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

Definitions
  Time Petri Net

Main Property
  State Space Reduction
  Dynamic Programming

Applications
  Reachability Graph
  Time Paths in bounded TPNs

Conclusion
Time Petri Net
Definition (informal)

Time Petri Net
Time Petri Net

Definition (informal)
Time Petri Net

Definition (informal)
Example

\[
Z_1 : \quad \begin{array}{c}
\text{P}_2 \\
\text{P}_1 \\
\text{P}_3 \\
\end{array} \\
\begin{array}{c}
[1,5] \\
[0,3] \\
[2,4] \\
[2,3] \\
\end{array} \\
\begin{array}{c}
t_1 \\
t_2 \\
t_3 \\
t_4 \\
\end{array} \\
\begin{array}{c}
2 \\
\end{array}
\]
Example

\[ m_0 = (2, 0, 1) \]
Example

$m_0 = (2, 0, 1)$ \quad p$-marking
Example

\[ m_0 = (2, 0, 1) \quad p\text{-marking} \]
\[ h_0 = (\#, 0, 0, 0) \quad t\text{-marking} \]
state

**Definition (state)**

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

1. \( m \) is a \( p \)-marking in \( Z \).
2. \( h \) is a \( t \)-marking in \( Z \).
Definition (state)

\[ z = (m, h) \] is called a state in a TPN \( Z \) iff:

- \( m \) is a \( p \)-marking in \( Z \).
**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)

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 / time elapsing.
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

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

Louchka Popova-Zeugmann  
TPN State Space Reduction Using DP and Time Paths
Example

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

\([1,5]\)

\([0,3]\)

\([2,4]\)

\([2,3]\)
**Example**

\[
\begin{pmatrix}
0 \\
\#
\end{pmatrix} \quad 1.3 \quad \rightarrow \quad \begin{pmatrix}
1.3 \\
\#
\end{pmatrix}
\]
Example

\[ z_0 \xrightarrow{1.3} (m_1, \begin{pmatrix} 1.3 \\ \# \\ \# \\ 1.3 \end{pmatrix}) \xrightarrow{1.0} (m_2, \begin{pmatrix} 2.3 \\ \# \\ \# \\ 2.3 \end{pmatrix}) \]
Example

Time Petri Net

\[ z_0 \xrightarrow{1.3} 1.0 (m_2, \begin{pmatrix} 2.3 \\ \# \\ 2.3 \end{pmatrix}) \xrightarrow{t_4} \]
Example

Time Petri Net

\[
z_0 \xrightarrow{1.3} z_1 \xrightarrow{1.0} (m_2, \begin{pmatrix} 2.3 \\ \# \\ \# \\ 2.3 \end{pmatrix}) \xrightarrow{t_4} (m_3, \begin{pmatrix} 2.3 \\ \# \\ \# \\ 0.0 \end{pmatrix})
\]
Time Petri Net

Example

\[ Z_1: \]

\[ P_1 \]

\[ P_2 \]

\[ P_3 \]

\[ z_0 \xrightarrow{1.3} t_1 \xrightarrow{1.0} t_4 \xrightarrow{2.0} (m_3, \begin{pmatrix} 2.3 \\ \# \\ 0.0 \end{pmatrix}) \]

\[ (m_4, \begin{pmatrix} 4.3 \\ \# \\ 2.0 \end{pmatrix}) \]
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**: $\sigma$ 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 exists 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 exists 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 $Z = [P, T, F, V, m_0, l]$ be a TPN and $\sigma = (t_1, \ldots, t_n)$ be a transition sequence in $Z$.

$\delta(\sigma) = [m_\sigma, \Sigma_\sigma, B_\sigma]$ is the parametric description of $\sigma$, 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 $\sigma$, if

$\quad m_0 \xrightarrow{\sigma} m_{\sigma}$
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 $\sigma$, if

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a sum of variables,
- $\Sigma_\sigma$ 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 $\sigma$, if

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

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a sum of variables,
- $\Sigma_\sigma$ is a parametrical $t$–marking
- $B_\sigma$ 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 $\sigma$, if

- $m_0 \xrightarrow{\sigma} m_\sigma$
- $\Sigma_\sigma(t)$ is a sum of variables,
  - $\Sigma_\sigma$ is a parametrical $t$–marking
- $B_\sigma$ 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

\[ \sigma = (t_4, t_3) \implies \delta(\sigma) = \{ \left( \begin{array}{c}
0 \\
1 \\
1
\end{array} \right), \left( \begin{array}{c}
x_1 + x_2 + x_3 \\
\# \\
\# \\
x_3
\end{array} \right) \mid 2 \leq x_1 \leq 3, \quad x_1 + x_2 \leq 5 \]

\[ 2 \leq x_2 \leq 4, \quad x_1 + x_2 + x_3 \leq 5 \}

0 \leq x_3 \leq 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

\[ \sigma = (t_1 \ t_3 \ t_4 \ t_2 \ t_3) \]
State Space Reduction

Example

\[ \sigma = (t_1 \ t_3 \ t_4 \ t_2 \ t_3) \]
State Space Reduction

Example

\[ \sigma = \left( t_1 \quad t_3 \quad t_4 \quad t_2 \quad t_3 \right) \]
Example

\[ \sigma = (t_1 \ t_3 \ t_4 \ t_2 \ t_3) \]

\[ \sigma(\tau) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_3 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_3 \xrightarrow{1.4} z \]
State Space Reduction

Example

\[ \sigma = (t_1 \ t_3 \ t_4 \ t_2 \ t_3) \]

\[ \sigma(\tau) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_3 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_3 \xrightarrow{1.4} z \]
State Space Reduction

Example

\[ \sigma = (t_1 t_3 t_4 t_2 t_3) \]

\[ m_\sigma = (1, 2, 2, 1, 1) \]
\[ \Sigma_\sigma = \begin{pmatrix} x_4 + x_5 \\ x_5 \\ x_5 \\ x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \end{pmatrix} \] and
State Space Reduction

Example (continuation)

\[ B_\sigma = \{ (x_0, x_1, x_2, x_3, x_4, x_5) \mid 
\begin{align*}
0 &\leq x_0, & x_0 &\leq 2, & x_0 + x_1 + x_2 &\leq 5, \\
0 &\leq x_1, & x_2 &\leq 2, & x_2 + x_3 &\leq 5, \\
1 &\leq x_2, & x_3 &\leq 2, & x_0 + x_1 + x_2 + x_3 &\leq 5, \\
1 &\leq x_3, & x_4 &\leq 2, & x_0 + x_1 + x_2 + x_3 + x_4 &\leq 5, \\
0 &\leq x_4, & x_5 &\leq 2, & x_0 + x_1 + x_2 + x_3 + x_4 + x_5 &\leq 5, \\
0 &\leq x_5, & x_0 + x_1 &\leq 5, & x_4 + x_5 &\leq 2 \} \]
State Space Reduction

Example (continuation)

The run $\sigma(\tau)$ with $\sigma(\tau) =$

\[
\begin{align*}
Z_0 &\xrightarrow{0.7} t_1 &\xrightarrow{0.0} t_3 &\xrightarrow{0.4} t_4 &\xrightarrow{1.2} t_2 &\xrightarrow{0.5} t_3 &\xrightarrow{1.4} (m_{\sigma}, \begin{pmatrix} 1.9 \\ 1.4 \\ 1.4 \\ 4.2 \\ \# \end{pmatrix})
\end{align*}
\]

is feasible.
State Space Reduction

Example ( continuation )

\[
\begin{pmatrix}
1.9 \\
1.4 \\
1.4 \\
1.4 \\
4.2 \\
\#
\end{pmatrix}
\]

\[
(m_\sigma, \begin{pmatrix}
1.9 \\
1.4 \\
1.4 \\
1.4 \\
4.2 \\
\#
\end{pmatrix})
\]

\[
z_0 \xrightarrow{\sigma(\beta)} z
\]
State Space Reduction

Example ( continuation )

\[
(m_\sigma, \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \\ 4.0 \end{pmatrix}) \xrightarrow{\sigma(\lambda)} [z]
\]

\[
(m_\sigma, \begin{pmatrix} 1.9 \\ 1.4 \\ 1.4 \\ 4.2 \end{pmatrix}) \xrightarrow{\sigma(\beta)} z
\]
## State Space Reduction

### Example (continuation)

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(m_{\sigma}, \begin{pmatrix} 1.0 \ 1.0 \ 1.0 \ 4.0 \ # \end{pmatrix})$</td>
<td>$\sigma(?)$</td>
<td>$z \rightarrow [z]$</td>
</tr>
<tr>
<td>$(m_{\sigma}, \begin{pmatrix} 1.9 \ 1.4 \ 1.4 \ # \end{pmatrix})$</td>
<td>$\sigma(\beta)$</td>
<td>$z \rightarrow z$</td>
</tr>
<tr>
<td>$(m_{\sigma}, \begin{pmatrix} 2.0 \ 2.0 \ 2.0 \ # \end{pmatrix})$</td>
<td>$\sigma(?)$</td>
<td>$z \rightarrow [z]$</td>
</tr>
</tbody>
</table>
Example ( continuation )

The runs
\[ \sigma(\tau_1^*) := z_0 \xrightarrow{1} t_1 \xrightarrow{0} t_3 \xrightarrow{1} t_4 \xrightarrow{1} t_2 \xrightarrow{0} t_3 \xrightarrow{1} [Z] \]

and
\[ \sigma(\tau_2^*) := z_0 \xrightarrow{1} t_1 \xrightarrow{0} t_3 \xrightarrow{0} t_4 \xrightarrow{2} t_2 \xrightarrow{0} t_3 \xrightarrow{2} [Z] \]

are feasible in \( Z \), too.
State Space Reduction

Example ( continuation )

The runs
\[ \sigma(\tau_1^*) := z_0 \xrightarrow{1} t_1 \xrightarrow{0} t_3 \xrightarrow{1} t_4 \xrightarrow{1} t_2 \xrightarrow{0} t_3 \xrightarrow{1} Z \]
\[ \sigma(\tau) = z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_3 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_3 \xrightarrow{1.4} Z \]
\[ \sigma(\tau_2^*) := z_0 \xrightarrow{1} t_1 \xrightarrow{0} t_3 \xrightarrow{0} t_4 \xrightarrow{2} t_2 \xrightarrow{0} t_3 \xrightarrow{2} Z \]

are feasible in \( Z \), too.
State Space Reduction

Theorem (1)

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

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

and all $\tau_i \in \mathbb{R}^+_0$.

Then, there exists a further feasible run $\sigma(\tau^*)$ of $\sigma$ with

$$z_0 \xrightarrow{\tau^*_0} t_0 \xrightarrow{\cdots} \tau^*_n \xrightarrow{t_n} z^*_n = (m^*_n, h^*_n).$$

such that
State Space Reduction

Theorem (1 – continuation)

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

\[ z_0 \xrightarrow{\tau_0^*} t_0 \xrightarrow{\tau_n^*} t_n \xrightarrow{\tau_i^*} \cdots \xrightarrow{\tau_n^*} z_n^* = (m_n^*, h_n^*) \]
State Space Reduction

Theorem (1 – continuation)

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

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

1. For each \( i, 0 \leq i \leq n \) the time \( \tau_i^* \) is a natural number.
State Space Reduction

Theorem (1 – continuation)

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

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

1. For each \( i, 0 \leq i \leq n \) the time \( \tau_i^* \) 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)

\[ z_0 \xrightarrow{\tau_0} t_0 \rightarrow \cdots \rightarrow z_n = (m_n, h_n), \tau_i \in \mathbb{R}_0^+ . \]

\[ z_0 \xrightarrow{\tau_0^*} t_0 \rightarrow \cdots \rightarrow z_n^* = (m_n^*, h_n^*), \tau_i^* \in \mathbb{N} . \]

1. For each \( i, 0 \leq i \leq n \) the time \( \tau_i^* \) 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 $\sigma$, i.e.

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

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

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

such that
Theorem (2 – continuation)

1. For each \( i, 0 \leq i \leq n \) the time \( \tau_i^* \) 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 \( \sum_{i=1}^{n} \tau_i^* = \lceil \sum_{i=1}^{n} \tau_i \rceil \)
3. For each transition \( t \in T \) holds:
   \( t \) is ready to fire in \( z_n \) if \( t \) is ready to fire in \( \lfloor z_n \rfloor \), too.
Dynamic programming

Where is the Dynamic Programming here?
Dynamic programming

Where is the Dynamic Programming here?

Let us consider the previous example again.
Input:

- The TPN $Z_2$, 

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
Input:

- The TPN $Z_2$,
- the transition sequence $\sigma = (t_1, t_3, t_4, t_2, t_3)$
- the six elapses of time $\hat{\beta}(x_0) = 0.7, \hat{\beta}(x_1) = 0.0, \hat{\beta}(x_2) = 0.4,$ $\hat{\beta}(x_3) = 1.2, \hat{\beta}(x_4) = 0.5, \hat{\beta}(x_5) = 1.4,$ 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 $Z_2$. 
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 $\mathbb{Z}_2$. 
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 $\mathbb{Z}_2$.

- 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 $\beta^*(x_0), \beta^*(x_1), \ldots, \beta^*(x_5)$ which are integers,
- $\sigma(\beta^*)$ is a feasible run in $\mathbb{Z}_2$.
- 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 $\sigma(\beta^*)$.

$\implies P^* : \text{Compute } \beta^*.$
Dynamic programming

\[ P^*(s) \]

Compute

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

Compute

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

\[ P^*(s) \]

Compute

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

<table>
<thead>
<tr>
<th></th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_{\sigma}(t_1)$</th>
<th>$\Sigma_{\sigma}(t_2)$</th>
<th>$\Sigma_{\sigma}(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>$\beta_0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
</tbody>
</table>

\[
\Sigma_{\sigma}(t_1) = x_4 + x_5, \\
\Sigma_{\sigma}(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
## Dynamic Programming

<table>
<thead>
<tr>
<th></th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\beta}$</td>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>( \hat{\beta} )</th>
<th>( x_0 )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( \Sigma_\sigma(t_1) )</th>
<th>( \Sigma_\sigma(t_2) )</th>
<th>( \Sigma_\sigma(t_5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\beta}_0 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>( \hat{\beta}_1 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]

\[
\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
Dynamic Programming

\[
\begin{array}{c|ccccccc}
\hat{\beta} & \beta_0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \\
\hline
\beta_0 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.4 \\
\beta_1 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.0 \\
\end{array}
\]

\[
\Sigma_\sigma(t_1) = \Sigma_\sigma(t_2) = \Sigma_\sigma(t_5) = x_5
\]

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
Dynamic Programming

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>( x_0 )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( \Sigma_\sigma(t_1) )</th>
<th>( \Sigma_\sigma(t_2) )</th>
<th>( \Sigma_\sigma(t_5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1</td>
<td>3.8</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5 \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
## Dynamic Programming

### I

<table>
<thead>
<tr>
<th>( \hat{\beta} )</th>
<th>( \beta_0 )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.7 0.0 0.4 1.2 0.5 1.4</td>
<td>1.9 1.4 4.2</td>
<td></td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.7 0.0 0.4 1.2 0.5 1</td>
<td>1.5 1.0 3.8</td>
<td></td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.7 0.0 0.4 1.2 0 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Equations

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]

\[
\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
## Dynamic Programming

\[
\begin{array}{c|ccccc|ccc}
\hat{\beta} & \beta_0 & \beta_1 & \beta_2 \\
\hline
\sum_{\sigma}(t_1) & 1.9 & 1.5 & 1.0 \\
\sum_{\sigma}(t_2) & 1.4 & 1.0 & 1.0 \\
\sum_{\sigma}(t_5) & 4.2 & 3.8 & 3.3 \\
\end{array}
\]

\[\sum_{\sigma}(t_1) = x_4 + x_5,\]
\[\sum_{\sigma}(t_5) = x_1 + x_2 + x_3 + x_4 + x_5\]

\[\sum_{\sigma}(t_2) = \sum_{\sigma}(t_3) = \sum_{\sigma}(t_4) = x_5\]
### Dynamic Programming

![Table](image)

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\textbf{l} & \textbf{x}_0 & \textbf{x}_1 & \textbf{x}_2 & \textbf{x}_3 & \textbf{x}_4 & \textbf{x}_5 & \sum_{\sigma}(t_1) & \sum_{\sigma}(t_2) & \sum_{\sigma}(t_5) \\
\hline
\hat{\beta} = \beta_0 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.4 & 1.9 & 1.4 & 4.2 \\
\beta_1 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1 & 1.5 & 1.0 & 3.8 \\
\beta_2 & 0.7 & 0.0 & 0.4 & 1.2 & 0 & 1 & 1.0 & & 3.3 \\
\beta_3 & 0.7 & 0.0 & 0.4 & & 0 & 1 & & & \\
\hline
\end{array}
\]

\[
\begin{align*}
\sum_{\sigma}(t_1) &= x_4 + x_5, \\
\sum_{\sigma}(t_5) &= x_1 + x_2 + x_3 + x_4 + x_5
\end{align*}
\]

\[
\sum_{\sigma}(t_2) = \sum_{\sigma}(t_3) = \sum_{\sigma}(t_4) = x_5
\]
Dynamic Programming

$$\hat{\beta} = \begin{bmatrix} \beta_0 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.4 \\ \beta_1 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1 \\ \beta_2 & 0.7 & 0.0 & 0.4 & 1.2 & 0 & 1 \\ \beta_3 & 0.7 & 0.0 & 0.4 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{\sigma}(t_1) \\ \Sigma_{\sigma}(t_2) \\ \Sigma_{\sigma}(t_3) \\ \Sigma_{\sigma}(t_4) \end{bmatrix} = \begin{bmatrix} 1.9 \\ 1.5 \\ 1.0 \\ 1.0 \end{bmatrix}$$

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

$$\Sigma_{\sigma}(t_5) = x_1 + x_2 + x_3 + x_4 + x_5$$
## Dynamic Programming

![Table Image]

\[
\begin{array}{|c|cccccc|ccc|}
\hline
I & x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & \Sigma_\sigma(t_1) & \Sigma_\sigma(t_2) & \Sigma_\sigma(t_5) \\
\hline
\hat{\beta} = \beta_0 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.4 & 1.9 & 1.4 & 4.2 \\
\beta_1 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1 & 1.5 & 1.0 & 3.8 \\
\beta_2 & 0.7 & 0.0 & 0.4 & 1.2 & 0 & 1 & 1.0 & 3.3 & \\
\beta_3 & 0.7 & 0.0 & 0.4 & 1 & 0 & 1 & 3.1 & & \\
\hline
\end{array}
\]

\[
\begin{align*}
\Sigma_\sigma(t_1) &= x_4 + x_5, \\
\Sigma_\sigma(t_5) &= x_1 + x_2 + x_3 + x_4 + x_5
\end{align*}
\]

\[
\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>( \hat{\beta} )</th>
<th>( x_0 )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( \Sigma_\sigma(t_1) )</th>
<th>( \Sigma_\sigma(t_2) )</th>
<th>( \Sigma_\sigma(t_5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>0.7</td>
<td>0.0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]

\[
\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>$\mathbf{I}$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\beta}$</td>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]

\[
\Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>$\hat{\beta}$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>3.7</td>
<td></td>
</tr>
</tbody>
</table>

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

\[
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>( \beta )</th>
<th>( x_0 )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( \Sigma_\sigma(t_1) )</th>
<th>( \Sigma_\sigma(t_2) )</th>
<th>( \Sigma_\sigma(t_5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_0 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>( \beta_1 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td>( \mathbf{1.0} )</td>
<td>3.3</td>
<td></td>
</tr>
<tr>
<td>( \beta_3 )</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>0.7</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

\[
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
Dynamic Programming

\[
\hat{\beta} = \begin{array}{c}
\beta_0 \\
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_5 \\
\end{array}
\begin{array}{cccccc}
0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.4 \\
0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1 \\
0.7 & 0.0 & 0.4 & 1.2 & 0 & 1 \\
0.7 & 0.0 & 0.4 & 1 & 0 & 1 \\
0.7 & 0 & 1 & 1 & 0 & 1 \\
0.7 & 0 & 1 & 1 & 0 & 1 \\
\end{array}
\begin{array}{c}
\Sigma_\sigma(t_1) \Sigma_\sigma(t_2) \Sigma_\sigma(t_5) \\
1.9 & 1.4 & 4.2 \\
1.5 & 1.0 & 3.8 \\
1.0 & & 3.3 \\
& & 3.1 \\
& & 3.7 \\
& & \\
\end{array}
\]

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5 \\
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5 
\]
# Dynamic Programming

<table>
<thead>
<tr>
<th>$\hat{\beta}$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td></td>
<td>3.3</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.1</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.7</td>
</tr>
<tr>
<td>$\beta_5$</td>
<td>0.7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.7</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>$I$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\beta}$</td>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_5$</td>
<td>0.7</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>$\beta^*$</td>
<td>$\beta_6$</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \Sigma_\sigma(t_1) = x_4 + x_5, \]
\[ \Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5 \]

\[ \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5 \]
### Dynamic Programming

<table>
<thead>
<tr>
<th>I</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\beta}$ = $\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
<td>4.2</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td><strong>1.0</strong></td>
<td></td>
<td>3.3</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td><strong>1</strong></td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.1</td>
</tr>
<tr>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0</td>
<td><strong>1</strong></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.7</td>
</tr>
<tr>
<td>$\beta_5$</td>
<td>0.7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>3.7</td>
</tr>
<tr>
<td>$\beta^*$ = $\beta_6$</td>
<td><strong>1</strong></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\Sigma_\sigma(t_1) = x_4 + x_5, \quad \Sigma_\sigma(t_2) = \Sigma_\sigma(t_3) = \Sigma_\sigma(t_4) = x_5
\]
\[
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
### Dynamic Programming

<table>
<thead>
<tr>
<th>$I$</th>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$\Sigma_\sigma(t_1)$</th>
<th>$\Sigma_\sigma(t_2)$</th>
<th>$\Sigma_\sigma(t_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\beta}$</td>
<td>$\beta_0$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1.4</td>
<td>1.9</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>$\beta_1$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>$\beta_2$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1.2</td>
<td>0</td>
<td>1</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>$\beta_3$</td>
<td>0.7</td>
<td>0.0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_4$</td>
<td>0.7</td>
<td>0.0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_5$</td>
<td>0.7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3.7</td>
<td></td>
</tr>
<tr>
<td>$\beta^*$</td>
<td>$\beta_6$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4.0</td>
<td></td>
</tr>
</tbody>
</table>

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

\[
\Sigma_\sigma(t_5) = x_1 + x_2 + x_3 + x_4 + x_5
\]
Dynamic Programming

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

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

The set of its critical states is the singleton \( S^0 = \{6\} \).

The set of its terminal states is the singleton \( S^t = \{0\} \).
Dynamic Programming

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

Louchka Popova-Zeugmann  TPN State Space Reduction Using DP and Time Paths
The state space (for P*) is the set \( S = \{0, 1, \ldots , 6\} \).

The set of its critical states is the singleton \( S^o = \{6\} \).

The set of its terminal states is the singleton \( S^t = \{0\} \).

The set of non-terminal states is \( S^{''} = S \setminus S^t = \{1, 2, \ldots , 6\} \).

The T-linker \( L_T \) has the form \( L_T(z(s^o)) = z^o = z(s^o) \).
Dynamic Programming

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

  \[ t(s) := s - 1, \quad s \in S''. \]
Dynamic Programming

- The *linker* $L$ is clearly given by

\[
\begin{align*}
z(s) & = L(s, \{(s', z(s')) | s' \in t(s)\}), \quad \forall s \in S'' \\
     & = L(s, z(t(s))) \\
     & = L(s, z(s-1)) := \beta_s
\end{align*}
\]
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 $\sigma(\beta)$ 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 $l_{\sigma(\beta^*)} = 5$
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 $\sigma$ in $Z$ can be realized with an "integer" run.
State Space Reduction

Corollary

- Each feasible t-sequence $\sigma$ in $Z$ can be realized with an "integer" run.
- Each reachable marking in $Z$ can be found using "integer" runs only.
Corollary

- Each feasible t-sequence $\sigma$ 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.
Corollary

- Each feasible $t$-sequence $\sigma$ 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

Reduced State Space
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.
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)

The reachability graph is a directed graph.
The reachability graph is a weighted directed graph.

Reduced State Space
Reachability Graph

Definition (informal)

The reachability graph is a directed graph.
The reachability graph is a weighted directed graph.
Reachability Graph

Definition (informal)

The reachability graph is a directed graph.

The reachability graph is a weighted directed graph.
Definition (informal)

⇒ The reachability graph is a directed graph.
Reachability Graph

Definition (informal)

⇒ The reachability graph is a weighted directed graph.
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 \( \mathbb{Z} \)).

**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

$$lp(z, z') := \begin{cases} \infty & \text{, if a cycle is reachable starting on } z \\ \max_{\sigma(\tau)} \sum_i \tau_i & \text{, else} \end{cases}$$
**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 \( RG(Z) \).
The State Space Reduction of a TPN is a nonoptimization truncated decision problem
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).
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).

$\Rightarrow$

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