

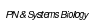
Definitions
Main Property
Approximation
Conclusion

Outline

- Definitions
 - Petri Net
 - Time Petri Net
- Main Property
 - State Space Reduction
- Applications
 - Reachability Graph
 - T-invariants
 - Time Paths in unbounded TPNs
 - Time Paths in bounded TPNs
 - Time PN and Timed PN
- Conclusion

A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other navigation functions.

PN & Systems Biology

$$2 \text{ NAD}^+ + 2 \text{ H}_2\text{O} \rightarrow 2 \text{ NADH} + 2 \text{ H}^+ + \text{O}_2$$
*PN & Systems Biology*

```

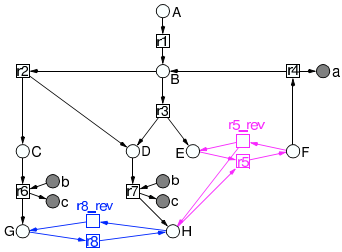
graph TD
    A((A)) -- black --> B((B))
    B -- blue --> C((C))
    B -- pink --> D((D))
    D -- pink --> E((E))
    C -- blue --> D
    B -- blue --> D

```

PN & Systems Biology

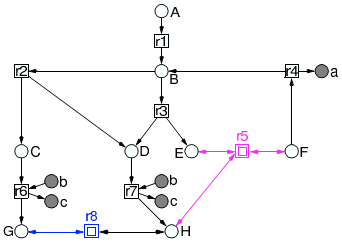
September 2004

- r1: A -> B
- r2: B -> C + D
- r3: B -> D + E
- r4: F -> B + a
- r5: E + H <-> F
- r6: C + b -> G + c
- r7: D + b -> H + c
- r8: H <-> G



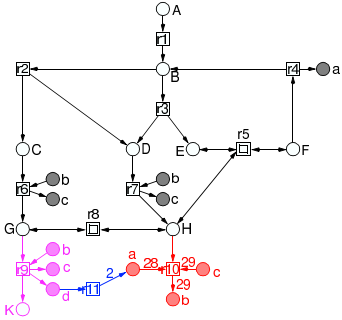
-> reversible reactions

- r1: A -> B
- r2: B -> C + D
- r3: B -> D + E
- r4: F -> B + a
- r5: E + H <-> F
- r6: C + b -> G + c
- r7: D + b -> H + c
- r8: H <-> G

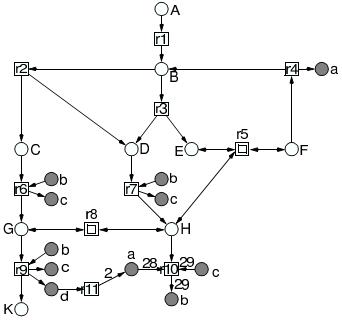


-> reversible reactions
- hierarchical nodes

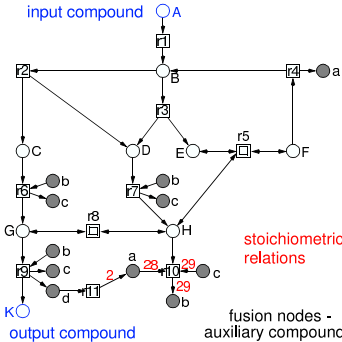
- r1: A -> B
- r2: B -> C + D
- r3: B -> D + E
- r4: F -> B + a
- r5: E + H <-> F
- r6: C + b -> G + c
- r7: D + b -> H + c
- r8: H <-> G
- r9: G + b -> K + c + d
- r10: H + 28a + 29c -> 29b
- r11: d -> 2a



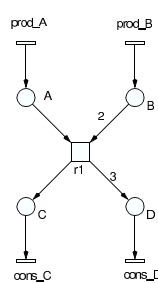
- r1: A -> B
- r2: B -> C + D
- r3: B -> D + E
- r4: F -> B + a
- r5: E + H <-> F
- r6: C + b -> G + c
- r7: D + b -> H + c
- r8: H <-> G
- r9: G + b -> K + c + d
- r10: H + 28a + 29c -> 29b
- r11: d -> 2a



- r1: A -> B
- r2: B -> C + D
- r3: B -> D + E
- r4: F -> B + a
- r5: E + H <-> F
- r6: C + b -> G + c
- r7: D + b -> H + c
- r8: H <-> G
- r9: G + b -> K + c + d
- r10: H + 28a + 29c -> 29b
- r11: d -> 2a



stoichiometric relations

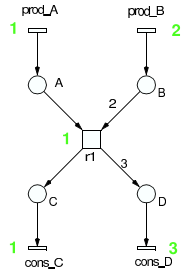


-> properties as time-less net

INA	ORD	HOM	NEB	PUR	CSV	SCF	CON	SC	Pt0	tF0	Pp0	Pf0	MG	SM	PC	EEC	ES
N	Y	N	Y	N	Y	Y	N	Y	N	Y	N	N	Y	N	Y	Y	Y
CP1	CT1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1	R1
N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y

TRANSFORMATION, Ex1

PN & Systems Biology



T-INVARIANTE

-> properties as time-less net

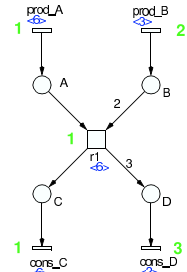
```
INA
ORD HOM NEM PUR CSV SCF CON SC Ff0 tF0 Fp0 pF0 MG SM FC EFC ES
N Y N Y N Y Y N Y N Y N N Y N Y Y Y
CPI CTI B SB REV DST BSt DTZ DCF L LV L&S
N Y N N Y N ? N Y Y Y N
```

popova@informatik.uni-bonn.de, monika.heiner@informatik.uni-cottbus.de, insuoch@ifh-bonn.de

September 2004

TRANSFORMATION, Ex1

PN & Systems Biology



T-INVARIANTE

-> properties as time net

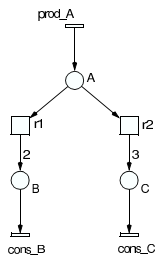
```
INA
ORD HOM NEM PUR CSV SCF CON SC Ff0 tF0 Fp0 pF0 MG SM FC EFC ES
N Y N Y N Y Y N Y N Y N N Y N Y Y Y
CPI CTI B SB REV DST BSt DTZ DCF L LV L&S
N Y N N Y N ? N Y Y Y N
```

popova@informatik.uni-bonn.de, monika.heiner@informatik.uni-cottbus.de, insuoch@ifh-bonn.de

September 2004

TRANSFORMATION, Ex2

PN & Systems Biology



-> properties as time-less net

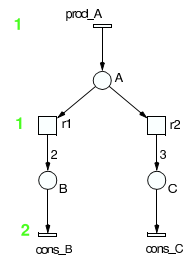
```
INA
ORD HOM NEM PUR CSV SCF CON SC Ff0 tF0 Fp0 pF0 MG SM FC EFC ES
N Y N Y N Y Y N Y N Y N N Y N Y Y Y
CPI CTI B SB REV DST BSt DTZ DCF L LV L&S
N Y N N Y N ? N N Y Y Y N
```

popova@informatik.uni-bonn.de, monika.heiner@informatik.uni-cottbus.de, insuoch@ifh-bonn.de

September 2004

TRANSFORMATION, Ex2

PN & Systems Biology



T-INVARIANTE1
T-INVARIANTE2

-> properties as time-less net

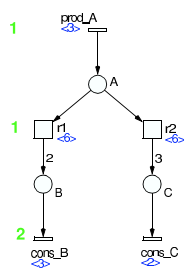
```
INA
ORD HOM NEM PUR CSV SCF CON SC Ff0 tF0 Fp0 pF0 MG SM FC EFC ES
N Y N Y N Y Y N Y N Y N N Y N Y Y Y
CPI CTI B SB REV DST BSt DTZ DCF L LV L&S
N Y N N Y N ? N N Y Y Y N
```

popova@informatik.uni-bonn.de, monika.heiner@informatik.uni-cottbus.de, insuoch@ifh-bonn.de

September 2004

TRANSFORMATION, Ex2

PN & Systems Biology



T-INVARIANTE1
T-INVARIANTE2

-> properties as time net

```
INA
ORD HOM NEM PUR CSV SCF CON SC Ff0 tF0 Fp0 pF0 MG SM FC EFC ES
N Y N Y N Y Y N Y N Y N N Y N Y Y Y
CPI CTI B SB REV DST BSt DTZ DCF L LV L&S
N Y N N Y N ? N N Y Y Y N
```

popova@informatik.uni-bonn.de, monika.heiner@informatik.uni-cottbus.de, insuoch@ifh-bonn.de

September 2004

Definition

Place, Transition, Edge, Label

Petri Net

Place, Transition, Edge, Label

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
- V - set of vertices
- $P \cap T = \emptyset, P \cup T \neq \emptyset$.
- 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)

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

Petri Net

Example

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

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

Firing transition

Definition

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

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

firing transition

Example

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

firing transition

Example

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

firing transition

Example

Definitions
Mark Property
Application
Conclusion

Print Net
Time Petri Net

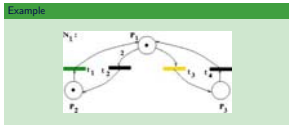
firing transition

Example

Definitions
What is a firing transition
Application
Conclusion

Print Net
Time Petri Net

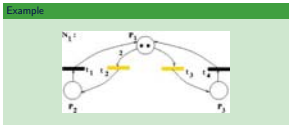
firing transition



Definitions
What is a firing transition
Application
Conclusion

Print Net
Time Petri Net

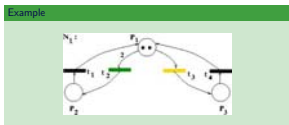
firing transition



Definitions
What is a firing transition
Application
Conclusion

Print Net
Time Petri Net

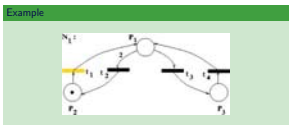
firing transition



Definitions
What is a firing transition
Application
Conclusion

Print Net
Time Petri Net

firing transition



Definitions
What is a Time Petri Net
Application
Conclusion

Print Net
Time Petri Net

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)
► $I: T \rightarrow Q_0^+ \times (Q_0^+ \cup \{\infty\})$ and
► $h(t) \leq b(t)$ for each $t \in T$, where $I(t) = (h(t), b(t))$.

Definitions
What is a Time Petri Net
Application
Conclusion

Print Net
Time Petri Net

Time Petri Net

Definition (FTPN)
A TPN is called finite Time Petri net (FTPN) iff
 $I: T \rightarrow Q_0^+ \times Q_0^+$.

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

Time Petri Net

Example

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

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

state

Definition (state)

Let $Z = (P, T, F, V, m_0, l)$ be a TPN and $h : T \rightarrow \mathbb{R}_0^+ \cup \{\#\}$. $z = (m, h)$ is called a **state** in Z iff:

- m is a p -marking in Z , e.g. m is a marking in $S(Z)$.
- h is a t -marking in Z , e.g.
 - $\forall t \in T \wedge t^- \leq m \rightarrow (h(t) \in \mathbb{R}_0^+ \wedge h(t) \leq l(t))$,
 - and
 - $\forall t \in T \wedge t^- \leq m \rightarrow h(t) = \#$.

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

Definition (state changing)

Let $Z = (P, T, F, V, m_0, l)$ be a TPN, \hat{t} be a transition in T and $z = (m, h)$, $z' = (m', h')$ be two states. Then

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

- $t^- \leq m$ and
- $eff(\hat{t}) \leq h(\hat{t})$.

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

state changing

Definition (state changing)

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

- \hat{t} is ready to fire in the state $z = (m, h)$
- $m' = m - \Delta \hat{t}$ and
- $\forall t \in T \xrightarrow{\hat{t}} \begin{cases} \# & \text{iff } t^- \not\leq m' \\ h(t) & \text{iff } t^- \leq m \wedge t^- \leq m' \wedge Ft \cap Ft = \emptyset \\ 0 & \text{otherwise} \end{cases}$

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

state changing

Definition (state changing)

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

- $m' = m$ and
- $\forall t \in T \wedge h(t) \neq \# \rightarrow h(t) + r \leq l(t)$ i.e. the time elapsing r is possible, and
- $\forall t \in T \rightarrow h'(t) = \begin{cases} h(t) + r & \text{iff } t^- \leq m' \\ \# & \text{iff } t^- \not\leq m' \end{cases}$.

Definitions
Mark Property
Boundedness
Consistency

Print this
Time Petri Net

Time Petri Net

Example

$(m_0, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix})$

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

$$\begin{aligned}
 & z_0 \xrightarrow{1.3} (m_1; \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}) \xrightarrow{1.3} (m_2; \begin{pmatrix} 1.3 \\ 2 \\ 1.3 \end{pmatrix})
 \end{aligned}$$

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

$$\begin{aligned}
 & z_0 \xrightarrow{1.3} (m_1; \begin{pmatrix} 1.3 \\ 2 \\ 1.3 \end{pmatrix}) \xrightarrow{1.0} (m_2; \begin{pmatrix} 2.3 \\ 2 \\ 2.3 \end{pmatrix})
 \end{aligned}$$

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

$$\begin{aligned}
 & z_0 \xrightarrow{1.3, 1.0} (m_2; \begin{pmatrix} 2.3 \\ 2 \\ 2.3 \end{pmatrix}) \xrightarrow{1.5}
 \end{aligned}$$

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

$$\begin{aligned}
 & z_0 \xrightarrow{1.3, 1.0} (m_2; \begin{pmatrix} 2.3 \\ 2 \\ 2.3 \end{pmatrix}) \xrightarrow{1.5} (m_3; \begin{pmatrix} 2.3 \\ 0.0 \\ 2.3 \end{pmatrix})
 \end{aligned}$$

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

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

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Application
Conclusion

Print this
Time Petri Net

Time Petri Net

Example

$$\begin{aligned}
 & z_0 \xrightarrow{1.3, 1.0, 1.5} (m_4; \begin{pmatrix} 4.3 \\ 2.0 \\ 2.3 \end{pmatrix}) \xrightarrow{1.5}
 \end{aligned}$$

Lauchlin Paterson Ziegmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

Time Petri Net

Example

$$z_0 \xrightarrow{1.3, 1.0, t_1, 2.0} (m_4, \begin{pmatrix} 4.3 \\ 2 \\ 2 \end{pmatrix}) \xrightarrow{t_2} (m_5, \begin{pmatrix} 2 \\ 0.0 \\ 2.0 \\ 2 \end{pmatrix})$$

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

Time Petri Net

Example

$$z_0 \xrightarrow{1.3, 1.0, t_1, 2.0, t_1} (m_5, \begin{pmatrix} 2 \\ 0.0 \\ 2.0 \\ 2 \end{pmatrix}) \xrightarrow{t_2}$$

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

Time Petri Net

Example

$$z_0 \xrightarrow{1.3, 1.0, t_1, 2.0, t_1, t_2} (m_6, \begin{pmatrix} 0.0 \\ 2 \\ 2 \\ 2 \end{pmatrix})$$

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

Transition sequences, Runs

Definition

- ▶ **transition sequence**: $\sigma = (t_1, \dots, t_n)$
- ▶ **run**: $\sigma(\tau) = (t_1, t_2, \dots, t_{n-1}, t_n)$
- ▶ **feasible run**: $z_0 \xrightarrow{t_1} z_1 \xrightarrow{t_2} z_2 \xrightarrow{t_3} \dots \xrightarrow{t_n} z_n$
- ▶ **feasible transition sequence**: σ is feasible if there ex. a feasible run $\sigma(\tau)$

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

Reachable state, Reachable marking, State space

Definition

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

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Marking Property
Reachability
Consistency
Time Petri Net

Time Petri Net

State class

Definition (state class)

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

Basis: $C_\sigma := \{z \mid \exists \tau (\tau \in R_0^+ \wedge z_0 \xrightarrow{\tau} z)\}$

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

$$C_{\sigma t} := \{z \mid \exists z_1 \exists z_2 \exists \tau (z_1 \in C_\sigma \wedge \tau \in R_0^+ \wedge z_1 \xrightarrow{t} z_2 \xrightarrow{\tau} z)\}$$

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

LucBrie Petri-Netzmann Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

Properties

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

Lecture Page Zoom
Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

Parametric Description of the State Space

Let $Z = [P, T, F, V, m_0, I]$ be a TPN and $\sigma = (t_1, \dots, t_n)$ be a transition sequence in Z .

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

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

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

Lecture Page Zoom
Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

State Space Reduction

Example

$\sigma = (e) \implies$

$$\delta(\sigma) = C_\sigma = \left(\underbrace{((0, 1, 1))}_{m_\sigma}, \underbrace{\{x_1, x_2, x_3\}}_{\Sigma_\sigma}, \underbrace{\{0 \leq x_1 \leq 3\}}_{B_\sigma} \right)$$

Lecture Page Zoom
Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

State Space Reduction

Example

Lecture Page Zoom
Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

State Space Reduction

Example

Lecture Page Zoom
Quantitative Analysis of TPNs

Definitions
Main Property
Verification
Quantification

Print
New
Time
Print
New

State Space Reduction

Example

Lecture Page Zoom
Quantitative Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

Example

$$\sigma = (t_1, t_2) \implies \delta(\sigma) = C_{\delta_0} t_1 =$$

$$\left(\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} x_1 + x_0 + x_3 \\ x_2 \\ x_3 \end{pmatrix} \right) \mid \begin{matrix} 2 \leq x_1 \leq 5, & x_1 + x_0 \leq 5 \\ 2 \leq x_2 \leq 4, & x_1 + x_2 \leq 5 \\ 0 \leq x_3 \leq 3 \end{matrix} \right).$$

LaTeX Page Zoom

Quantification Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

State Space Reduction

Theorem (1)

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

$$z_0 \xrightarrow{t_1} z_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} z_n = (m_n, h_n),$$

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

$$z_0 \xrightarrow{\tau_1^*} z_1 \xrightarrow{\tau_2^*} \dots \xrightarrow{\tau_n^*} z_n^* = (m_n^*, h_n^*),$$

such that

LaTeX Page Zoom

Quantification Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

State Space Reduction

Theorem (1 – continuation)

$$z_0 \xrightarrow{t_1} z_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} z_n = (m_n, h_n), \tau_i \in \mathbb{R}_0^+.$$

$$z_0 \xrightarrow{\tau_1^*} z_1 \xrightarrow{\tau_2^*} \dots \xrightarrow{\tau_n^*} z_n^* = (m_n^*, h_n^*), \tau_i^* \in \mathbb{N}.$$

- For each $i, 0 \leq i \leq n$ holds: $\tau_i^* \in \mathbb{N}$.
- For each enabled transition t at marking $m_n (= m_n^*)$ it holds:
 - $h_n(t)^* = [h_n(t)]$.
 - $\sum_{i=1}^n \tau_i^* = \lceil \sum_{i=1}^n \tau_i \rceil$.

LaTeX Page Zoom

Quantification Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

State Space Reduction

Theorem (2 – similar to 1)

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

$$z_0 \xrightarrow{t_1} z_1 \xrightarrow{t_2} \dots \xrightarrow{t_n} z_n = (m_n, h_n),$$

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

$$z_0 \xrightarrow{\tau_1^*} z_1 \xrightarrow{\tau_2^*} \dots \xrightarrow{\tau_n^*} z_n^* = (m_n^*, h_n^*),$$

such that

LaTeX Page Zoom

Quantification Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

State Space Reduction

Theorem (2 – continuation)

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

LaTeX Page Zoom

Quantification Analysis of TPNs

Navigation

Main Property

Quantification

Conclusion

State Space Reduction

State Space Reduction

Example

LaTeX Page Zoom

Quantification Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

- Quantitative
- Main Property
- Approximation
- Classification

State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_2 t_5)$

$\sigma(r) := z_0 \xrightarrow{0.7} t_1 \xrightarrow{0.0} t_2 \xrightarrow{0.4} t_4 \xrightarrow{1.2} t_2 \xrightarrow{0.5} t_5 \xrightarrow{1.4} z$

Lucia Pérez Zengaro

Quantitative Analysis of TPNs

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example

$\sigma = (t_1 t_2 t_4 t_5)$
 $m_\sigma = (1, 2, 2, 1, 1)$

Lucia de Paiva - University of Porto

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example (continuation)

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

Lucia de Paiva - University of Porto

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example (continuation)

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

Lucia de Paiva - University of Porto

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example (continuation)

The run $\sigma(r)$ with

$\sigma(r) := s_0 \xrightarrow{0.7, t_1} s_1 \xrightarrow{0.0, t_2} s_2 \xrightarrow{0.4, t_4} s_3 \xrightarrow{1.2, t_5} s_4 \xrightarrow{0.5, t_1} s_5 \xrightarrow{1.4, t_2} s_6$

is feasible.

Lucia de Paiva - University of Porto

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example (continuation)

The run $\sigma(r)$ with

$\sigma(r) := s_0 \xrightarrow{0.7, t_1} s_1 \xrightarrow{0.0, t_2} s_2 \xrightarrow{0.4, t_4} s_3 \xrightarrow{1.2, t_5} s_4 \xrightarrow{0.5, t_1} s_5 \xrightarrow{1.4, t_2} s_6$

is feasible.

Lucia de Paiva - University of Porto

Quantitative
Main Property
Verification
State Space Reduction

State Space Reduction

Example (continuation)

The run $\sigma(r)$ with

$s_0 \xrightarrow{0.7, t_1} s_1 \xrightarrow{0.0, t_2} s_2 \xrightarrow{0.4, t_4} s_3 \xrightarrow{1.2, t_5} s_4 \xrightarrow{0.5, t_1} s_5 \xrightarrow{1.4, t_2} s_6 \xrightarrow{1.9, t_1} s_7 \xrightarrow{1.4, t_2} s_8 \xrightarrow{1.4, t_2} s_9 \xrightarrow{4.2, t_2} s_{10}$

is feasible.

Lucia de Paiva - University of Porto

Introduction
Mark Property
 Countability
 Continuity

State Space Reduction

State Space Reduction

Example (continuation)

	x_1	x_2	x_3	x_4	$\Sigma_1(x_1)$	$\Sigma_2(x_2)$	$\Sigma_3(x_3)$
$j = j_0$	0.0	0.4	1.2	0.5	1.9	1.4	4.2
$j = j_1$	0.7	0.0	0.4	1.2	2.5	2.0	4.8
$j = j_2$	0.7	0.0	1.2	0	1	2.0	4.3
$j = j_3$	0.7	0.0	0.4	2	0	1	5.1
$j = j_4$	0.7	0.0	0	1	1	0	4.7
$j = j_5$	0.7	0	1	1	0	1	4.7
$j = j_6$	1	0	1	1	0	1	5.0

Conclusion
 Countability
 Continuity

State Space Reduction

Introduction
State Property
Reachability
Computation

State Space Reduction


State Space Reduction

Corollary

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

Linearization Property Equivalence

Quantitative Analysis of TPNs


 Definition
 Application
 Conclusion

Reachability Graph

When Paths are unbounded (TMs)
 When Paths are bounded (FSMs)
 When PNs and Petri Nets

Reachability Graph

Definition

Basis


$\mathcal{R}_0 \in RG(Z)$

Step

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

1. for $i=1$ to n do
- if $z \xrightarrow{a_i}$, z' possible in Z then $z' \in RG(Z)$ end
2. if $z \xrightarrow{a_i}$, z' possible in Z then $z' \in RG(Z)$

⇒ The reachability graph is a weighted directed graph.


 Lecture 3: Reachability
 Reachability Graphs

Questions? Ask at the end of the lecture
 Questions? Ask at the end of the lecture

Inductive Step:
Make Progress:
Variant:

Reachability Graph
 Transitions:
 Time Paths in subselected TTMs:
 Time Paths in Selected TTMs:
 Time Pts and Guard Pts

A TPN and its full Reachability Graph

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

<ul style="list-style-type: none"> Inductive Reachability Applications Conclusion 	<p>Reachability Graph</p> <ul style="list-style-type: none"> Reachability Time Points in advanced TSPs Time Points in bounded TSPs Time PNs and Event PNs
---	--

Example (The reduced reachability graphs $RG^{(1)}(Z)$ and $RG(Z)$)

Definition:
 In the Program
 Application
 Combination

Reachability Graph
 T invariant.
 Time Point is referenced TPN.
 Time Point is referenced TPN.
 Time PPN and Time PPN

Example [The reachability graph $RG(Z_1)$]

The diagram shows a reachability graph with two states, s_1 and s_2 . State s_1 is represented by a circle with a double border and contains the expression $x = 0 \wedge y = 0$. State s_2 is represented by a circle with a double border and contains the expression $x = 1 \wedge y = 0$. There is a transition from s_1 to s_2 labeled $x := 1$. There is a transition from s_2 back to s_1 labeled $x := 0$. The graph is set against a light green background.

Definition	Reachability Graph
Example	T-Invariants
Application	From Petri to generalized TPNs
Conclusion	From TPN to General TPN From TPN and General TPN

Definition:
In the Program
Application
Context

Reachability graph
T invariants
Two TPNs is unconfusable TPNs
Two TPNs are unconfusable TPNs
Two TPN and Trans TPN

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

⇒

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

<ul style="list-style-type: none"> Optimization Combinatorics Applications Complexity 	<ul style="list-style-type: none"> Branch-and-Bound Branch-and-Cut Time Limits in unbounded TSPs Time Limits in Integer TSPs Time PDB and Time PDB
---	---

Let $Z = (P, T, F, V, I, m_0)$ be a TPN.

Then the following problems can be decided/computed without knowledge of its RC:

Result 1:

Input: The time function I is fixed,
 σ is an arbitrary transition sequence.

Output: Feasibility of σ in Z ?

Solution: Solve a linear system of inequalities in \mathbb{R}_0^+ .

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

Result 2:

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

Let $Z = (P, T, F, V, I, m_0)$ be a TPN. Then the following problems can be decided/computed without knowledge of its RG:

- Result 3:**
- Input:** The time function I is fixed, σ is an arbitrary transition sequence.
- Output:** \min / \max length of σ .
- Solution:** Solve a linear program in \mathbb{R}_+^n . (Actually, the solution is in \mathbb{N}_+^n .)

Let $Z = (P, T, F, V, I, m_0)$ be a TPN.
Then the following problems can be decided/computed without knowledge of its RG:

Result 4:

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

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

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

Result 5:

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

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

Solution: Solve

$$\max_{\text{linear program in } \sigma_1^Z} \{ \langle A', x \rangle \mid A' \cdot x \leq b' \} < \min_{\text{linear program in } \sigma_2^Z} \{ \langle A'', x \rangle \mid A'' \cdot x \leq b'' \},$$

Let $Z = (P, T, F, V, I, m_0)$ be a bounded TPN. Additionally the following problems can be decided/computed with the knowledge of its RG, amongst others:

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

Result 6:

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

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

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Let $Z = (P, T, F, V, I, m_0)$ be a bounded TPM. Additionally the following problems can be decided/computed with the knowledge of its RG, amongst others:

Result 7:

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

Output:

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

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

LaTeX Page Zoom

Quantitative Analysis of TPNs

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Definition

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

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

LaTeX Page Zoom

Quantitative Analysis of TPNs

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Result 8:

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

Output:

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

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

LaTeX Page Zoom

Quantitative Analysis of TPNs

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Result 9:

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

Output:

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

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

LaTeX Page Zoom

Quantitative Analysis of TPNs

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Transformation Timed PN \longrightarrow Time PN

LaTeX Page Zoom

Quantitative Analysis of TPNs

Solutions

Mark Property

Applications

Conclusion

Reachability Graph

T-invariant

Time PN is bounded TPNs

Time PN and Timed PN

Conclusion

- theoretical approach
BN \implies modelling \implies PN \implies modelling of steady state \implies
DPN \implies analysing \implies TPN
- experimental approach
BN \implies modelling & analysing \implies TPN

LaTeX Page Zoom

Quantitative Analysis of TPNs