Time Petri Net State Space Reduction Using Dynamic Programming and Time Paths

Louchka Popova-Zeugmann

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

IFORS 2005, Hawaii
July 11-15, 2005

Berlin - Brandenburger Tor

Time Petri Net State Space Reduction Using Dynamic Programming and Time Paths

Louchka Popova-Zeugmann

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

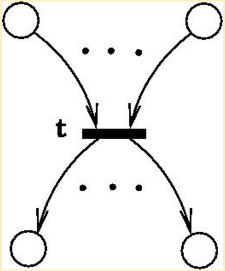
IFORS 2005, Hawaii
July 11-15, 2005

Outline

Time Petri Net

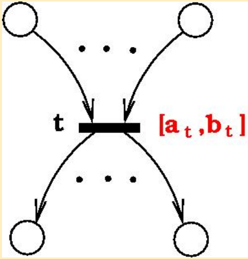
Time Petri Net

Definition (informal)



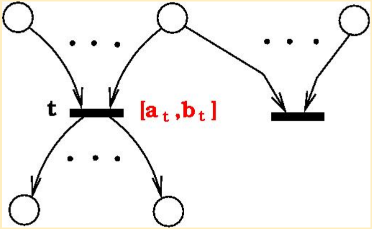
Time Petri Net

Definition (informal)



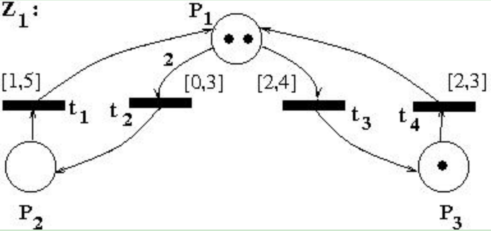
Time Petri Net

Definition (informal)



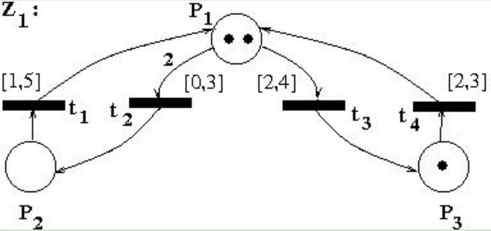
Time Petri Net

Example



Time Petri Net

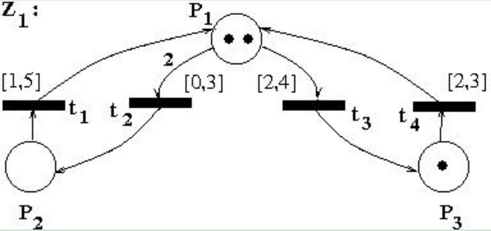
Example



- $m_0 = (2, 0, 1)$

Time Petri Net

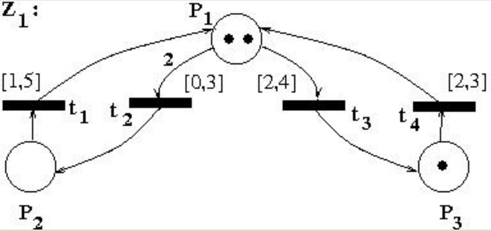
Example



- $m_0 = (2, 0, 1)$ *p*-marking

Time Petri Net

Example



- $m_0 = (2, 0, 1)$ *p*-marking
- $h_0 = (\sharp, 0, 0, 0)$ *t*-marking

state

Definition (state)

$z = (m, h)$ is called a **state** in a TPN Z iff:

state

Definition (state)

$z = (m, h)$ is called a **state** in a TPN Z iff:

- m is a p-marking in Z .

state

Definition (state)

$z = (m, h)$ is called a **state** in a TPN Z iff:

- m is a p-marking in Z .
- h is a t-marking in Z.

Definition (state changing)

Definition (state changing)

Let Z be a TPN, and $z = (m, h)$, $z' = (m', h')$ be two states.

Definition (state changing)

Let Z be a TPN, and $z = (m, h)$, $z' = (m', h')$ be two states. Then

$$z = (m, h) \text{ changes into } z' = (m', h')$$

Definition (state changing)

Let Z be a TPN, and $z = (m, h)$, $z' = (m', h')$ be two states. Then

$z = (m, h)$ changes into $z' = (m', h')$ by

firing
a transition /

Definition (state changing)

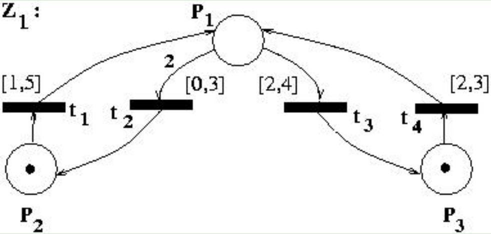
Let Z be a TPN, and $z = (m, h)$, $z' = (m', h')$ be two states. Then

$$z = (m, h) \text{ changes into } z' = (m', h') \text{ by}$$

firing a transition / \ **time elapsing**

Time Petri Net

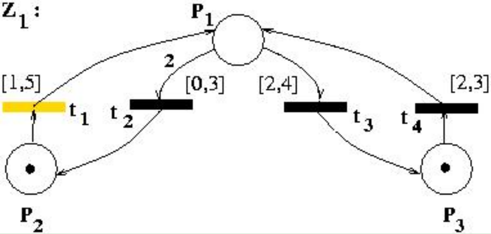
Example



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

Time Petri Net

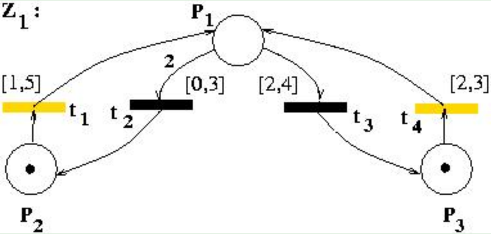
Example



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

Time Petri Net

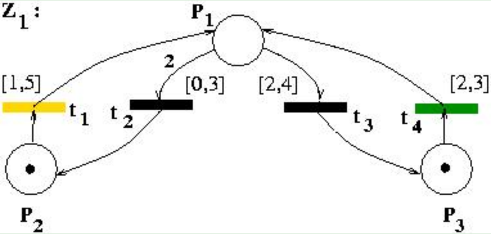
Example



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

Time Petri Net

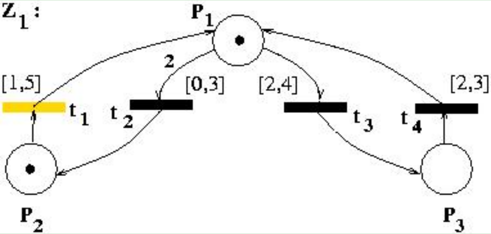
Example



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

Time Petri Net

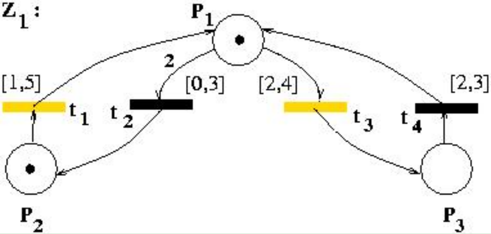
Example



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

Time Petri Net

Example



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

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:** $\sigma(\tau) = (\tau_0, t_1, \tau_1, \cdots, \tau_{n-1}, t_n, \tau_n)$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$
- **run:** $\sigma(\tau) = (\tau_0, t_1, \tau_1, \cdots, \tau_{n-1}, t_n, \tau_n)$
- **feasible run:** $z_0 \xrightarrow{\tau_0} z_0^* \xrightarrow{t_1} z_1 \xrightarrow{\tau_1} z_1^* \cdots \xrightarrow{t_n} z_n \xrightarrow{\tau_n} z_n^*$

Transition sequences, Runs

Definition

- **transition sequence:** $\sigma = (t_1, \cdots, t_n)$
- **run:** $\sigma(\tau) = (\tau_0, t_1, \tau_1, \cdots, \tau_{n-1}, t_n, \tau_n)$
- **feasible run:** $z_0 \xrightarrow{\tau_0} z_0^* \xrightarrow{t_1} z_1 \xrightarrow{\tau_1} z_1^* \cdots \xrightarrow{t_n} z_n \xrightarrow{\tau_n} z_n^*$
- **feasible transition sequence :** σ is feasible if there ex. a feasible run $\sigma(\tau)$

Reachable state, Reachable marking, State space

Definition

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

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 sum of variables,
  Σ_σ is a parametrical $t-$marking

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 sum of variables,
  Σ_σ is a parametrical $t-$marking
- B_σ is a set of conditions (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 sum of variables,
  Σ_σ is a parametrical $t-$marking
- B_σ is a set of conditions (a system of inequalities)

Obviously

- $z_0 \xrightarrow{\sigma} (m_\sigma, \Sigma_\sigma) =: z_\sigma$,

Parametric Description of the State Space

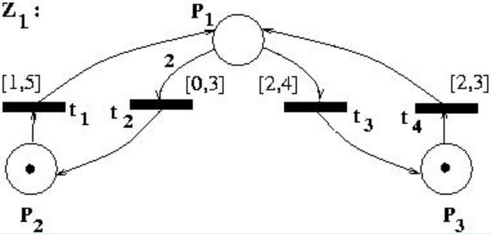
Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \cdots, t_n)$ be a transition sequence in Z.
$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of σ, if

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a sum of variables,
  Σ_σ is a parametrical $t-$marking
- B_σ is a set of conditions (a system of inequalities)

Obviously

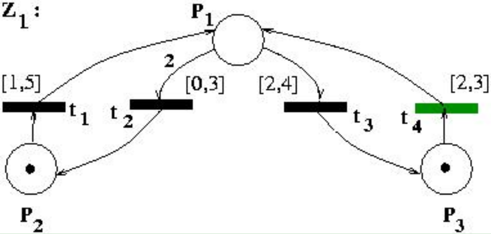
- $z_0 \xrightarrow{\sigma} (m_\sigma, \Sigma_\sigma) =: z_\sigma$,
- $StSp(Z) = \bigcup_\sigma z_\sigma$.

Example



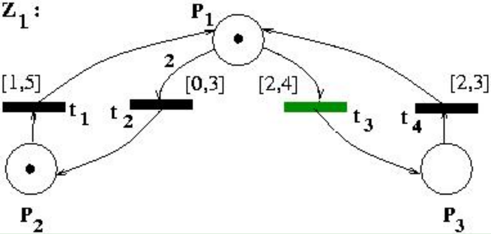
$$\sigma = (t_4, t_3)$$

Example



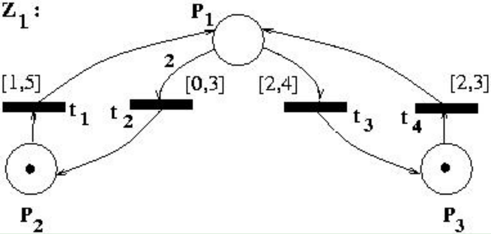
$$\sigma = (t_4, t_3)$$

Example



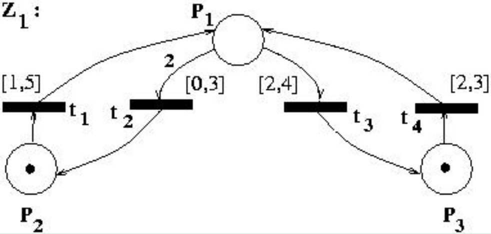
$$\sigma = (t_4, t_3)$$

Example



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

Example

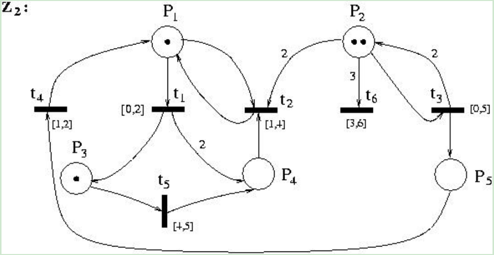


$$\sigma = (t_4, t_3) \quad \Longrightarrow \quad \delta(\sigma) =$$

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

State Space Reduction

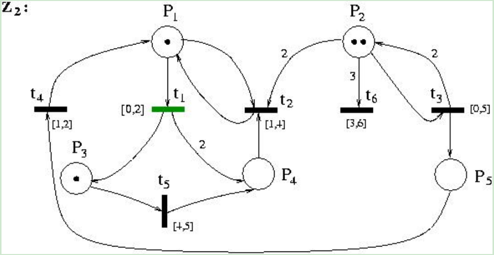
Example



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

State Space Reduction

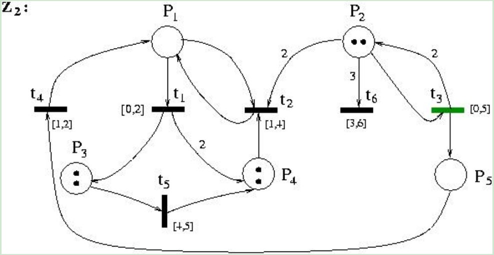
Example



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

State Space Reduction

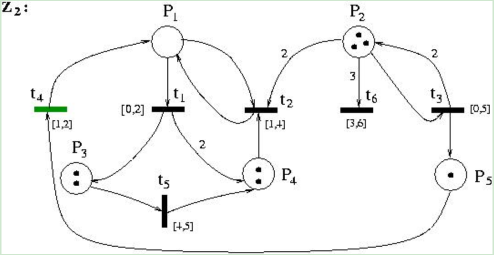
Example



$$\sigma = (t_1 \; {\color{green}t_3} \; t_4 \; t_2 \; t_3)$$

State Space Reduction

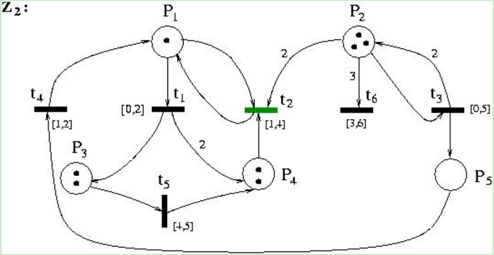
Example



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

State Space Reduction

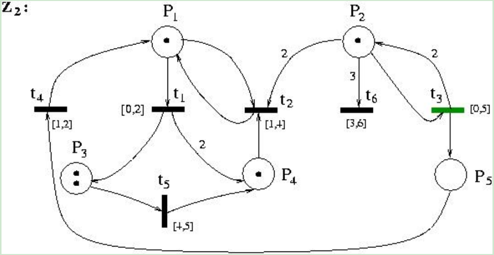
Example



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

State Space Reduction

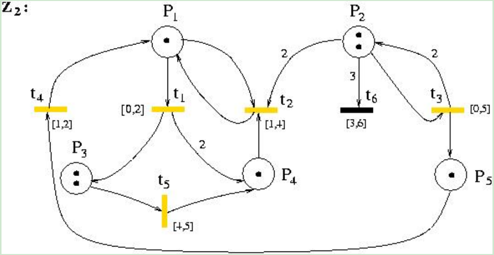
Example



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

State Space Reduction

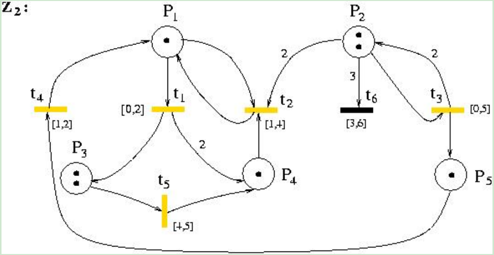
Example



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

State Space Reduction

Example

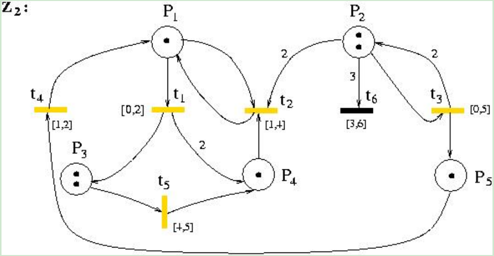


$$\sigma = (t_1 \;\; t_3 \;\; t_4 \;\; t_2 \;\; t_3)$$

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

State Space Reduction

Example

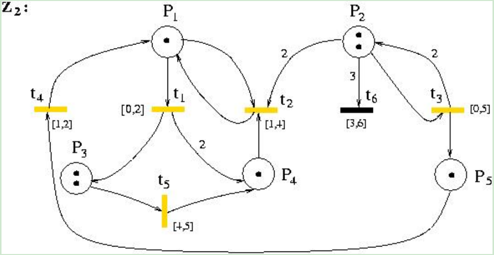


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

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

State Space Reduction

Example

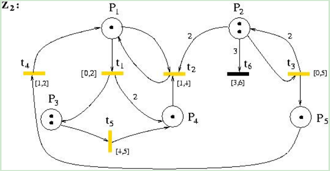


$$\sigma = (t_1 t_3 t_4 t_2 t_3)$$

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

State Space Reduction

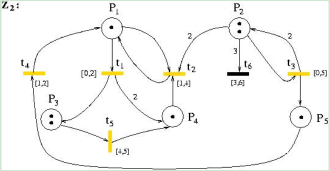
Example (continuation)



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

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

$\sigma(\tau) =$

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

is feasible.

State Space Reduction

Example (continuation)

$$\underbrace{\left(m_\sigma, \begin{pmatrix} \mathbf{1.9} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{4.2} \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(\hat{\beta})} z}$$

State Space Reduction

Example (continuation)

$$\underbrace{\left(m_\sigma, \begin{pmatrix} \mathbf{1.0} \\ \mathbf{1.0} \\ \mathbf{1.0} \\ \mathbf{1.0} \\ \mathbf{4.0} \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(?)} \lfloor z \rfloor} \qquad \underbrace{\left(m_\sigma, \begin{pmatrix} \mathbf{1.9} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{1.4} \\ \mathbf{4.2} \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(\hat{\beta})} z}$$

State Space Reduction

Example (continuation)

$$\underbrace{\left(m_\sigma, \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 4.0 \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(?)} \lfloor z \rfloor} \qquad \underbrace{\left(m_\sigma, \begin{pmatrix} 1.9 \\ 1.4 \\ 1.4 \\ 1.4 \\ 4.2 \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(\hat{\beta})} z} \qquad \underbrace{\left(m_\sigma, \begin{pmatrix} 2.0 \\ 2.0 \\ 2.0 \\ 2.0 \\ 5.0 \\ \sharp \end{pmatrix}\right)}_{z_0 \xrightarrow{\sigma(?)} \lceil z \rceil}$$

State Space Reduction

Example (continuation)

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

Example (continuation)

The runs

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

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

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

are feasible in Z, too.

State Space Reduction

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)

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

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

1. *For each $i, 0 \leq i \leq n$ the time τ_i^* is a natural number.*

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 $i, 0 \leq i \leq n$ the time τ_i^* is a natural number.*
2. *For each enabled transition t at marking $m_n(= m_n^*)$ it holds:*
   - 2.1 $h_n(t)^* = \lfloor h_n(t) \rfloor$.
   - 2.2 $\sum_{i=1}^{n} \tau_i^* = \lfloor \sum_{i=1}^{n} \tau_i \rfloor$

State Space Reduction

Theorem (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 $i, 0 \leq i \leq n$ the time τ_i^* is a natural number.*
2. *For each enabled transition t at marking $m_n (= m_n^*)$ it holds:*
   2.1 $h_n(t)^* = \lfloor h_n(t) \rfloor$.
   2.2 $\sum_{i=1}^{n} \tau_i^* = \lfloor \sum_{i=1}^{n} \tau_i \rfloor$
3. *For each transition $t \in T$ holds:*
   t is ready to fire in z_n iff t is ready to fire in $\lfloor z_n \rfloor$, too.

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 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 (2 – continuation)

1. *For each* $i, 0 \le i \le n$ *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 $\sum_{i=1}^{n} \tau_i^* = \lceil \sum_{i=1}^{n} \tau_i \rceil$
3. *For each transition* $t \in T$ *holds:*
   t *is ready to fire in* z_n *i* t *is ready to fire in* $\lceil z_n \rceil$, *too.*

Dynamic programming

Where is
the Dynamic Programming
here?

Dynamic programming

Where is the Dynamic Programming here?

Let us consider the previous example again

Dynamic programming

Input:

- **The TPN Z_2,**

Dynamic programming

Input:

- **The TPN Z_2,**
- **the transition sequence $\sigma = (t_1, t_3, t_4, t_2, t_3)$**

Dynamic programming

Input:

- **The TPN Z_2,**
- **the transition sequence $\sigma = (t_1, t_3, t_4, t_2, t_3)$**
- **the six elapses of time**
  $\hat{\beta}(x_0) = 0.7,\quad \hat{\beta}(x_1) = 0.0,\quad \hat{\beta}(x_2) = 0.4,$
  $\hat{\beta}(x_3) = 1.2,\quad \hat{\beta}(x_4) = 0.5,\quad \hat{\beta}(x_5) = 1.4,$
  which are real numbers and

Dynamic programming

Input:

- **The TPN Z_2,**
- **the transition sequence $\sigma = (t_1, t_3, t_4, t_2, t_3)$**
- **the six elapses of time**
  $\hat{\beta}(x_0) = 0.7, \quad \hat{\beta}(x_1) = 0.0, \quad \hat{\beta}(x_2) = 0.4,$
  $\hat{\beta}(x_3) = 1.2, \quad \hat{\beta}(x_4) = 0.5, \quad \hat{\beta}(x_5) = 1.4$**,**
  which are real numbers and
- **the run**
  $\sigma(\hat{\beta}) = (0.7,\ t_1,\ 0.0,\ t_3,\ 0.4,\ t_4,\ 1.2,\ t_2,\ 0.5,\ t_3,\ 1.4)$
  is a feasible one in Z_2.

Dynamic programming

Output:

- **Six elapses of time $\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_5)$ which are integers,**

Dynamic programming

Output:

- **Six elapses of time** $\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_5)$ **which are integers,**
- $\sigma(\beta^*)$ **is a feasible run in** Z_2**.**

Dynamic programming

Output:

- **Six elapses of time $\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_5)$ which are integers,**
- **$\sigma(\beta^*)$ is a feasible run in Z_2.**
- **The set of transitions which are ready to fire after $\sigma(\hat{\beta})$ is the same as the set of transitions which are ready to fire after $\sigma(\beta^*)$.**

Dynamic programming

Output:

- **Six elapses of time $\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_5)$ which are integers,**
- **$\sigma(\beta^*)$ is a feasible run in Z_2.**
- **The set of transitions which are ready to fire after $\sigma(\hat{\beta})$ is the same as the set of transitions which are ready to fire after $\sigma(\beta^*)$.**

$\Longrightarrow$ P* : Compute β^*.

Dynamic programming

$P^*(s)$

Compute

- six elapses of time $\beta_s(x_0), \beta_s(x_1), \cdots, \beta_s(x_5)$,

Dynamic programming

$P^*(s)$

Compute

- six elapses of time $\beta_s(x_0), \beta_s(x_1), \cdots, \beta_s(x_5)$,
- at least s of them are integers,

Dynamic programming

$P^*(s)$

Compute

- six elapses of time $\beta_s(x_0), \beta_s(x_1), \cdots, \beta_s(x_5)$,
- at least s of them are integers,
- the modified run is a feasible one.

Dynamic programming

$z^*(s)$

- modifies one elapse of time which is not integer in $P^*(s-1)$ to such an integer that the modified run remains feasible.

Dynamic programming

$z^*(s)$

- modifies one elapse of time which is not integer in $P^*(s-1)$ to such an integer that the modified run remains feasible.
- Each row s $(s = 0, 1, \cdots, 6)$ in the next tableau I is a solution of one modified problem $P^*(s)$.

Dynamic programming

$z^*(s)$

- modifies one elapse of time which is not integer in $P^*(s-1)$ to such an integer that the modified run remains feasible.
- Each row s $(s = 0, 1, \cdots, 6)$ in the next tableau I is a solution of one modified problem $P^*(s)$.

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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				

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\quad$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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**			

$$\Sigma_\sigma(t_1) = x_4 + x_5, \qquad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$$

$$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$$

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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		1			

$$\Sigma_\sigma(t_1) = x_4 + x_5, \qquad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$$
$$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$$

Dynamic Programming

I	x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta} = \beta_0$	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			

$$\Sigma_\sigma(t_1) = x_4 + x_5, \qquad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$$

$$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$$

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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		0	1			

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\quad\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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			

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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	0	1			

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I	x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta} = \beta_0$	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			

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5,$ $\quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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		1	1	0	1			

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I	x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta} = \beta_0$	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			

$$\Sigma_\sigma(t_1) = x_4 + x_5, \qquad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$$

$$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$$

Dynamic Programming

I			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

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

$\mathbf{I}$	x_0	x_1	x_2	x_3	x_4	x_5	$\Sigma_\sigma(t_1)$	$\Sigma_\sigma(t_2)$	$\Sigma_\sigma(t_5)$
$\hat{\beta} = \beta_0$	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
$\beta^* = \beta_6$		0	1	1	0	1			

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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^*	$=$	β_6	**1**	0	1	1	0	1			

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

I			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^*	=	β_6	**1**	0	1	1	0	1			**4.0**

$\Sigma_\sigma(t_1) = x_4 + x_5$, $\quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5$

$\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$

Dynamic Programming

- The *state space* (for P*) is the set $S = \{0, 1, \dots, 6\}$.

Dynamic Programming

- The *state space* (for P*) is the set $S = \{0, 1, \ldots, 6\}$.
- The *set of its critical states* is the singleton $\mathrm{S}^{\mathrm{O}} = \{6\}$.

Dynamic Programming

- The *state space* (for P*) is the set $S = \{0, 1, \ldots, 6\}$.
- The *set of its critical states* is the singleton $\mathsf{S}^{\mathsf{o}} = \{6\}$.
- The *set of its terminal states* is the singleton $\mathsf{S}^{\mathsf{t}} = \{0\}$.

Dynamic Programming

- The *state space* (for P*) is the set $S = \{0, 1, \ldots, 6\}$.
- The *set of its critical states* is the singleton $\mathsf{S}^{\mathsf{o}} = \{6\}$.
- The *set of its terminal states* is the singleton $\mathsf{S}^{\mathsf{t}} = \{0\}$.
- The *set of non-terminal states* is $\mathsf{S}'' = \mathsf{S} \setminus \mathsf{S}^{\mathsf{t}} = \{1, 2, \ldots, 6\}$.

Dynamic Programming

- The *state space* (for P^*) is the set $S = \{0, 1, \ldots, 6\}$.
- The *set of its critical states* is the singleton $\mathsf{S}^{\mathsf{o}} = \{6\}$.
- The *set of its terminal states* is the singleton $\mathsf{S}^{\mathsf{t}} = \{0\}$.
- The *set of non-terminal states* is $\mathsf{S}'' = \mathsf{S} \setminus \mathsf{S}^{\mathsf{t}} = \{1, 2, \ldots, 6\}$.
- The T-*linker* $\mathsf{L_T}$ has the form $\mathsf{L_T}(\mathsf{z}(\mathsf{s}^{\mathsf{o}})) = \mathsf{z}^{\mathsf{o}} = \mathsf{z}(\mathsf{s}^{\mathsf{o}})$.

Dynamic Programming

- The *state space* (for P*) is the set $S = \{0, 1, \ldots, 6\}$.
- The *set of its critical states* is the singleton $\mathsf{S}^{\mathsf{o}} = \{6\}$.
- The *set of its terminal states* is the singleton $\mathsf{S}^{\mathsf{t}} = \{0\}$.
- The *set of non-terminal states* is $\mathsf{S}'' = \mathsf{S} \setminus \mathsf{S}^{\mathsf{t}} = \{1, 2, \ldots, 6\}$.
- The T-*linker* L_T has the form $\mathsf{L}_\mathsf{T}(\mathsf{z}(\mathsf{s}^\mathsf{o})) = \mathsf{z}^\mathsf{o} = \mathsf{z}(\mathsf{s}^\mathsf{o})$.
- The *transition function t* is defined as

$$\mathsf{t}(\mathsf{s}) := \mathsf{s} - 1, \qquad \mathsf{s} \in \mathsf{S}''.$$

Dynamic Programming

- The *linker* L is clearly given by

$$\begin{aligned} \mathsf{z(s)} &= \mathsf{L(s, \{(s',z(s')) \mid s' \in t(s)\})}, \qquad \forall s \in \mathsf{S}'' \\ &= \mathsf{L(s, z(t(s)))} \\ &= \mathsf{L(s,z(s\text{-}1))} := \beta_s \end{aligned}$$

Dynamic Programming

The time length of the run $\sigma(\hat{\beta})$ is

$$l_{\sigma(\beta^*)} = \hat{\beta}(x_0) + \hat{\beta}(x_1) + \hat{\beta}(x_2) + \hat{\beta}(x_3) + \hat{\beta}(x_4) + \hat{\beta}(x_5) = 4.2$$

Dynamic Programming

The time length of the run $\sigma(\hat{\beta})$ is

$$l_{\sigma(\beta^*)} = \hat{\beta}(x_0) + \hat{\beta}(x_1) + \hat{\beta}(x_2) + \hat{\beta}(x_3) + \hat{\beta}(x_4) + \hat{\beta}(x_5) = 4.2$$

In tableau I: The time length of the run $\sigma(\beta^*)$ is $l_{\sigma(\beta^*)} = 4$

Dynamic Programming

The time length of the run $\sigma(\hat{\beta})$ is

$l_{\sigma(\beta^*)} = \hat{\beta}(x_0) + \hat{\beta}(x_1) + \hat{\beta}(x_2) + \hat{\beta}(x_3) + \hat{\beta}(x_4) + \hat{\beta}(x_5) = 4.2$

In tableau I: The time length of the run $\sigma(\beta^*)$ is $l_{\sigma(\beta^*)} = 4$

In tableau II: The time length of the run $\sigma(\beta^*)$ is $l_{\sigma(\beta^*)} = 5$

Dynamic Programming

The time length of the run $\sigma(\hat{\beta})$ is

$l_{\sigma(\beta^*)} = \hat{\beta}(x_0) + \hat{\beta}(x_1) + \hat{\beta}(x_2) + \hat{\beta}(x_3) + \hat{\beta}(x_4) + \hat{\beta}(x_5) = 4.2$

In tableau I: The time length of the run $\sigma(\beta^*)$ is $l_{\sigma(\beta^*)} = 4$

In tableau II: The time length of the run $\sigma(\beta^*)$ is $l_{\sigma(\beta^*)} = 5$

i.e. $l_{\sigma(\beta^*)} = 4 \leq 4.2 = l_{\sigma(\beta^*)} = 4.2 \leq 5 = l_{\sigma(\beta^*)}$

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.*

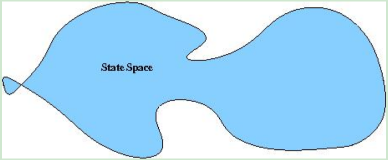
State Space Reduction

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 p-markings are natural numbers.*

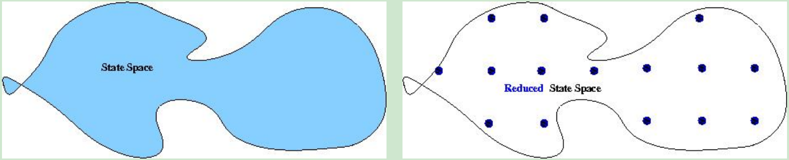
State Space Reduction

Example (State Space Reduction)



State Space Reduction

Example (State Space Reduction)



State Space Reduction

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 $p-$markings in Z is finite.

State Space Reduction

Theorem (3)

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

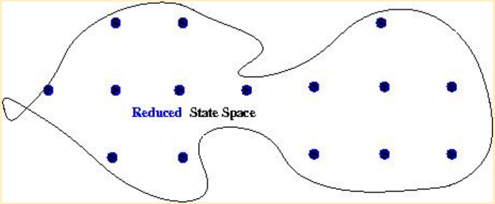
if and only if

the set of all reachable p−markings in Z is finite.

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

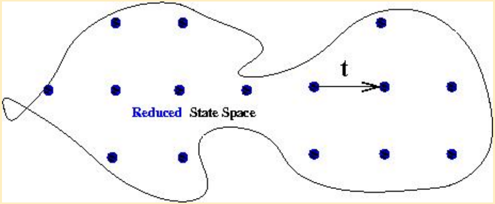
Reachability Graph

Definition (informal)



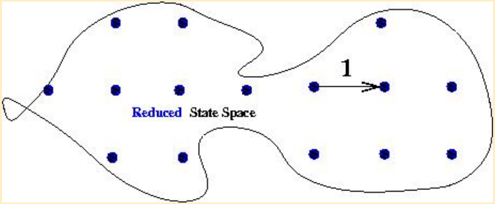
Reachability Graph

Definition (informal)



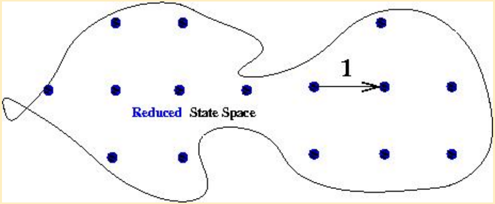
Reachability Graph

Definition (informal)



Reachability Graph

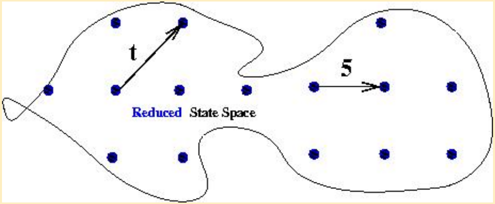
Definition (informal)



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

Reachability Graph

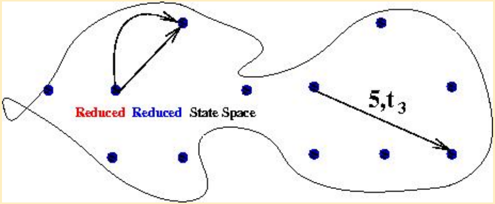
Definition (informal)



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

Reachability Graph

Definition (informal)



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

Let Z be a bounded TPN. The following problems can be decided/computed with the knowledge of its RG, **amongst others**:

Let Z be a bounded TPN. The following problems can be decided/computed with the knowledge of its RG, **amongst others**:

Result:

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 $RG(Z) = (V, E)$)

Result:

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

Output: – Is there a path between m and m'?
– 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

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

Result:

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:

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

Output: – Is there a path between z and 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.
or by computing all strongly connected components of $RG(Z)$.

Conclusion

- The State Space Reduction of a TPN is a nonoptimization truncated decision problem

Conclusion

- The State Space Reduction of a TPN is a nonoptimization truncated decision problem
- The minimal and the maximal time length of a path between two markings in a TPN is a natural number (if finite)

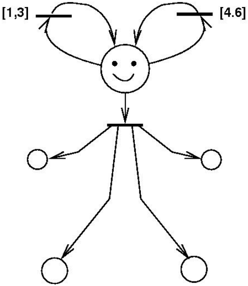
Conclusion

- The State Space Reduction of a TPN is a nonoptimization truncated decision problem
- The minimal and the maximal time length of a path between two markings in a TPN is a natural number (if finite)

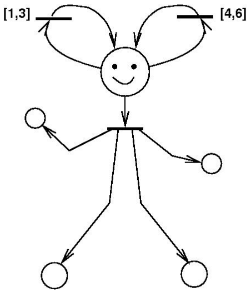
  $$\Longrightarrow$$

  it can be computed in polynomial/linear time (with res. to the RG)

Thank you!



Thank you!



Thank you!