Quantitative Analysis of Time Petri Nets Used for Modelling Biochemical Networks

Louchka Popova-Zeugmann

Humboldt-Universität zu Berlin
Institut of Computer Science
Unter den Linden 6, 10099 Berlin, Germany

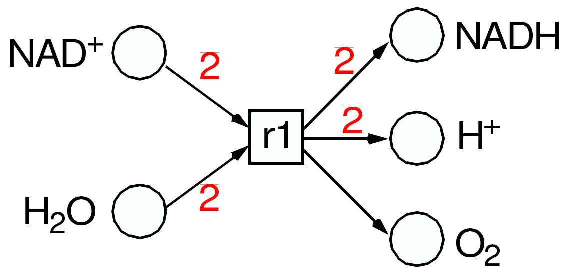
Max-Planck-Institut für Molekulare Pflanzenphysiologie
May 04, 2005

Outline

- **chemical reactions -> atomic actions -> Petri net transitions**

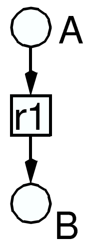
$$2\ NAD^+ + 2\ H_2O \rightarrow 2\ NADH + 2\ H^+ + O_2$$

input compounds



output compounds

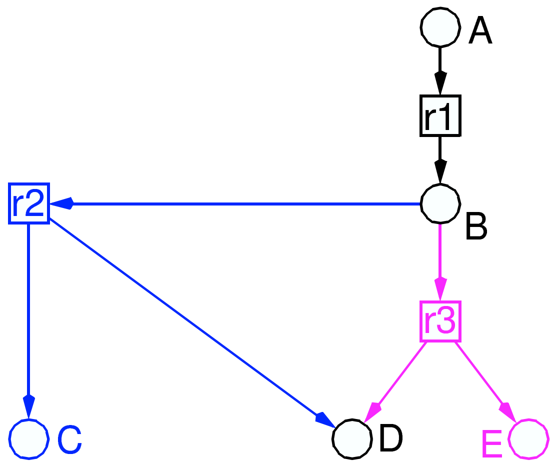
r1: A -> B



r1: A -> B

r2: B -> C + D

r3: B -> D + E



-> alternative reactions

r1: A -> B

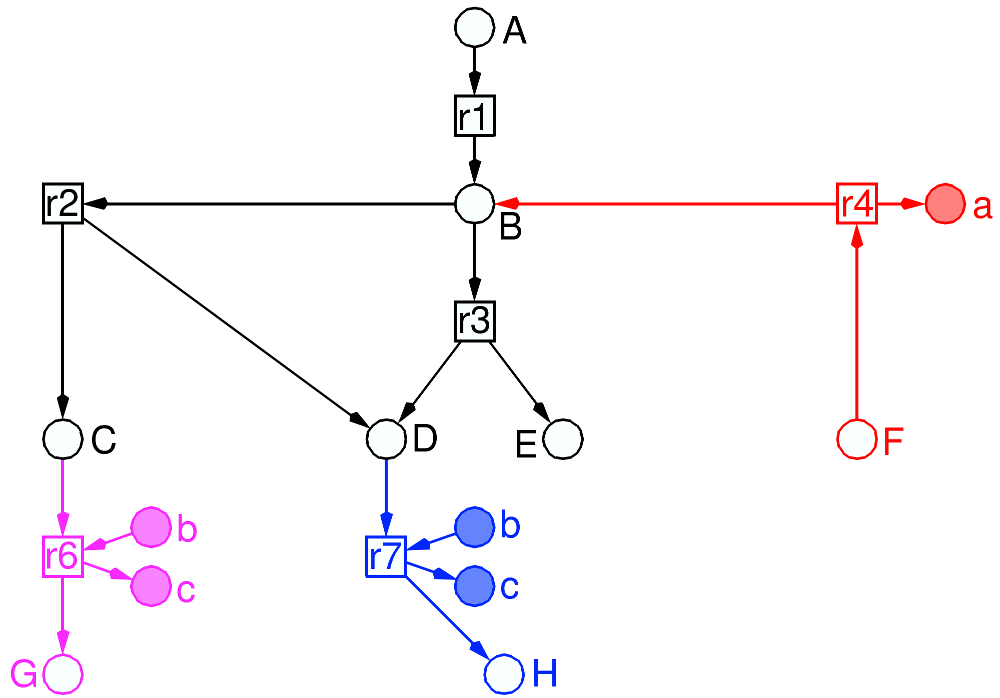
r2: B -> C + D

r3: B -> D + E

r4: F -> B + a

r6: C + b -> G + c

r7: D + b -> H + c



-> concurrent reactions

r1: A -> B

r2: B -> C + D

r3: B -> D + E

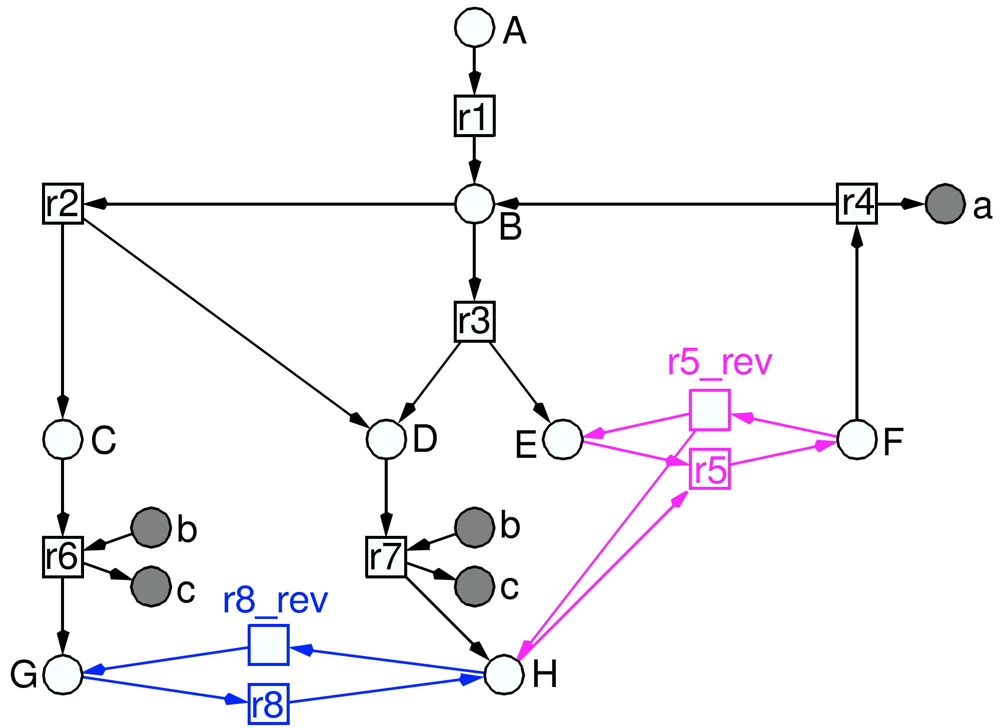
r4: F -> B + a

r5: E + H <-> F

r6: C + b -> G + c

r7: D + b -> H + c

r8: H <-> G



-> reversible reactions

r1: A -> B

r2: B -> C + D

r3: B -> D + E

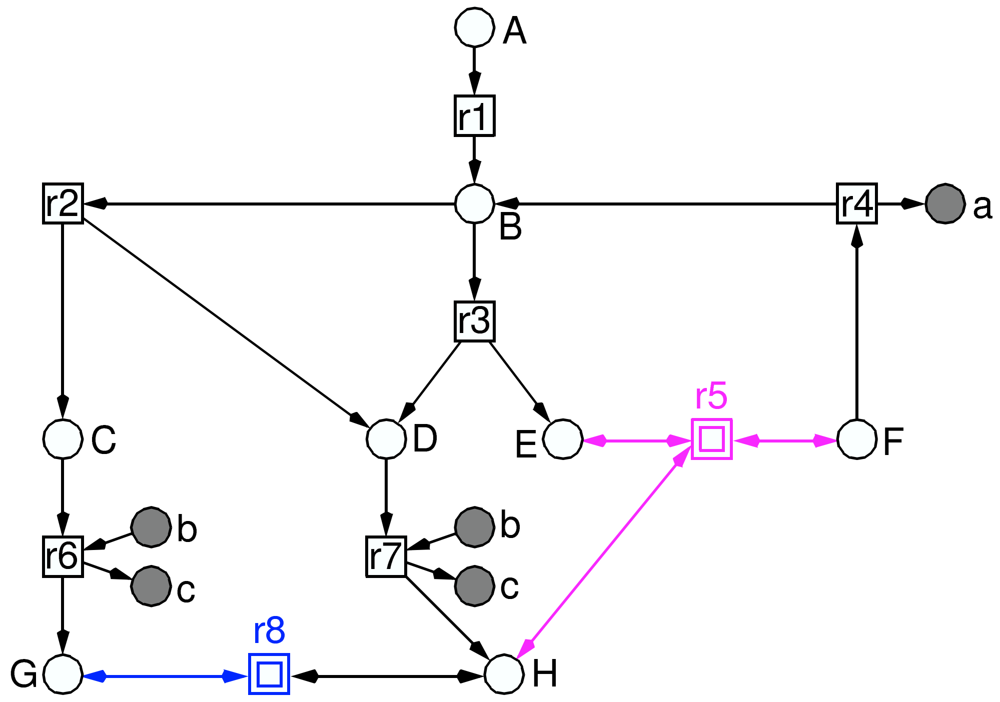
r4: F -> B + a

r5: E + H <-> F

r6: C + b -> G + c

r7: D + b -> H + c

r8: H <-> G



-> reversible reactions
- hierarchical nodes

r1: A -> B

r2: B -> C + D

r3: B -> D + E

r4: F -> B + a

r5: E + H <-> F

r6: C + b -> G + c

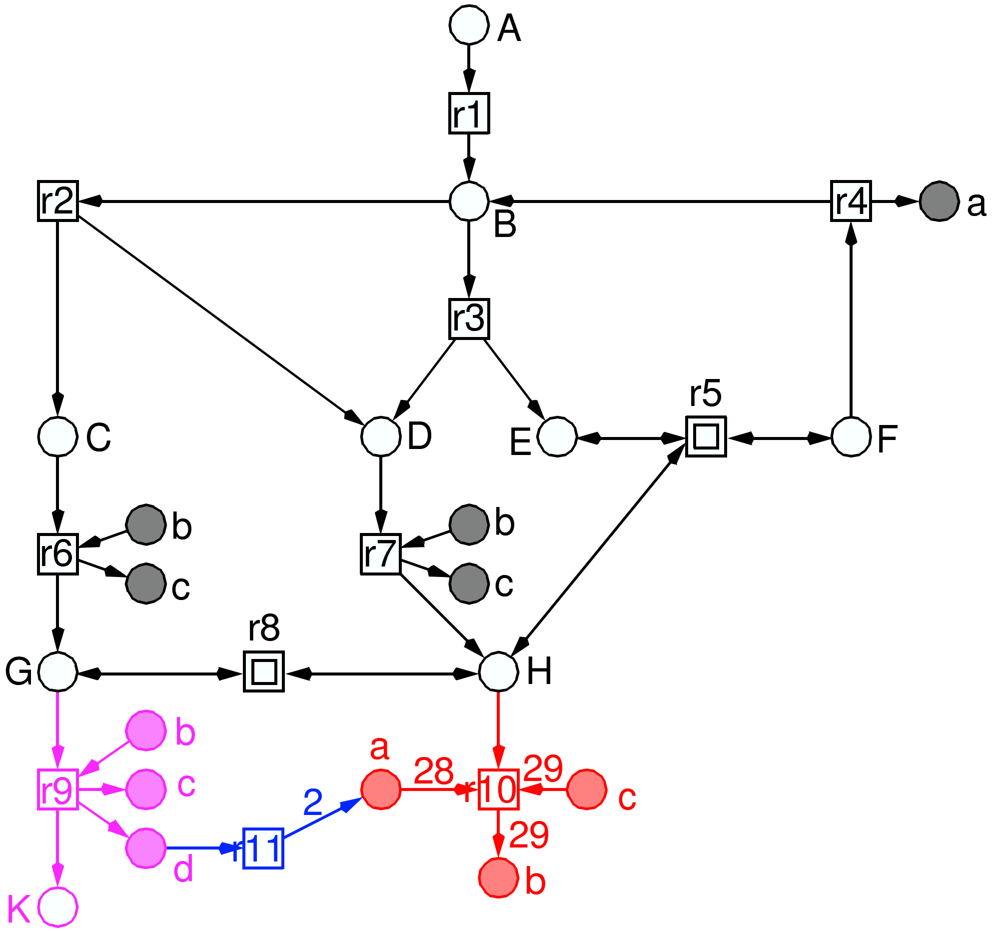
r7: D + b -> H + c

r8: H <-> G

r9: G + b -> K + c + d

r10: H + 28a + 29c -> 29b

r11: d -> 2a



r1: A -> B

r2: B -> C + D

r3: B -> D + E

r4: F -> B + a

r5: E + H <-> F

r6: C + b -> G + c

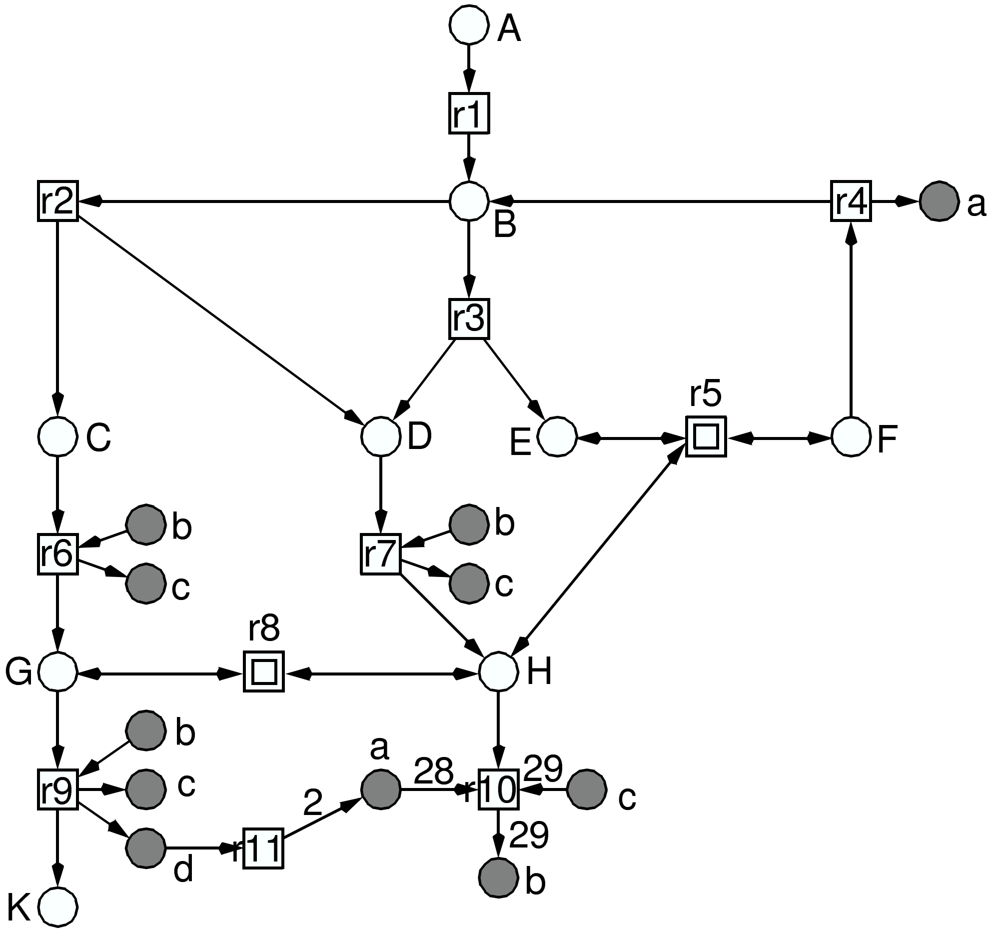
r7: D + b -> H + c

r8: H <-> G

r9: G + b -> K + c + d

r10: H + 28a + 29c -> 29b

r11: d -> 2a



r1: A -> B

r2: B -> C + D

r3: B -> D + E

r4: F -> B + a

r5: E + H <-> F

r6: C + b -> G + c

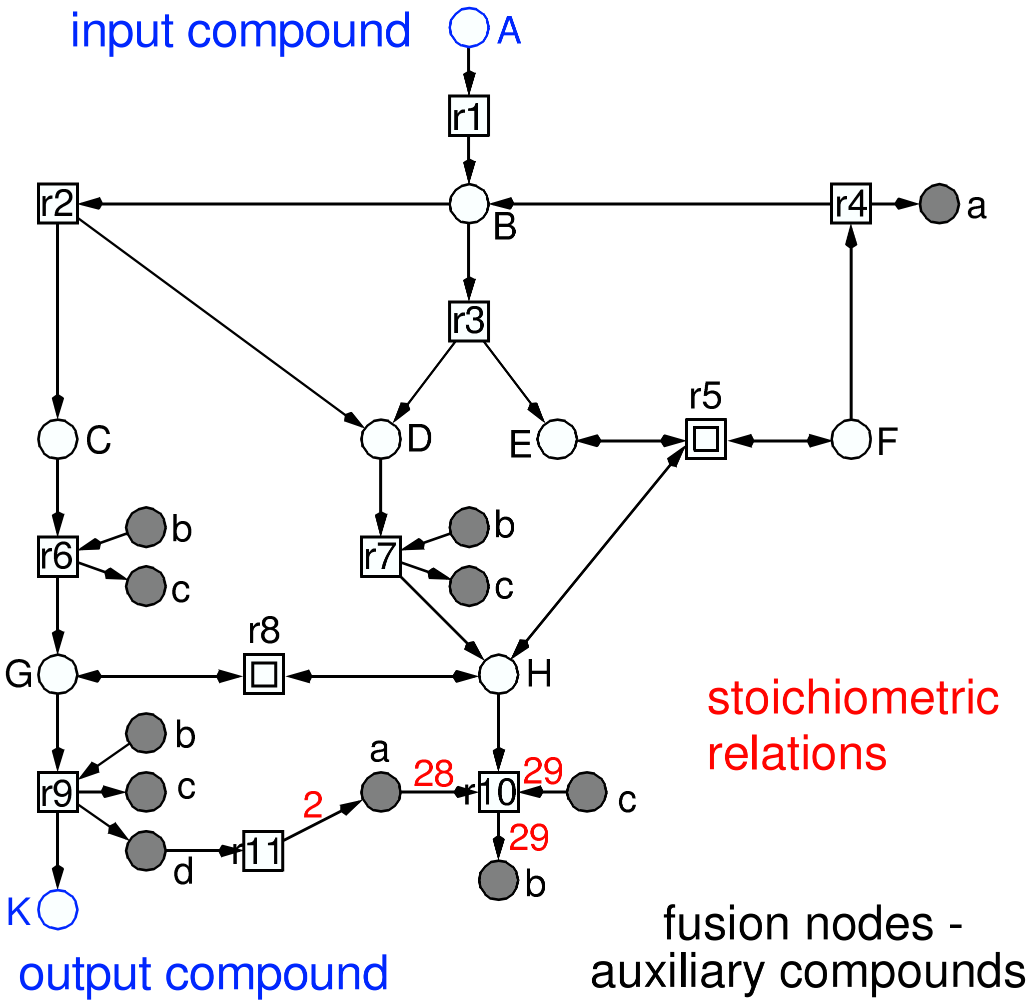
r7: D + b -> H + c

r8: H <-> G

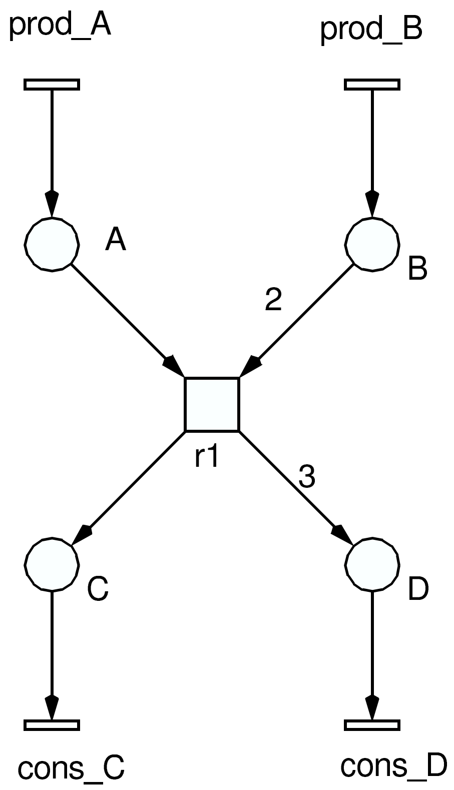
r9: G + b -> K + c + d

r10: H + 28a + 29c -> 29b

r11: d -> 2a



TRANSFORMATION, EX1



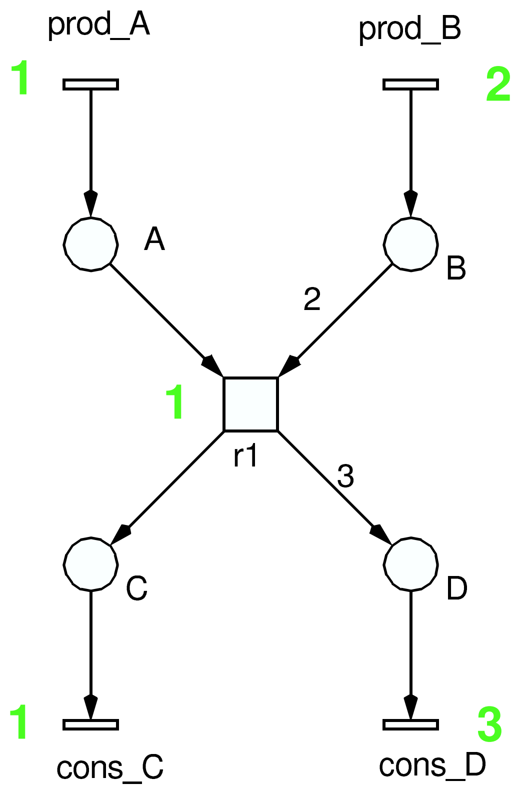
-> properties as time-less net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	N	N	Y	N	?	N	Y	Y	Y	N

TRANSFORMATION, EX1



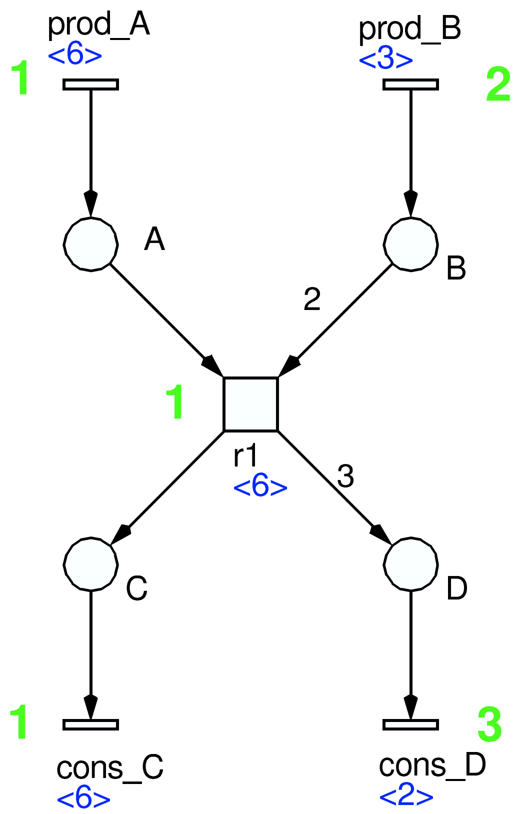
T-INVARIANTE

-> properties as time-less net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	N	N	Y	N	?	N	Y	Y	Y	N



T-INVARIANTE

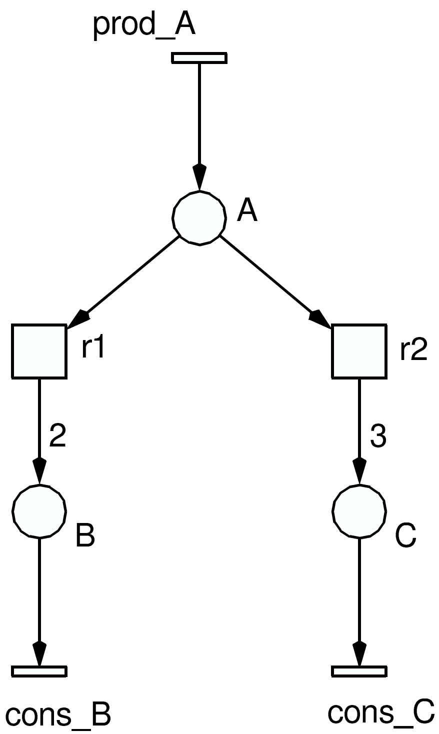
-> properties as time net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	Y	N	N	N	?	N	Y	Y	Y	N

TRANSFORMATION, EX2



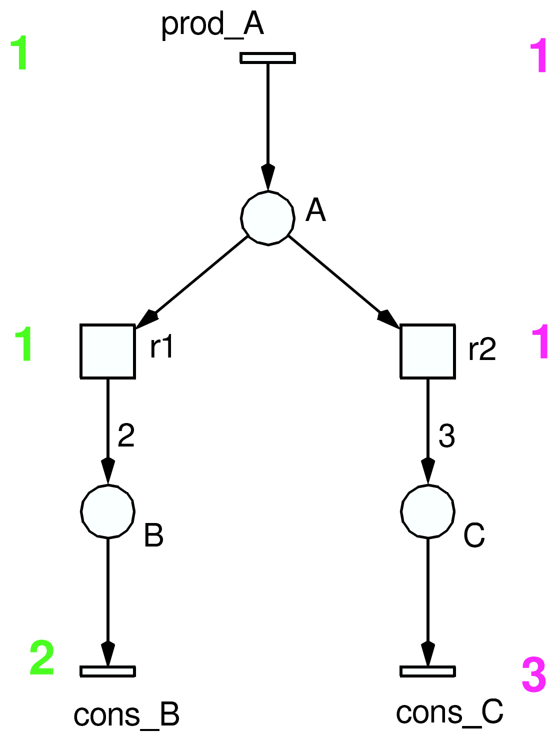
-> properties as time-less net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	N	N	Y	N	?	N	N	Y	Y	N

TRANSFORMATION, EX2



T-INVARIANTE1

T-INVARIANTE2

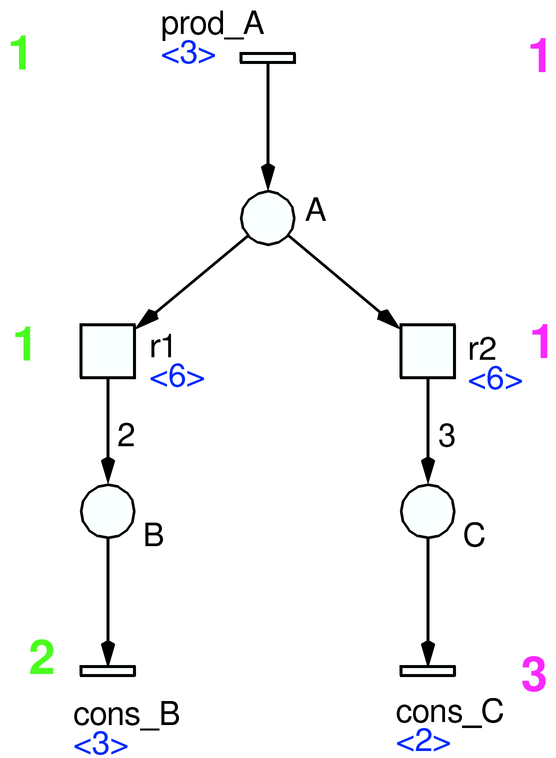
-> properties as time-less net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	N	N	Y	N	?	N	N	Y	Y	N

TRANSFORMATION, EX2



T-INVARIANTE1

T-INVARIANTE2

-> properties as time net

INA

ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
N	Y	N	Y	N	Y	Y	N	Y	Y	N	N	Y	N	Y	Y	Y

CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
N	Y	Y	N	N	N	?	N	Y	Y	Y	N

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,
  P—set of places

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,
  P−set of places
  T−set of transitions

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,

$$\left.\begin{array}{l}\text{P}-\text{set of places}\\ \text{T}-\text{set of transitions}\end{array}\right\}\text{set of vertices}$$

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,
  $$\left.\begin{array}{l}\text{P}-\text{set of places}\\ \text{T}-\text{set of transitions}\end{array}\right\}\text{set of vertices}$$
  $P \cap T = \emptyset, \quad P \cup T \neq \emptyset,$

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,

  $$\left.\begin{array}{l}\text{P}-\text{set of places}\\ \text{T}-\text{set of transitions}\end{array}\right\}\text{set of vertices}$$

  $P \cap T = \emptyset, \quad P \cup T \neq \emptyset,$

  F – set of edges (arcs)

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,

  $\left.\begin{array}{l}\text{P}-\text{set of places}\\ \text{T}-\text{set of transitions}\end{array}\right\}$ set of vertices

  $P \cap T = \emptyset, \quad P \cup T \neq \emptyset,$

  F – set of edges (arcs)

  $F \subseteq (P \times T) \cup (T \times P)$ und $dom(F) \cup cod(F) = P \cup T$

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,
  $\left.\begin{array}{l}\text{P}-\text{set of places}\\ \text{T}-\text{set of transitions}\end{array}\right\}$ set of vertices
  $P \cap T = \emptyset, \quad P \cup T \neq \emptyset,$
  F – set of edges (arcs)
  $F \subseteq (P \times T) \cup (T \times P)$ und $dom(F) \cup cod(F) = P \cup T$
- $V : F \longrightarrow \mathbb{N}^+$ (weights of edges)

Petri Net

Definition (Petri Net)

The structure $N = (P, T, F, V, m_0)$ is a **Petri Net (PN)**, iff

- P, T und F are finite sets,

  $$\left.\begin{array}{l}\text{P—set of places}\\ \text{T—set of transitions}\end{array}\right\}\text{set of vertices}$$

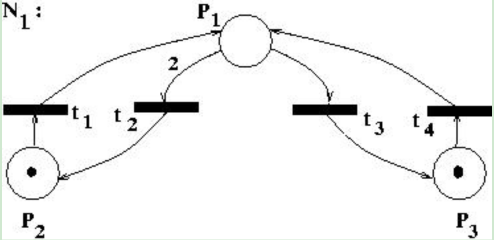
  $P \cap T = \emptyset, \quad P \cup T \neq \emptyset,$

  F – set of edges (arcs)

  $F \subseteq (P \times T) \cup (T \times P)$ und $dom(F) \cup cod(F) = P \cup T$
- $V : F \longrightarrow \mathbb{N}^+$ (weights of edges)
- $m_0 : P \longrightarrow \mathbb{N}$ (initial marking)

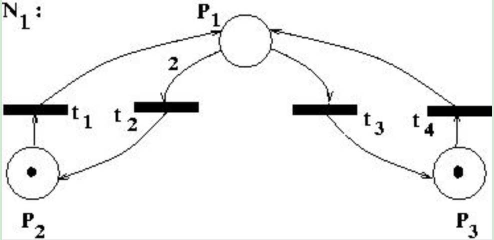
Petri Net

Example



Petri Net

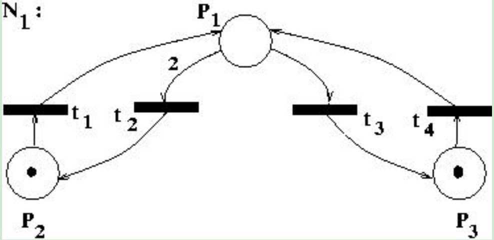
Example



- $m_0 = (0, 1, 1)$

Petri Net

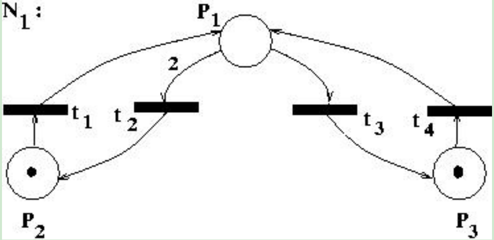
Example



- $m_0 = (0, 1, 1)$
- $t_1^- = (0, 1, 0)$

Petri Net

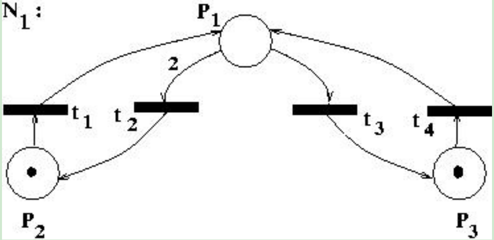
Example



- $m_0 = (0,1,1)$
- $t_1^- = (0,1,0)$ $t_1^+ = (1,0,0)$

Petri Net

Example



- $m_0 = (0, 1, 1)$
- $t_1^- = (0, 1, 0)$ $\quad$ $t_1^+ = (1, 0, 0)$
- $\Delta(t_1) = -t_1^- + t_1^+ = (1, -1, 0)$

Firing transition

Definition

- A transition $t \in T$ is **enabled (may fire)** at a marking m iff all input-places of t have enough tokens

Firing transition

Definition

- A transition $t \in T$ is **enabled (may fire)** at a marking m iff all input-places of t have enough tokens
  e.g. $t^{-} \leq m$.

Firing transition

Definition

- A transition $t \in T$ is **enabled (may fire)** at a marking m iff all input-places of t have enough tokens
  e.g. $t^{-} \leq m$.
- When an enabled transition t at a marking m fires, a **successor** marking m' is reached

Firing transition

Definition

- A transition $t \in T$ is **enabled (may fire)** at a marking m iff all input-places of t have enough tokens
  e.g. $t^- \leq m$.
- When an enabled transition t at a marking m fires, a **successor** marking m' is reached
  given by $m' := m + \Delta t$

Firing transition

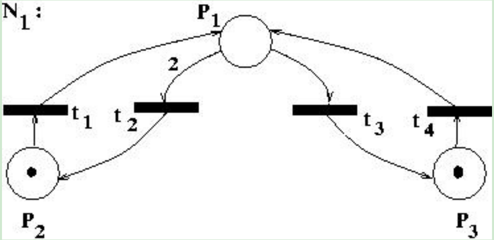
Definition

- A transition $t \in T$ is **enabled (may fire)** at a marking m iff all input-places of t have enough tokens
  e.g. $t^- \leq m$.
- When an enabled transition t at a marking m fires, a **successor** marking m' is reached
  given by $m' := m + \Delta t$

  denoted by $m \xrightarrow{t} m'$.

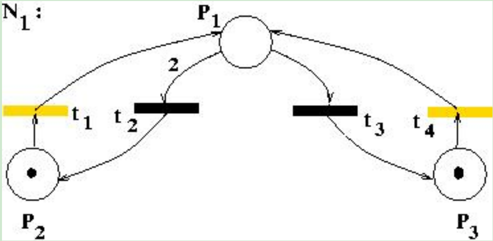
firing transition

Example



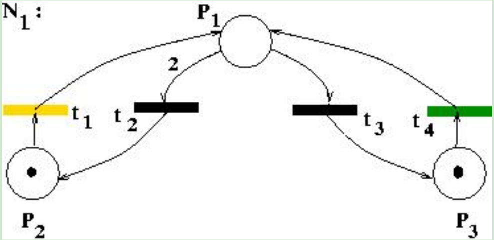
firing transition

Example



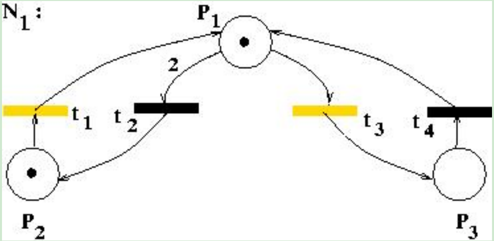
firing transition

Example



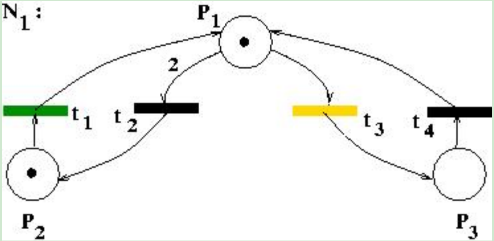
firing transition

Example



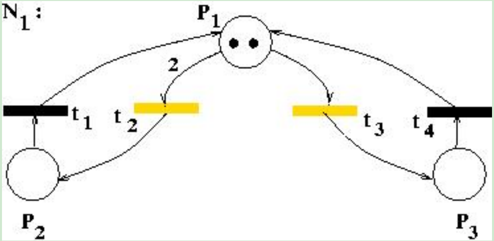
firing transition

Example



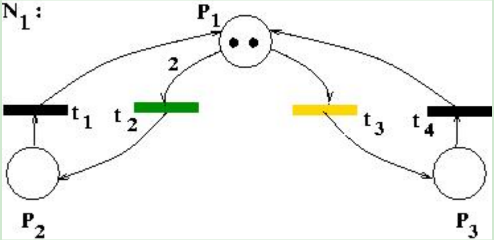
firing transition

Example



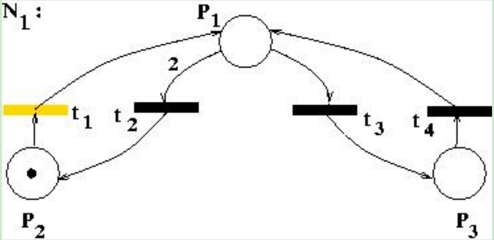
firing transition

Example



firing transition

Example



Time Petri Net

Definition (Time Petri net)

The structure $Z = (P, T, F, V, m_o, I)$ is called a **Time Petri net (TPN)** iff:

Time Petri Net

Definition (Time Petri net)

The structure $Z = (P, T, F, V, m_o, I)$ is called a **Time Petri net (TPN)** iff:

- $S(Z) := (P, T, F, V, m_o)$ is a PN (skeleton of Z)

Time Petri Net

Definition (Time Petri net)

The structure $Z = (P, T, F, V, m_o, I)$ is called a **Time Petri net (TPN)** iff:

- $S(Z) := (P, T, F, V, m_o)$ is a PN (skeleton of Z)
- $I : T \longrightarrow \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\})$ and

Time Petri Net

Definition (Time Petri net)

The structure $Z = (P, T, F, V, m_o, I)$ is called a **Time Petri net (TPN)** iff:

- $S(Z) := (P, T, F, V, m_o)$ is a PN (skeleton of Z)
- $I : T \longrightarrow \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\})$ and $I_1(t) \leq I_2(t)$ for each $t \in T$, where $I(t) = (I_1(t), I_2(t))$.

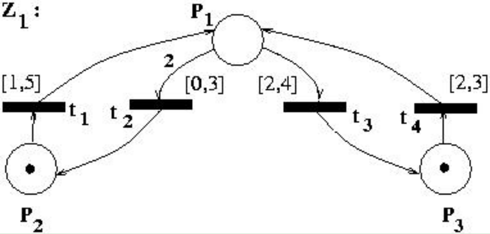
Time Petri Net

Definition (FTPN)

A TPN is called finite Time Petri net (FTPN) iff
$I : T \longrightarrow \mathbb{Q}_0^+ \times \mathbb{Q}_0^+$.

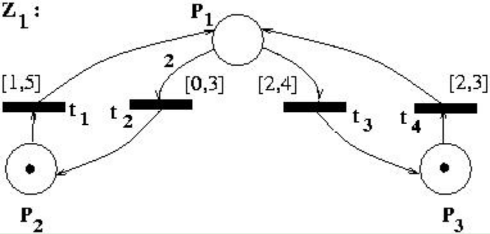
Time Petri Net

Example



Time Petri Net

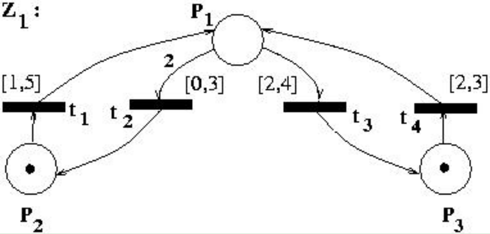
Example



- $m_0 = (0, 1, 1)$ p-marking

Time Petri Net

Example



- $m_0 = (0, 1, 1)$ p-marking
- $h_0 = (0, \sharp, \sharp, 0)$ t-marking

state

Definition (state)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN and $h : T \longrightarrow \mathbb{R}_0^+ \cup \{\#\}$.
$z = (m, h)$ is called a **state** in Z iff:

state

Definition (state)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN and $h : T \longrightarrow \mathbb{R}_0^+ \cup \{\#\}$.
$z = (m, h)$ is called a **state** in Z iff:

- m is a p-marking in Z, e.g. m is a marking in $S(Z)$.

state

Definition (state)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN and $h : T \longrightarrow \mathbb{R}_0^+ \cup \{\#\}$.
$z = (m, h)$ is called a **state** in Z iff:

- m is a p-marking in Z, e.g. m is a marking in $S(Z)$.
- h is a t-marking in Z,

state

Definition (state)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN and $h : T \longrightarrow \mathbb{R}_0^+ \cup \{\#\}$.
$z = (m, h)$ is called a **state** in Z iff:

- m is a p-marking in Z, e.g. m is a marking in $S(Z)$.
- h is a t-marking in Z, e.g.
  $\forall t \; (\; (t \in T \wedge t^- \leq m) \; \longrightarrow \; (h(t) \in \mathbb{R}_0^+ \wedge h(t) \leq lft(t))),$

state

Definition (state)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN and $h : T \longrightarrow \mathbb{R}_0^+ \cup \{\#\}$.
$z = (m, h)$ is called a **state** in Z iff:

- m is a p-marking in Z, e.g. m is a marking in $S(Z)$.
- h is a t-marking in Z, e.g.
  $\forall t \; (\; (t \in T \wedge t^- \leq m) \; \longrightarrow \; (h(t) \in \mathbb{R}_0^+ \wedge h(t) \leq lft(t))),$
  and
  $\forall t \; (\; (t \in T \wedge t^- \not\leq m) \; \longrightarrow \; h(t) = \#).$

Definition (state changing)

Definition (state changing)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN,
$\hat{t}$ be a transition in T and
$z = (m, h)$, $z' = (m', h')$ be two states.

Definition (state changing)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN,
$\hat{t}$ be a transition in T and
$z = (m, h)$, $z' = (m', h')$ be two states.
Then

(a) the transition $\hat{t}$ is **ready** to fire in the state $z = (m, h)$, denoted by $z \xrightarrow{\hat{t}}$, iff

Definition (state changing)

Let $Z = (P, T, F, V, m_o, I)$ be a TPN,
$\hat{t}$ be a transition in T and
$z = (m, h)$, $z' = (m', h')$ be two states.
Then

(a) the transition $\hat{t}$ is **ready** to fire in the state $z = (m, h)$, denoted by $z \xrightarrow{\hat{t}}$, iff

  (i) $\hat{t}^- \leq m$ and
  (ii) $eft(\hat{t}) \leq h(\hat{t})$.

state changing

Definition (state changing)

(b) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by firing the transition** $\hat{t}$, denoted by $z \xrightarrow{\hat{t}} z'$, iff

state changing

Definition (state changing)

(b) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by firing the transition** $\hat{t}$, denoted by $z \xrightarrow{\hat{t}} z'$, iff

(i) t is ready to fire in the state $z = (m, h)$

state changing

Definition (state changing)

(b) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by firing the transition** $\hat{t}$, denoted by $z \xrightarrow{\hat{t}} z'$, iff

(i) t is ready to fire in the state $z = (m, h)$

(ii) $m' = m + \Delta\hat{t}$ and

state changing

Definition (state changing)

(b) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by firing the transition** $\hat{t}$, denoted by $z \xrightarrow{\hat{t}} z'$, iff

(i) t is ready to fire in the state $z = (m, h)$

(ii) $m' = m + \Delta\hat{t}$ and

(iii) $\forall t\ (\ t \in T \longrightarrow$

$$h'(t) =: \begin{cases} \# & \text{iff} \quad t^- \not\leq m' \\ h(t) & \text{iff} \quad t^- \leq m \wedge t^- \leq m' \wedge Ft \cap F\hat{t} = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad).$$

state changing

Definition (state changing)

(c) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by the time elapsing** $\tau \in \mathbb{R}_0^+$, denoted by $z \xrightarrow{\tau} z'$, iff

state changing

Definition (state changing)

(c) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by the time elapsing** $\tau \in \mathbb{R}_0^+$, denoted by $z \xrightarrow{\tau} z'$, iff

(i) $m' = m$ and

state changing

Definition (state changing)

(c) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by the time elapsing** $\tau \in \mathbb{R}_0^+$, denoted by $z \xrightarrow{\tau} z'$, iff

(i) $m' = m$ and

(ii) $\forall t \; (\; t \in T \wedge h(t) \neq \# \longrightarrow h(t) + \tau \leq \mathit{lft}(t) \;)$ i.e. the time elapsing τ is possible, and

state changing

Definition (state changing)

(c) the state $z = (m, h)$ is **changed** into the state $z' = (m', h')$ **by the time elapsing** $\tau \in \mathbb{R}_0^+$, denoted by $z \xrightarrow{\tau} z'$, iff

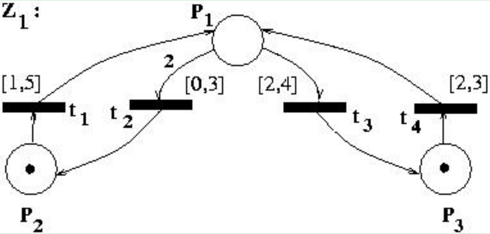
(i) $m' = m$ and

(ii) $\forall t \; (\; t \in T \wedge h(t) \neq \# \longrightarrow h(t) + \tau \leq lft(t) \;)$ i.e. the time elapsing τ is possible, and

(iii) $\forall t \; (\; t \in T \longrightarrow h'(t) := \begin{cases} h(t) + \tau & \text{iff} \quad t^- \leq m' \\ \# & \text{iff} \quad t^- \not\leq m' \end{cases} \;).$

Time Petri Net

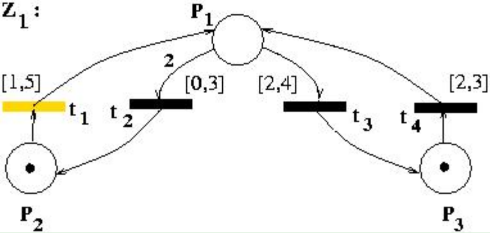
Example



$$(m_0, \begin{pmatrix} 0 \\ \sharp \\ \sharp \\ 0 \end{pmatrix})$$

Time Petri Net

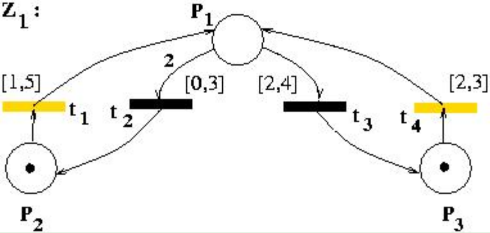
Example



$$(m_0, \begin{pmatrix} 0 \\ \sharp \\ \sharp \\ 0 \end{pmatrix}) \xrightarrow{1.3} (m_1, \begin{pmatrix} 1.3 \\ \sharp \\ \sharp \\ 1.3 \end{pmatrix})$$

Time Petri Net

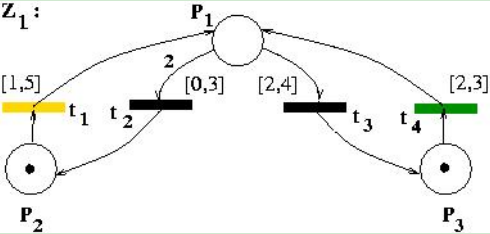
Example



$$z_0 \xrightarrow{1.3} (m_1, \begin{pmatrix} 1.3 \\ \sharp \\ \sharp \\ 1.3 \end{pmatrix}) \xrightarrow{1.0} (m_2, \begin{pmatrix} 2.3 \\ \sharp \\ \sharp \\ 2.3 \end{pmatrix})$$

Time Petri Net

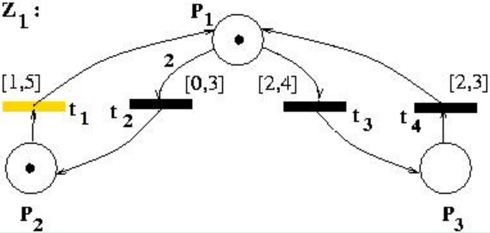
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \left(m_2, \begin{pmatrix} 2.3 \\ \sharp \\ \sharp \\ 2.3 \end{pmatrix}\right) \xrightarrow{t_4}$$

Time Petri Net

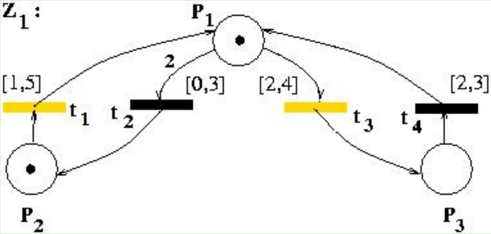
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \left(m_2, \begin{pmatrix} 2.3 \\ \sharp \\ \sharp \\ 2.3 \end{pmatrix}\right) \xrightarrow{t_4} \left(m_3, \begin{pmatrix} 2.3 \\ \sharp \\ 0.0 \\ \sharp \end{pmatrix}\right)$$

Time Petri Net

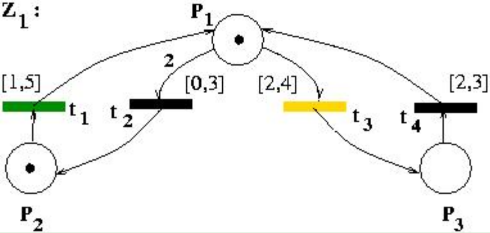
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \xrightarrow{t_4} (m_3, \begin{pmatrix} 2.3 \\ \sharp \\ 0.0 \\ \sharp \end{pmatrix}) \xrightarrow{2.0} (m_4, \begin{pmatrix} 4.3 \\ \sharp \\ 2.0 \\ \sharp \end{pmatrix})$$

Time Petri Net

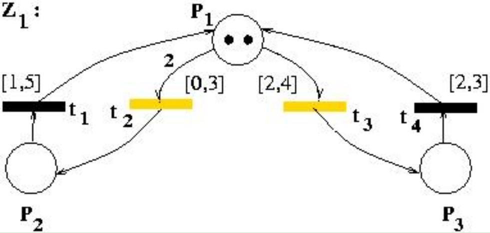
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \xrightarrow{t_4} \xrightarrow{2.0} \left(m_4, \begin{pmatrix} 4.3 \\ \sharp \\ 2.0 \\ \sharp \end{pmatrix}\right) \xrightarrow{t_1}$$

Time Petri Net

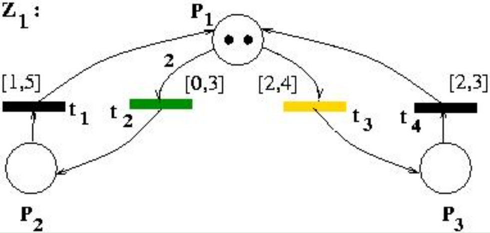
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \xrightarrow{t_4} \xrightarrow{2.0} \left(m_4, \begin{pmatrix} 4.3 \\ \sharp \\ 2.0 \\ \sharp \end{pmatrix}\right) \xrightarrow{t_1} \left(m_5, \begin{pmatrix} \sharp \\ 0.0 \\ 2.0 \\ \sharp \end{pmatrix}\right)$$

Time Petri Net

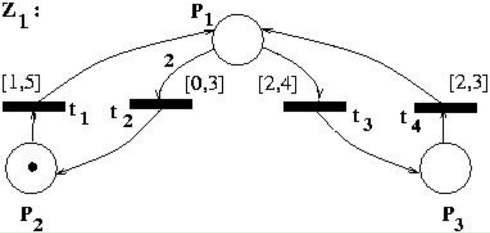
Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \xrightarrow{t_4} \xrightarrow{2.0} \xrightarrow{t_1} \left(m_5, \begin{pmatrix} \sharp \\ 0.0 \\ 2.0 \\ \sharp \end{pmatrix}\right) \xrightarrow{t_2}$$

Time Petri Net

Example



$$z_0 \xrightarrow{1.3} \xrightarrow{1.0} \xrightarrow{t_4} \xrightarrow{2.0} \xrightarrow{t_1} \left(m_5, \begin{pmatrix} \sharp \\ 0.0 \\ 2.0 \\ \sharp \end{pmatrix}\right) \xrightarrow{t_2} \left(m_6, \begin{pmatrix} 0.0 \\ \sharp \\ \sharp \\ \sharp \end{pmatrix}\right)$$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$
- **run:** $\sigma(\tau) = (t_1, \tau_1, \cdots, \tau_{n-1}, t_n)$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$
- **run:** $\sigma(\tau) = (t_1, \tau_1, \cdots, \tau_{n-1}, t_n)$
- **feasable run:** $z_0 \xrightarrow{\tau_1} z_0^* \xrightarrow{t_1} z_1 \xrightarrow{\tau_2} \cdots \xrightarrow{t_n} z_n$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$
- **run:** $\sigma(\tau) = (t_1, \tau_1, \cdots, \tau_{n-1}, t_n)$
- **feasable run:** $z_0 \xrightarrow{\tau_1} z_0^* \xrightarrow{t_1} z_1 \xrightarrow{\tau_2} \cdots \xrightarrow{t_n} z_n$
- **feasable transition sequence :** σ is feasable if there ex. a feasable run $\sigma(\tau)$

Reachable state, Reachable marking, State space

Definition

- z is **reachable state** in Z if there ex. a feasable run $\sigma(\tau)$ and $z_0 \xrightarrow{\sigma(\tau)} z$

Reachable state, Reachable marking, State space

Definition

- z is **reachable state** in Z if there ex. a feasable run $\sigma(\tau)$ and $z_0 \xrightarrow{\sigma(\tau)} z$
- m is **reachable marking** in Z if there ex. a reachable state z in Z with $z = (m, h)$

Reachable state, Reachable marking, State space

Definition

- z is **reachable state** in Z if there ex. a feasable run $\sigma(\tau)$ and $z_0 \xrightarrow{\sigma(\tau)} z$
- m is **reachable marking** in Z if there ex. a reachable state z in Z with $z = (m, h)$
- The set of all reachable states in Z is the **state space** of Z (denoted: $StSp(Z)$).

State class

Definition (state class)

Let Z be a TPN and σ be a feasable transition sequence. The set C_σ is called a state class, iff

State class

Definition (state class)

Let Z be a TPN and σ be a feasable transition sequence. The set C_σ is called a state class, iff

$$\text{Basis: } C_e := \{z \mid \exists \tau (\tau \in \mathbb{R}_0^+ \wedge z_0 \xrightarrow{\tau} z)\}$$

State class

Definition (state class)

Let Z be a TPN and σ be a feasable transition sequence. The set C_σ is called a state class, iff

Basis: $C_e := \{z \mid \exists \tau (\tau \in \mathbb{R}_0^+ \wedge z_0 \xrightarrow{\tau} z)\}$

Step: Let C_σ be already defined. Then $C_{\sigma t}$ is derived from C_σ by firing t ($C_\sigma \xrightarrow{t} C_{\sigma t}$), iff

$$C_{\sigma t} := \{z \mid \exists z_1 \exists z_2 \exists \tau (z_1 \in C_\sigma \wedge \tau \in \mathbb{R}_0^+ \wedge z_1 \xrightarrow{t} z_2 \xrightarrow{\tau} z)\}.$$

State class

Definition (state class)

Let Z be a TPN and σ be a feasable transition sequence. The set C_σ is called a state class, iff

Basis: $C_e := \{z \mid \exists \tau(\tau \in \mathbb{R}_0^+ \wedge z_0 \xrightarrow{\tau} z)\}$

Step: Let C_σ be already defined. Then $C_{\sigma t}$ is derived from C_σ by firing t ($C_\sigma \xrightarrow{t} C_{\sigma t}$), iff

$$C_{\sigma t} := \{z \mid \exists z_1 \exists z_2 \exists \tau(z_1 \in C_\sigma \wedge \tau \in \mathbb{R}_0^+ \wedge z_1 \xrightarrow{t} z_2 \xrightarrow{\tau} z)\}.$$

Obviously: $StSp(Z) = \bigcup_\sigma C_\sigma$

Properties

- static properties:

- dynamic properties:

Properties

- static properties: being
  - pure
  - ordinary
  - free choice
  - extended simple
  - conservative, etc.

- dynamic properties:

Properties

- static properties: being
  - pure
  - ordinary
  - free choice
  - extended simple
  - conservative, etc.
- dynamic properties: being/having
  - bounded
  - live
  - reachable marking/state
  - place- or transitions invariants
  - deadlocks, etc.

Properties

- static properties: being
  - pure
  - ordinary
  - free choice
  - extended simple
  - conservative, etc.

  decidable **without knowledge** of the state space!
- dynamic properties: being/having
  - bounded
  - live
  - reachable marking/state
  - place- or transitions invariants
  - deadlocks, etc.

Properties

- static properties: being
  - pure
  - ordinary
  - free choice
  - extended simple
  - conservative, etc.

  decidable **without knowledge** of the state space!
- dynamic properties: being/having
  - bounded
  - live
  - reachable marking/state
  - place- or transitions invariants
  - deadlocks, etc.

  decidable, if at all (TPN is equiv. to TM!),
  with implicit/explicit knowledge of the state space

Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

- $m_0 \xrightarrow{\sigma} m_\sigma$

Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a term (in a FO Logic), "1/2–interpreted" as a sum of variables for each transition t

Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a term (in a FO Logic), "1/2–interpreted" as a sum of variables for each transition t
- B_σ is a set of formulae (in a FO Logic), "1/2–interpreted" as a system of inequalities.

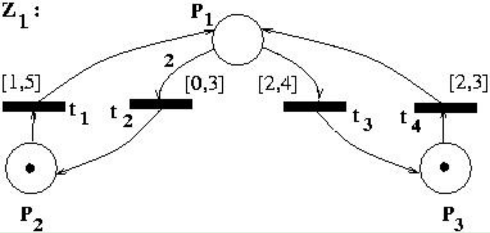
Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

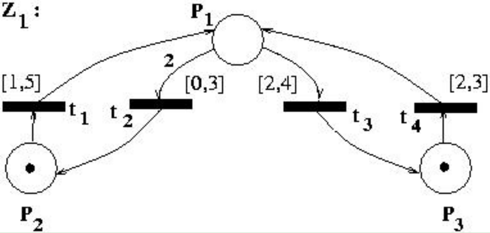
- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a term (in a FO Logic), "1/2–interpreted" as a sum of variables for each transition t
- B_σ is a set of formulae (in a FO Logic), "1/2–interpreted" as a system of inequalities.

Obviously $\delta(\sigma) = C_\sigma$.

Example



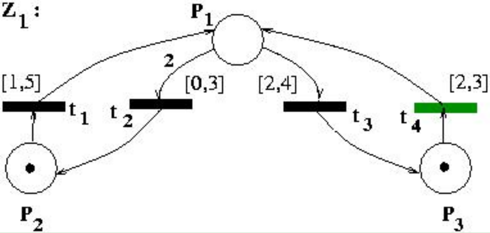
Example



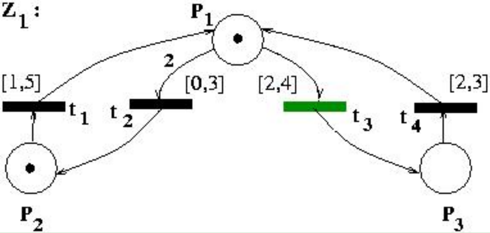
$$\sigma = (e) \qquad \Longrightarrow$$

$$\delta(\sigma) = C_e = \{(\underbrace{(0,1,1)}_{m_\sigma}, \underbrace{(x_1, \sharp, \sharp, x_1)}_{\Sigma_\sigma}) \mid \underbrace{0 \leq x_1 \leq 3}_{B_\sigma}\}$$

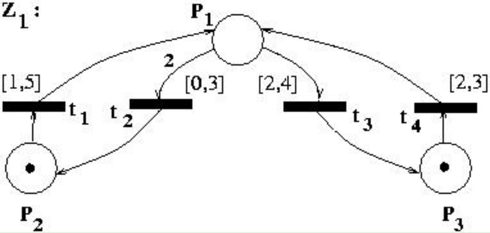
Example



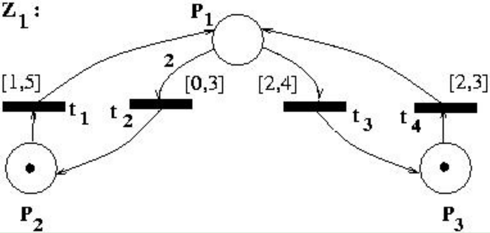
Example



Example

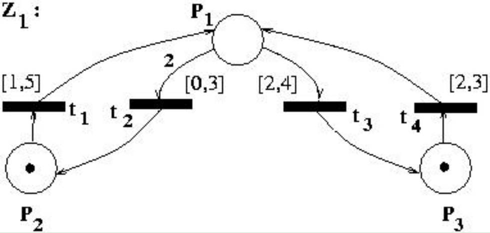


Example



$$\sigma = (t_4, t_3)$$

Example



$$\sigma = (t_4, t_3) \quad \Longrightarrow \quad \delta(\sigma) = C_{t_4 t_3} =$$

$$\{ (\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} x_1 + x_2 + x_3 \\ \sharp \\ \sharp \\ x_3 \end{pmatrix}) \mid \begin{array}{ll} 2 \leq x_1 \leq 3, & x_1 + x_2 \leq 5 \\ 2 \leq x_2 \leq 4, & x_1 + x_2 + x_3 \leq 5 \\ 0 \leq x_3 \leq 3 & \end{array} \}.$$

State Space Reduction

Theorem (1)

Let Z be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a feasible transition sequence in Z, with a run $\sigma(\tau)$ as an execution of σ, i.e.

$$z_0 \xrightarrow{\tau_0} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n} \xrightarrow{t_n} z_n = (m_n, h_n),$$

and all $\tau_i \in \mathbb{R}_0^+$.
Then, there exists a further feasible run $\sigma(\tau^*)$ of σ with

$$z_0 \xrightarrow{\tau_0^*} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*} \xrightarrow{t_n} z_n^* = (m_n^*, h_n^*).$$

such that

State Space Reduction

Theorem (1 – continuation)

$$z_0 \xrightarrow{\tau_0} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n} \xrightarrow{t_n} z_n = (m_n, h_n), \ \tau_i \in \mathbb{R}_0^+.$$

$$z_0 \xrightarrow{\tau_0^*} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*} \xrightarrow{t_n} z_n^* = (m_n^*, h_n^*)$$

State Space Reduction

Theorem (1 – continuation)

$z_0 \xrightarrow{\tau_0} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n} \xrightarrow{t_n} z_n = (m_n, h_n),\ \tau_i \in \mathbb{R}_0^+.$

$z_0 \xrightarrow{\tau_0^*} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*} \xrightarrow{t_n} z_n^* = (m_n^*, h_n^*)$

1. *For each* $i, 0 \leq i \leq n$ *holds:* $\tau_i^* \in \mathbb{N}$.

State Space Reduction

Theorem (1 – continuation)

$$z_0 \xrightarrow{\tau_0} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n} \xrightarrow{t_n} z_n = (m_n, h_n),\ \tau_i \in \mathbb{R}_0^+.$$

$$z_0 \xrightarrow{\tau_0^*} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*} \xrightarrow{t_n} z_n^* = (m_n^*, h_n^*),\ \tau_i^* \in \mathbb{N}.$$

1. *For each* $i, 0 \leq i \leq n$ *holds:* $\tau_i^* \in \mathbb{N}$.

State Space Reduction

Theorem (1 – continuation)

$$z_0 \xrightarrow{\tau_0} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n} \xrightarrow{t_n} z_n = (m_n, h_n),\ \tau_i \in \mathbb{R}_0^+.$$

$$z_0 \xrightarrow{\tau_0^*} \xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*} \xrightarrow{t_n} z_n^* = (m_n^*, h_n^*),\ \tau_i^* \in \mathbb{N}.$$

1. *For each* $i, 0 \leq i \leq n$ *holds:* $\tau_i^* \in \mathbb{N}$.
2. *For each enabled transition* t *at marking* $m_n (= m_n^*)$ *it holds:*

   2.1 $h_n(t)^* = \lfloor h_n(t) \rfloor$.

   2.2 $\sum_{i=1}^{n} \tau_i^* = \lfloor \sum_{i=1}^{n} \tau_i \rfloor$

State Space Reduction

Theorem (2 – similar to 1)

Let Z be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a feasible transition sequence in Z, with a run $\sigma(\tau)$ as an executuion of σ, i.e.

$$z_0 \xrightarrow{\tau_0}\xrightarrow{t_0} \cdots \xrightarrow{\tau_n}\xrightarrow{t_n} z_n = (m_n, h_n),$$

and all $\tau_i \in \mathbb{R}_0^+$.
Then, there exists a further feasible run $\sigma(\tau^*)$ of σ with

$$z_0 \xrightarrow{\tau_0^*}\xrightarrow{t_0} \cdots \xrightarrow{\tau_n^*}\xrightarrow{t_n} z_n^* = (m_n^*, h_n^*).$$

such that

State Space Reduction

Theorem (2 – continuation)

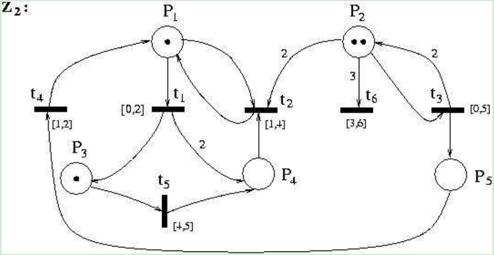
1. *For each* $i, 0 \leq i \leq n$ *the time* τ_i^* *is a natural number.*
2. *For each enabled transition* t *at marking* $m_n (= m_n^*)$ *it holds:*

    2.1 $h_n(t)^* = \lceil h_n(t) \rceil$.

    2.2 $\sum_{i=1}^{n} \tau_i^* = \lceil \sum_{i=1}^{n} \tau_i \rceil$

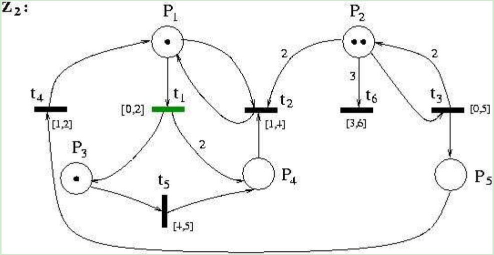
State Space Reduction

Example



State Space Reduction

Example

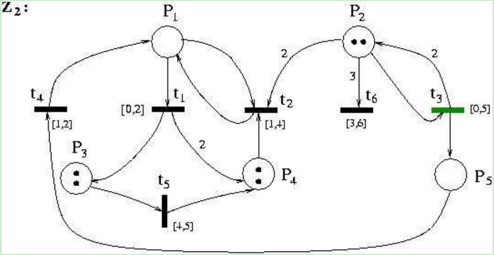


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

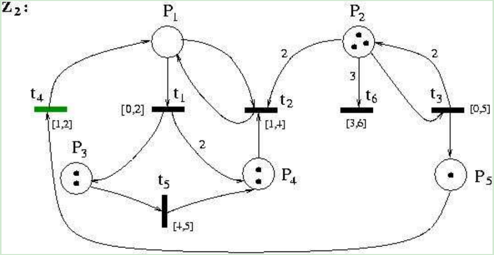


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

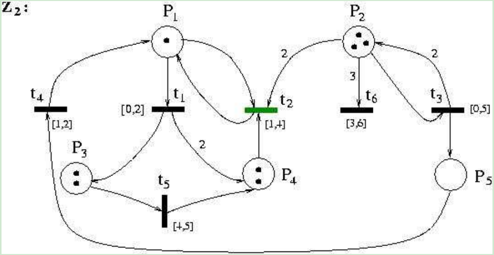


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

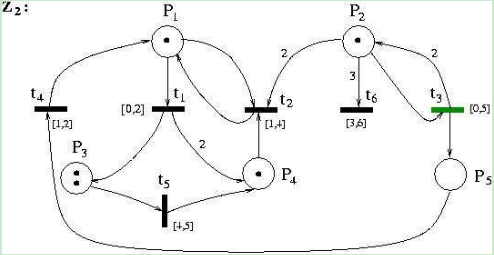


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

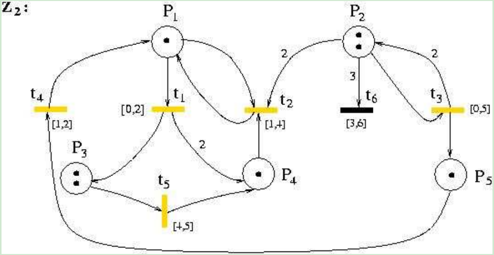


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

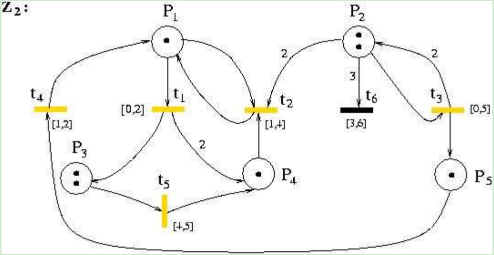


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

State Space Reduction

Example

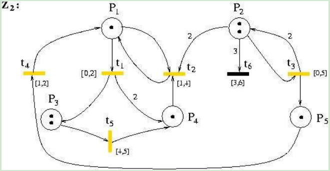


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

$$m_\sigma = (1, 2, 2, 1, 1)$$

State Space Reduction

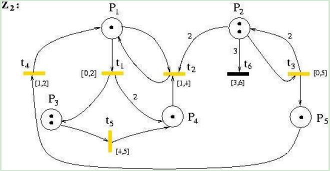
Example (continuation)



$$\Sigma_\sigma = \begin{pmatrix} x_4 + x_5 \\ x_5 \\ x_5 \\ x_5 \\ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \\ \sharp \end{pmatrix} \quad \text{and}$$

State Space Reduction

Example (continuation)



$$B_\sigma = \left\{ \begin{array}{lll} 0 \le x_0, & x_0 \le 2, & x_0 + x_1 + x_2 \le 5 \\ 0 \le x_1, & x_2 \le 2, & x_2 + x_3 \le 5 \\ 1 \le x_2, & x_3 \le 2, & x_0 + x_1 + x_2 + x_3 \le 5 \\ 1 \le x_3, & x_4 \le 2, & x_0 + x_1 + x_2 + x_3 + x_4 \le 5 \\ 0 \le x_4, & x_5 \le 2, & x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \le 5 \\ 0 \le x_5, & x_0 + x_1 \le 5 & x_4 + x_5 \le 2 \end{array} \right\}.$$

State Space Reduction

Example (continuation)

The run $\sigma(\tau)$ with

$$\sigma(\tau) := z_0 \xrightarrow{0.7} \xrightarrow{t_1} \xrightarrow{0.0} \xrightarrow{t_3} \xrightarrow{0.4} \xrightarrow{t_4} \xrightarrow{1.2} \xrightarrow{t_2} \xrightarrow{0.5} \xrightarrow{t_3} \xrightarrow{1.4} z$$

is feasible.

State Space Reduction

Example (continuation)

The run $\sigma(\tau)$ with

$$\sigma(\tau) := z_0 \xrightarrow{\mathbf{0.7}} \xrightarrow{t_1} \xrightarrow{\mathbf{0.0}} \xrightarrow{t_3} \xrightarrow{\mathbf{0.4}} \xrightarrow{t_4} \xrightarrow{\mathbf{1.2}} \xrightarrow{t_2} \xrightarrow{\mathbf{0.5}} \xrightarrow{t_3} \xrightarrow{\mathbf{1.4}} z$$

is feasible.

State Space Reduction

Example (continuation)

The run $\sigma(\tau)$ with

$$z_0 \xrightarrow{\mathbf{0.7}} \xrightarrow{t_1} \xrightarrow{\mathbf{0.0}} \xrightarrow{t_3} \xrightarrow{\mathbf{0.4}} \xrightarrow{t_4} \xrightarrow{\mathbf{1.2}} \xrightarrow{t_2} \xrightarrow{\mathbf{0.5}} \xrightarrow{t_3} \xrightarrow{\mathbf{1.4}} \left(m, \begin{pmatrix} \mathbf{1.9} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{4.2} \\ \sharp \end{pmatrix}\right)$$

is feasible.

State Space Reduction

Example (continuation)

		x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta}$ =	β_0	0.7	0.0	0.4	1.2	0.5	1.4	1.9	1.4	4.2
	β_1	0.7	0.0	0.4	1.2	0.5	**1**	1.5	1.0	3.8
	β_2	0.7	0.0	0.4	1.2	**0**	1	1.0		3.3
	β_3	0.7	0.0	0.4	**1**	0	1			3.1
	β_4	0.7	0.0	**1**	1	0	1			3.7
	β_5	0.7	**0**	1	1	0	1			3.7
	β_6	**1**	0	1	1	0	1			4.0

State Space Reduction

Example (continuation)

		x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta} =$	β_0	0.7	0.0	0.4	1.2	0.5	1.4	1.9	1.4	4.2
	β_1	0.7	0.0	0.4	1.2	0.5	**2**	2.5	2.0	4.8
	β_2	0.7	0.0	0.4	1.2	**0**	1	2.0		4.3
	β_3	0.7	0.0	0.4	**2**	0	1			5.1
	β_4	0.7	0.0	**0**	1	0	1			4.7
	β_5	0.7	**0**	1	1	0	1			4.7
	β_6	**1**	0	1	1	0	1			5.0

State Space Reduction

Example (continuation)

Hence, the runs

$$\sigma(\tau_1^*) := z_0 \xrightarrow{\mathbf{1}} \xrightarrow{t_1} \xrightarrow{\mathbf{0}} \xrightarrow{t_3} \xrightarrow{\mathbf{1}} \xrightarrow{t_4} \xrightarrow{\mathbf{1}} \xrightarrow{t_2} \xrightarrow{\mathbf{0}} \xrightarrow{t_3} \xrightarrow{\mathbf{1}} \lfloor z \rfloor$$

and

$$\sigma(\tau_2^*) := z_0 \xrightarrow{\mathbf{1}} \xrightarrow{t_1} \xrightarrow{\mathbf{0}} \xrightarrow{t_3} \xrightarrow{\mathbf{0}} \xrightarrow{t_4} \xrightarrow{\mathbf{2}} \xrightarrow{t_2} \xrightarrow{\mathbf{0}} \xrightarrow{t_3} \xrightarrow{\mathbf{2}} \lceil z \rceil$$

are feasible in Z, too.

State Space Reduction

Corollary

- *Each feasible t-sequence σ in Z can be realized with an "integer" run.*

State Space Reduction

Corollary

- *Each feasible t-sequence σ in Z can be realized with an "integer" run.*
- *Each reachable marking in Z can be found using "integer" runs only.*

State Space Reduction

Corollary

- *Each feasible t-sequence σ in Z can be realized with an "integer" run.*
- *Each reachable marking in Z can be found using "integer" runs only.*
- *If z is reachable in Z, then $\lfloor z \rfloor$ and $\lceil z \rceil$ are reachable in Z, too.*

State Space Reduction

Corollary

- *Each feasible t-sequence σ in Z can be realized with an "integer" run.*
- *Each reachable marking in Z can be found using "integer" runs only.*
- *If z is reachable in Z, then $\lfloor z \rfloor$ and $\lceil z \rceil$ are reachable in Z, too.*
- *The length of the shortest and longest time path between two arbitrary states are natural numbers.*

State Space Reduction

Definition

A state $z = (m, h)$ in a TPN is **integer** one iff
for all enabled transitions t at m holds: $h(t) \in \mathbb{N}$.

State Space Reduction

Definition

A state $z = (m, h)$ in a TPN is **integer** one iff
for all enabled transitions t at m holds: $h(t) \in \mathbb{N}$.

Theorem (3)

Let Z be a FTPN.
The set of all reachable integer states in Z is finite

if and only if

the set of all reachable markings in Z is finite.

State Space Reduction

Definition

A state $z = (m, h)$ in a TPN is **integer** one iff
for all enabled transitions t at m holds: $h(t) \in \mathbb{N}$.

Theorem (3)

Let Z be a FTPN.
The set of all reachable integer states in Z is finite

if and only if

the set of all reachable markings in Z is finite.

Remark: Theorem 3 can be generalized for all TPNs (applying a further reduction).

Reachability Graph

Definition

Basis)

$z_0 \in RG(Z)$

Reachability Graph

Definition

Basis)

$z_0 \in RG(Z)$

Step)

Let z be in $RG(Z)$ already.

Reachability Graph

Definition

Basis)

$z_0 \in RG(Z)$

Step)

Let z be in $RG(Z)$ already.

1. *for* $i=1$ *to* n *do*

   if $z \xrightarrow{t_i} z'$ *possible in* Z *then* $z' \in RG(Z)$ *end*

Reachability Graph

Definition

Basis)

$z_0 \in RG(Z)$

Step)

Let z be in $RG(Z)$ *already.*

1. $\underline{for}$ $i{=}1$ $\underline{to}$ n $\underline{do}$

   $\underline{if}$ $z \xrightarrow{t_i} z'$ *possible in* Z $\underline{then}$ $z' \in RG(Z)$ $\underline{end}$

2. $\underline{if}$ $z \xrightarrow{1} z'$ *possible in* Z $\underline{then}$ $z' \in RG(Z)$

Reachability Graph

Definition

Basis)

$z_0 \in RG(Z)$

Step)

Let z *be in* $RG(Z)$ *already.*

1. $\underline{for}$ $i=1$ $\underline{to}$ n $\underline{do}$

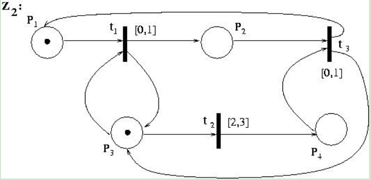
   $\underline{if}$ $z \xrightarrow{t_i} z'$ *possible in* Z $\underline{then}$ $z' \in RG(Z)$ $\underline{end}$

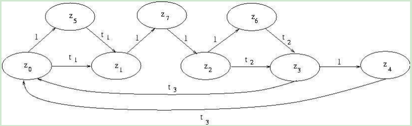
2. $\underline{if}$ $z \xrightarrow{1} z'$ *possible in* Z $\underline{then}$ $z' \in RG(Z)$

$\Longrightarrow$ The reachability graph is a weighted directed graph.

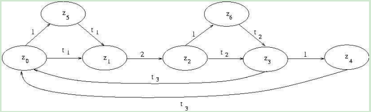
A TPN and its full Reachability Graph

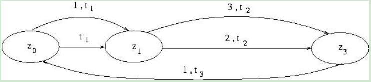
Example (A TPN Z and its full reachability graph $RG^{(1)}(Z)$)



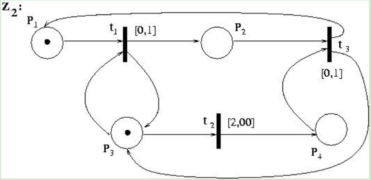


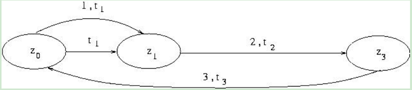
Example (The reduced reachability graphs $RG^{(2)}(Z)$ and $RG(Z)$)





Example (The reachability graph $RG(Z_3)$)





Definition

The transition sequence σ is a **feasible T-invariant** in a TPN Z if for each marking m in Z holds: $m \xrightarrow{\sigma} m$.

Definition

The transition sequence σ is a **feasible T-invariant** in a TPN Z if for each marking m in Z holds: $m \xrightarrow{\sigma} m$.

For **timeless PN:** σ is a feasible T-invariant iff
$m = m + C \cdot \psi(\sigma)$ and $\psi(\sigma)$ - the Parikh-vektor of σ.
$\Longrightarrow$ easy to be found.

Lemma

Let Z be a TPN, $S(Z)$ be the skeleton of Z and σ be a feasible T-invariant in $S(Z)$.
σ is a feasible T-invariant in Z **iff** *B_σ has a solution.*

Lemma

Let Z be a TPN, $S(Z)$ be the skeleton of Z and σ be a feasible T-invariant in $S(Z)$.
σ is a feasible T-invariant in Z **iff** *B_σ has a solution.*

Computing the T-invariants of a Z:

- Solve the linear system of equations $C \cdot x = 0$ for $x \in \mathbb{N}$.

Lemma

Let Z be a TPN, $S(Z)$ be the skeleton of Z and σ be a feasible T-invariant in $S(Z)$.
σ is a feasible T-invariant in Z **iff** *B_σ has a solution.*

Computing the T-invariants of a Z:

- Solve the linear system of equations $C \cdot x = 0$ for $x \in \mathbb{N}$.
- Decide feasibility of a T-invariant σ with $\text{Parikh}(\sigma) = x$.

Lemma

Let Z be a TPN, $S(Z)$ be the skeleton of Z and σ be a feasible T-invariant in $S(Z)$.
σ is a feasible T-invariant in Z **iff** *B_σ has a solution.*

Computing the T-invariants of a Z:

- Solve the linear system of equations $C \cdot x = 0$ for $x \in \mathbb{N}$.
- Decide feasibility of a T-invariant σ with Parikh$(\sigma) = x$.
- If σ is feasible, then solve the linear system of inequalities B_σ in $\mathbb{R}_0^+$.

Remark: The reachability graph of a TPN is not used for computing the feasible T-invariants of Z

$$\Longrightarrow$$

feasible T-invariants for **unbounded** nets can be computed!

Let $Z = (P, T, F, V, I, m_o)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Let $Z = (P, T, F, V, I, m_o)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Result 1:

Input:	The time function I is fixed, σ is an arbitrary transition sequence.
Output:	Feasibility of σ in Z?
Solution:	Solve a linear system of inequalities in $\mathbb{R}_0^+$.

Let $Z = (P, T, F, V, I, m_o)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Result 2:

Input:	The time function I is not fixed, σ is an arbitrary transition sequence.
Output:	Feasibility of σ in Z for a fixed I?
Solution:	Solve a linear system of inequalities in $\mathbb{Q}_0^+$.

Let $Z = (P, T, F, V, I, m_o)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Result 3:

Input:	The time function I is fixed, σ is an arbitrary transition sequence.
Output:	min / max-length of σ.
Solution:	Solve a linear program in $\mathbb{R}_0^+$. (Actually, the solution is in $\mathbb{N}$.)

Let $Z = (P, T, F, V, I, m_o)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Result 4:

Input: The time function I is not fixed,
σ is an arbitrary transition sequence,
λ is an arbitrary real number.

Output: Existence of a fixed I and a run $\sigma(\tau)$ in Z and the length of $\sigma(\tau) \leq \lambda$?

Solution: Solve a linear program in $\mathbb{Q}_0^+$.

Result 5:

Input: The time function I is not fixed,
$\sigma_1 = (\sigma, t')$ is a arbitrary t-sequence and
$\sigma_2 = (\sigma, t'')$ is a arbitrary t-sequence.

Output: Existence of a fixed I so that σ_1 is feasible in Z and σ_2 is not feasible in Z?

Solution: Solve

$$\underbrace{\max\{< c', x >| \ A' \cdot x \leq b'\}}_{\text{linear program in } \mathbb{Q}_0^+} < \underbrace{\min\{< c'', x >| \ A'' \cdot x \leq b''\}}_{\text{linear program in } \mathbb{Q}_0^+}.$$

Let $Z = (P, T, F, V, I, m_o)$ be a bounded TPN. Additionally the following problems can be decided/computed with the knowledge of its RG, amongst others:

Let $Z = (P, T, F, V, I, m_o)$ be a bounded TPN. Additionally the following problems can be decided/computed with the knowledge of its RG, amongst others:

Result 6:

Input: z and z' - two states (in Z).

Output: – Is there a path between z and z' in $RG(Z)$?
– If yes, compute the path with the shortest time length.

Solution: By means of prevalent methods of the graph theory, e.g. Bellman-Ford algorithm (the running time is $\mathcal{O}(|V| \cdot |E|)$ and $RG(Z) = (V, E)$)

Let $Z = (P, T, F, V, I, m_o)$ be a bounded TPN. Additionally the following problems can be decided/computed with the knowledge of its RG, amongst others:

Result 7:

Input: m and m' - two markings (in Z).

Output:
- Is there a path between m and m' in $RG(Z)$?
- If yes, compute the path with the shortest time length.

Solution: By means of prevalent methods of the graph theory, for computing all-pairs shortest paths.
The running time is polynomial, too.

Definition

The **longest path** between two states (vertices in $RG(Z)$) z and z' is $lp(z, z')$ with

$$lp(z,z') := \begin{cases} \infty & \text{, if a cycle is reachable starting on } z \\ \max \sum\limits_{\sigma(\tau)} \tau_i & \text{, if } z \xrightarrow{\sigma(\tau)} z' \end{cases}$$

Result 8:

Input: z and z' - two states (in Z).

Output:
– Is there a path between z and z' in $RG(Z)$?
– If yes, compute the path with the longest time length.

Solution: By means of prevalent methods of the graph theory, e.g. Bellman-Ford algorithm (polyn. running time). or by computing all strongly connected components of $RG(Z)$. (linear running time)

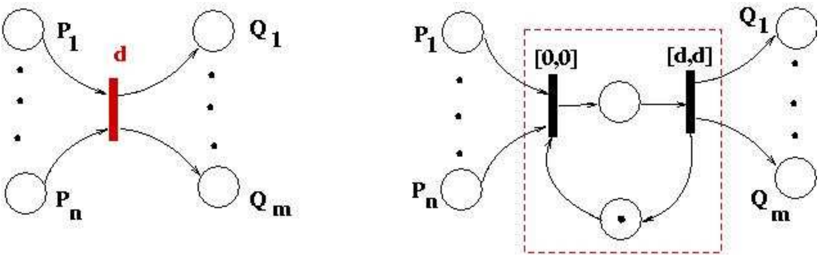
Result 9:

Input: m and m' - two states (in Z).

Output:
– Is there a path between z and z' in $RG(Z)$?
– If yes, compute the path with the longest time length.

Solution: By means of prevalent methods of the graph theory, e.g. Bellman-Ford algorithm (polyn. running time). or by computing all strongly connected components of $RG(Z)$. (linear running time)

Transformation Timed PN $\longrightarrow$ Time PN



Conclusion

- theoretical approach

  BN $\Longrightarrow$ *modelling* $\Longrightarrow$ PN $\Longrightarrow$ *modelling of steady state* $\Longrightarrow$

  DPN $\Longrightarrow$ *analysing* $\Longrightarrow$ TPN

- experimental approach

  BN $\Longrightarrow$ *modelling & analysing* $\Longrightarrow$ TPN