Definitions

Petri Net

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

- $P$, $T$ and $F$ are finite sets,
- $P$ – set of places
- $T$ – set of transitions
- $F$ – set of edges (arcs)
- $F \subseteq (P \times T) \cup (T \times P)$ and $\text{dom}(F) \cup \text{cod}(F) = P \cup T$
- $V : F \rightarrow \mathbb{N}^+$ (weights of edges)
- $m_0 : P \rightarrow \mathbb{N}$ (initial marking)

Time Petri Net

Definition (Time Petri net)
The structure $Z = (P, T, F, V, m_0, I)$ is called a Time Petri net (TPN) iff:

- $S(Z) := (P, T, F, V, m_0)$ is a PN (skeleton of $Z$)
- $I : T \rightarrow \mathbb{Q}_+^d \times (\mathbb{Q}_+^d \cup \{\infty\})$ and
  $I_1(t) \leq I_2(t)$ for each $t \in T$, where $I(t) = (I_1(t), I_2(t))$.

Example

Example

$m_0 = (0, 1, 1)$

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

Example

Example

Outline

Definitions

Petri Net

Main Property

State Space Reduction

Dynamic Programming

Applications

Reachability Graph

Time Paths in bounded TPNs

Conclusion
**Definition (state)**

Let \( Z = (P, T, F, V, m_0, I) \) be a TPN and \( h : T \rightarrow \mathbb{R}_+^* \cup \{\#\} \).

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

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

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

**Example**

1. \( z_0 \rightarrow z_1 \), where \( z_0 = (m_0, h_0) \) and \( z_1 = (m_1, h_1) \).

2. \( z_1 \rightarrow z_2 \), where \( z_2 = (m_2, h_2) \).

**Definition (state changing)**

Let \( Z = (P, T, F, V, m_0, I) \) be a TPN, \( z = (m, h) \), \( z' = (m', h') \) be two states. Then \( z \) changes into \( z' \) by:

- firing a transition
- time elapsing

\( z = (m, h) \) changes into \( z' = (m', h') \) by:

- firing a transition
- time elapsing
Transition sequences, Runs

Definition

- **transition sequence**: \( \sigma = (t_1, \ldots, t_n) \)
- **run**: \( \sigma(\tau) = (\tau_0, t_1, \tau_1, \ldots, \tau_n, t_n) \)
- **feasible run**: \( z_0 \xrightarrow{\tau_0} z^*_0 \xrightarrow{t_1} z_1 \xrightarrow{\tau_1} \cdots \xrightarrow{t_n} z_n \xrightarrow{\tau_n} z^*_n \)
- **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 ex. a feasible run \( \sigma(\tau) \) and \( z_0 \xrightarrow{\tau_0} z \)
- **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: \( \text{StSp}(Z) \)).

Qualitative Properties

- **static properties**: being/having
  - homogeneous
  - ordinary
  - free choice
  - extended simple
  - conservative
  - deadlocks, etc.
  - decidable without knowledge of the state space!
- **dynamic properties**: being/having
  - bounded
  - live
  - reachable marking/state
  - place- or transitions invariants, etc.
  - decidable, if at all (TPN is equiv. to TMI),
    with implicit/explicit knowledge of the state space

Quantitative Properties

- each time proposition as having/computing
  - (min-/max) time length of path
  - path between two states/markings with min-/max time length
  - set of all reachable markings within a period
  - looking for efts and lfts leading to certain qualitative/quantitative properties etc.
  - decidable, if at all, 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 \( \sigma = (t_1, \ldots, t_n) \) be a transition sequence in \( Z \).

\[ \delta(\sigma) = [m_0, \Sigma, B] \]

is the parametric description of \( \sigma \), if

- \( m_0 \xrightarrow{t_i} m_0 \)
- \( \Sigma(t) \) is a parametrical \( t \)-marking
- \( B(t) \) is a set of conditions (a system of inequalities)

Obviously

- \( z_0 \xrightarrow{t_i} (m_0, \Sigma, B) =: z_0 \)
- \( \text{StSp}(Z) = \bigcup_{\sigma} z_\sigma \).

**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 \( \sigma(t) \) as an execution of \( \sigma \), i.e.

\[ z_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} z_n = (m_n, h_n), \]

and all \( t_i \in R_0^+ \).

Then, there exists a further feasible run \( \sigma(t_\ast) \) of \( \sigma \) with

\[ z_0 \xrightarrow{t_{i_1}} \cdots \xrightarrow{t_{i_k}} \ast \xrightarrow{t_{i_k+1}} \cdots \xrightarrow{t_n} z_n = (m_n, h_n), \]

such that

**Example**

![Example Diagram](image)

\[ \sigma = (e) \implies \delta(\sigma) = C_\sigma = \{(0, 1, 1, 3, 3, 3, 3)| 0 \leq x_i \leq 3 \} \]

**State Space Reduction**

**Theorem (1 – continuation)**

1. For each \( i, 0 \leq i \leq n \) the time \( t_i^\ast \) is a natural number.
2. For each enabled transition \( t \) at marking \( m_n = m_n^* \) it holds:
   
   \[ 2.1 \ h_0(t) = \lfloor h_0(t) \rfloor \]
   
   \[ 2.2 \ \sum_{i=1}^{k} t_i^\ast = \lfloor \sum_{i=1}^{k} t_i \rfloor \]
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.

**State Space Reduction**

**Theorem (2 – similar to 1)**

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

\[ z_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} z_n = (m_n, h_n), \]

and all \( t_i \in R_0^+ \).

Then, there exists a further feasible run \( \sigma(t_\ast) \) of \( \sigma \) with

\[ z_0 \xrightarrow{t_{i_1}} \cdots \xrightarrow{t_{i_k}} \ast \xrightarrow{t_{i_k+1}} \cdots \xrightarrow{t_n} z_n = (m_n, h_n), \]

such that

**Theorem (2 – continuation)**

1. For each \( i, 0 \leq i \leq n \) the time \( t_i^\ast \) is a natural number.
2. For each enabled transition \( t \) at marking \( m_n = m_n^* \) it holds:
   
   \[ 2.1 \ h_0(t)^\ast = \lfloor h_0(t) \rfloor \]
   
   \[ 2.2 \ \sum_{i=1}^{k} t_i^\ast = \lfloor \sum_{i=1}^{k} t_i \rfloor \]
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.
The theorem 1 solves the following problem:

**Input:** a TPN, a transition sequence \( \sigma = (t_1, \ldots, t_n) \) and a sequence of \((n+1)\) real numbers, \((\beta(x_0), \beta(x_1), \cdots, \beta(x_n))\) subject to a certain finite set \(VC\) of conditions (inequalities).

**Output:** a sequence of \((n+1)\) integers, \((\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_n))\) subject to \(VC\).

The solving of the output is the problem \(P^*\):

**Problem \(P^*\):** Compute a sequence of \((n+1)\) integers, \((\beta^*(x_0), \beta^*(x_1), \cdots, \beta^*(x_n))\) subject to \(VC^*\).

The solution strategy for the problem \(P^*\) is a typical dynamic programming’s one.

\(^1\)\(VC^*\) is a certain finite superset of the set \(VC\)
State Space Reduction

Example (continuation)

\[ B_x = \begin{cases} 
0 & \leq x_0, x_0 \leq 2, \quad x_0 + x_1 + x_2 \leq 5 \\
0 & \leq x_1, x_2 \leq 2, \quad x_2 + x_3 \leq 5 \\
1 & \leq x_2, x_3 \leq 2, \quad x_0 + x_1 + x_2 + x_3 \leq 5 \\
1 & \leq x_3, x_4 \leq 2, \quad x_0 + x_1 + x_2 + x_3 + x_4 \leq 5 \\
0 & \leq x_4, x_5 \leq 2, \quad x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \leq 5 \\
0 & \leq x_5, x_0 + x_5 \leq 5 
\end{cases} \]

State Space Reduction

Example (continuation)

The run \( \sigma(t) \) with 
\[
\sigma(t) = \begin{pmatrix} 0.7 & 1 & 0.0 & 0.4 & 1.2 & 0.4 & 1 & 0.5 & 1.4 \end{pmatrix} \]
is feasible.

State Space Reduction

Example (continuation)

Hence, the runs
\[
\sigma(t_1) := z_0 1 \to t_1 0 \to t_1 1 \to t_1 1 \to t_1 0 \to t_1 1 \to [z] \\
\sigma(t) := z_0 0.7 \to t_1 0.0 \to t_1 0.4 \to t_1 1.2 \to t_1 0.5 \to t_1 1.4 \to [z] \\
\sigma(t_2) := z_0 1 \to t_1 0 \to t_1 0 \to t_1 2 \to t_1 0 \to t_1 2 \to [z] 
\]
are feasible in \( Z \), too.
Where is the Dynamic Programming here?

Let us consider the tableau 1 again!

Output:

- Six elapses of time \( \beta^*(x_0), \beta^*(x_1), \ldots, \beta^*(x_3) \) which are integers.
- \( \sigma(\beta^*) \) is a feasible run in \( Z_2 \).
- The set of transitions which are ready to fire after \( \sigma(\beta^*) \) is the same as the set of transitions which are ready to fire after \( \sigma(\hat{\beta}) \).

\[ \text{Input:} \]

- The TPN \( Z_2 \).
- the transition sequence \( \sigma = (t_1, t_3, t_4, t_2, t_3) \).
- the six \( (6 = n + 1, \text{i.e. } n = 5) \) 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
- 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 \( Z_2 \).

\[ \begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{I} & x_0 & x_1 & x_2 & x_3 & x_4 & \Sigma_x(t_1) \Sigma_x(t_2) \Sigma_x(t_3) \\
\hline
\beta_1 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1.9 & 1.4 & 4.2 \\
\beta_2 & 0.7 & 0.0 & 0.4 & 1.2 & 0.5 & 1 & 1 & 1 \\
\beta_2 & 0.7 & 0.0 & 0.4 & 1.2 & 0 & 0 & 1 & 1 \\
\beta_2 & 0.7 & 0.0 & 0.4 & 1 & 0 & 1 & 1 & 1 \\
\hline
\end{array} \]
Dynamic Programming

- The set of its critical states is the singleton $S^0 = \{5\}$.
- The set of its terminal states is the singleton $S^t = \{0\}$.
- The set of non-terminal states is $S'' = S \setminus S^t = \{1, 2, \ldots, 5\}$.
- The T-linker $L_T$ has the form $L_T(z(s^0)) = z^0 = z(s^0)$.
- The transition function $t$ is defined as $t(s) := s - 1$, $s \in S''$.

State Space Reduction

- 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.

Definition

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

Example (State Space Reduction)

The time length of the run $\sigma(\beta)$ is $l_{\sigma(\beta)} = \beta(x_0) + \beta(x_1) + \beta(x_2) + \beta(x_3) + \beta(x_4) = 4.2$

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

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**)
1. $z_0 \in RG(Z)$

**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 $z' \in RG(Z)$ stop

2. if $z \xrightarrow{t_1} z'$ possible in $Z$ then $z' \in RG(Z)$ stop

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

**Example (The FTPN $Z_3$ and its reachability graph(s))**

Let $Z = (P, T, F, V, I, m_0)$ 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 $O(|V| \cdot |E|)$ and $RG(Z) = (V, E)$).

**Definition**

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

$$lp(z, z') := \begin{cases} 
\infty & \text{if a cycle is reachable starting on } z \text{ before reaching } z' \\
\max_{i} \sum_{\tau_i} \tau_i & \text{if } z \xrightarrow{\sigma(t)} z' 
\end{cases}$$
Result:

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 (poly. running time), or by computing all strongly connected components of \( RG(Z) \). (linear running time)

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

Thank you!