Verification of Infinite-State Systems

Backward Reachability Analysis for Monotonic Systems

Parosh Aziz Abdulla
Uppsala University
Sweden
Background
Parameterized Systems
Petri Nets
Lossy Channel Systems
Timed Petri Nets
Background

Classical Approach
Finite-State Systems

Model Checking
Model |= (safety) property

Challenge:
Infinite-State Systems

Sources of “Infiniteness”: 

Unbounded Data Structures
• stacks (recursion)
• queues (protocols)
• counters (programs)
• clocks (time)
• lists, trees, graphs (heaps)

Unbounded Control Structures
• parameterized systems
• multithreaded programs
• concurrent libraries
• Petri nets

Multiple Sources:
• timed Petri nets
• recursive programs with unbounded data
• channels with time stamps
• etc
Infinite-State Systems
Infinite-State Systems

Unbounded Number of Processes
Infinite-State Systems

Unbounded Number of Processes

Cache Coherence Protocol

- unbounded number of processes
- correctness:
  - exclusive ownership: at most one process
Infinite-State Systems

Unbounded Data Structures
Infinite-State Systems

Unbounded Data Structures

Unbounded Channels

- unbounded FIFO channels
- correctness:
  - regardless of channels size
Infinite-State Systems

Unbounded Data Structures

Unbounded Stack

- push (a);
- push (b);
- push (c);
- pop ();
- push (d);

- unbounded stack
- correctness:
  - regardless of stack size
Infinite-State Systems

Unbounded Data Structures

Unbounded Counters

- unbounded counters
- correctness:
  - regardless of counter values
Infinite-State Systems
Unbounded Data Structures

Clocks

- timed systems
- real-value clocks
Background

Classical Approach
Finite-State Systems

Model Checking
Model $\models$ (safety) property

Challenge:
Infinite-State Systems

Sources of "Infiniteness":

Unbounded Data Structures
- stacks (recursion)
- queues (protocols)
- counters (programs)
- clocks (time)
- lists, trees, graphs (heaps)

Unbounded Control Structures
- parameterized systems
- multithreaded programs
- concurrent libraries
- Petri nets

Multiple Sources:
- timed Petri nets
- recursive programs with unbounded data
- queues with time stamps
- etc
Background
Parameterized Systems
Petri Nets
Lossy Channel Systems
Timed Petri Nets
Parameterized Systems
Parameterized Systems

$L = \text{free?}$

$C$

$L = \text{busy}$

$L := \text{free}$
Parameterized Systems

- Specification
  - Mutual Exclusion (MutEx):
  - At most one process in C

I

\[ L = \text{free?} \]
\[ L := \text{busy} \]
\[ L := \text{free} \]
Parameterized Systems

$P^n|L$

- **Specification**
  - Mutual Exclusion (MutEx):
  - At most one process in C
Task = Parameterized Verification

- Verify correctness regardless of the number of processes
- \( \forall n. (P^n | L) \models MutEx \)

Specification

- Mutual Exclusion (MutEx):
  - At most one process in C
Parameterized Systems

Infinite-State System

\[ P^n | L \]

- Specification
  - Mutual Exclusion (MutEx):
    - At most one process in C

- Task = Parameterized Verification
  - Verify correctness regardless of the number of processes
  - \( \forall n. (P^n | L) \models \text{MutEx} \)
Background
Parameterized Systems
Petri Nets
Lossy Channel Systems
Timed Petri Nets
Petri Nets
Petri Nets
Petri Nets

\[ t_1 \quad \text{places} \quad t_2 \]
Petri Nets

- Configurations
- Ordering
- Monotonicity
- Upward Closed Sets
- Backward Reachability
- Transitions
- Computing Predecessors

Model
Petri Nets

Markings

Diagram showing Petri net with transitions $t_1$ and $t_2$ and tokens at places.
Markings
Markings

Petri Nets

$t_1$ $t_2$

marking
Markings

Petri Nets

$t_1$  $t_2$

marking  multiset
Markings
Transitions
Petri Nets

Transitions
Transitions

$t_1$ transition

$t_2$
Petri Nets

Transitions

$\text{transition}$

$t_1$ $t_2$
Encoding (counter abstraction)

- # tokens in $\text{red}$ = # processes in $\text{red}$
- # tokens in $\text{green}$ = # processes in $\text{green}$
- one/no token in $\text{blue}$ = lock free/busy
Petri Nets

Transitions

\[
\begin{align*}
t_1 & \rightarrow \text{Transition} \\
t_2 & \rightarrow \text{Transition}
\end{align*}
\]
Petri Nets

Transitions

$t_1$ $t_2$ $t_1$

$\text{Transitions}$

$t_1$ $t_2$ $t_1$
Petri Nets

Transitions

t_1 \rightarrow t_2
Petri Nets

Transitions

$t_1$
Petri Nets

Transitions

$t_1 \quad t_2$

$t_1$

Green circle

Red circle

Black circle

Blue, Green, Red, Black circles
Transitions

Petri Nets

$t_1$, $t_2$
Petri Nets

Transitions

$t_1$  $t_2$
Petri Nets

Safety Properties
Safety Properties

Initial Markings (Init)

- infinitely many
- one
- arbitrarily many
Safety Properties

**Initial Markings (Init)**
- infinitely many
- one
- arbitrarily many

**Bad Markings (Bad)**
- infinitely many
- at least two
Safety Properties

Initial Markings ($Init$)
- infinitely many
- one
- arbitrarily many

Bad Markings ($Bad$)
- infinitely many
- at least two

Safety Property
$Init \overset{*}{\rightarrow} Bad$
Safety Properties

Petri Nets

Initial Markings (\textit{Init})
- infinitely many
- one
- arbitrarily many

Bad Markings (\textit{Bad})
- infinitely many
- at least two

Safety Property
\[ \text{Init} \rightarrow^* \text{Bad} \]
Safety Properties

Initial Markings (Init)
- infinitely many
- one
- arbitrarily many

Bad Markings (Bad)
- infinitely many
- at least two

Safety Property

\( \text{Init} \rightarrow ^* \text{Bad} \)
Safety Properties

Initial Markings (Init)
- infinitely many
- one
- arbitrarily many

Bad Markings (Bad)
- infinitely many
- at least two

Safety Property

Init * Bad ?
Safety Properties

**Initial Markings (**$\text{Init}$**)**
- infinitely many
- one
- arbitrarily many

**Bad Markings (**$\text{Bad}$**)**
- infinitely many
- at least two

Safety Property

$\text{Init} \xrightarrow{*} \text{Bad}$ ?
Safety Properties

How to check safety properties?

Initial Markings (Init)
- infinitely many
- one
- arbitrarily many

Bad Markings (Bad)
- infinitely many
- at least two

Safety Property

Init ➡️* Bad ?
Safety Properties

How to check safety properties?

- Initial Markings (Init)
  - one
  - arbitrarily many

- Bad Markings (Bad)
  - at least two
  - infinitely many

Safety Property

Init * Bad ?

- Ordering
- Monotonicity
- Upward Closed sets
- Predecessors
- Backward Reachability
Petri Nets
Ordering
Petri Nets

Ordering

$\tau_1 \rightarrow \tau_2$
Ordering

Petri Nets
Petri Nets

- Configurations
- Transitions
- Ordering
- Monotonicity
- Upward Closed Sets
- Computing Predecessors
- Backward Reachability

Model

- ✓
Petri Nets

Monotonicity

Monotonicity
Petri Nets

Monotonicity

[Diagram of Petri Nets showing transitions and tokens]

Monotonicity
Petri Nets

Monotonicity
Monotonicity

larger configuration "simulate" smaller configurations
Petri Nets

- Model
  - Configurations
  - Transitions
  - Ordering
  - Monotonicity
    - Upward Closed Sets
    - Computing Predecessors
    - Backward Reachability
Upward Closed Set (UC)

- if $m_1 \in U$ and $m_1 \subseteq m_2$
- then $m_2 \in U$
Upward Closed Set (UC)

- if \( m_1 \in U \) and \( m_1 \subseteq m_2 \)
- then \( m_2 \in U \)

Why UC?

- Bad sets of markings are UC
  - checking safety properties = reachability of bad markings
- Uniquely characterized by generator
  - simple representation = finite multiset
implies Petri Nets
Upward Closed Sets

generator

$\subseteq$

$\subseteq$

genenerator
Petri Nets

Predecessors

$m_3$

$m_1$

$U$
Petri Nets

Predecessors

\[ \text{Pre}(U) \rightarrow m_2 \rightarrow m_4 \rightarrow m_3 \rightarrow m_1 \rightarrow U \]
Monotonicity: UC persevered by Pre
Monotonicity: UC persevered by $\text{Pre}$
Monotonicity: UC persevered by $Pre$

$Pre(U)$

upward closed?
Petri Nets

Predecessors

Monotonicity: UC persevered by $Pre$

$Pre(U)$

upward closed

$m_1 \rightarrow m_2$

$m_3 \rightarrow m_4$

$m_3$

$\square$

$m_1$

$\exists$

$U$

$\square$

upward closed

$Pre(U)$

upward closed
Monotonicity: UC persevered by $\text{Pre}$
Monotonicity: UC persevered by $Pre$

$Pre(U)$ is upward closed?
Petri Nets
Computing Predecessors

$Pre_{t_1} =$
Petri Nets
Computing Predecessors

\[ \text{Pre}_{t_1} \begin{bmatrix} \text{red} \end{bmatrix} = \begin{bmatrix} \text{red}, \text{green}, \text{blue} \end{bmatrix} \]
Petri Nets
Computing Predecessors

\[ \text{Pre}_{t_1} \begin{bmatrix} \text{red} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \\ \text{green} \\ \text{blue} \end{bmatrix} \]

\[ \text{Pre}_{t_2} \begin{bmatrix} \text{red} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \end{bmatrix} \]
Petri Nets
Computing Predecessors

\[
Pre_{t_1} \left[ \begin{array}{c}
\text{(initial configuration)}
\end{array} \right] = \begin{array}{c}
\text{(final configuration)}
\end{array}
\]

\[
Pre_{t_2} \left[ \begin{array}{c}
\text{(initial configuration)}
\end{array} \right] = \begin{array}{c}
\text{(final configuration)}
\end{array}
\]
Petri Nets

- Model
  - Configurations
    - Monotoncity
      - Upward Closed Sets
        - Computing Predecessors
  - Transitions
  - Ordering
    - Backward Reachability
Petri Nets

Backward Reachability
Backward Reachability
Petri Nets

Backward Reachability
Petri Nets

Backward Reachability

$\text{Petri Net}$

$\text{Backward Reachability}$
Petri Nets

Backward Reachability
Petri Nets

Backward Reachability

$\text{Pre}^*(\mathcal{N})$
Petri Nets

Backward Reachability

System Safe!

Symbolic representation = finite multisets

Termination: multisets well quasi-ordered
Petri Nets

Well Quasi-Ordering

$m_0, m_1, m_2, \ldots m_i, \ldots m_j, \ldots$

$\exists i < j : m_i \sqsubseteq m_j$

infinite sequence of markings
Ordering:
  - monotonicity
  - computing predecessors
  - well quasi-ordering
Background
Parameterized Systems
Petri Nets
Lossy Channel Systems
Timed Petri Nets
Lossy Channel Systems

Model

Ordering

Transitions

Configurations

Monotoncity

Upward Closed Sets

Computing Predecessors

Backward Reachability
Lossy Channel Systems

P2

finite-state process

lossy unbounded FIFO buffer

P1

c1

lossy unbounded FIFO buffer

P2

C1

P3

c2

finite-state process

lossy unbounded FIFO buffer
Lossy Channel Systems

Weak Memory Models

- \( x = 2 \)
- \( y = 1 \)
- \( z = 3 \)
Lossy Channel Systems

Weak Memory Models

thread

P₁

y=2

P₂

z=1

P₃

thread

x=2

y=1

z=3

Shared Memory

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer
Lossy Channel Systems

Weak Memory Models

Shared Memory

x=2

y=1

z=3

P_1

P_2

P_3

ty=2

tz=1

thread

thread

thread

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer
Lossy Channel Systems

Weak Memory Models

Shared Memory

P1

P2

P3

thread

thread

thread

Lossy unbounded FIFO buffer

Lossy unbounded FIFO buffer

Lossy unbounded FIFO buffer

x=3

y=2

z=1

x=2

y=1

z=3

P1

P2

P3
Lossy Channel Systems

Weak Memory Models

Shared Memory

thread

P1

P2

thread

P3

lossy unbounded FIFO buffer

x=4  x=3  y=2

z=1

lossy unbounded FIFO buffer

x=2

y=1

z=3
Lossy Channel Systems

Weak Memory Models

Lossy unbounded FIFO buffer

thread

P₁

x=4  x=3  y=2

P₂

z=1  y=2  z=1

P₃

x=0  x=1  y=0

Shared Memory

x=2

y=1

z=3
Lossy Channel Systems

Weak Memory Models

thread

P₁

lossy unbounded FIFO buffer

x=4 x=3

P₂

z=1 y=2 z=1

P₃

x=0 x=1 y=0

Shared Memory

x=2 y=2 z=3

thread

thread

thread

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer
Lossy Channel Systems

Weak Memory Models

Shared Memory

P1

P2

P3

thread

thread

thread

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer

lossy unbounded FIFO buffer
Lossy Channel Systems

- Lossy unbounded FIFO buffer

sender

receiver

c

d
Lossy Channel Systems

\[ P_1 \]

\[ P_2 \]

lossy unbounded FIFO buffer

\[ 0 \ 0 \ 0 \]

\[ 0 \ 0 \ 0 \]
Lossy Channel Systems

\[ P_1 \]

\[ P_2 \]

lossy unbounded FIFO buffer

0 0 0

0 0 0 1 1

0 0 0 1 1

\[ d?1 \] \[ c!0 \]

\[ d?0 \] \[ c!1 \]

\[ c?1 \] \[ d!0 \]

\[ c?0 \] \[ d!1 \]
Lossy Channel Systems

$P_1$

$P_2$

Lossy unbounded FIFO buffer

0 0 0 1 1

0 0
Lossy Channel Systems

Configurations

- State of $P_1$
- State of $P_2$
- Content of $c$
- Content of $d$

Configuration: 00 110
Lossy Channel Systems

Configurations

- State of $P_1$
- State of $P_2$
- Content of $c$
- Content of $d$

Configuration: 00 101
Lossy Configurations

Configurations

Initial state of $P_1$
Initial state of $P_2$
Content of $c$
Content of $d$

Initial configuration

State of $P_1$
State of $P_2$
Content of $c$
Content of $d$

Configuration
Lossy Channel Systems

- Model
  - Configurations
  - Backward Reachability
- Transitions
  - Ordering
  - Upward Closed Sets
  - Computing Predecessors
- Monotonicity
  - Backward Reachability
Lossy Channel Systems

Transitions

\[ t_4 \]
Lossy Channel Systems

Transitions

\[ c \quad \delta \]

\[ d \quad \delta \]

\[ t_1 \quad t_2 \quad t_3 \quad t_4 \]

\[ t_5 \quad t_6 \quad t_7 \quad t_8 \]

\[ t_9 \quad t_{10} \quad t_{11} \quad t_{12} \]

\[ \epsilon \quad \epsilon \quad 0 \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]

\[ \epsilon \quad \epsilon \]
Lossy Channel Systems

Transitions

\[
\begin{align*}
\text{(a)} & : c!0 \\
\text{(b)} & : c?0 \\
\text{(c)} & : d?0 \\
\text{(d)} & : t_1, t_2, t_3 \\
\text{(e)} & : t_4, t_5, t_6 \\
\end{align*}
\]
Lossy Channel Systems

Transitions

\[ d?1 (t_3, t_4) c!0 \]
\[ d?1 (t_1, t_2, d?0) \]
\[ d?0 (t_5, t_6) c!1 \]

\[ c?1 (t_7, t_8) c?0 \]
\[ c?0 (t_{11}, t_{12}) d!1 \]

\[ t_{10} \]

\[ t_4 \]
\[ t_4 \]
\[ t_{10} \]
Lossy Channel Systems

Transitions

\[ \begin{align*}
  &d?1(t_3, t_4, c^0) \\
  &d?1(t_1, t_2, d^0) \\
  &d?0(t_5, t_6, c^1) \\
  &c?1(t_9, t_{10}, d^0) \\
  &c?1(t_7, t_8, c^0) \\
  &c?0(t_{11}, t_{12}, d^1) \\
  &t_10 \rightarrow d!0 \\
\end{align*} \]
Lossy Channel Systems

Transitions
Lossy Channel Systems

Transitions

\[ c \rightarrow \begin{array}{c}
  t_1 \\
  t_2 \\
  t_3 \\
  t_4 \\
  t_5 \\
  t_6 \\
  t_7 \\
  t_8 \\
  t_9 \\
  t_{10} \\
  t_{11} \\
  t_{12}
\end{array}
\]

\[ d \rightarrow \begin{array}{c}
  \epsilon \\
  0 \\
  \epsilon \\
  0 \\
  \epsilon \\
  0 \\
  \epsilon \\
  0 \\
  \epsilon \\
  0 \\
  \epsilon \\
  0 \\
  \epsilon
\end{array}\]
Lossy Channel Systems

Transitions

\[ \begin{align*}
\text{Transition 1:} & \quad d?1 \quad t_3 \quad t_4 \quad c!0 \\
\text{Transition 2:} & \quad d?1 \quad t_1 \quad t_2 \quad d?0 \\
\text{Transition 3:} & \quad d?0 \quad t_5 \quad t_0 \quad c!1 \\
\text{Transition 4:} & \quad c?1 \quad t_9 \quad t_{10} \quad d!0 \\
\text{Transition 5:} & \quad c?1 \quad t_7 \quad t_8 \quad c?0 \\
\text{Transition 6:} & \quad c?0 \quad t_{11} \quad t_{12} \quad d!1 \\
\end{align*} \]
Lossy Channel Systems

Transitions

\[
\begin{align*}
\text{Lossy Channel Systems} & \quad \text{Transitions} \\
\text{\begin{align*}
& d?1(t_3 t_4) c?0 \\
& d?1 t_1 t_2 d?0 \\
& d?0 t_5 t_6 c?1 \\
& c?1 t_9 t_{10} d?0 \\
& c?0 t_7 t_{11} t_{12} d?1
\end{align*}} & \quad \begin{align*}
& \epsilon \quad \epsilon \\
& 0 \quad \epsilon \\
& 00 \quad \epsilon \\
& 00 \quad 00 \\
& 00 \quad 00 \\
& 0 \quad 00 \\
& 0 \quad 100 \\
& 0 \quad 10 \\
\end{align*} & \quad \begin{align*}
& t_4 \\
& t_4 \\
& t_{10} \\
& t_{10} \\
& t_{10} \\
& t_{12} \\
& t_2 \\
& t_2 \\
\end{align*}
\end{align*}
\]
Lossy Channel Systems

Transitions

- Transition $t_4$: $\epsilon \rightarrow \epsilon$
- Transition $t_4$: $0 \rightarrow \epsilon$
- Transition $t_{10}$: $00 \rightarrow \epsilon$
- Transition $t_{10}$: $00 \rightarrow 0$
- Transition $t_8$: $00 \rightarrow 0$
- Transition $t_{12}$: $00 \rightarrow 0$
- Transition $t_2$: $0 \rightarrow 10$
- Transition $t_2$: $0 \rightarrow 10$
- Transition $\text{loss}$: $0 \rightarrow 10$

Diagram:

- Node $d$: $0, 1$
- Node $c$: $0, 1$
- Node $e$: $0, 1$
- Node $f$: $0$
Lossy Channel Systems
Ordering
Subword Relation

ab \subseteq xaycz
Lossy Ordering

Subword Relation

\[ ab \sqsubseteq xaycz \]
Lossy Channel Systems

Ordering

Subword Relation

\[ ab \sqsubseteq xaycz \]

Ordering

Monotonicity
Lossy Upward-Closed Sets

generator
Lossy Channel Systems

- Model
- Configurations
- Ordering
- Transitions
- Monotoncity
- Upward Closed Sets
- Computing Predecessors
- Backward Reachability
Monotonicity: UC persevered by $Pre$

$Pre(U)$ upward closed?
Lossy Computing Predecessors
Lossy Computing Predecessors

$P_{\text{ret}_2}$ =

$\begin{array}{c}
\begin{array}{c}
\text{Pre}_2 \\
= \\
\end{array}
\end{array}$
Lossy Computing Predecessors

\[ \text{Pre}_{t_2} = \]
Lossy Channel Systems

Model

Configurations

Ordering

Transitions

Monotoncity

Upward Closed Sets

Computing Predecessors

Backward Reachability
Lossy Backward Reachability

\[ d?1 \quad t_3 \quad t_4 \quad c!0 \]
\[ d?1 \quad t_1 \quad t_2 \quad d?0 \]
\[ d?0 \quad t_5 \quad t_6 \quad c!1 \]

\[ c?1 \quad t_9 \quad t_{10} \quad d!0 \]
\[ c?1 \quad t_7 \quad t_8 \quad c!0 \]
\[ c?0 \quad t_11 \quad t_12 \quad d!1 \]

\[ t_2 \quad d?0 \]
Lossy Backward Reachability
Backward Reachability

Lossy Channel Systems
Lossy Backward Reachability
Lossy Backward Reachability

d?1 \ t_3 \ t_4 \ c!0
d?1 \ t_1 \ t_2 \ d?0
d?0 \ t_5 \ t_6 \ c!1

c?1 \ t_9 \ t_{10} \ d!0
c?1 \ t_7 \ t_8 \ c?0
c?0 \ t_1 \ t_2 \ d!1

t_4 \ c!0

\text{01} \ \epsilon
\text{01} \ 0
\text{010} \ \epsilon
\epsilon \ 01
\epsilon \ 010
Lossy Backward Reachability
Lossy Backward Reachability

c

\[
\begin{array}{c}
\begin{array}{c}
\text{c0} \\
\text{t3} \\
\text{t2} \\
\text{t1} \\
\text{c1} \\
\text{c2} \\
\text{c3} \\
\text{t5} \\
\text{t4} \\
\text{c0} \\
\end{array}
\end{array}
\text{c0}
\end{array}
\]

d

\[
\begin{array}{c}
\begin{array}{c}
\text{c1} \\
\text{t9} \\
\text{t8} \\
\text{t7} \\
\text{c0} \\
\text{c1} \\
\text{c2} \\
\text{t10} \\
\text{t11} \\
\text{c0} \\
\end{array}
\end{array}
\text{c1}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{d0} \\
\text{t3} \\
\text{t2} \\
\text{t1} \\
\text{c0} \\
\text{c1} \\
\text{c2} \\
\text{t5} \\
\text{t4} \\
\text{c0} \\
\end{array}
\end{array}
\text{d1}
\end{array}
\]

c

\[
\begin{array}{c}
\begin{array}{c}
\text{d0} \\
\text{t3} \\
\text{t2} \\
\text{t1} \\
\text{c0} \\
\text{c1} \\
\text{c2} \\
\text{t5} \\
\text{t4} \\
\text{c0} \\
\end{array}
\end{array}
\text{d1}
\end{array}
\]

d

\[
\begin{array}{c}
\begin{array}{c}
\text{c1} \\
\text{t9} \\
\text{t8} \\
\text{t7} \\
\text{c0} \\
\text{c1} \\
\text{c2} \\
\text{t10} \\
\text{t11} \\
\text{c0} \\
\end{array}
\end{array}
\text{c1}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{d0} \\
\text{t3} \\
\text{t2} \\
\text{t1} \\
\text{c0} \\
\text{c1} \\
\text{c2} \\
\text{t5} \\
\text{t4} \\
\text{c0} \\
\end{array}
\end{array}
\text{d1}
\end{array}
\]
symbolic representation = finite words
Lossy Backward Reachability

symbolic representation = finite words

Termination: words well quasi-ordered
Well Quasi-Ordering

\[ w_0, w_1, w_2, \ldots, w_i, \ldots, w_j, \ldots \]

\[ \exists i < j : w_i \sqsubseteq w_j \]
Well Quasi-Ordering

\( w_0, w_1, w_2, \ldots, w_i, \ldots, w_j, \ldots \)

\[ \exists i < j : w_i \sqsubseteq w_j \]

Well Quasi-Ordering

\( c_0, c_1, c_2, \ldots, c_i, \ldots, c_j, \ldots \)

\[ \exists i < j : c_i \sqsubseteq c_j \]
Ordering:
- monotonicity
- computing predecessors
- well quasi-ordering
Timed Petri Nets
Timed Petri Nets
Timed Petri Nets

Model

Ordering

Transitions

Configurations

Monotonicity

Upward Closed Sets

Computing Predecessors

Backward Reachability
Markings
Timed Petri Nets

Transitions

$0.0$

$0.0$
Timed Petri Nets

Transitions

0.0

1.7

timed transition
Timed Petri Nets

Transitions

Timed transition
Timed Petri Nets

Transitions

Discrete transition
Timed Petri Nets

Transitions

Discrete transition
Timed Petri Nets

Transitions

Timed transition
Timed Petri Nets

Transitions

Timed transition
Timed Petri Nets

Transitions

Discrete transition

Time points:
- $t_1 = 1.7$
- $t_2 = 2.3$
- $t_3 = 1.0$
- $t_4 = 0.1$
- $t_5 = 3.1$

Transition times:
- $2.4$ to $5.4$

Interval times:
- $(1..3)$
- $(1..4)$
- $(1..5)$
- $(2..3)$
- $(2..5)$
- $(3..4)$
- $(4..5)$
- $(5..6)$
- $(1..2)$
- $(1..\infty)$
Timed Petri Nets

Transitions

- Transition 1: 0.0 → 1.7
- Transition 2: 2.4 → 3.6
- Transition 3: 5.4 → 5.4

Discrete transition
Timed Petri Nets

Transitions

Graphical representation of Timed Petri Nets with labeled transitions and token allocation.
Timed Petri Nets

cmax=6
signature: “sequence of multisets of colored natural numbers”

s: signature

cmax = 6
signature: “sequence of multisets of colored natural numbers”
Timed Petri Nets

Signatures

multiset

- integer part = 1
- place: 1

integer part = 5
place: 5

signatures

"sequence of multisets of colored natural numbers"

cmax = 6
signature: “sequence of multisets of colored natural numbers”

Timed Petri Nets

Signatures

cmax=6

multiset

• integer part = 1
• place: 1

• integer part = 5
• place: 5

• integer part > cmax
• place: ω

s: signature
signature: “sequence of multisets of colored natural numbers”
signature:
“sequence of multisets of colored natural numbers”
Timed Petri Nets

Signatures

cmax = 6

zero fractional part

increasing fractional parts

\( \text{sig}(c) = s \)

\( c \): configuration

\( s \): signature

6.7 5.0 3.5 4.7 8.2 1.9 4.3 2.9 2.2 0.8 1.0
Timed Petri Nets

Signatures

zero fractional part

increasing fractional parts

\[ \text{cmax}=6 \]

\[ \text{sig}(c) = s \]

configuration

S: signature
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s \]

\[ c_{\text{max}} = 6 \]

0: zero fractional part
1: increasing fractional parts

s: signature

c: configuration
Timed Petri Nets

Signatures

sig(c) = s

cmax = 6

zero fractional part

increasing fractional parts

s: signature

c: configuration
Timed Petri Nets

Signatures

$c_{max}=6$

$\text{sig}(c) = s$

$\omega$

zero fractional part

increasing fractional parts

$s$: signature

$c$: configuration
Timed Petri Nets

Signatures

cmax = 6

zero fractional part
increasing fractional parts

\[ \text{sig}(c) = s \]

\[ c_{\text{max}} = 6 \]
Timed Petri Nets

Signatures

zero fractional part

increasing fractional parts

$c_{max}=6$

$s: signature$

c: configuration

$\text{sig}(c) = s$

6.7 5.0 3.5 4.7 8.2 1.9 4.3 2.9 2.2 0.8 1.0
1.9 0.8 1.1
5.1 4.7 2.9
6.7 3.5 2.2
8.2 4.3 0.8
4.3 2.2 1.1

sig(c) = s

s: signature

c: configuration

cmax=6

Timed Petri Nets
Signatures
Timed Petri Nets

Signatures

\[ \text{sig}(c) \neq s \]

\[ c_{\text{max}} = 6 \]

\[ s: \text{signature} \]

\[ c: \text{configuration} \]
Timed Petri Nets

Signatures

$c_{\text{max}} = 6$

$s: \text{signature}$

$c: \text{configuration}$

$s(c) \neq s$
Timed Petri Nets

Signatures

\[ \text{cmax} = 6 \]

\[ \text{sig}(c) \neq s \]

\[ s: \text{signature} \]

\[ c: \text{configuration} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s \]

\[ c: \text{configuration} \]

\[ s: \text{signature} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s \]

\[ c: \text{configuration} \]

\[ s: \text{signature} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s \]
sig(c) = s: c and s are "bisimilar"
$\text{sig}(c) = s$: $c$ and $s$ are "bisimilar"
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \text{ } c \text{ and } s \text{ are "bisimilar"} \]
sig(c) = s: c and s are "bisimilar"
sig(c)=s: c and s are “bisimilar”
\textbf{Timed Petri Nets}

\textbf{Signatures}

\[\text{sig}(c) = s:\]
\[c \text{ and } s \text{ are "bisimilar"}\]
sig(c)=s: c and s are "bisimilar"
sig(c) = s: c and s are "bisimilar"
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \text{c and s are "bisimilar"} \]

\[ t_1 \]

\[ [0..1) \]
\[ [2..5) \]

\[ t_1 \]

\[ [0..1) \]
\[ [2..5) \]
sig(c)=s: c and s are "bisimilar"
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \text{c and s are "bisimilar"} \]
sig(c) = s: c and s are “bisimilar”
sig(c)=s:
c and s are "bisimilar"
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \]  
\[ c \text{ and } s \text{ are "bisimilar"} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \]

\[ c \text{ and } s \text{ are “bisimilar”} \]
Timed Petri Nets

\[ \text{Sig}(c) = s: \]
\[ c \text{ and } s \text{ are "bisimilar"} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \]

\[ c \text{ and } s \text{ are “bisimilar”} \]
Timed Petri Nets
Signatures

\[ \text{sig}(c) = s: \]
\[ c \text{ and } s \text{ are “bisimilar”} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \ c \text{ and } s \text{ are "bisimilar"} \]
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \]
c and s are “bisimilar”
Timed Petri Nets

Signatures

\[ \text{sig}(c) = s: \]

\(c\) and \(g\) are “bisimilar”

\[
\begin{array}{cccccccc}
5.0 & 3.7 & 8.2 & 0.7 & 2.2 & 6.5 & 1.0 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
5.1 & 3.8 & 8.3 & 0.8 & 2.3 & 6.6 & 1.1 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
5.2 & 3.9 & 8.4 & 0.9 & 2.4 & 6.7 & 1.2 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
5.3 & 4.0 & 8.5 & 1.0 & 2.5 & 6.8 & 1.3 \\
\end{array}
\]
Timed Petri Nets

Signatures
Timed Petri Nets
Signatures

The image contains two states of a Timed Petri Net, labeled with tokens as follows:

1. State 1: Tokens 1, 2, 3, 5, 6, and 0.
2. State 2: Tokens 1, 2, 3, 5, 6, and 0 with an additional token labeled with ω.
Timed Petri Nets

Signatures

![Diagram of Timed Petri Nets with signatures and time labels]
Timed Petri Nets

Signatures
Timed Petri Nets

Signatures
Timed Petri Nets

Model
- Configurations
- Transitions
- signatures Ordering

Monotoncity
- Upward Closed Sets
- Computing Predecessors
- Backward Reachability
Timed Petri Nets
Equivalence

\[ c_1 \equiv c_2 : \quad \text{sig}(c_1) = \text{sig}(c_2) \]
Timed Petri Nets

Equivalence

\[ c_1 \equiv c_2: \quad \text{sig}(c_1) = \text{sig}(c_2) \]
$c_1 \equiv c_2 : \quad \text{sig}(c_1) = \text{sig}(c_2)$

$\text{sig}(c_1)$

$\text{sig}(c_2)$
$c_1 \sqsubseteq c_2$:

$\exists c_3. (c_1 \equiv c_3) \land (c_3 \subseteq c_2)$
Timed Petri Nets

Ordering

$c_1 \sqsubseteq c_2$:

$\exists c_3. (c_1 \equiv c_3) \land (c_3 \subseteq c_2)$
Timed Petri Nets

Ordering

\[ c_1 \sqsubseteq c_2 : \exists c_3. (c_1 \equiv c_3) \land (c_3 \subseteq c_2) \]
s₁ ⊆ s₂: Derive s₁ from s₂ by:
• removing elements from multisets
• removing multisets
Derive $s_1$ from $s_2$ by:

- removing elements from multisets
- removing multisets
$s_1 \subseteq s_2$: Derive $s_1$ from $s_2$ by:
- removing elements from multisets
- removing multisets
Timed Petri Nets

Ordering

\[ c \models s : \exists c'. (c' \sqsubseteq c) \land (\text{sig}(c') = s) \]
\[ c \models s : \exists c'. (c' \sqsubseteq c) \land (\text{sig}(c') = s) \]
### Timed Petri Nets

**Ordering**

\[
c ⊨ s \quad \exists c'. (c' ⊆ c) \land (\text{sig}(c') = s)
\]

<table>
<thead>
<tr>
<th>5.0</th>
<th>1.7</th>
<th>8.2</th>
<th>4.7</th>
<th>3.2</th>
<th>6.5</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.8</td>
<td>9.1</td>
<td>9.1</td>
<td>1.1</td>
<td>9.9</td>
<td>6.6</td>
<td></td>
</tr>
</tbody>
</table>

\[\text{sig}(c')\]

\[\omega\]
Timed Petri Nets

Ordering

\[ c \models s : \exists c'. (c' \subseteq c) \land (\text{sig}(c') = s) \]
Timed Petri Nets

Denotation

\[ [s] = \{ c \mid c \models s \} \]
Timed Petri Nets

Denotation

\[ [s] = \{ c \mid c \models s \} \]
Timed Petri Nets

Denotation

\[ [s] = \{ c \mid c \models s \} \]
$[s] = \{c \mid c \models s\}$
\[ [s] = \{ c \mid c \models s \} \]
Timed Petri Nets

Denotation

\[ [s] = \{ c \mid c \models s \} \]
Timed Petri Nets

\[ [s] = \{ c \mid c \models s \} \]

\[ \mathbb{J}_s \mathbb{K}_s = \{ c \mid c \models s \} \]

infinite

upward closed wrt. \( \subseteq \)
Timed Petri Nets

Denotation

\[ [s] = \{c | c \models s\} \]

\( \omega \)

\( \leq \)

\( \subseteq \)

\( s_1 \subseteq s_2 \)

\( [s_2] \subseteq [s_1] \)

infinite

upward closed wrt. \( \subseteq \)
Timed Petri Nets

Monotonicity

5.0 1.7 8.2 4.7 3.2 6.5 1.0
Timed Petri Nets

### Monotonicity

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>1.7</td>
<td>8.2</td>
<td>4.7</td>
<td>3.2</td>
<td>6.5</td>
</tr>
<tr>
<td>5.0</td>
<td>3.2</td>
<td>2.5</td>
<td>4.8</td>
<td>3.1</td>
<td>6.6</td>
</tr>
<tr>
<td>1.8</td>
<td>9.1</td>
<td>9.1</td>
<td>1.1</td>
<td>9.9</td>
<td>6.6</td>
</tr>
</tbody>
</table>
Timed Petri Nets
Monotonicity

\[ \text{time} = 0.3 \]
Timed Petri Nets

Monotonicity

time=0.3

5.0 1.7 8.2 4.7 3.2 6.5 1.0

5.3 2.0 8.5 5.0 3.5 6.8 1.3

time=0.2

5.0 3.2 2.5 4.8 3.1 6.6 1.0

5.2 3.4 2.7 5.0 3.3 6.8 1.2

1.8 9.1 9.1 1.1 9.9 6.6

2.0 9.3 9.3 1.3 10.1 6.8
Computing Predecessors

\( \text{time} = 0.3 \)

\( \text{time} = 0.2 \)
Timed Petri Nets

- Model
  - Configurations
  - Ordering
    - Monotonicity
      - Upward Closed Sets
      - Backward Reachability
  - Transitions
    - Computing Predecessors
Timed Petri Nets

Computing Predecessors
Computing predecessors

Timed Petri Nets
Timed Petri Nets

Computing Predecessors

\[
\text{Pre}_{\text{time}} = \begin{bmatrix}
2 & 0 & 6 & 1 & 0 \\
\omega & 4 & 4 & 4 & 4
\end{bmatrix}
\]
Timed Petri Nets

Computing Predecessors

\[ P_{\text{time}} = \]

[Diagram showing Timed Petri Nets with places and transitions labeled with numbers and time values.]
Pre-time = Timed Petri Nets
Computing Predecessors

Pre-time

Time Petri Nets

\[ \text{Time} \text{ Petri Nets} \]
Timed Petri Nets

Computing Predecessors

Pre\text{time}
Timed Petri Nets

- Model
  - Configurations
  - Ordering
    - Monotonocity
      - Upward Closed Sets
      - Backward Reachability
    - Computing Predecessors
  - Transitions
Timed Petri Nets

Backward Reachability
Timed Petri Nets

Backward Reachability
symbolic representation = finite words over finite multisets

Termination:
finite words over finite multisets well quasi-ordered