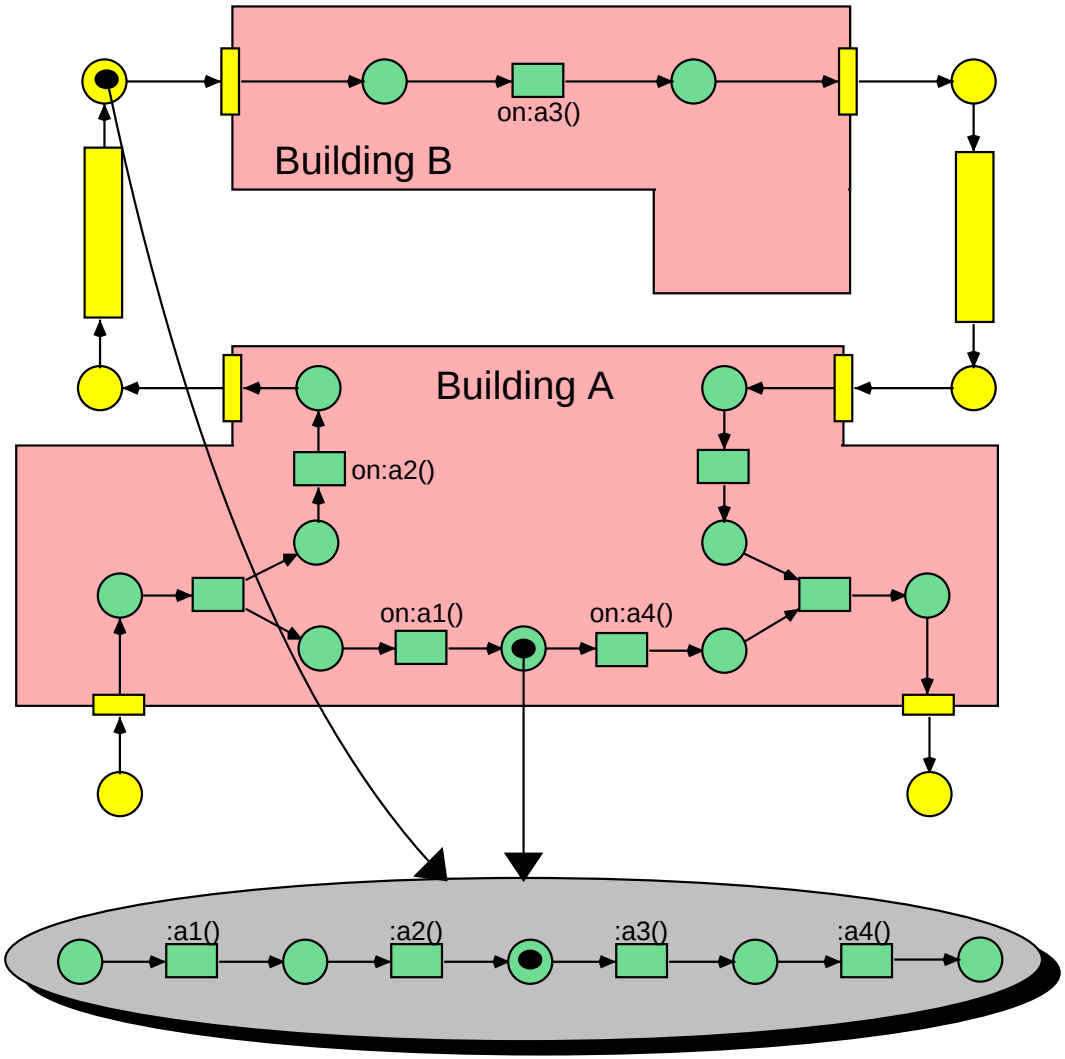
Reference Semantics



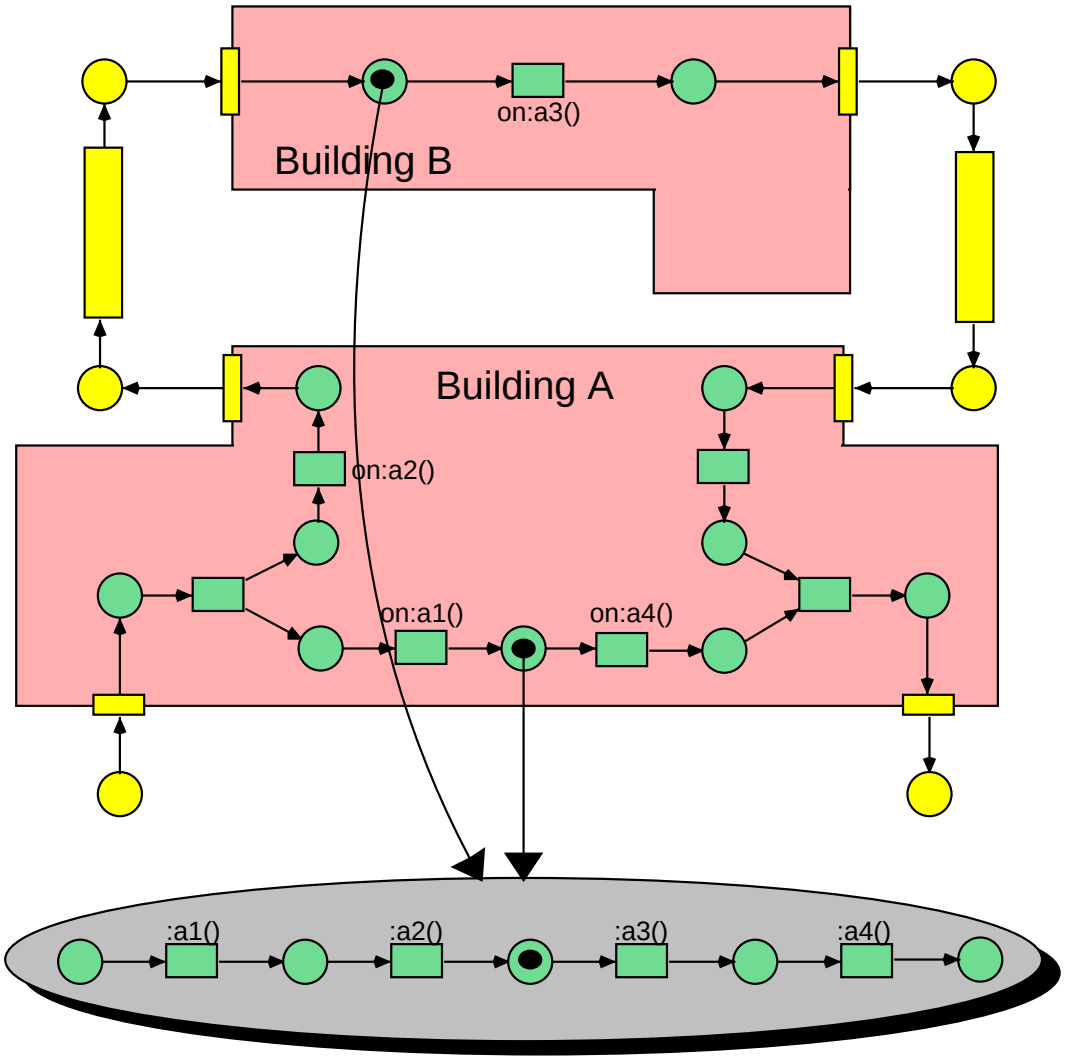
Reference Semantics



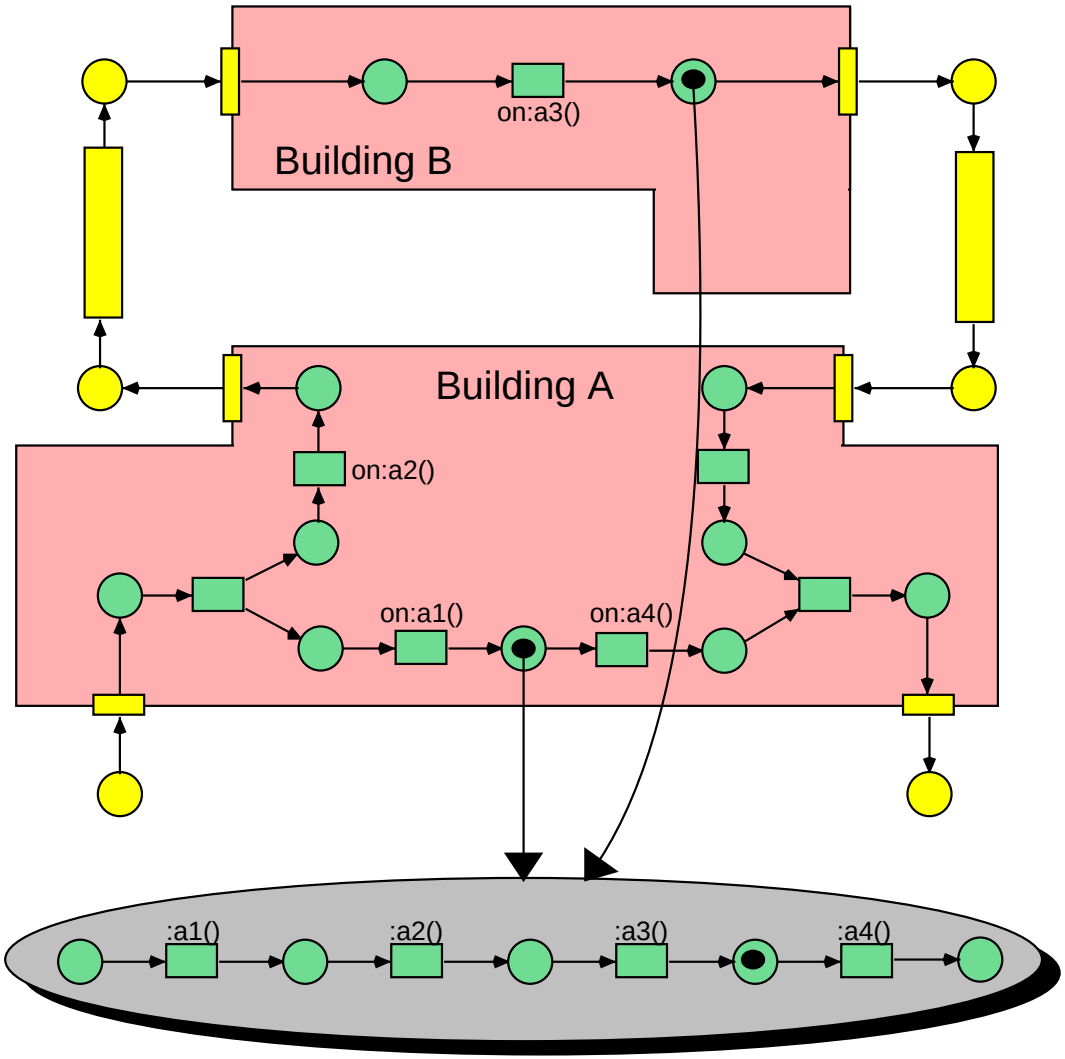
Reference Semantics



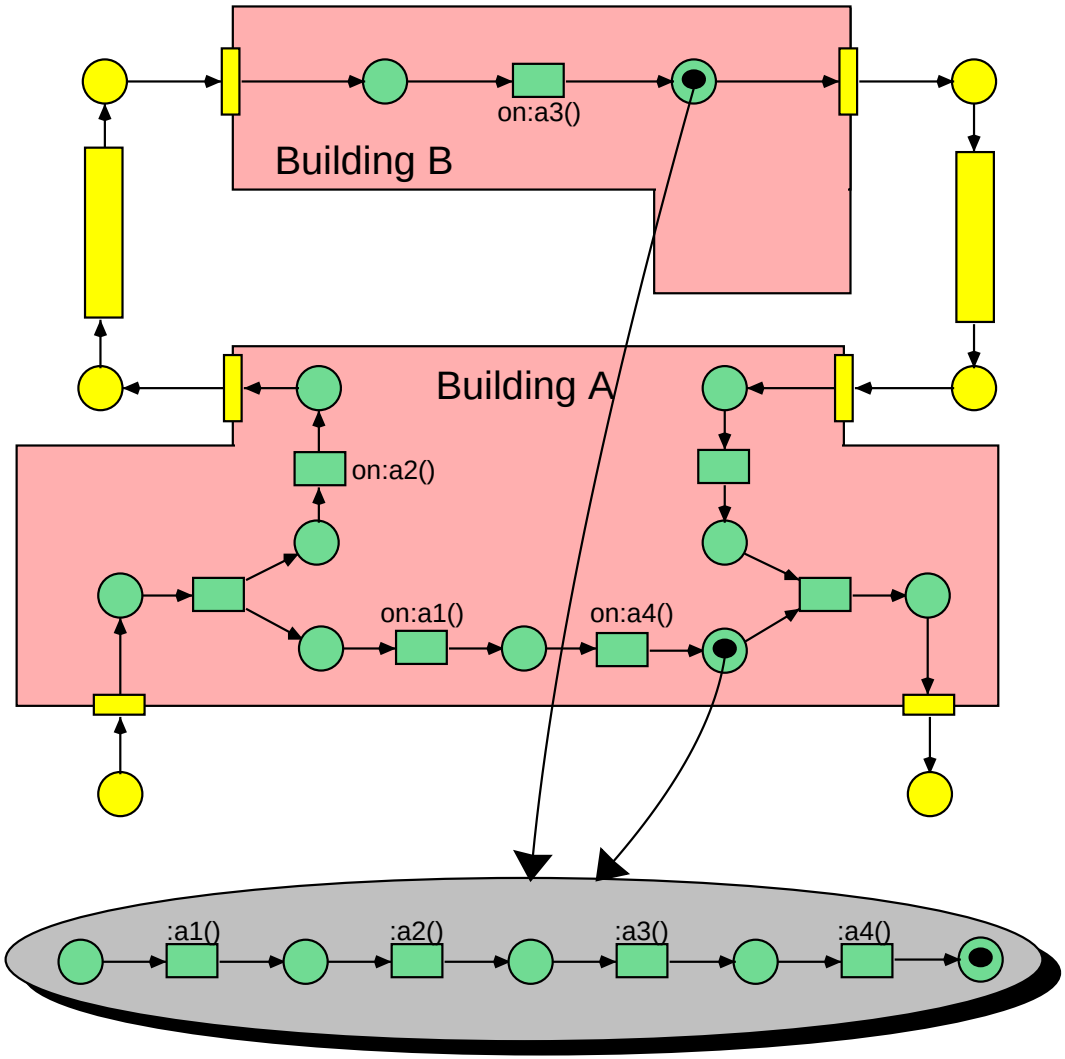
Reference Semantics



Reference Semantics



Reference Semantics

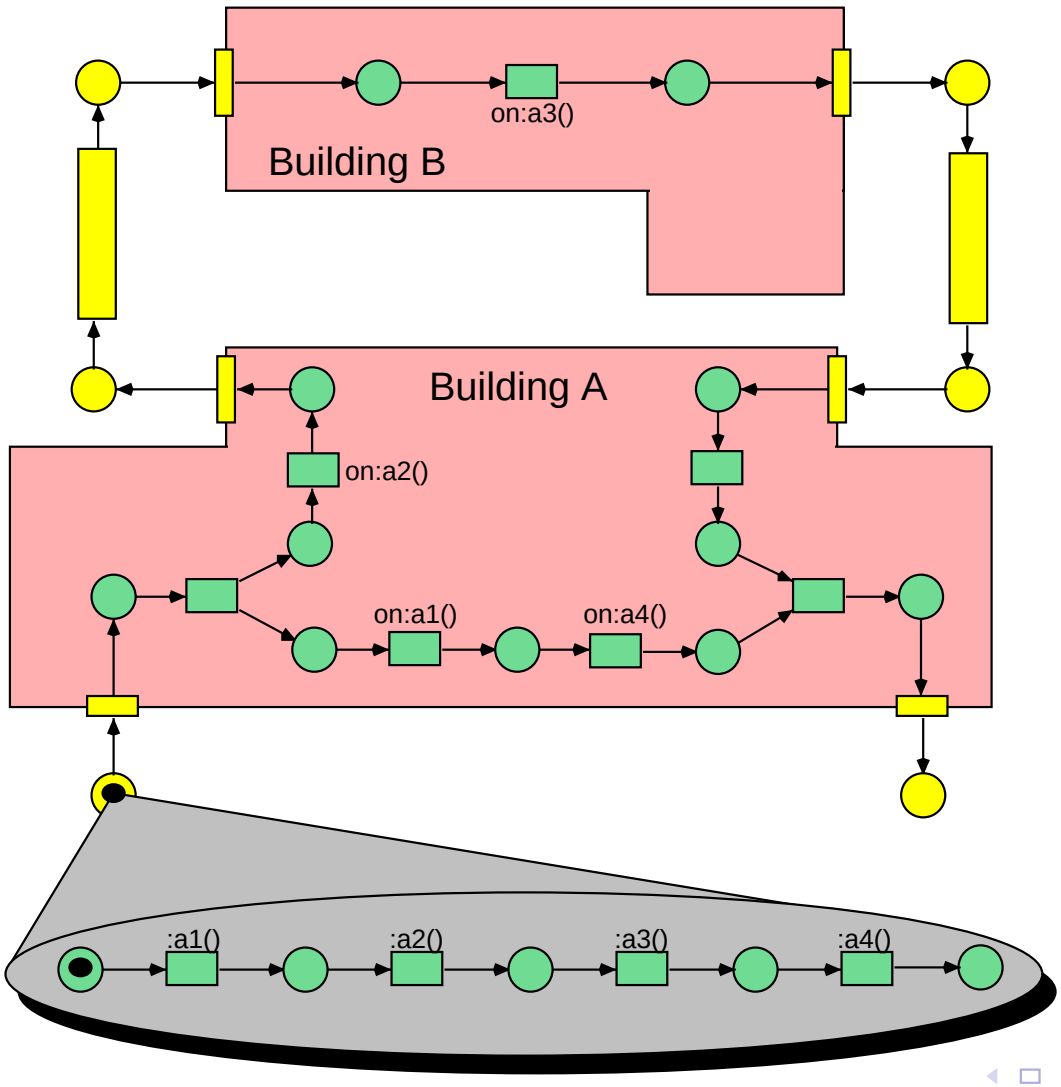


Net Tokens

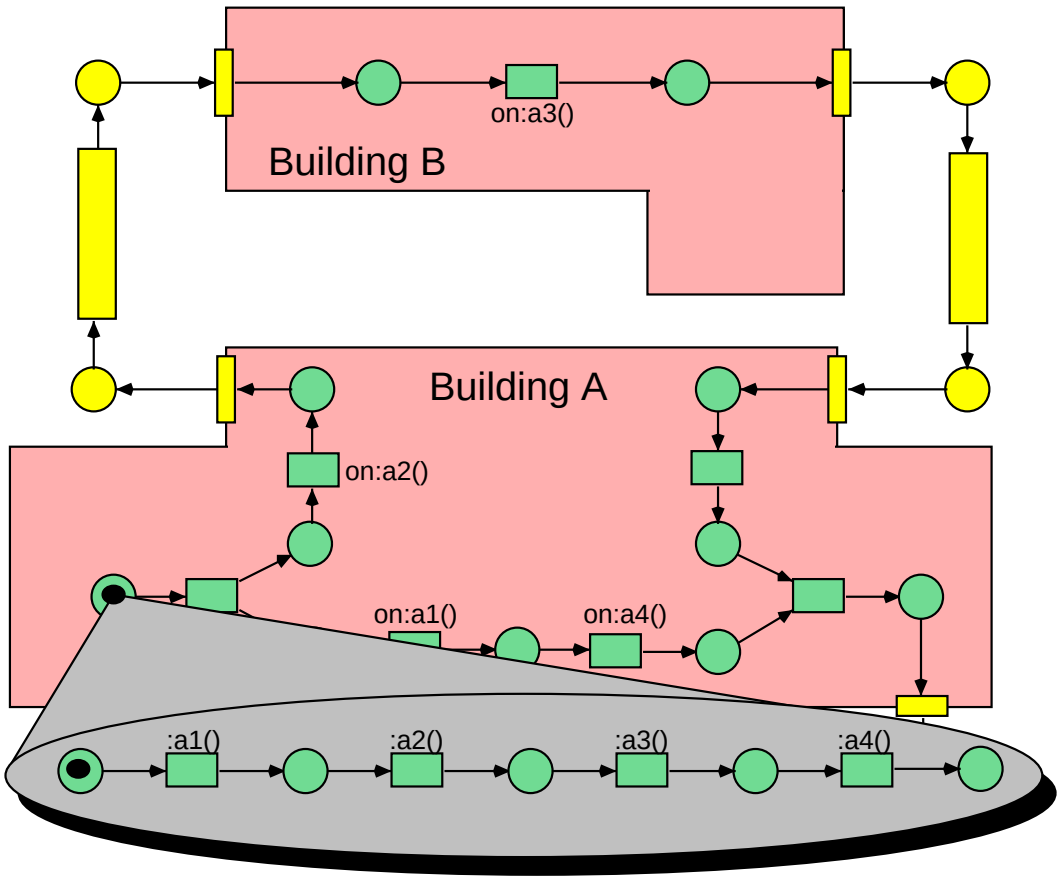
How to define Net Tokens?

1. Net Tokens as **References**:
   - Assumes a global name space ($\rightarrow$ side effects)
   - Easy to implement
   - Closely related to programming languages
2. Net Tokens as **Values**:
   - Assumes a distributed space
   - Easy to understand
   - Adequate for mobility

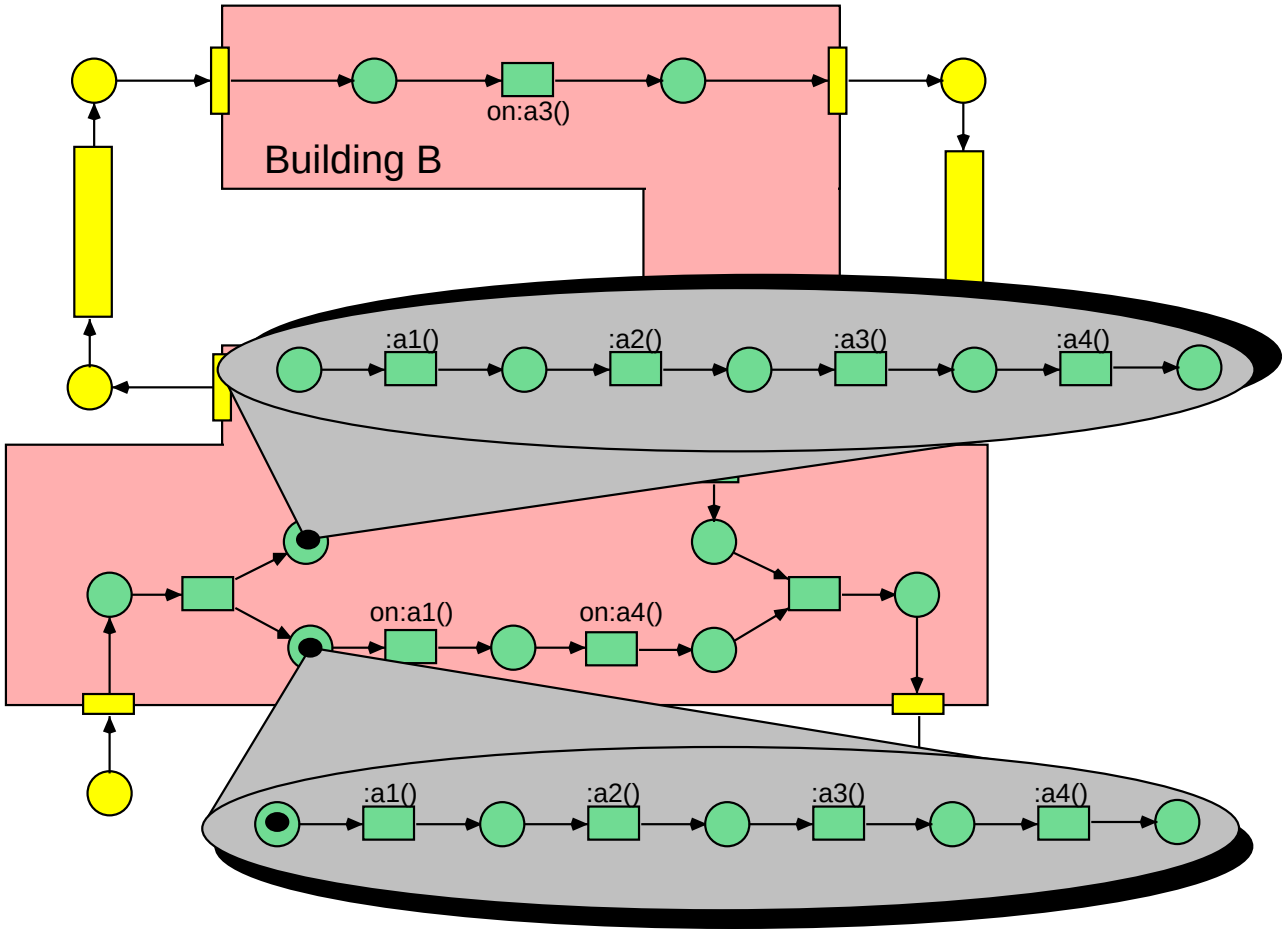
Value Semantics



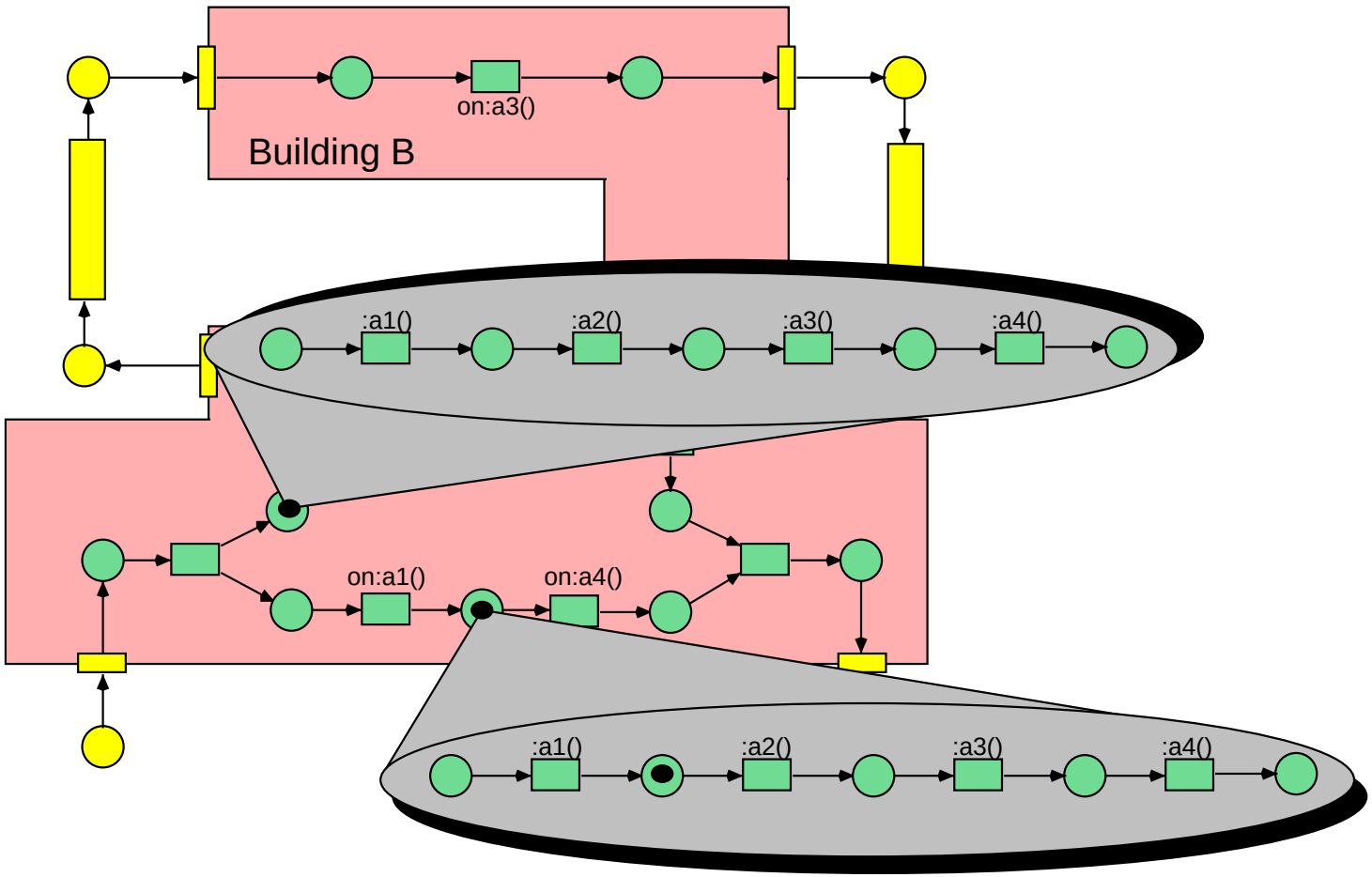
Value Semantics



Value Semantics



Value Semantics



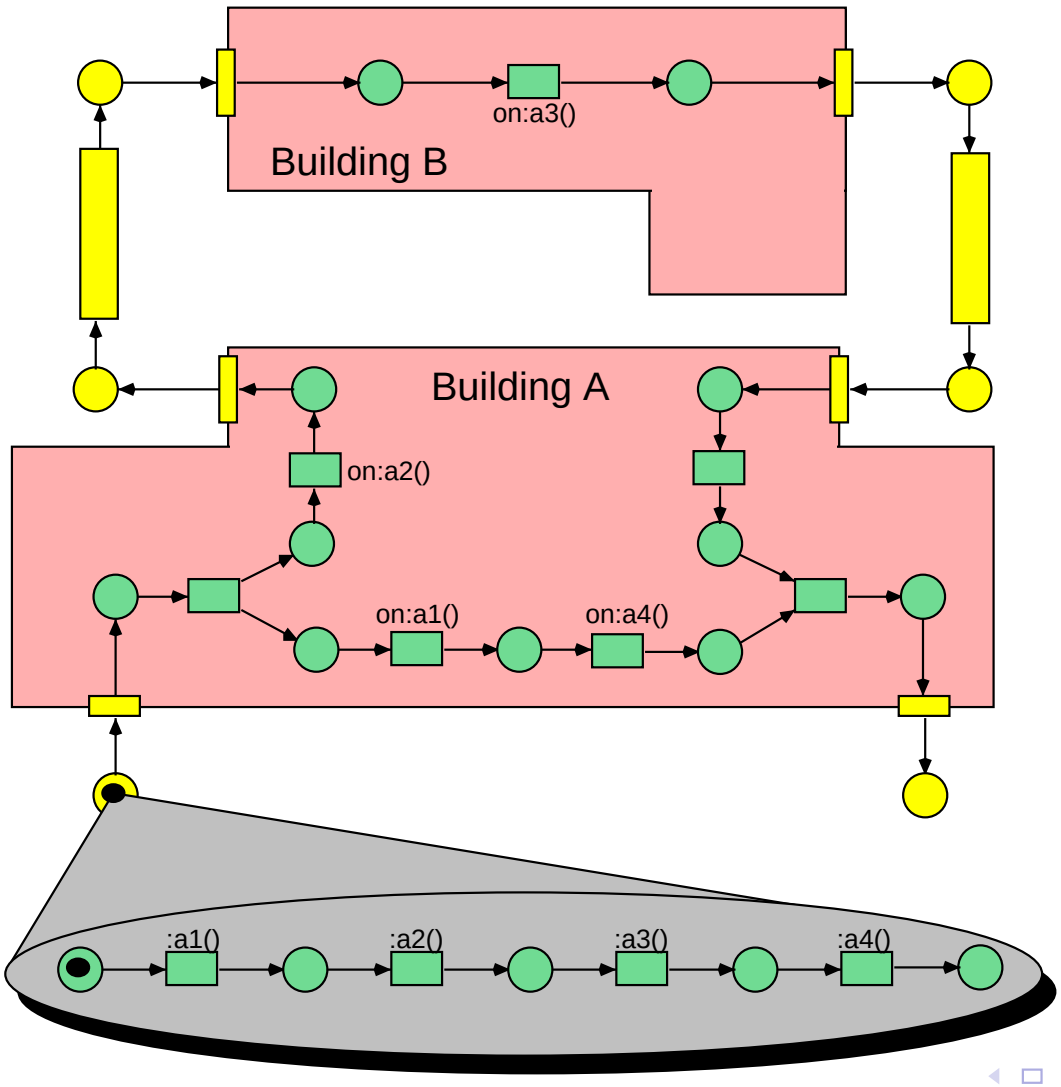
References/Values

- Reference semantics creates one single namespace.
- Reference semantics allows side-effects between different namespaces.
- Value semantics restricts access to one single place.
- Value semantics inhibits global access within the same namespace.

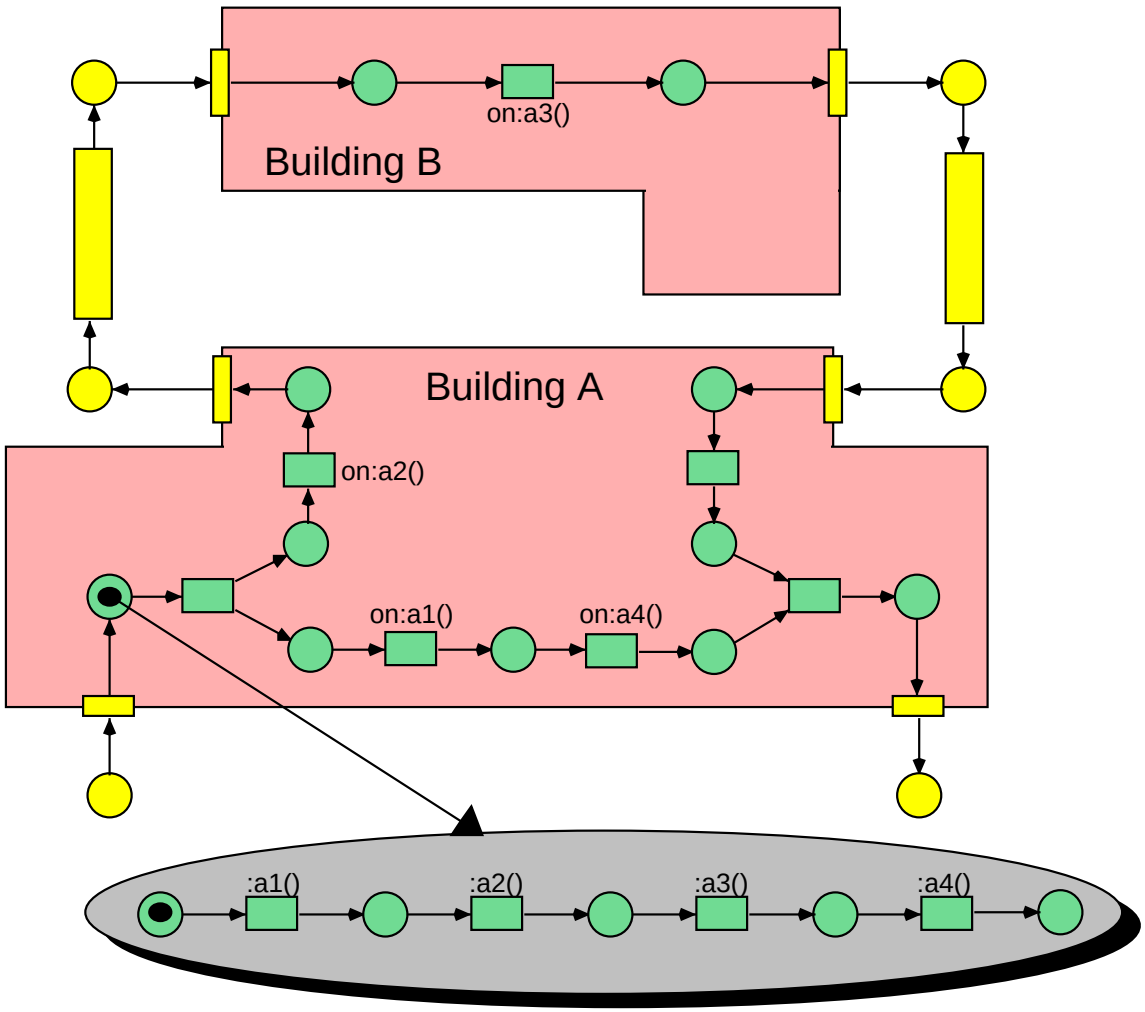
$\Longrightarrow$ A generalised semantics is needed: mobility semantics

- Mobility semantics "switches" between value and reference semantics.

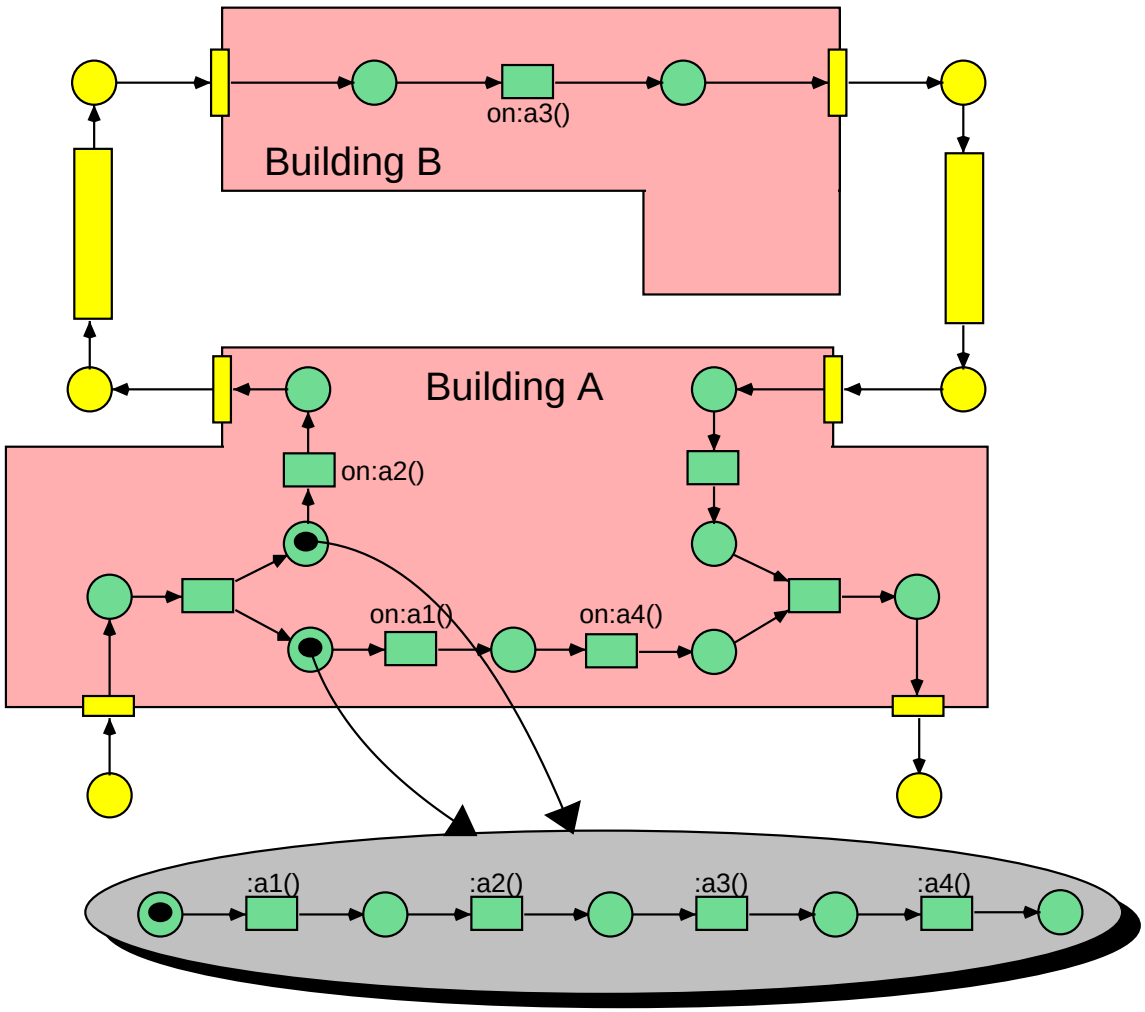
Generalised Semantics



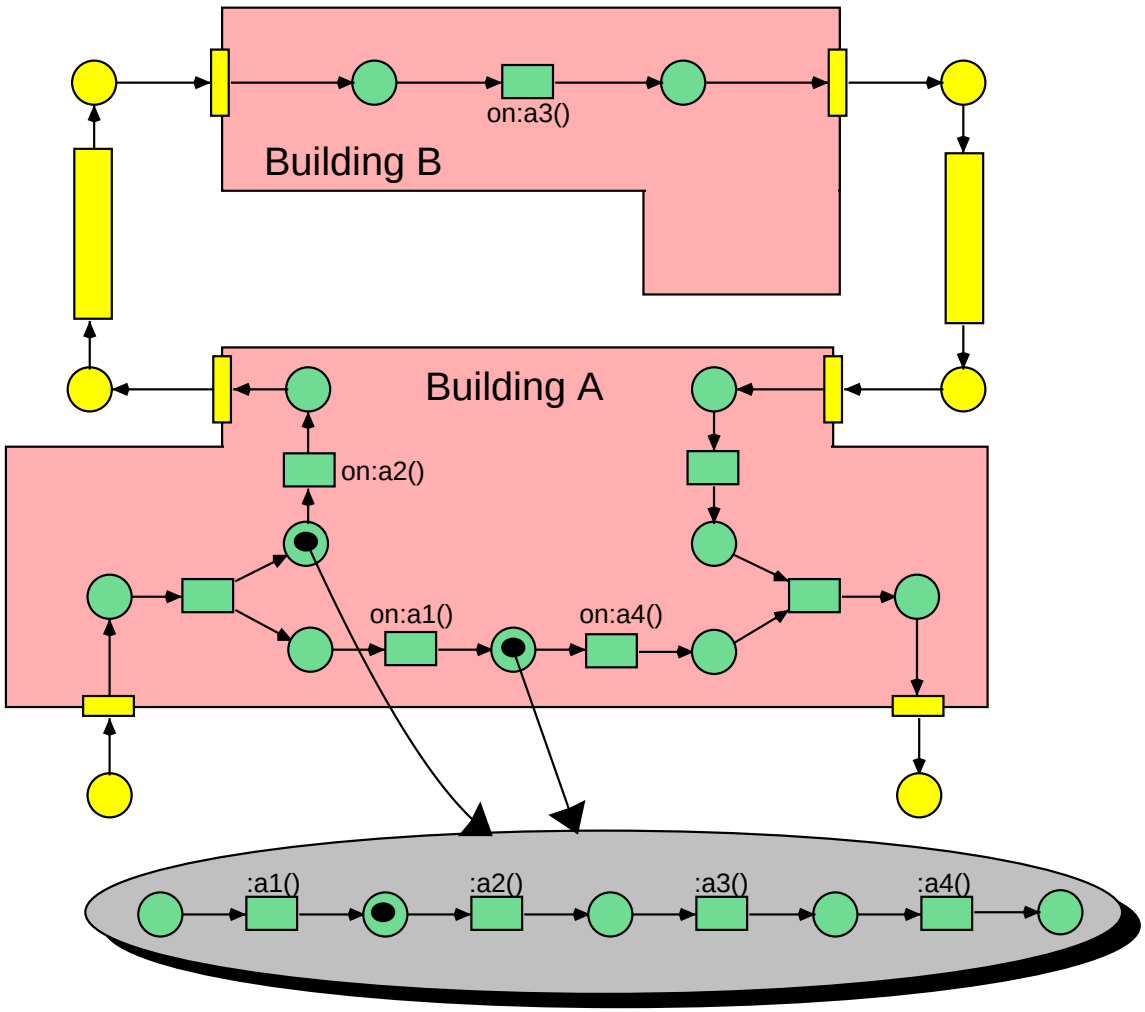
Generalised Semantics



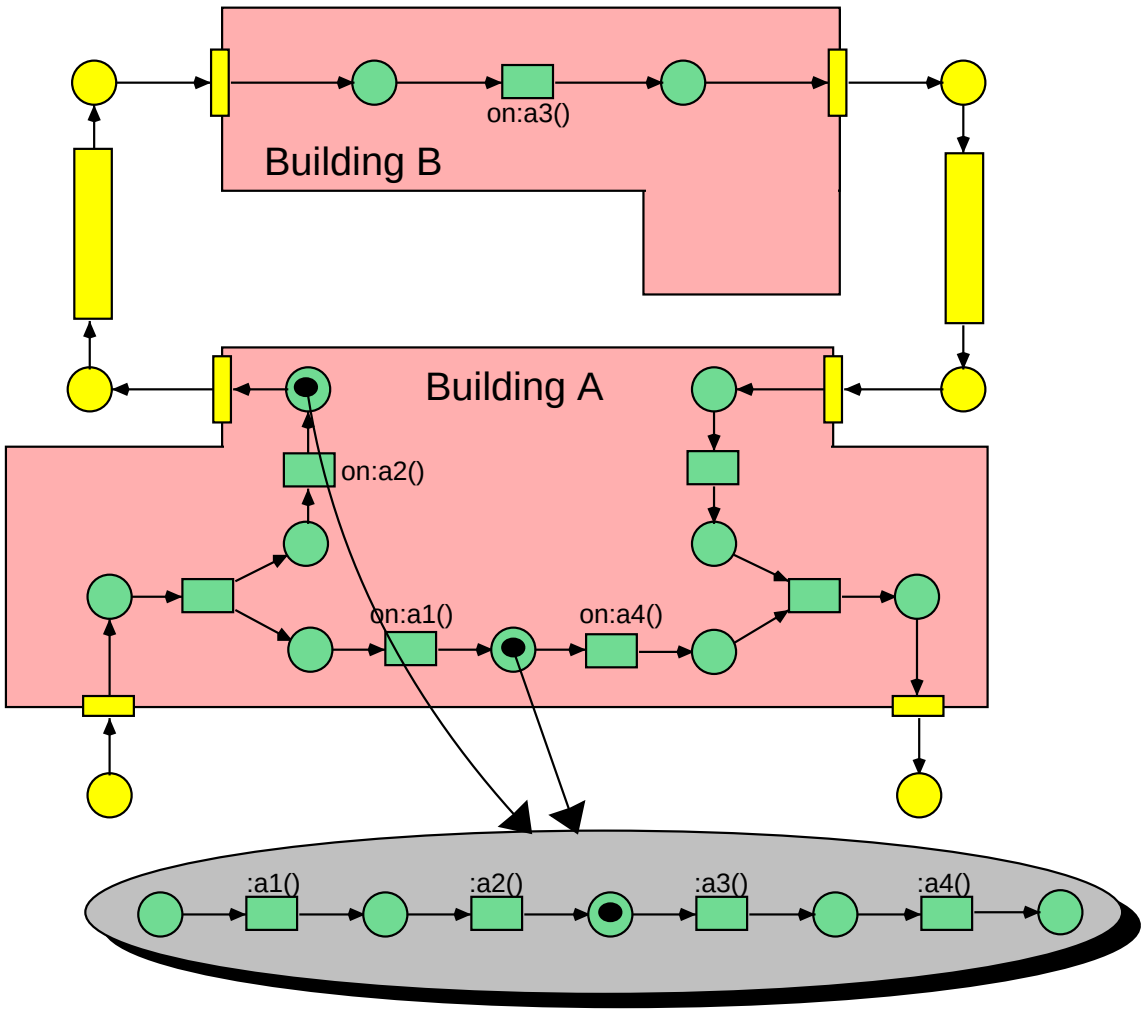
Generalised Semantics



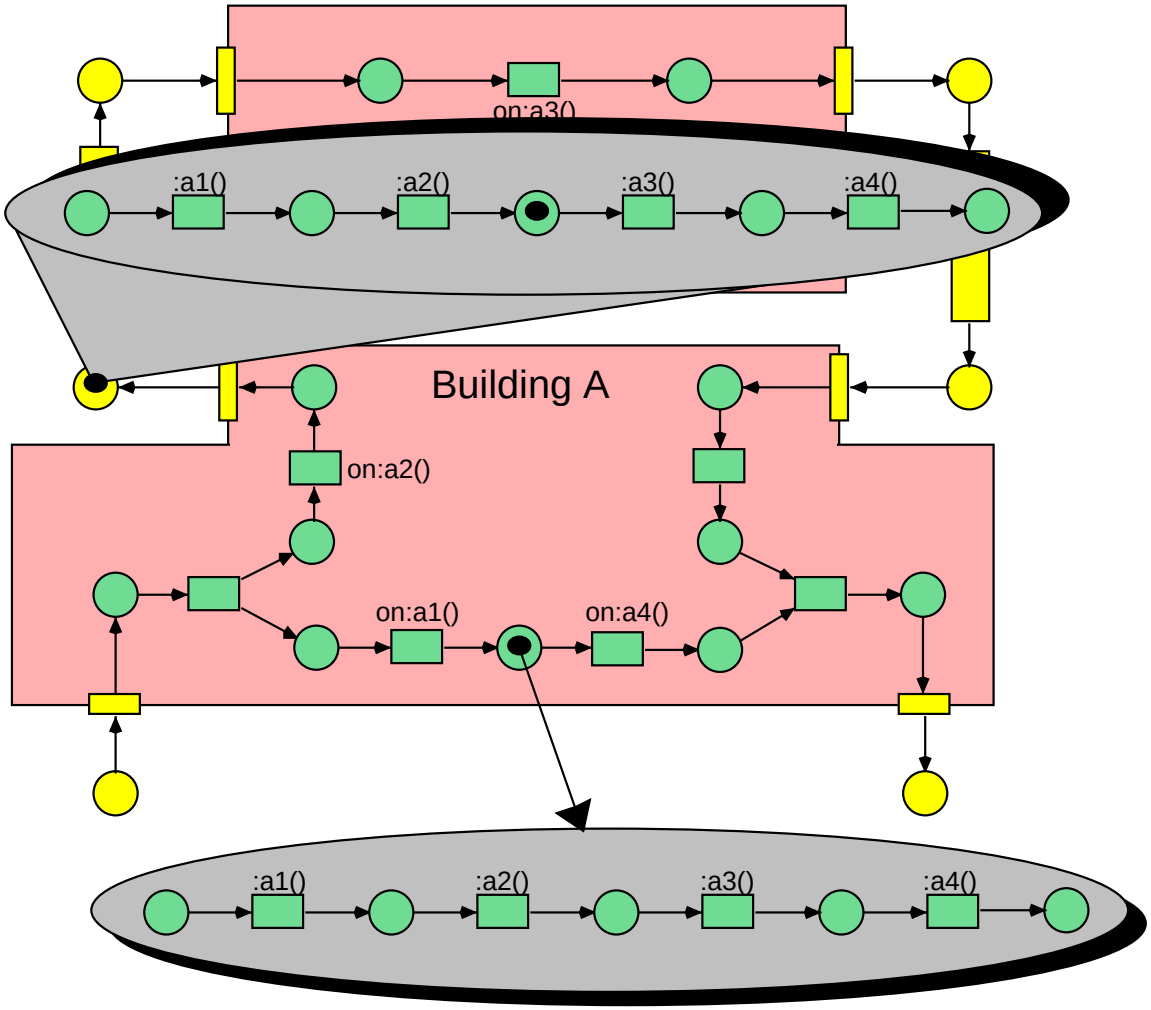
Generalised Semantics



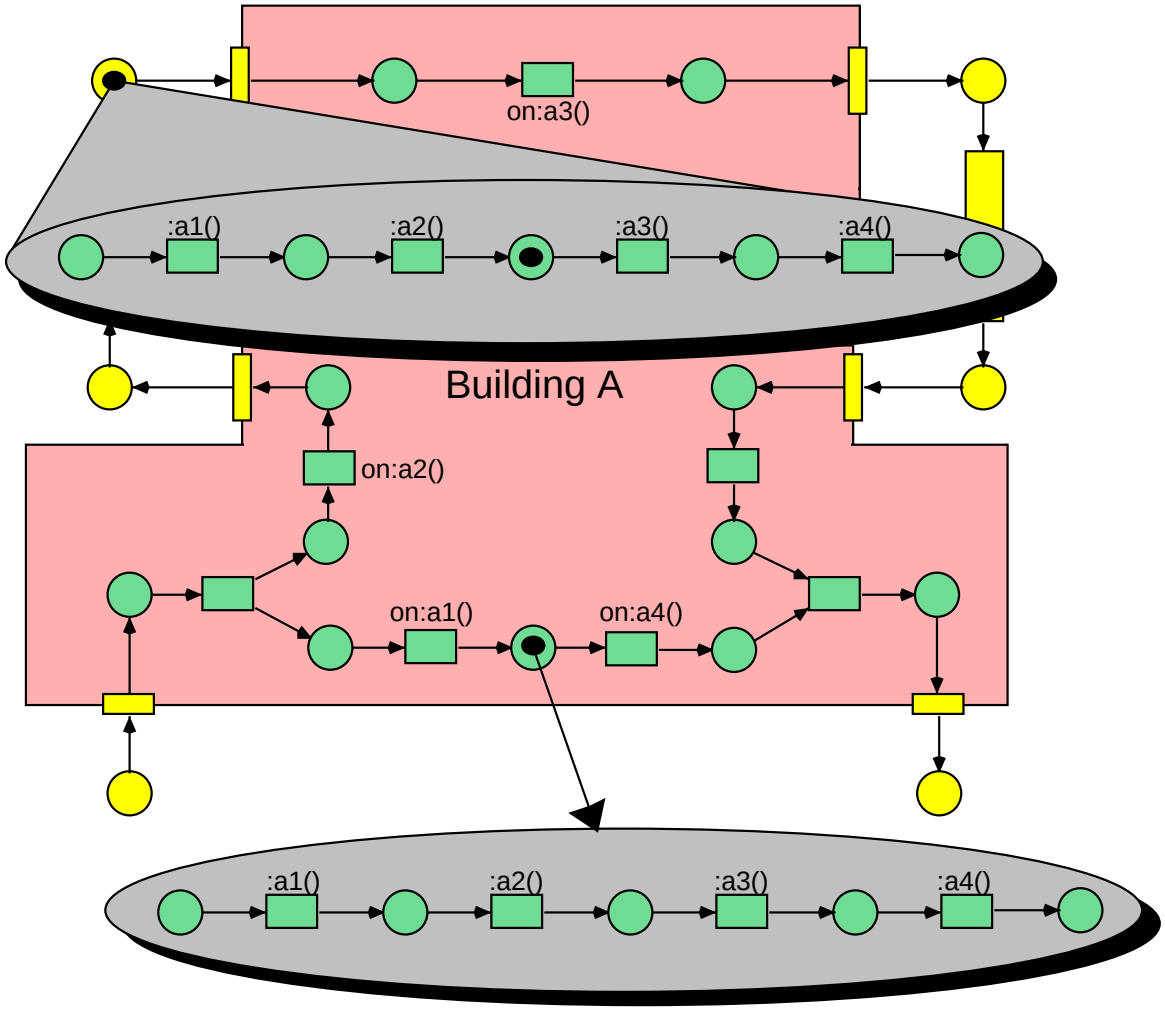
Generalised Semantics



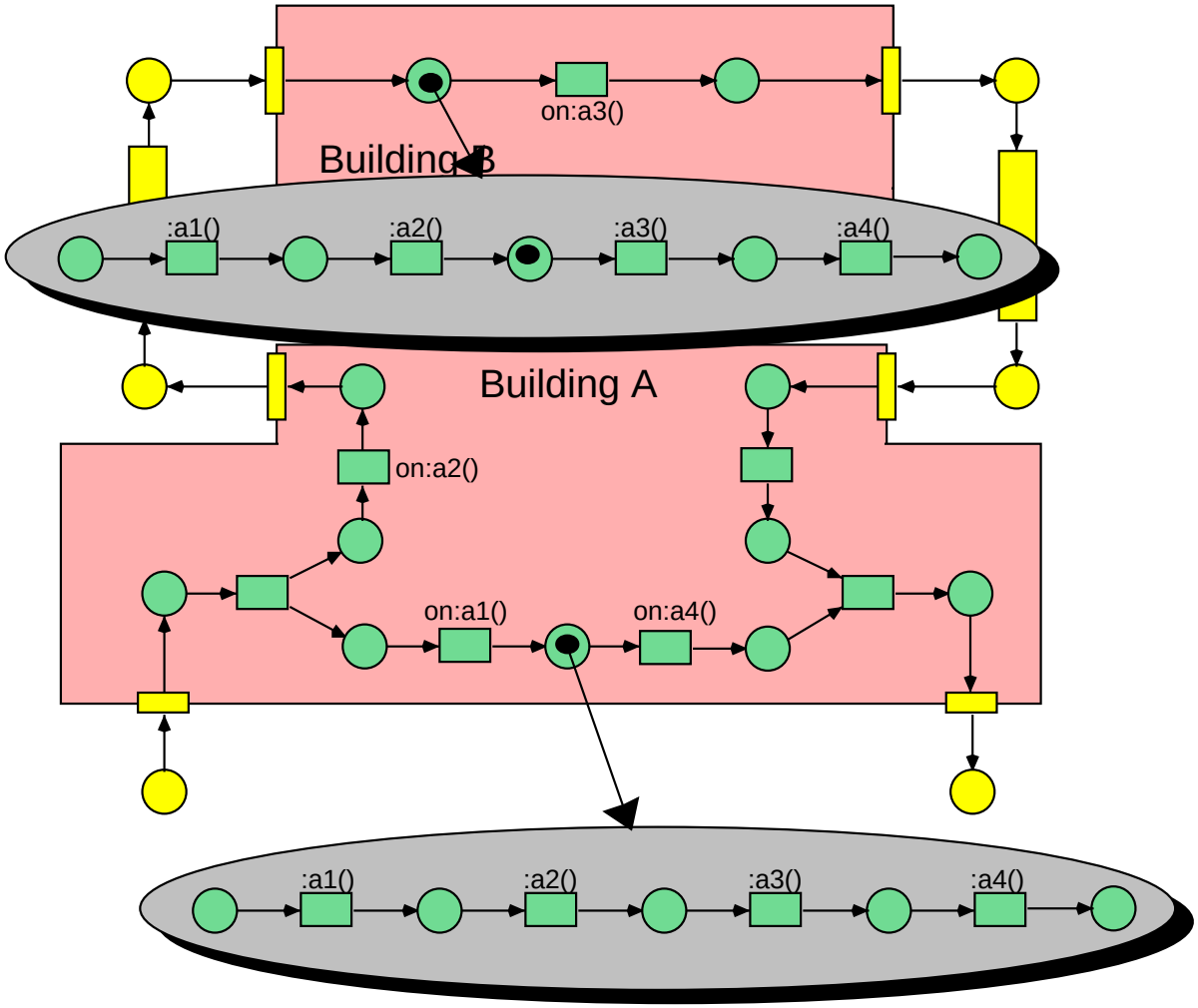
Generalised Semantics



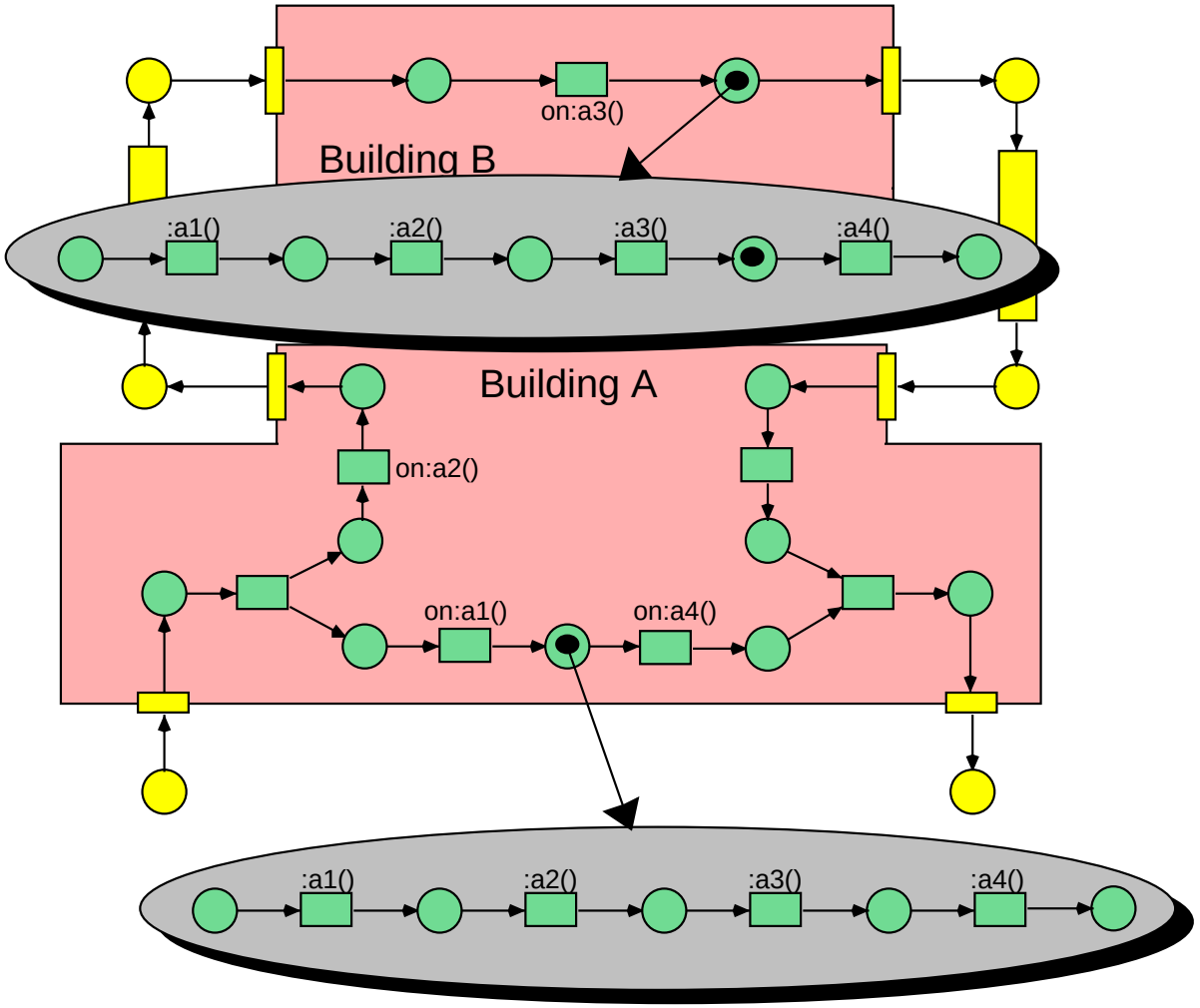
Generalised Semantics



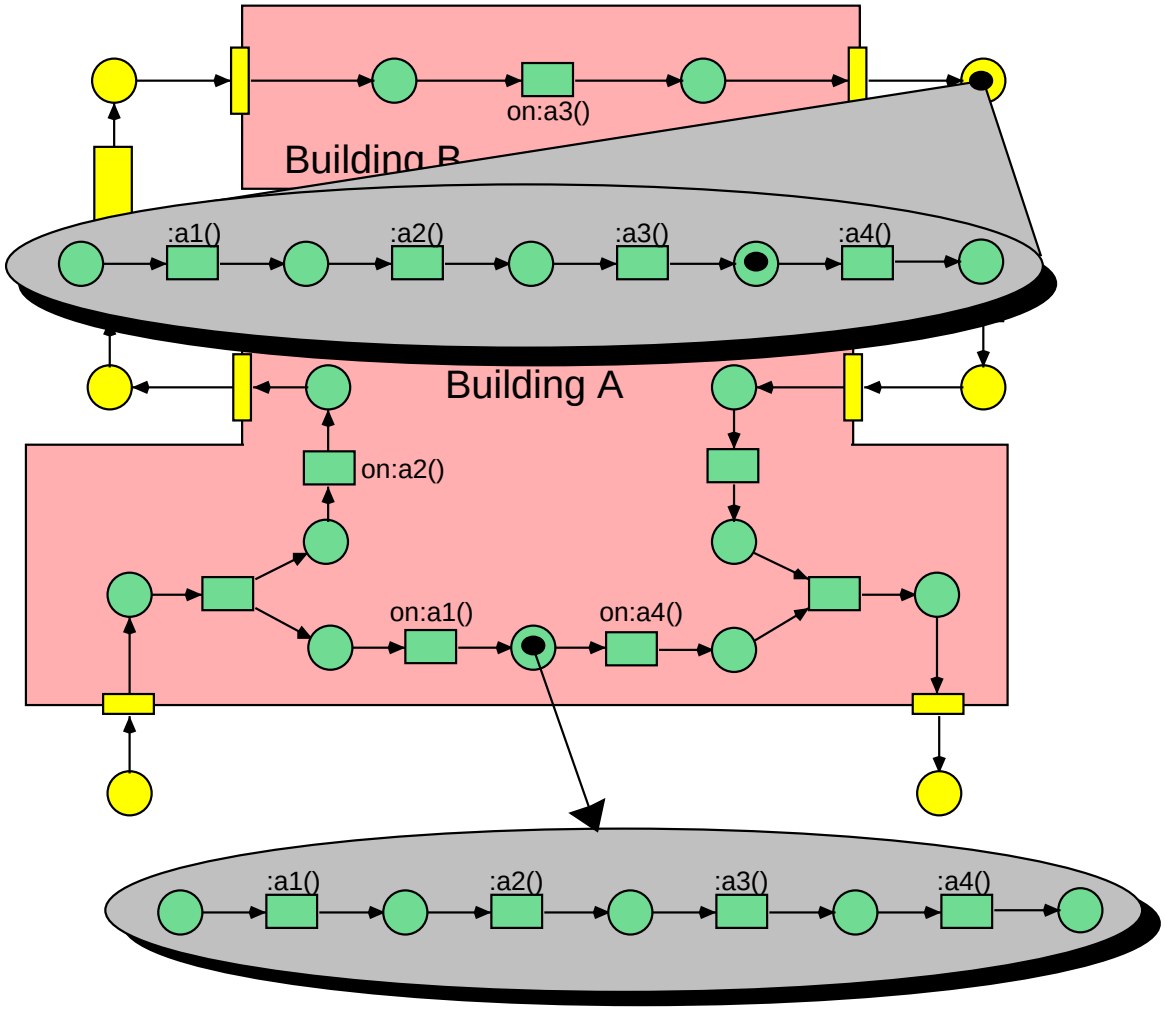
Generalised Semantics



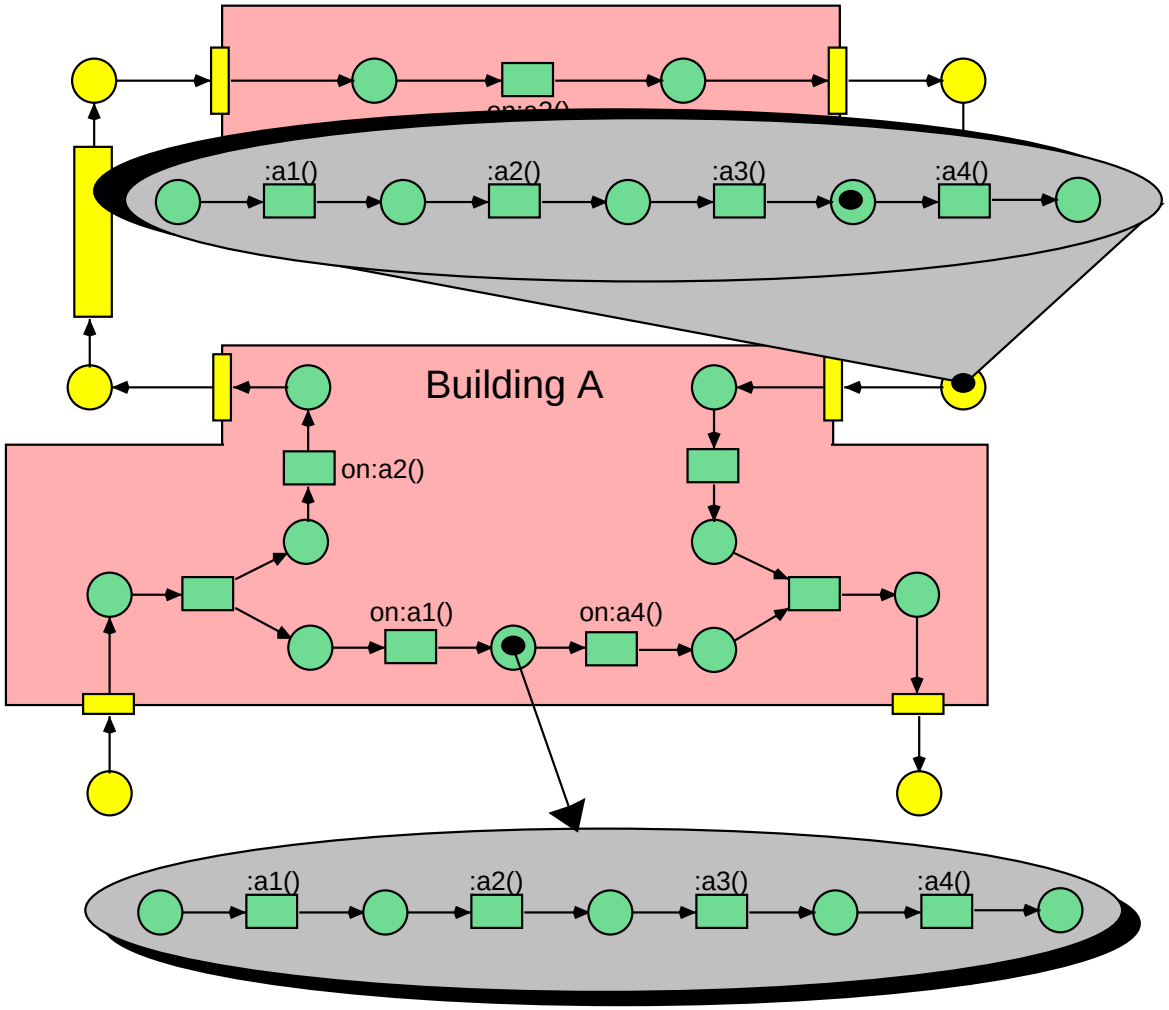
Generalised Semantics



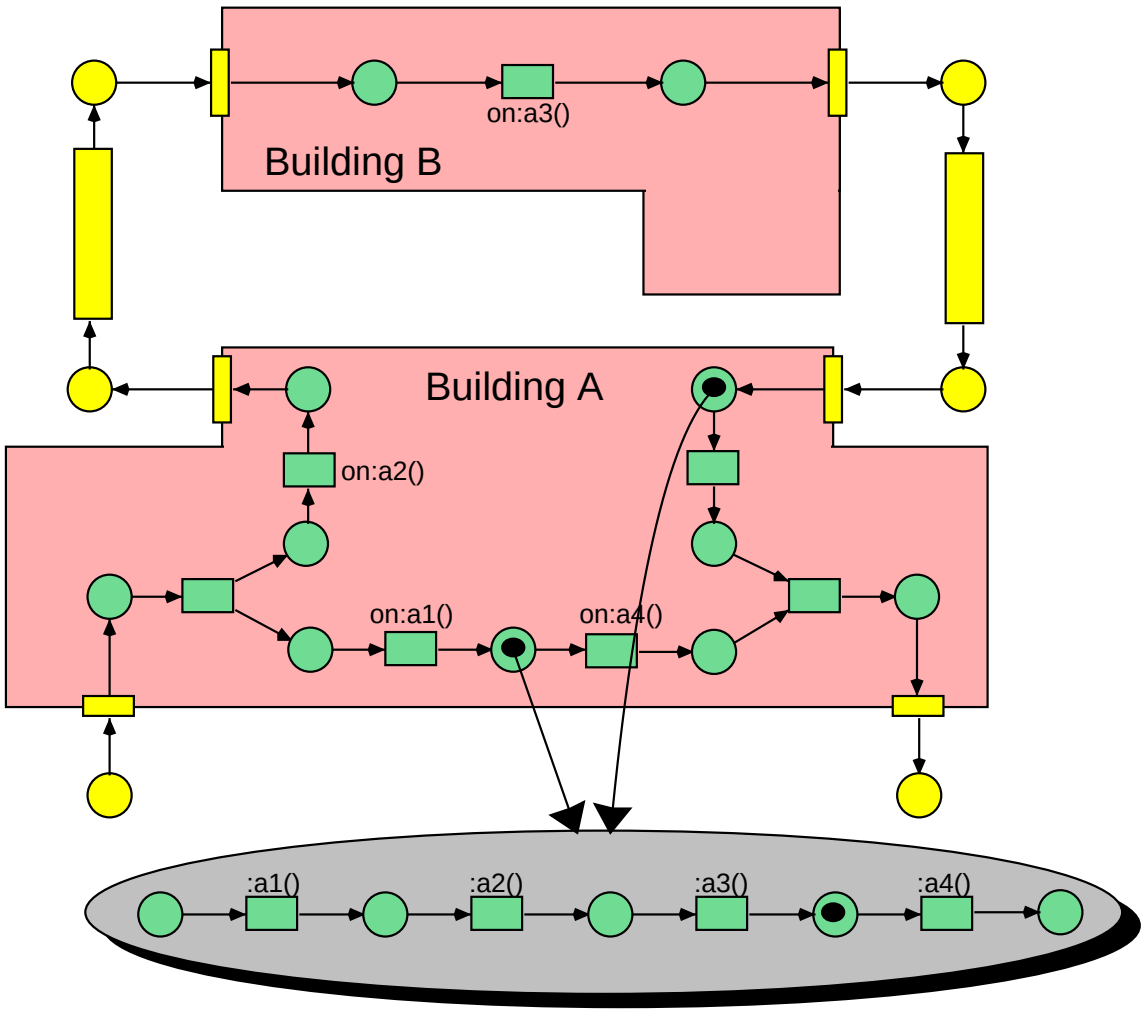
Generalised Semantics



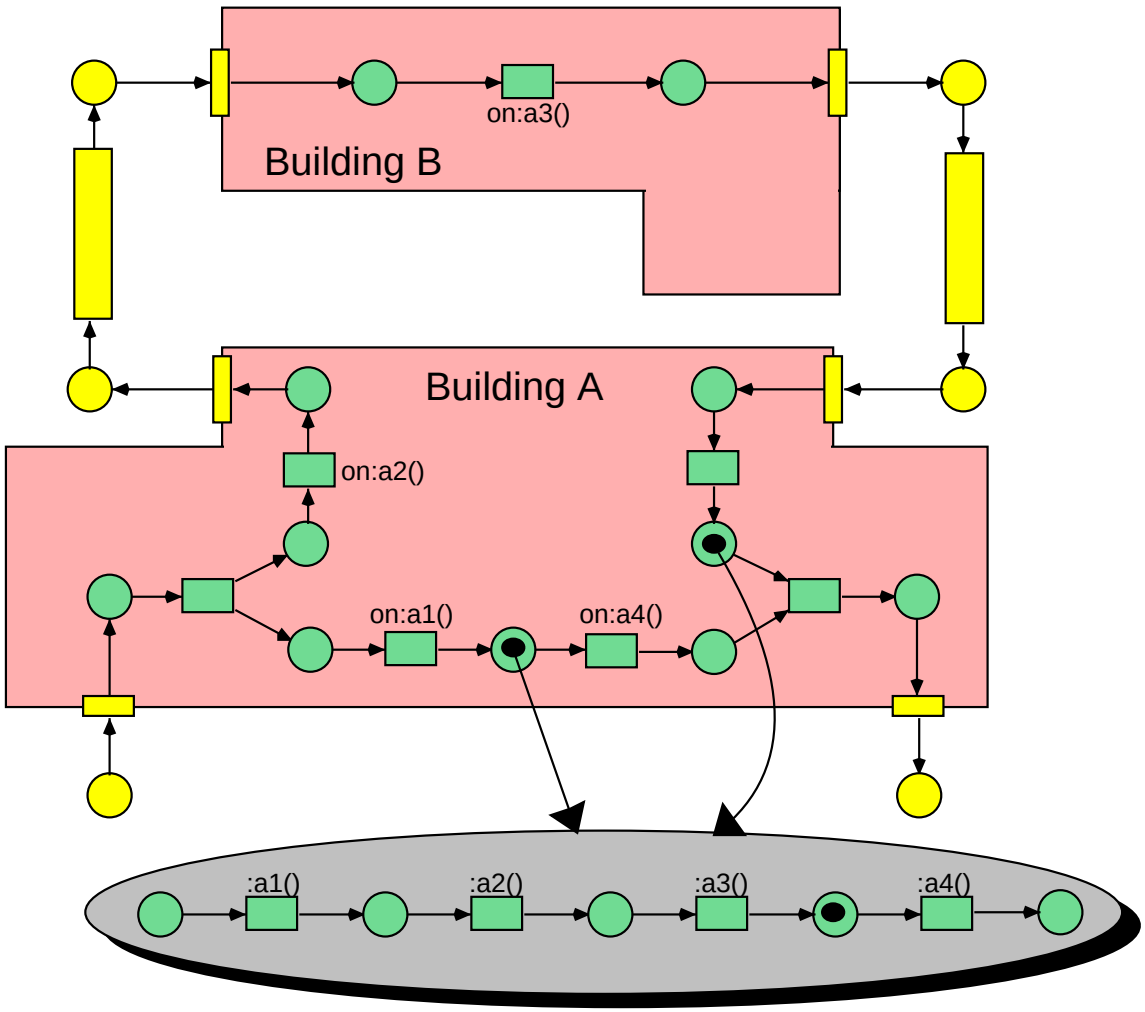
Generalised Semantics



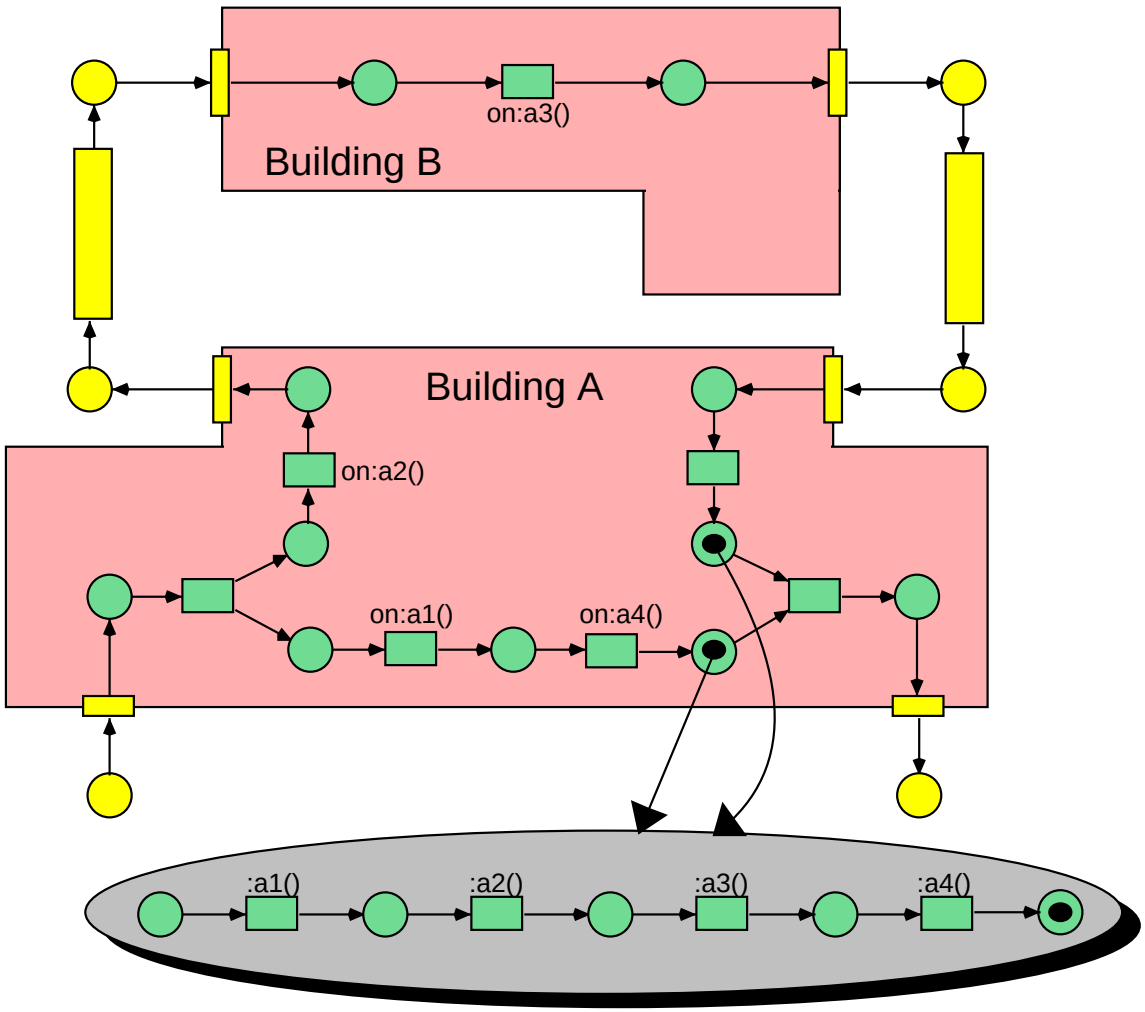
Generalised Semantics



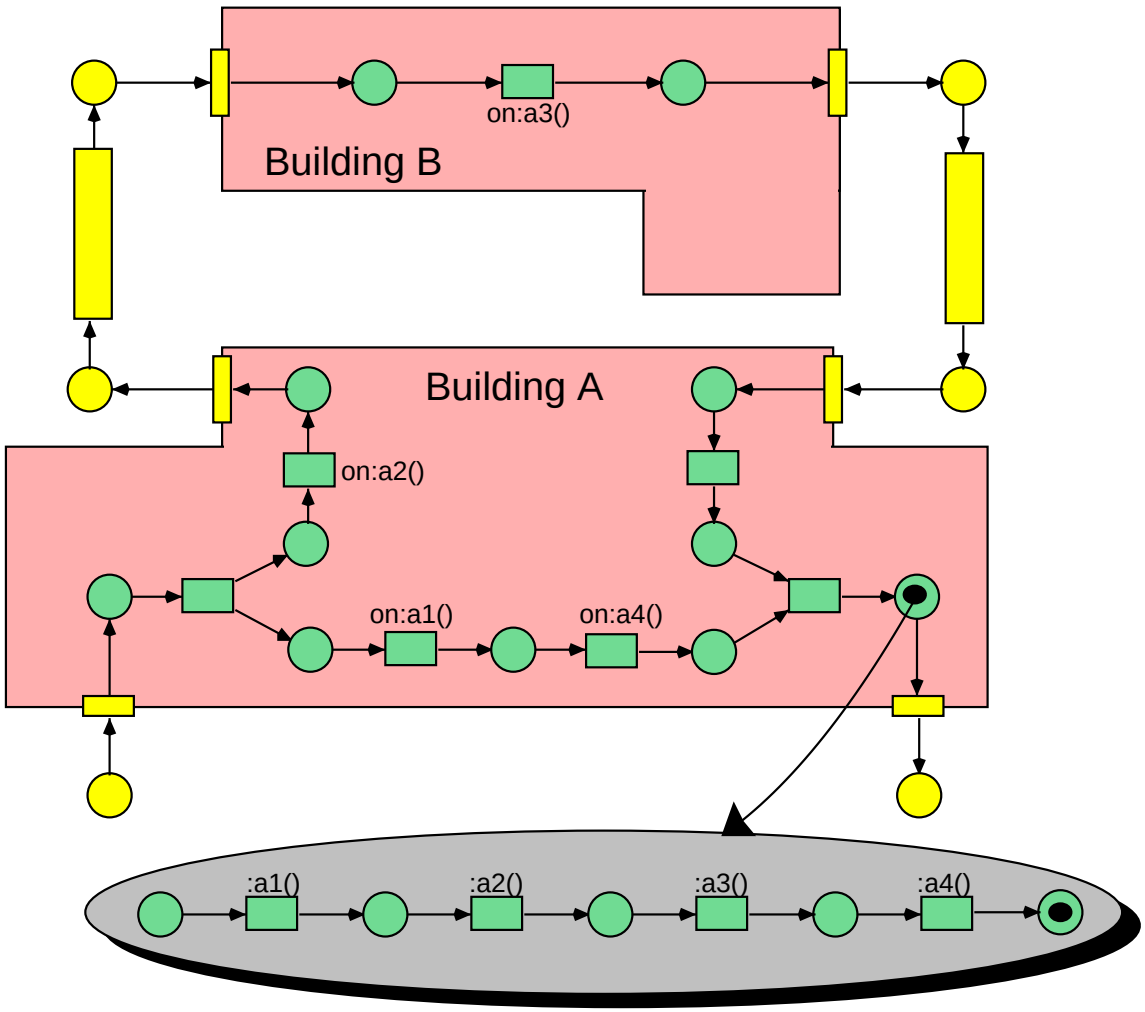
Generalised Semantics



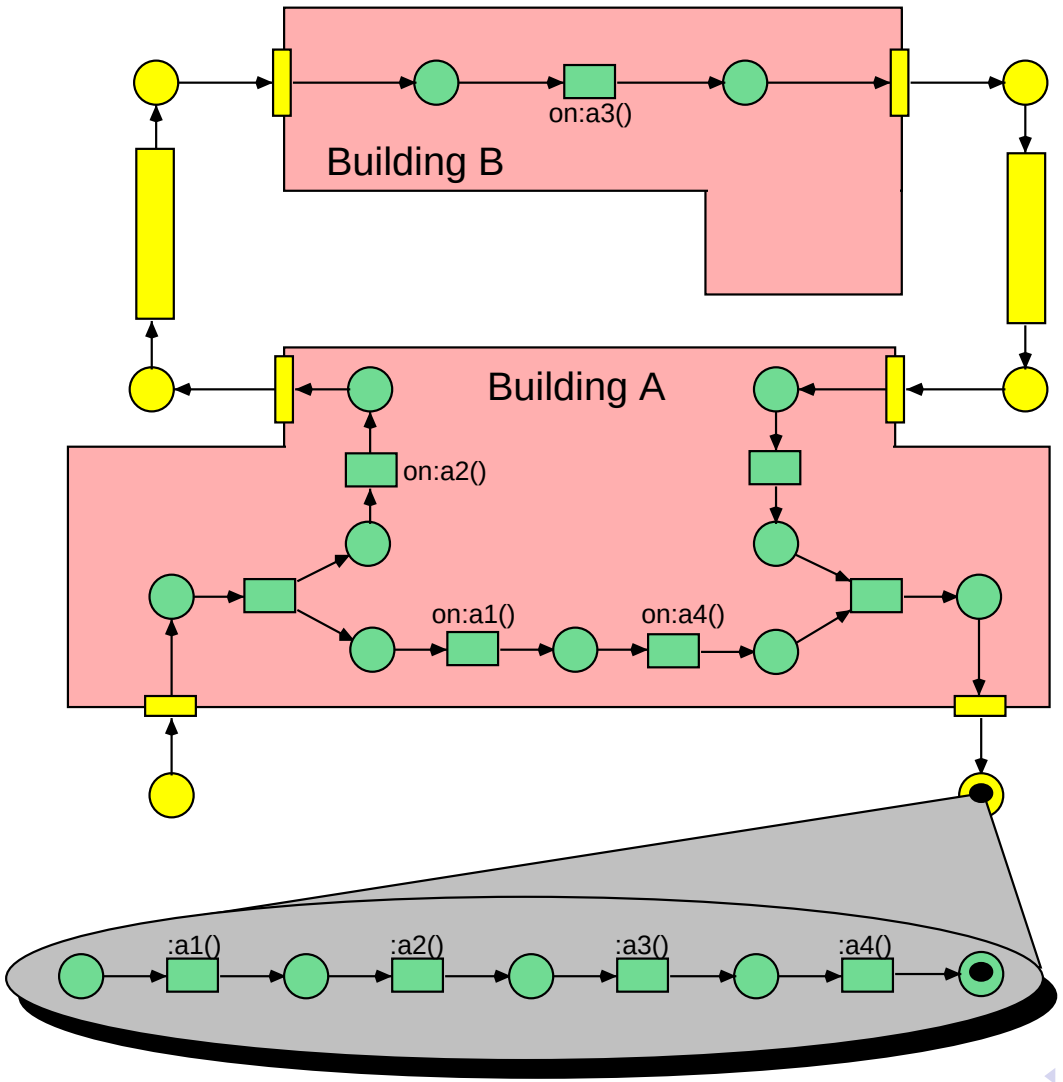
Generalised Semantics



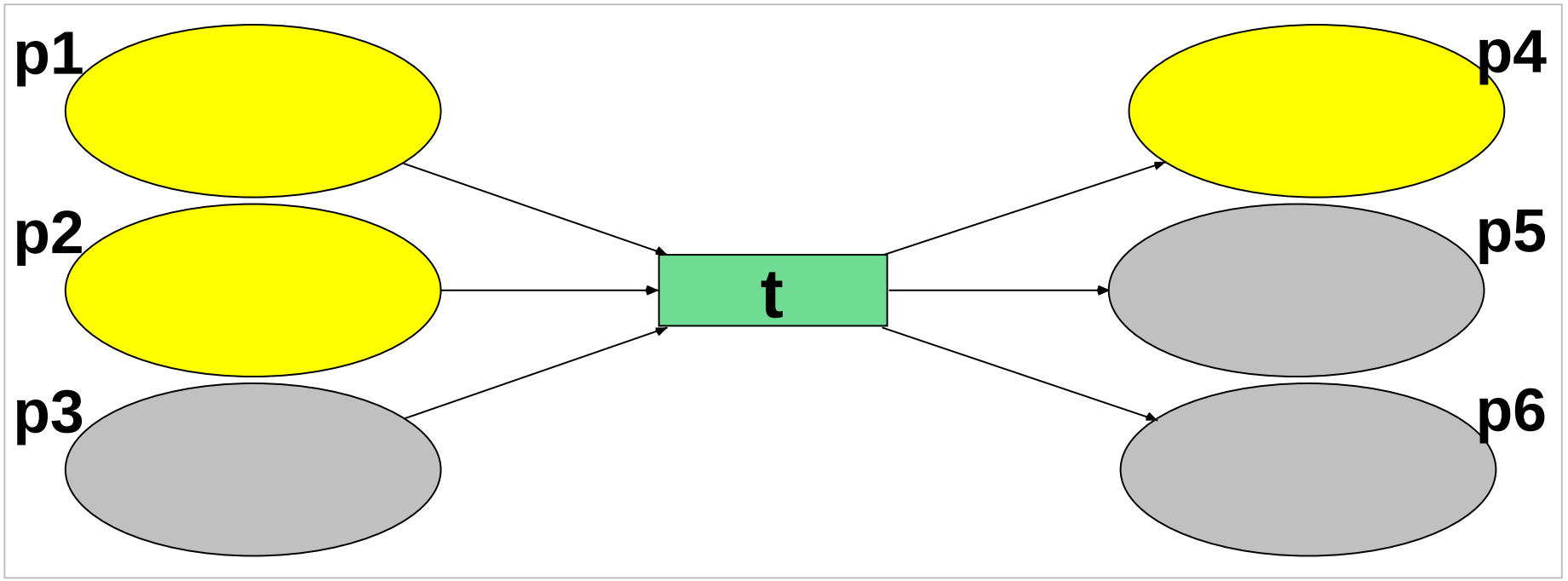
Generalised Semantics



Generalised Semantics



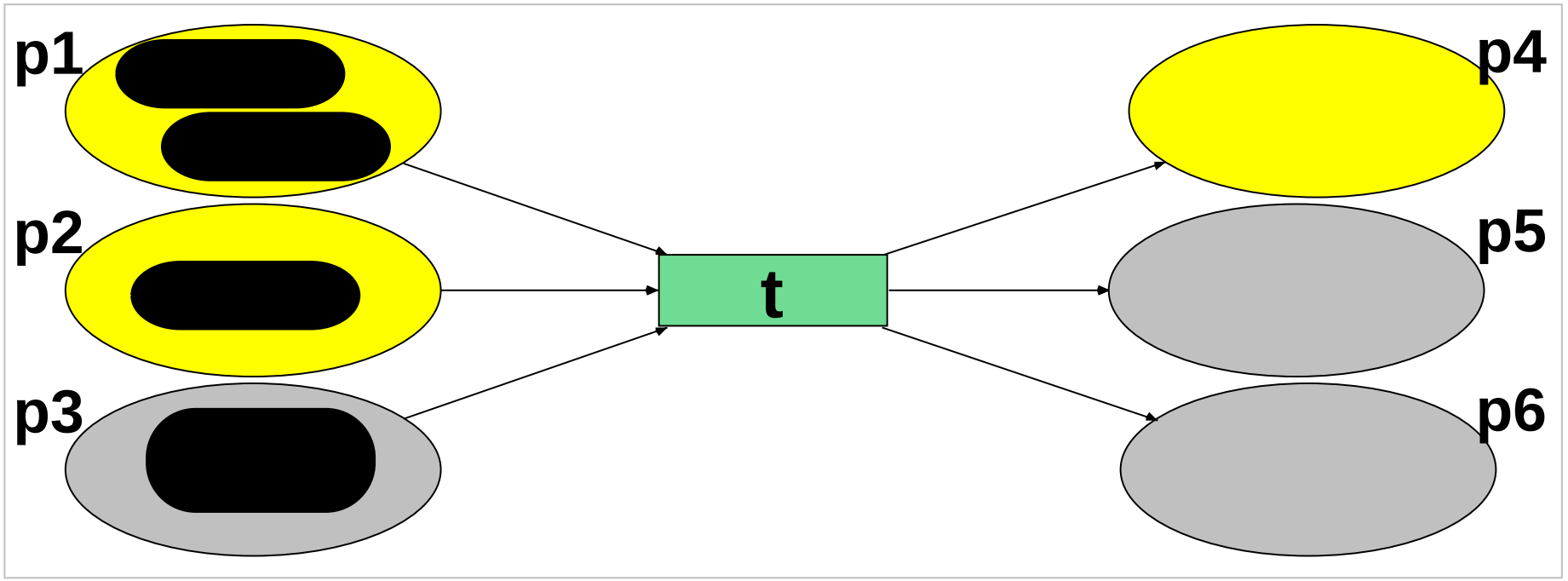
Example



Places p with different types:

$$\begin{aligned} d(p_1) &= d(p_1) = d(p_4) = N_1 \\ d(p_3) &= d(p_5) = d(p_6) = N_2 \end{aligned}$$

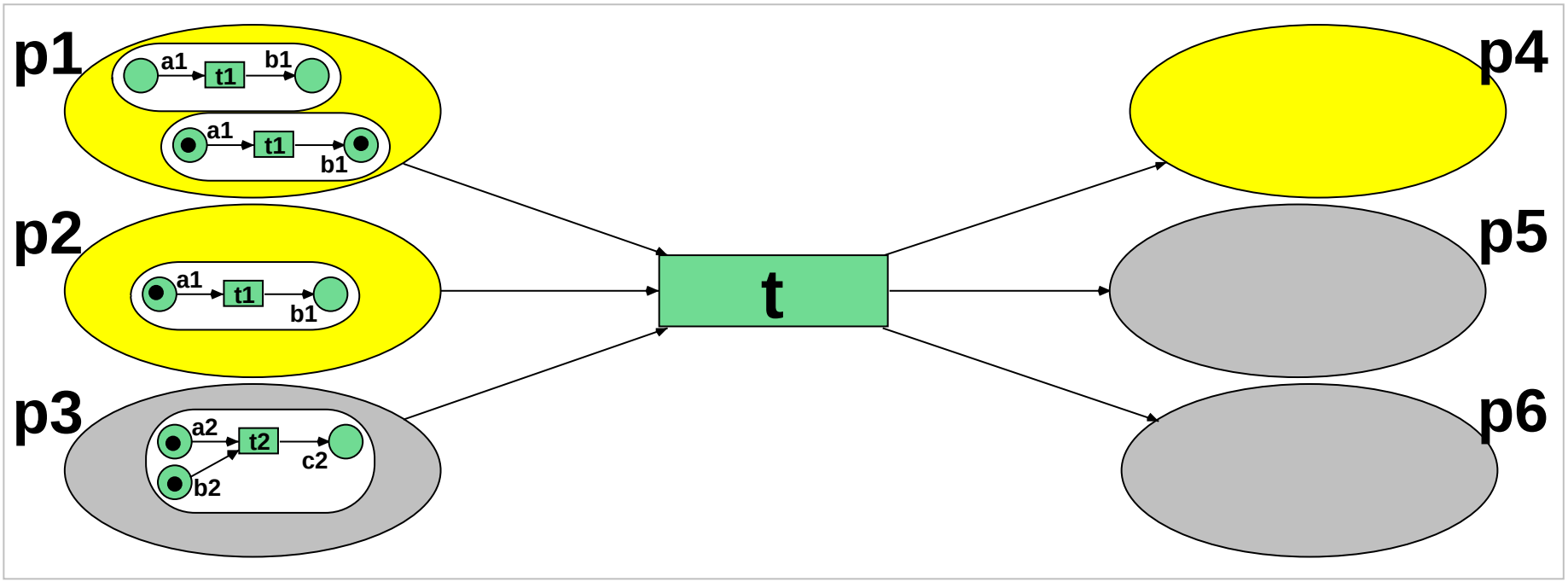
Example



Marking of the system net:

$$\begin{aligned} \mu &= p_1 \\ &+ p_1 \\ &+ p_2 \\ &+ p_3 \end{aligned}$$

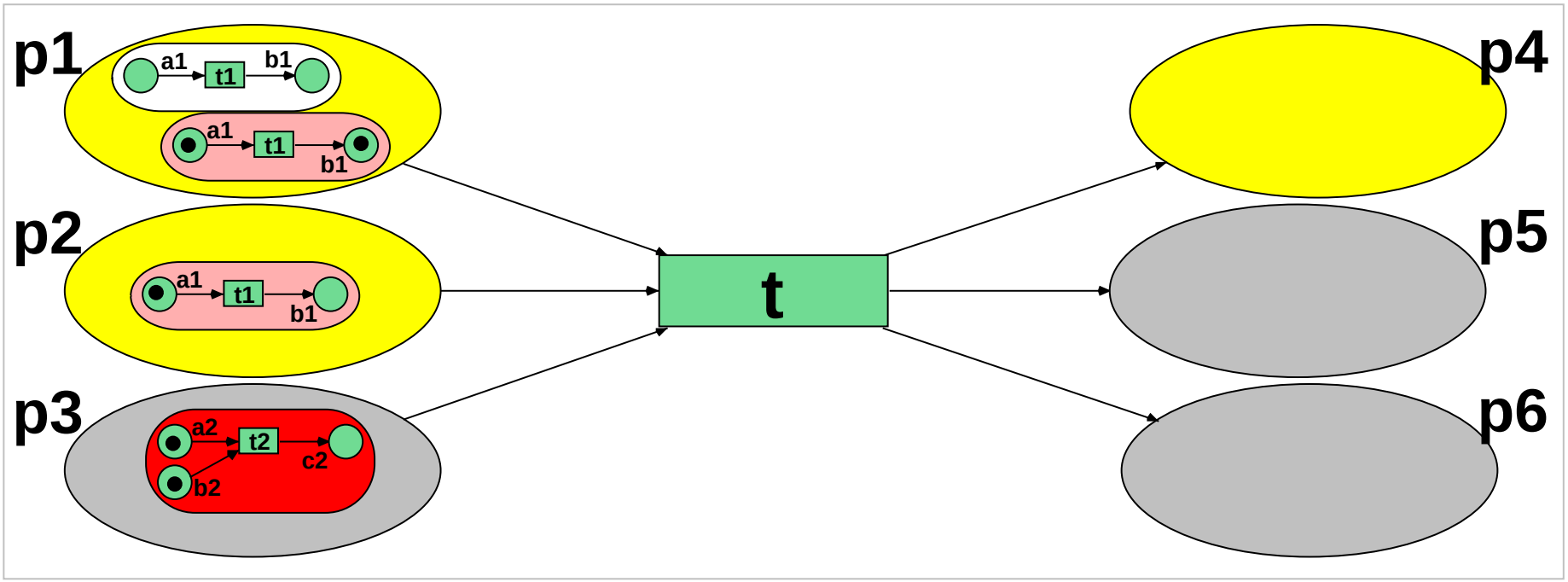
Example



Tokens are net-tokens, which have independent (nested) markings:

$$\begin{aligned} \mu &= (p_1, \mathbf{0}) \\ &+ (p_1, a_1 + b_1) \\ &+ (p_2, a_1) \\ &+ (p_3, a_2 + b_2) \end{aligned}$$

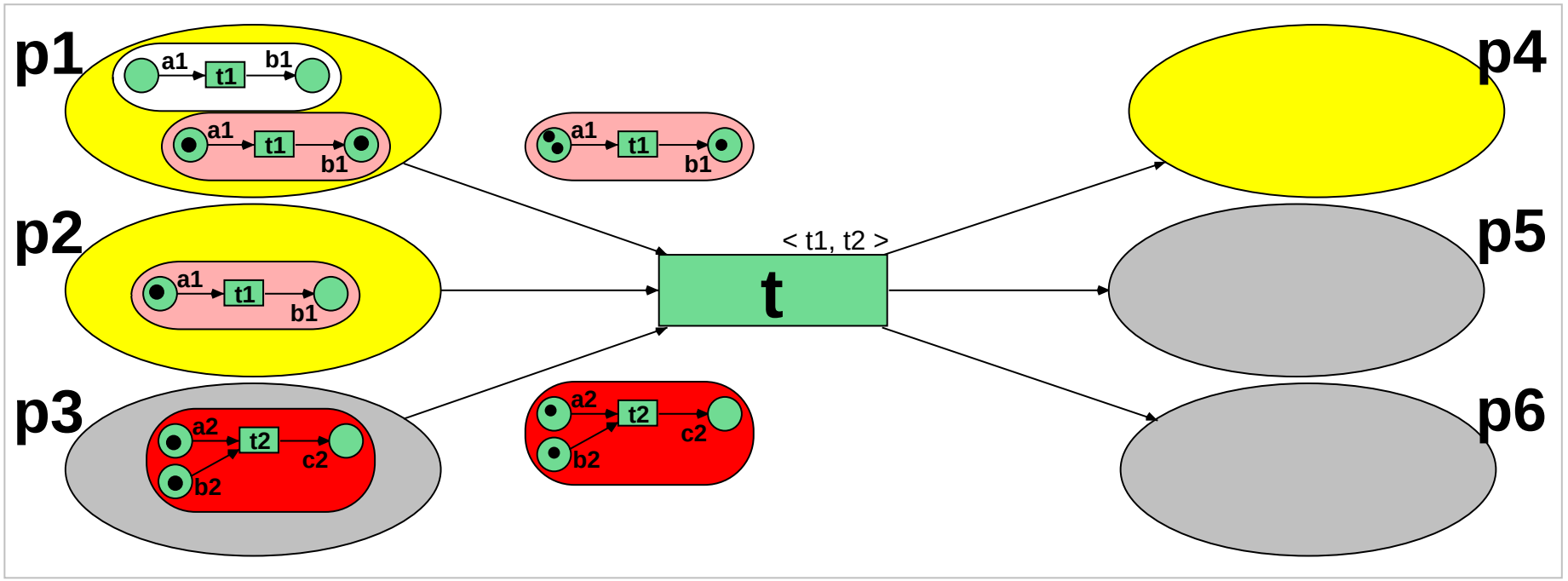
Example



Firing of t selects a subset λ of μ:

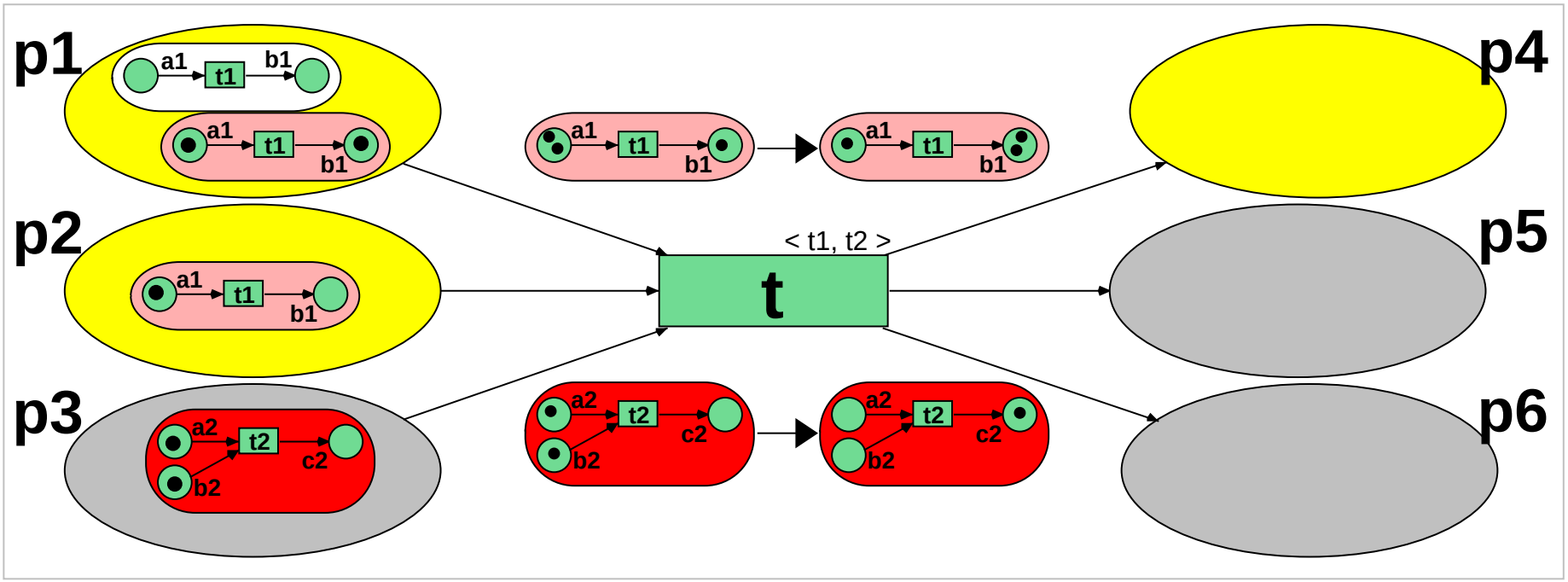
$$\begin{aligned} \lambda &= \cdots \\ &+ (p_1, a_1 + b_1) \\ &+ (p_2, a_1) \\ &+ (p_3, a_2 + b_2) \end{aligned}$$

Example



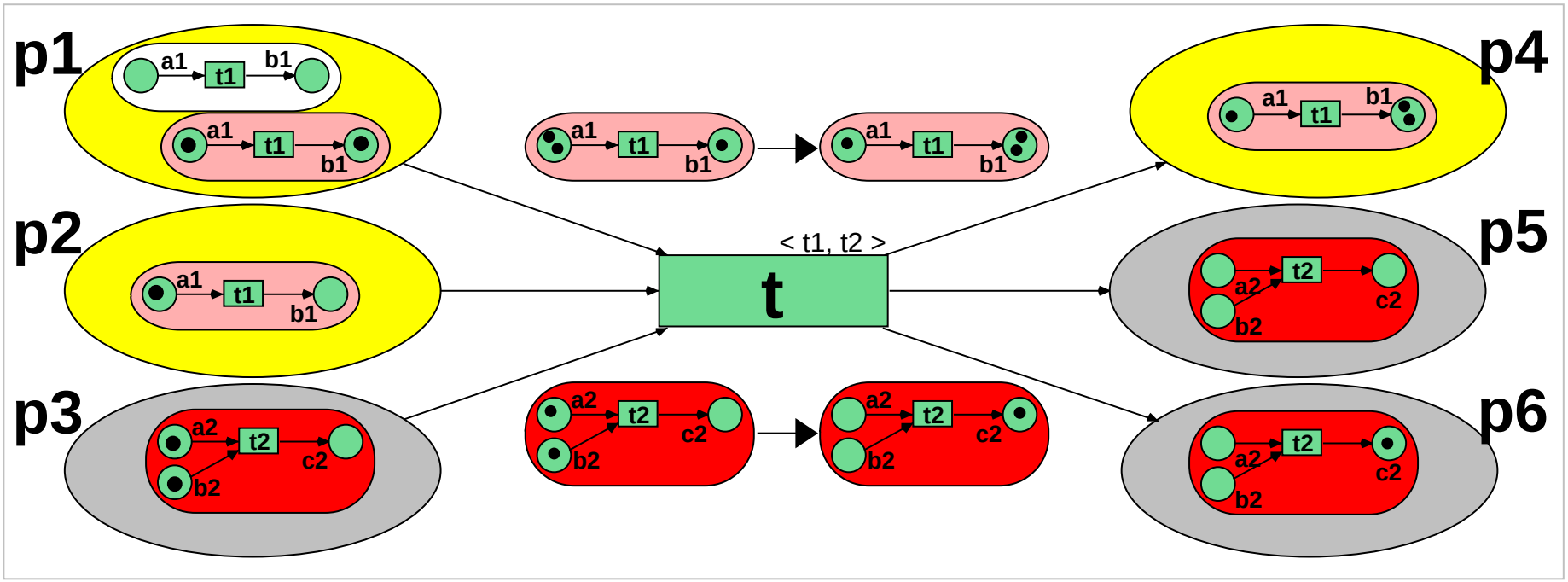
The markings of the net-tokens are added.

Example



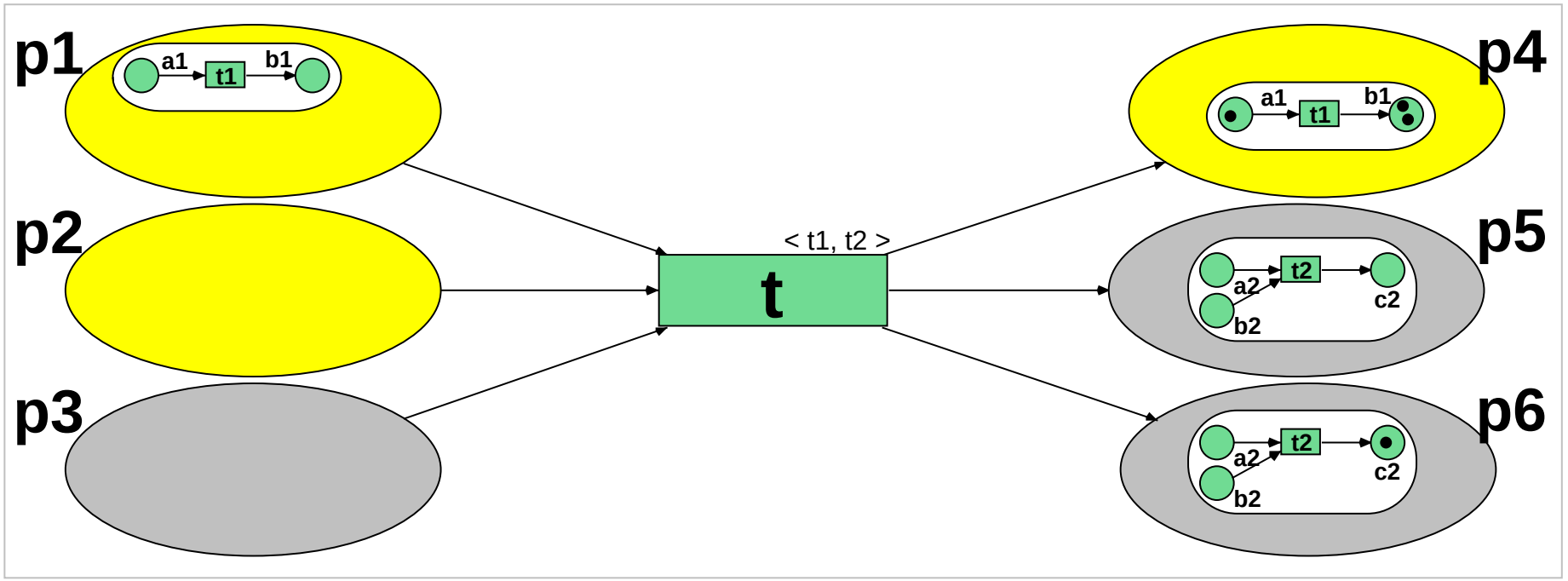
Synchronisation $\theta = (t, \langle t_1, t_2 \rangle)$ of t with t_1 and t_2.

Example



The resulting markings are distributed resulting in ρ.

Example



Successor marking $\mu' = \mu - \lambda + \rho$:

$$\mu \xrightarrow[OS]{(t,\langle t_1,t_2\rangle)(\lambda,\rho)} \mu'$$

Definition

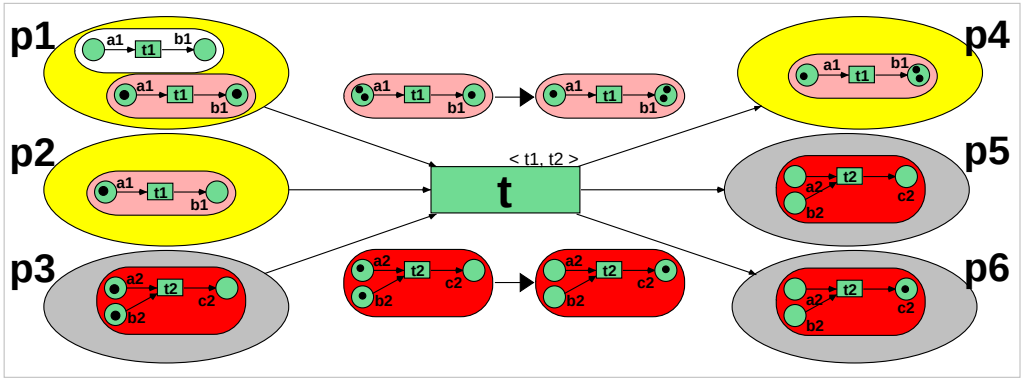
An elementary object system has two nesting levels.

Definition

An *elementary object system* is a tuple $OS = (\widehat{N}, \mathcal{N}, d, \Theta, \mu_0)$

1. $\widehat{N}$ is a P/T-net, called the *system-net*.
2. $\mathcal{N} = \{N_1, \ldots, N_n\}$ is a set of P/T-nets, called *object-nets*.
3. $d : \widehat{P} \to \{\bullet\} \cup \mathcal{N}$ is the typing of the system-net places.
4. $\Theta \subseteq \mathcal{T}$ defines synchronised transitions (system/object).
5. $\mu_0 \in \mathcal{M}$ is the initial marking.

Enabling Predicate



The enabling predicate $\phi((\widehat{\tau}, C), \lambda, \rho)$ holds iff:

1. System: $\Pi^1(\lambda) = \mathbf{pre}(\widehat{\tau})$
2. System: $\Pi^1(\rho) = \mathbf{post}(\widehat{\tau})$
3. Object: $\forall N \in \mathcal{N} : \Pi^2_N(\lambda) \geq \mathbf{pre}(C(N))$
4. Object:
   $\forall N \in \mathcal{N} : \Pi^2_N(\rho) = \Pi^2_N(\lambda) - \mathbf{pre}(C(N)) + \mathbf{post}(C(N))$

Projection Equivalence

The equivalence $\cong$ relates nested markings which coincide in their projections:

$$\alpha \cong \beta \iff \begin{aligned} \Pi^1(\alpha) &= \Pi^1(\beta) \wedge \\ \forall N \in \mathcal{N} : \Pi^2_N(\alpha) &= \Pi^2_N(\beta) \end{aligned}$$

Lemma

The enabling predicate is invariant w.r.t. $\cong$:

$$\phi((\widehat{\tau}, C), \lambda, \rho) \iff \forall \lambda', \rho' : \lambda' \cong \lambda \wedge \rho' \cong \rho \Longrightarrow \phi((\widehat{\tau}, C), \lambda', \rho')$$

Firing Rule

Definition

The event $(\widehat{\tau}, C) \in \mathcal{T}$ is *enabled* in mode (λ, ρ) if:

$$\lambda \leq \mu \wedge \exists \lambda', \rho' : \quad \begin{array}{c} (\lambda' \cong \lambda) \wedge \\ (\rho' \cong \rho) \wedge \\ \phi((\widehat{\tau}, C), \lambda', \rho') \end{array}$$

Successor marking: $\mu' := \mu - \lambda + \rho$.

Notation: $\mu \xrightarrow[OS]{(\widehat{\tau}, C)(\lambda, \rho)} \mu'$.

Synchronisation and autonomous firing

Synchronisation is the general case.

- System-autonomous firing: All the object nets fire the idle event ϵ_N with $\mathbf{pre}(\epsilon_N) = \mathbf{post}(\epsilon_N) = \mathbf{0}$.
- Object-autonomous firing: The object net fires in the place $\widehat{p}$ which is expressed by the idle event of the system net: $\epsilon_{\widehat{p}}$ where $\widehat{\mathbf{pre}}(\epsilon_{\widehat{p}}) = \widehat{\mathbf{post}}(\epsilon_{\widehat{p}}) = \widehat{p}$.

1. Object Nets
   - Reference Semantics
   - Value Semantics
   - New: Mobility Semantics
2. Elementary Object Systems
   - Synchronous Firing
   - Enabling Predicate
   - Firing Rule
3. Mobile EOS
   - Conversion
   - Expressiveness

Mobile EOS

- For Eos each place is its own name space.
- Clusters of places belonging to the same name space.
- An equivalence $\leftrightarrow$ on the object net's places $P = \bigcup_{N \in \mathcal{N}} P_N$ is called a *conversion equivalence*.

Definition

A *mobile* EOS is a pair $(OS, \leftrightarrow)$ where OS is an Eos and $\leftrightarrow$ is a conversion equivalence on the object nets' places $P = \bigcup_{N \in \mathcal{N}} P_N$.

Conversion Equivalence

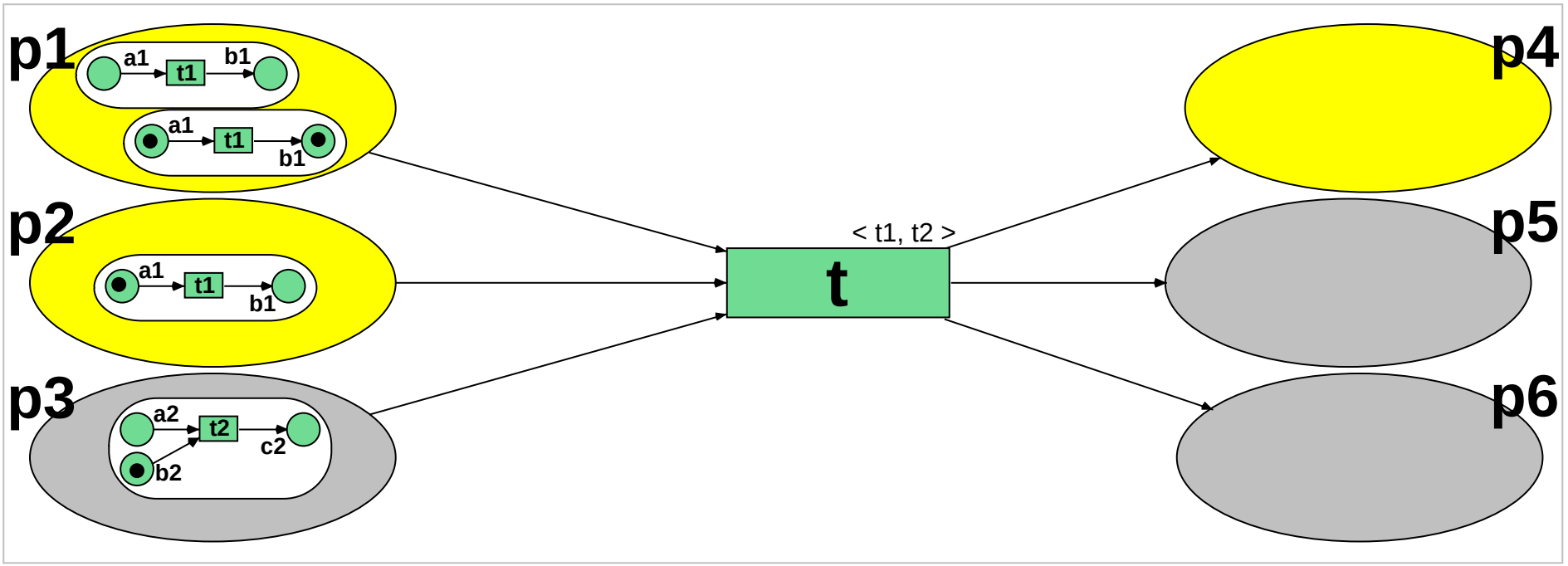
Conversion equivalence for nested markings:

$$\alpha \leftrightarrow \beta \quad \Longleftrightarrow \quad \begin{aligned} \Pi^1(\alpha) &= \Pi^1(\beta) \wedge \\ \forall N \in \mathcal{N} : \Pi^2_N(\alpha) &= \Pi^2_N(\beta) \pmod{\leftrightarrow} \end{aligned}$$

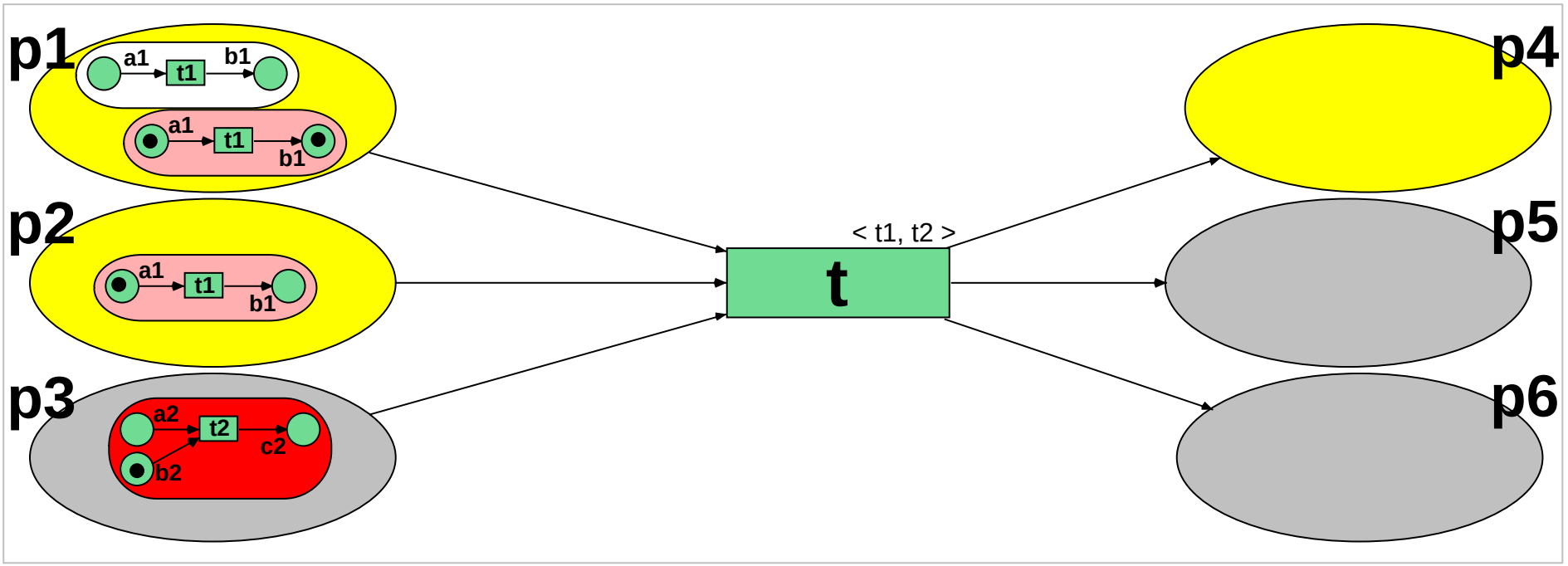
Lemma

For all conversions $\leftrightarrow$ we have $\alpha \cong \beta \Longrightarrow \alpha \leftrightarrow \beta$.
In particular, for $\leftrightarrow = id_P$ we have $\alpha \cong \beta \iff \alpha \leftrightarrow \beta$.

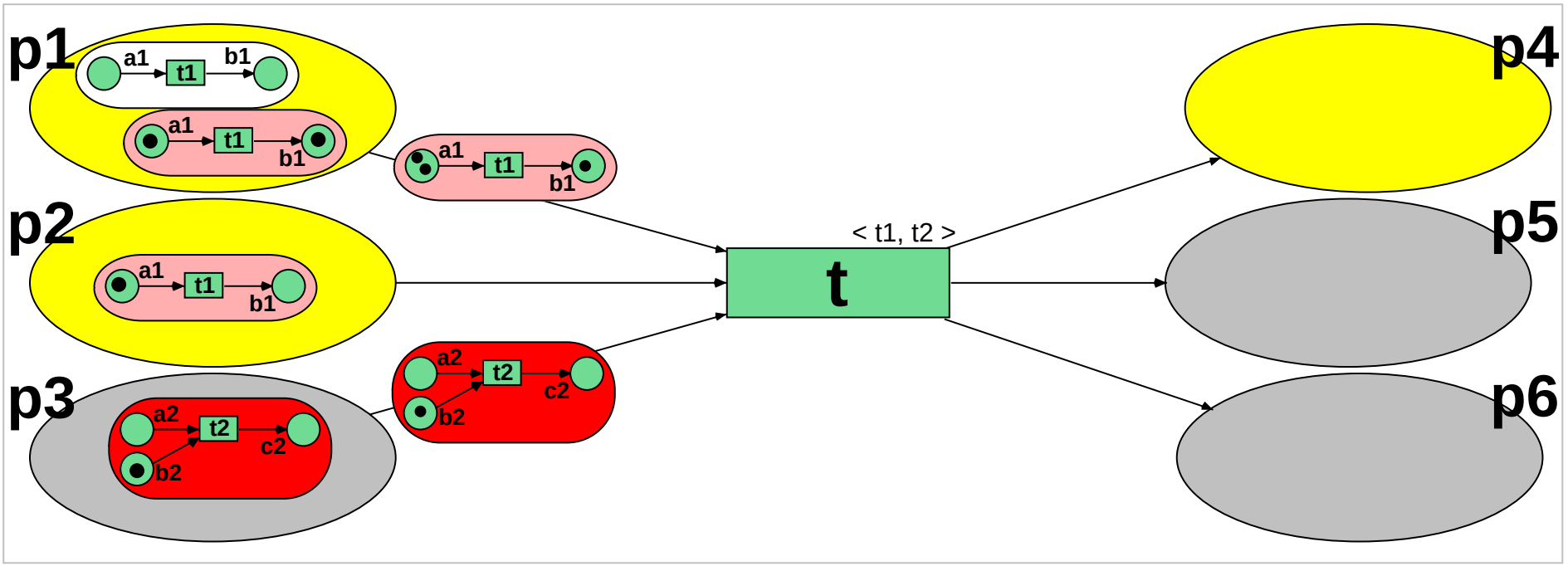
Example



Example

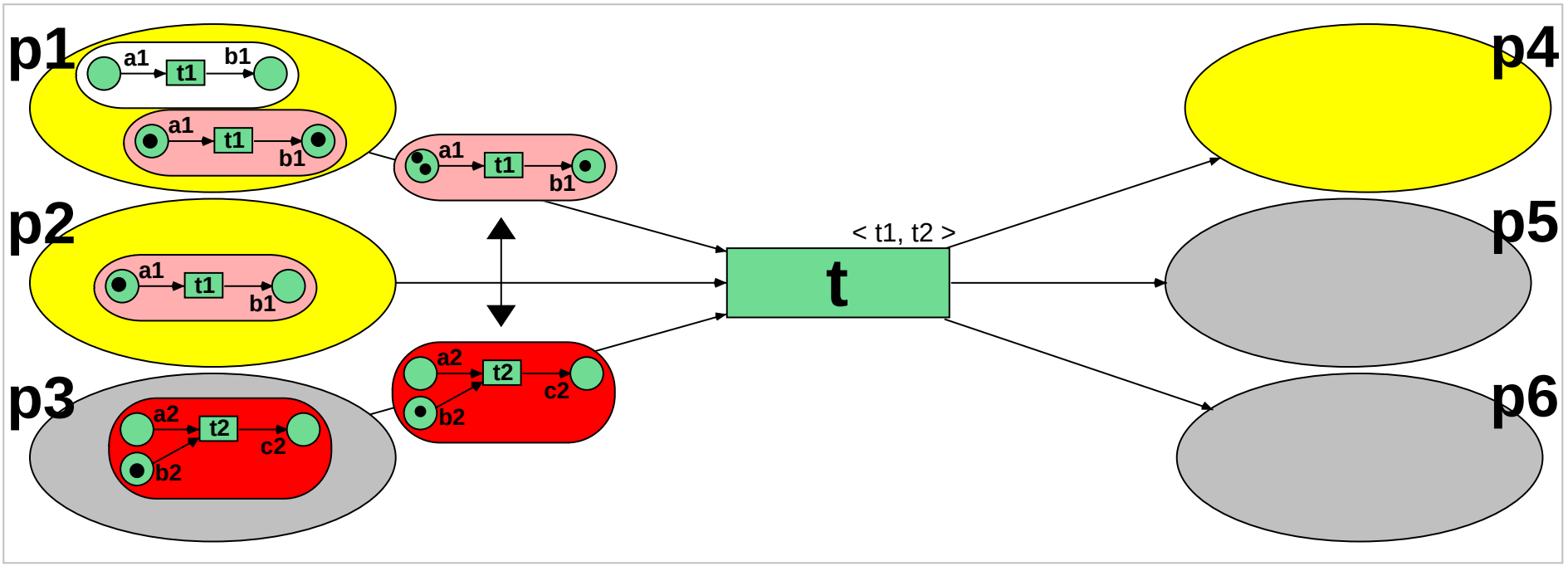


Example



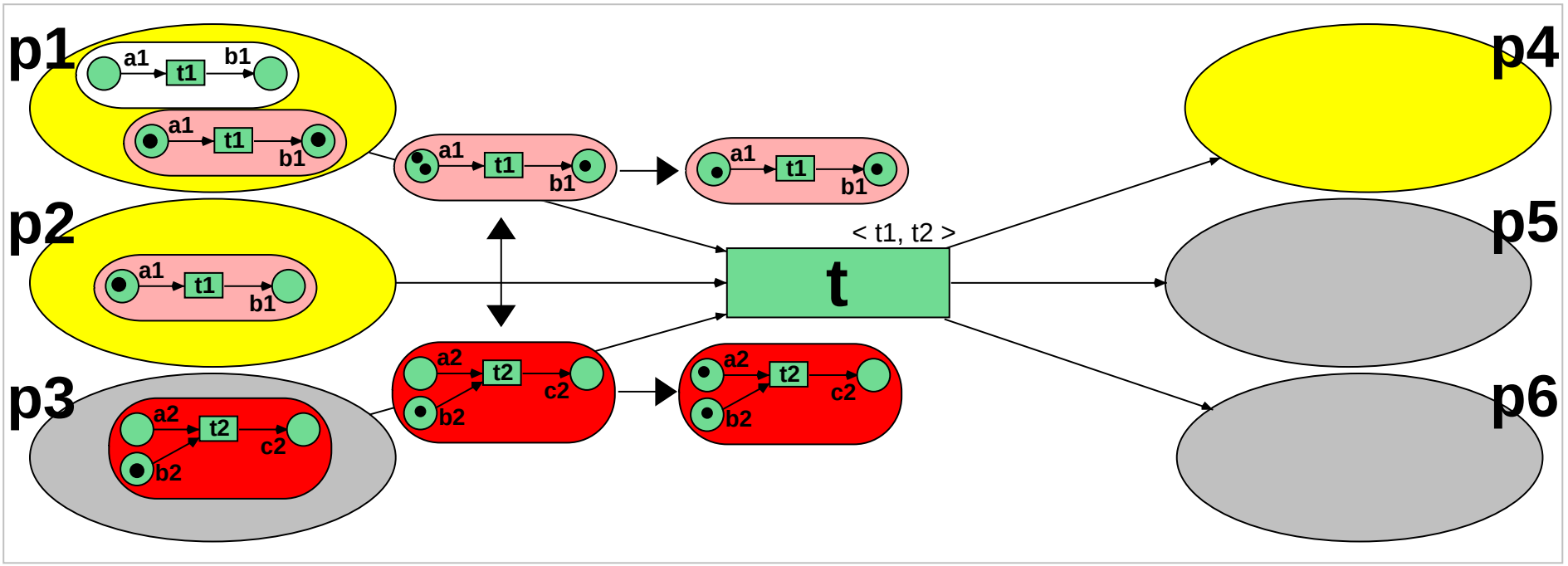
Synchronisation with t_2 is disabled.

Example



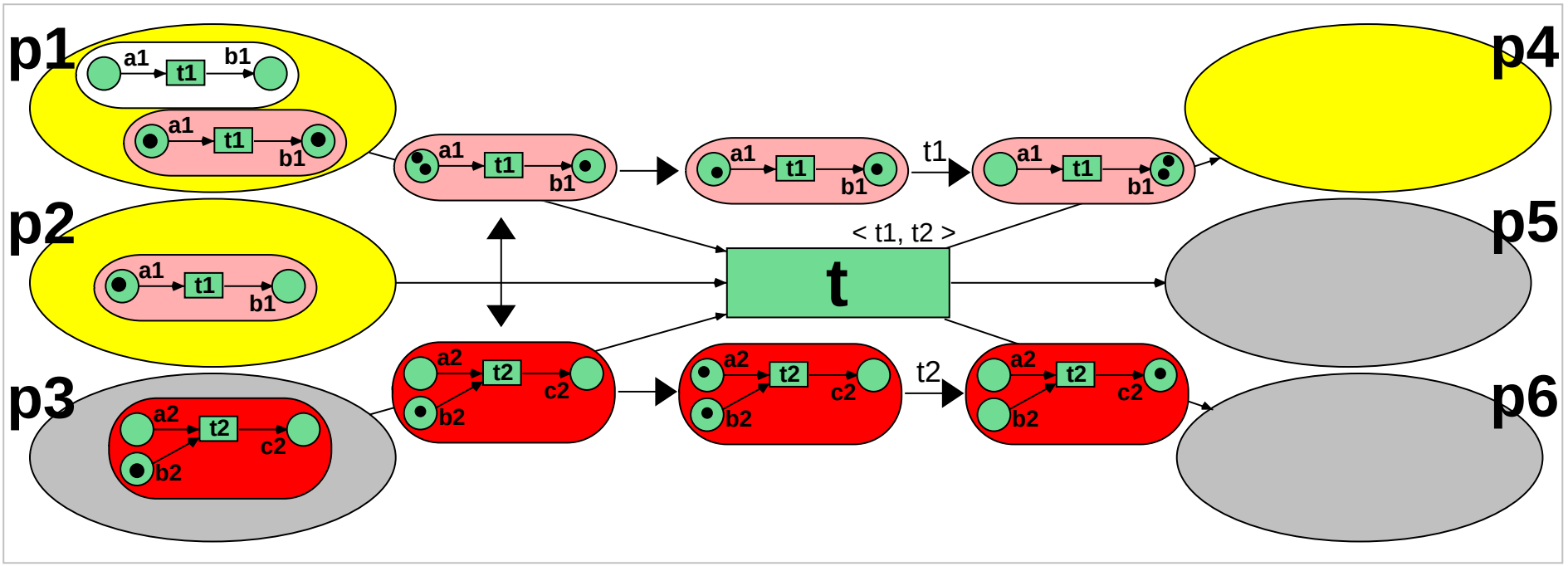
Conversion: $a_1 \leftrightarrow a_2$

Example



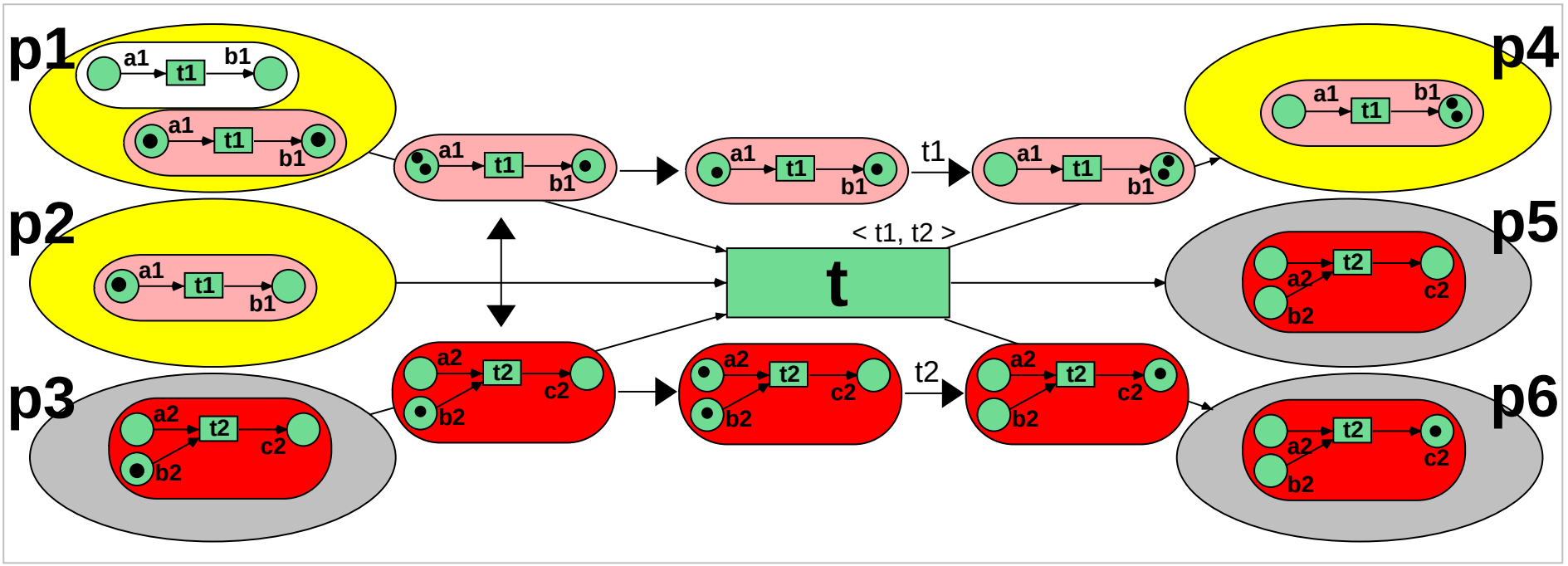
One token is converted from a_1 to a_2.

Example



Synchronisation is enabled.

Example



Firing Rule

Definition

$(\widehat{\tau}, C) \in \mathcal{T}$ is $\leftrightarrow$-*enabled* in mode (λ, ρ) if:

$$\lambda \leq \mu \wedge \exists \lambda', \rho' : \quad (\lambda' \leftrightarrow \lambda) \wedge (\rho' \leftrightarrow \rho) \wedge \phi((\widehat{\tau}, C), \lambda', \rho')$$

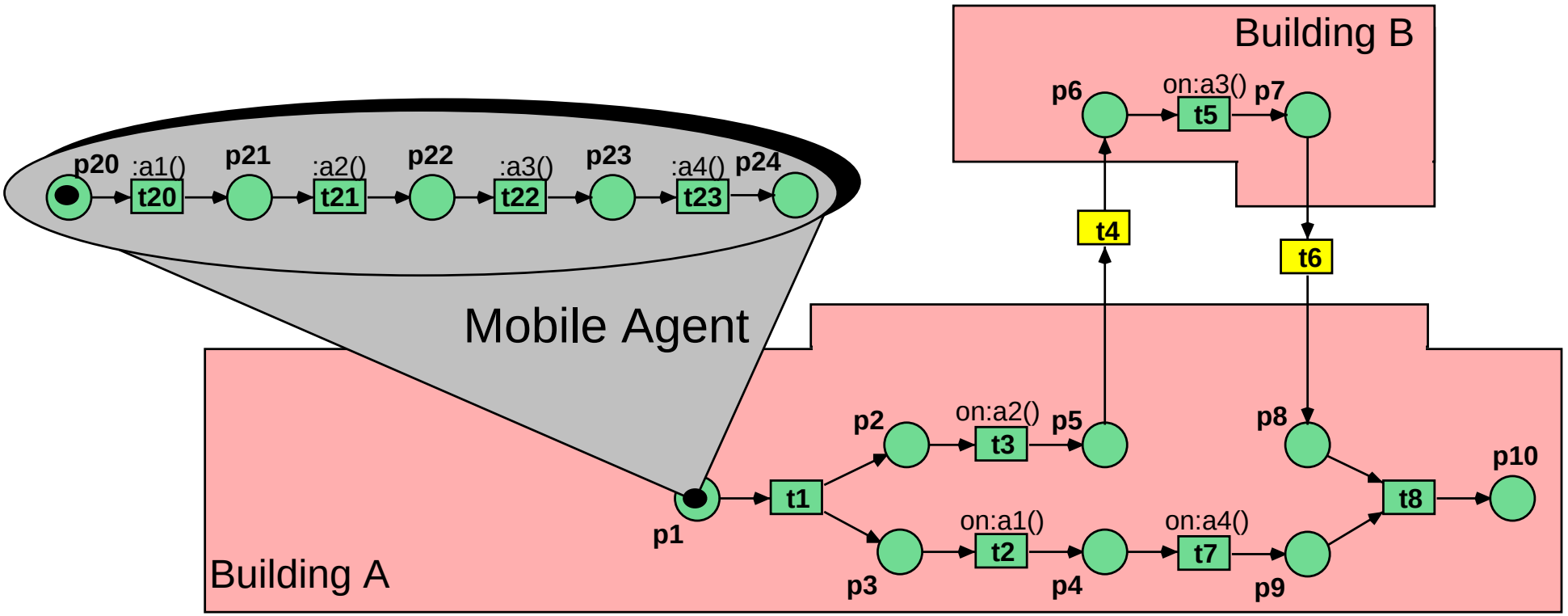
The successor marking is defined as $\mu' := \mu - \lambda + \rho$.

The mobile EOS (OS, id_P) behaves like the Eos OS:

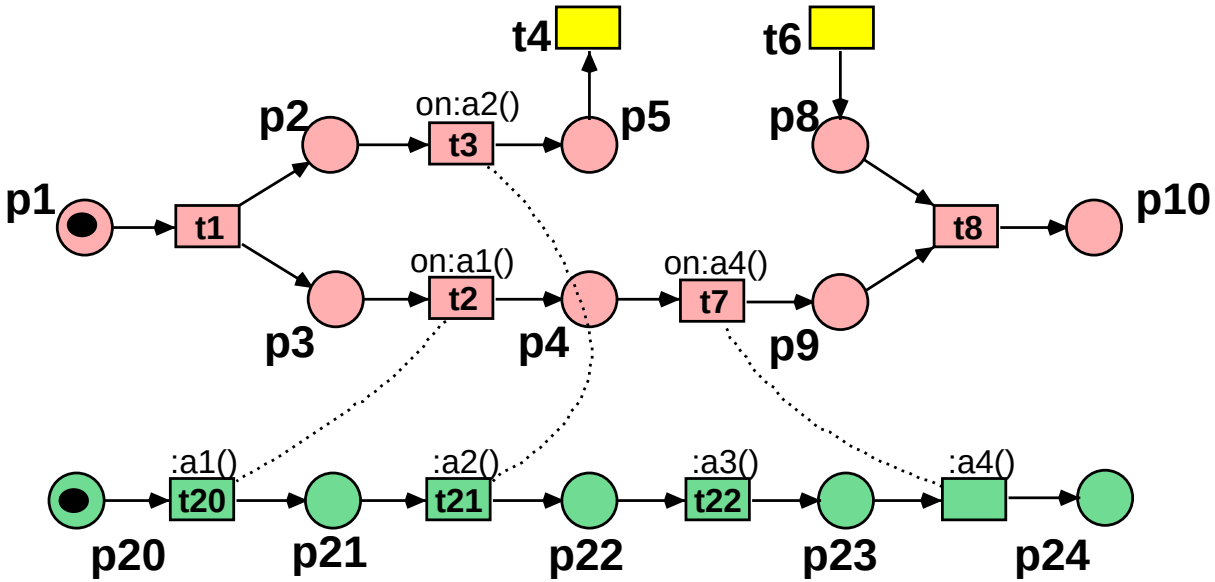
Theorem

The transition $(\widehat{\tau}, C) \in \mathcal{T}$ is enabled in mode (λ, ρ) for OS iff it is id_P-enabled in mode (λ, ρ) for (OS, id_P).

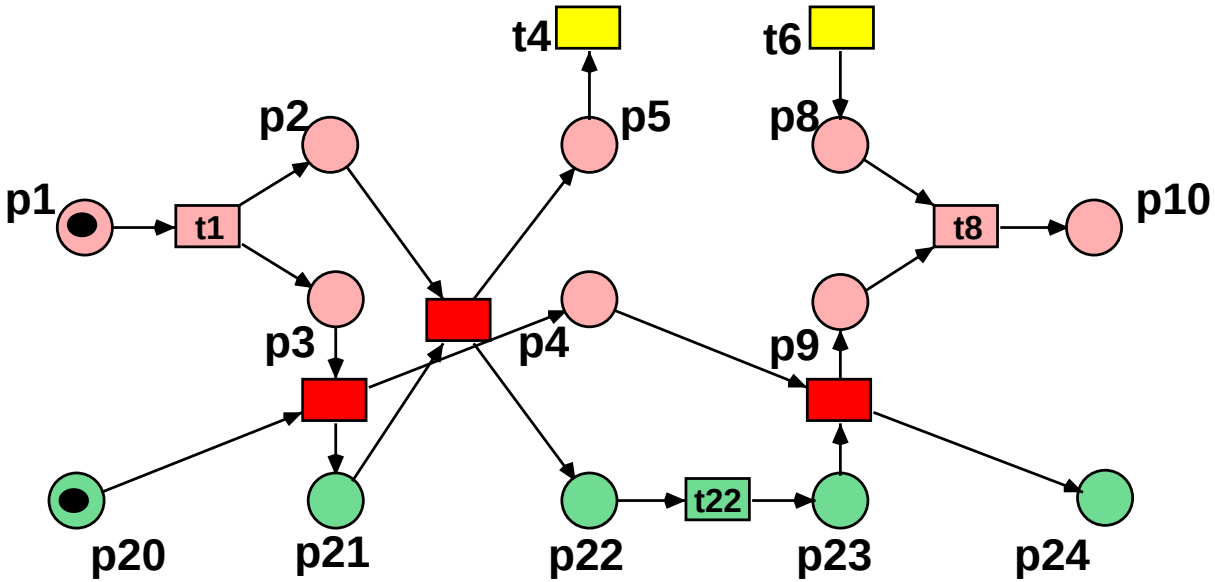
Simulation of Namespaces



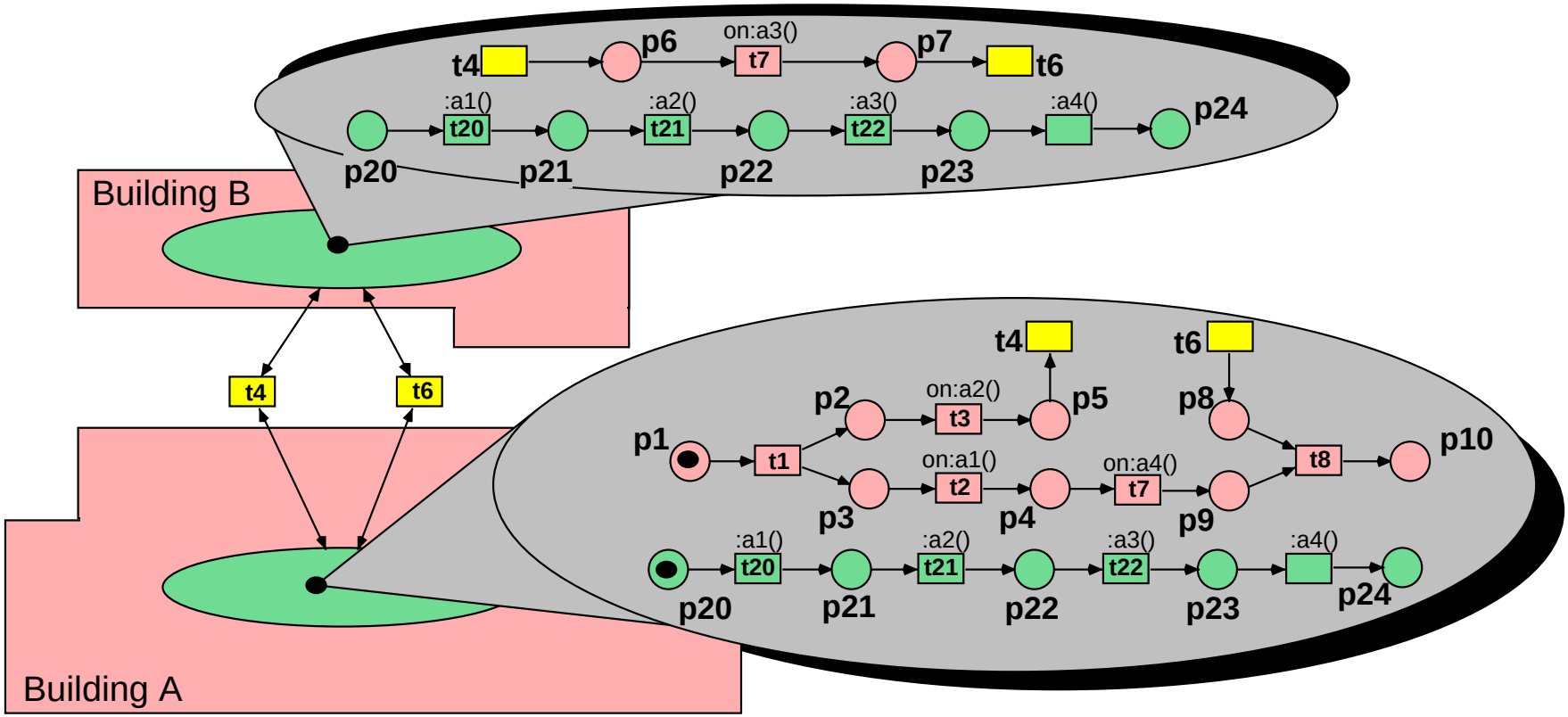
Simulation of Namespaces



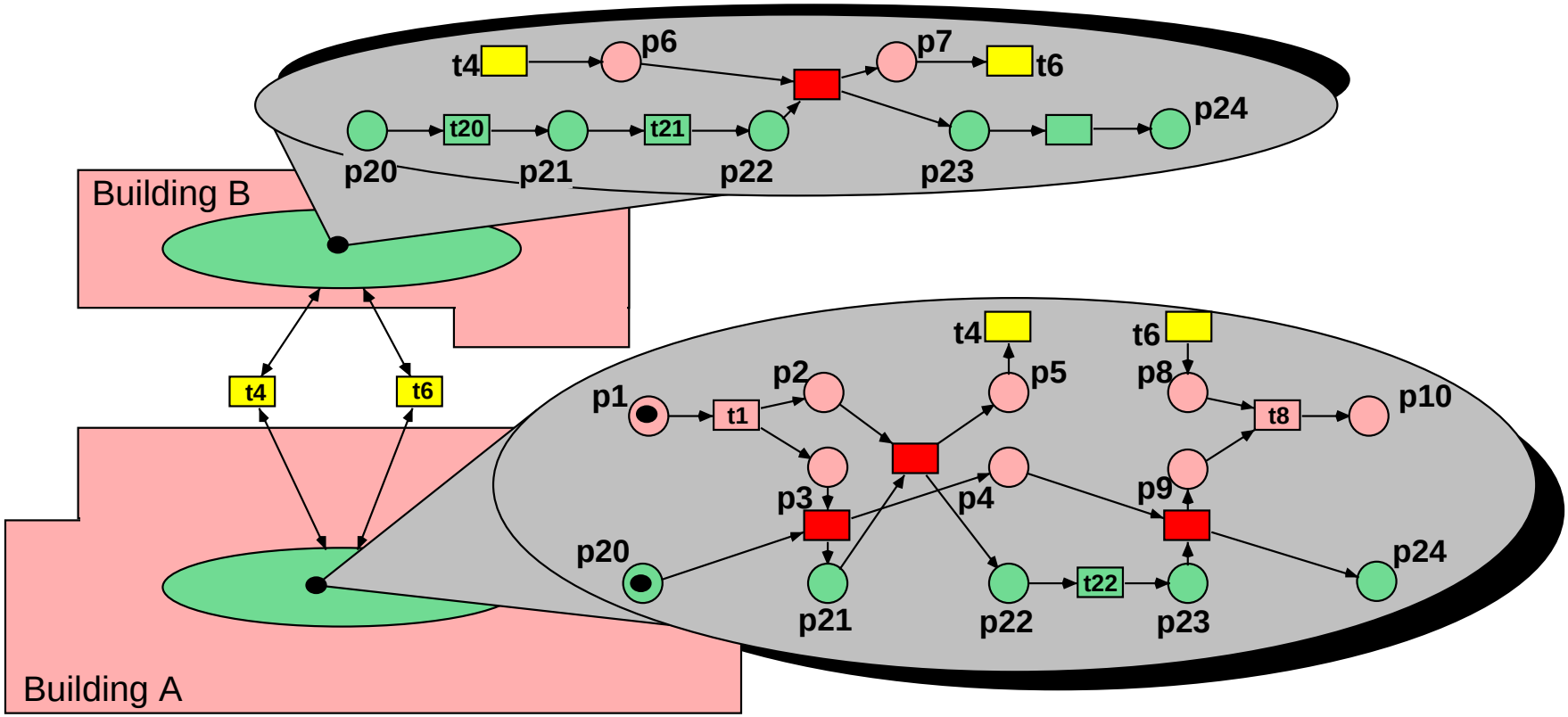
Simulation of Namespaces



Simulation of Namespaces



Simulation of Namespaces



Expressiveness

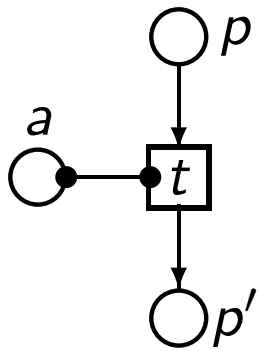
Theorem

Mobile Eos *can simulate Petri nets with inhibitor arcs.*

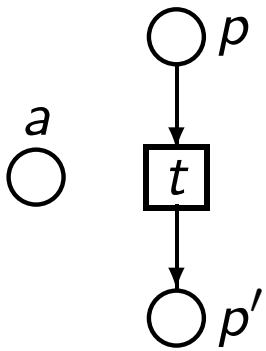
Expressiveness

Theorem

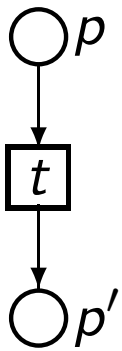
Mobile Eos *can simulate Petri nets with inhibitor arcs.*



(a) Inhibitor net (N, A)



(b) Net N

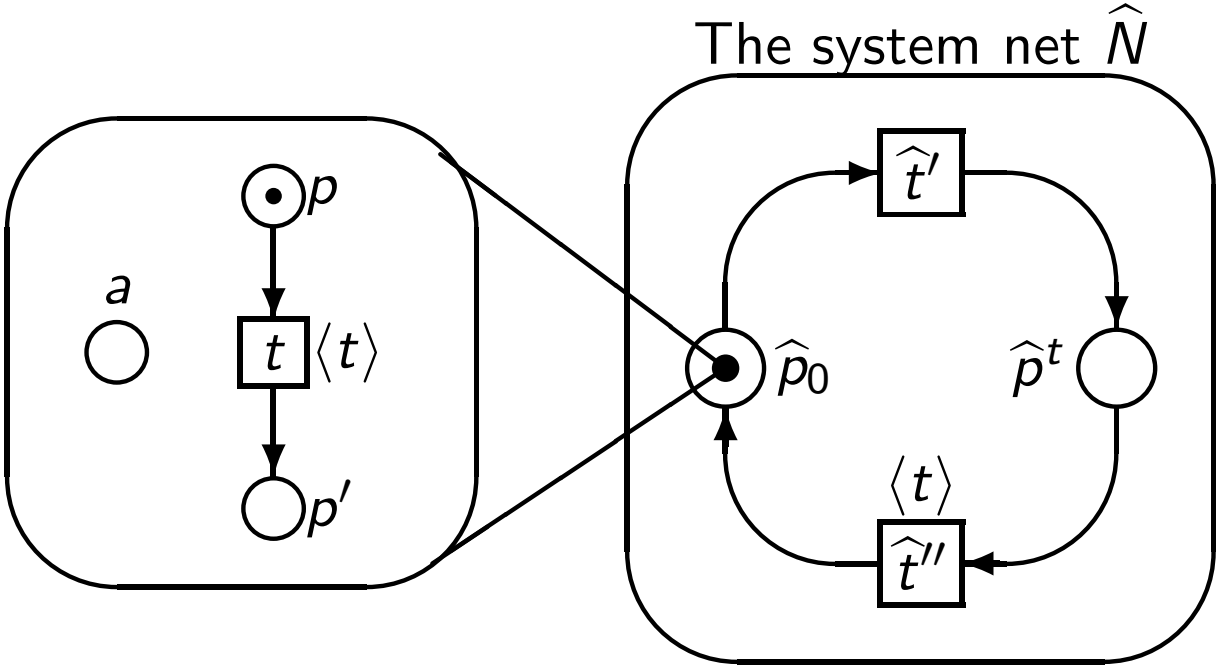


(c) Net N_t

Expressiveness

Theorem

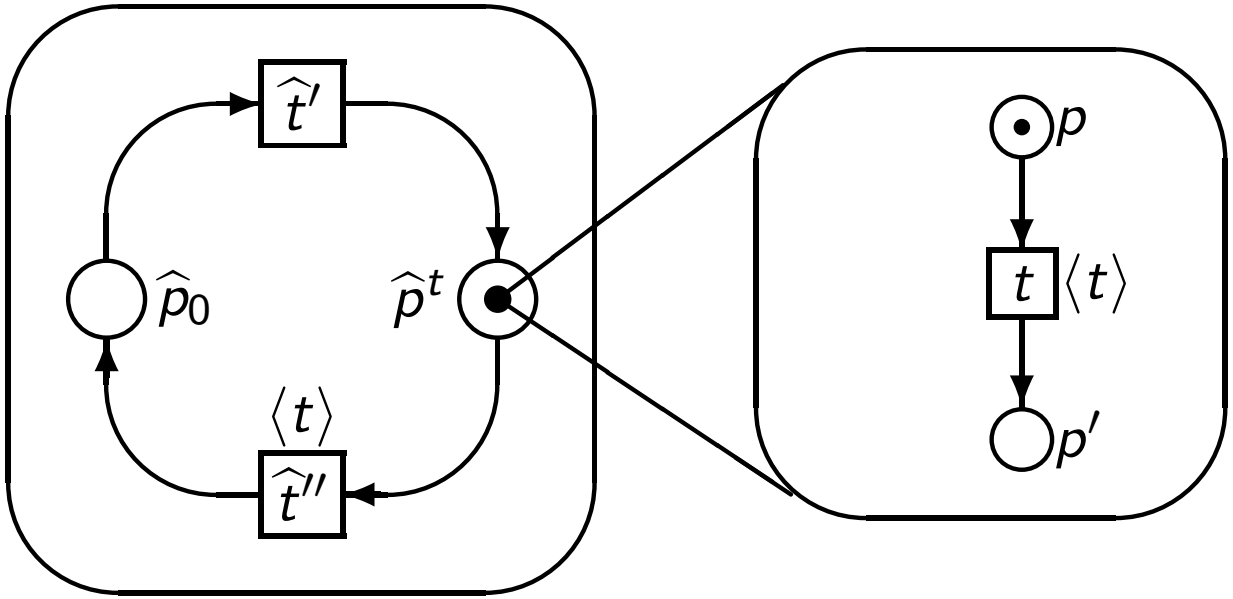
Mobile Eos *can simulate Petri nets with inhibitor arcs.*



Expressiveness

Theorem

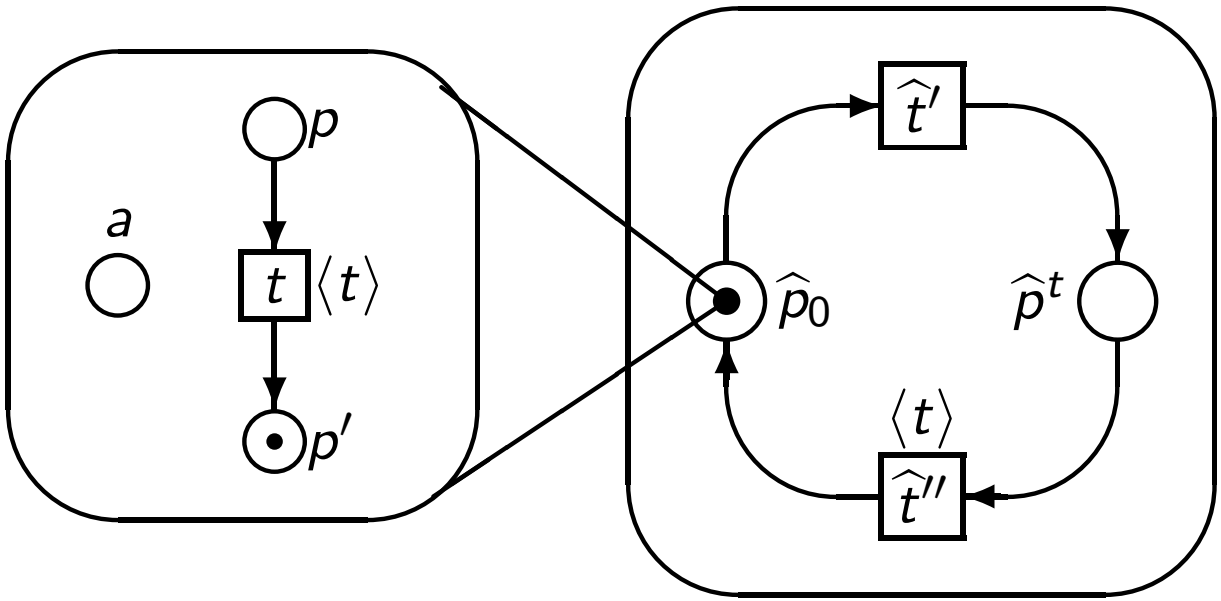
Mobile Eos *can simulate Petri nets with inhibitor arcs.*



Expressiveness

Theorem

Mobile Eos *can simulate Petri nets with inhibitor arcs.*



Expressiveness

Theorem

Mobile Eos *can simulate Petri nets with inhibitor arcs.*

Since boundedness is decidable for Eos and undecidable for inhibitor nets, we have that mobile Eos are more expressive than Eos.

Conclusion

- Object Nets
- Reference Semantics
- Value Semantics
- Here: Mobility Semantics
- Mobile EOS
- Conversion


Michael Köhler, Daniel Moldt, and Heiko Rölke.
Modelling mobility and mobile agents using nets within nets.
In W. v. d. Aalst and E. Best, editors, *International Conference on Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 121–140. Springer-Verlag, 2003.


Michael Köhler and Heiko Rölke.
Properties of Object Petri Nets.
In J. Cortadella and W. Reisig, editors, *International Conference on Application and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 2004.


Michael Köhler and Heiko Rölke.
Reference and value semantics are equivalent for ordinary Object Petri Nets.

In P. Darondeau and G. Ciardo, editors, *International Conference on Application and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 309–328. Springer-Verlag, 2005.