On Tools for Coverability

Michael Blondin, Christoph Haase, Grégoire Sutre

Part I: QCover

Michael Blondin

Joint work with Alain Finkel, Christoph Haase and Serge Haddad

Verifying safety with Petri nets

Process 1

Process 2

Lamport mutual exclusion "1-bit algorithm"

Verifying safety with Petri nets

Process 1

critical section

Process 2

critical section

Lamport mutual exclusion "1-bit algorithm"

Verifying safety with Petri nets

```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```

```
  while True:
★   y = True
    if x then:
      y = False
      while x: pass
      goto ★
    # critical section
    y = False
```

Verifying safety with Petri nets

```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```

```
while True:
★ y = True
  if x then:
    y = False
    while x: pass
    goto ★
  # critical section
  y = False
```

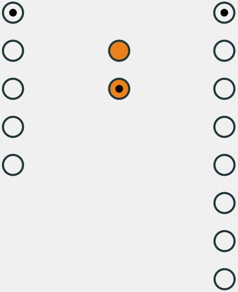
Verifying safety with Petri nets

```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```

```
while True:
★ y = True
  if x then:
    y = False
    while x: pass
    goto ★
  # critical section
  y = False
```

Verifying safety with Petri nets

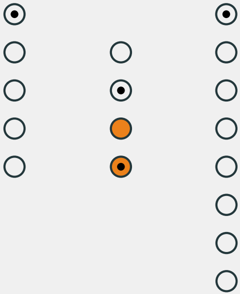
```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```



```
while True:
★ y = True
  if x then:
    y = False
    while x: pass
    goto ★
  # critical section
  y = False
```

Verifying safety with Petri nets

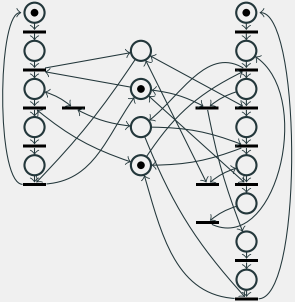
```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```



```
while True:
  y = True
    if x then:
      y = False
      while x: pass
      goto
  # critical section
  y = False
```

Verifying safety with Petri nets

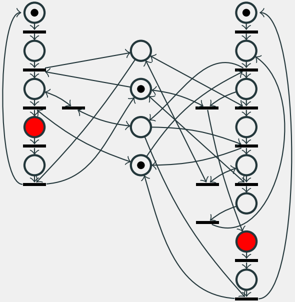
```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```



```
while True:
★  y = True
   if x then:
     y = False
     while x: pass
     goto ★
   # critical section
   y = False
```

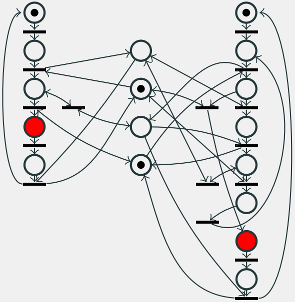
Verifying safety with Petri nets

```
while True:
  x = True
  while y: pass
  # critical section
  x = False
```



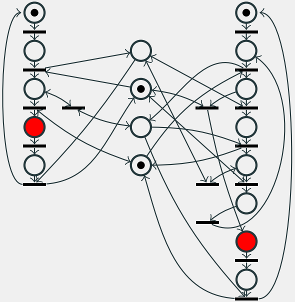
```
while True:
★ y = True
  if x then:
    y = False
    while x: pass
    goto ★
  # critical section
  y = False
```

Verifying safety with Petri nets



Processes at both critical sections $\iff$ each  ≥ 1

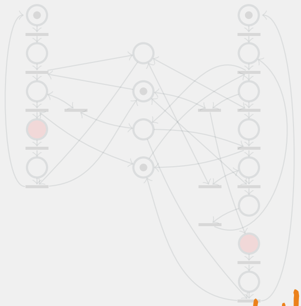
Verifying safety with Petri nets



Processes at both critical sections $\iff$ each  ≥ 1

 ≥ 0

Verifying safety with Petri nets



Coverability problem

Processes at both critical sections $\Longleftrightarrow$ each ● ≥ 1 ○ ≥ 0

Coverability problem

Problem

Input:	Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$
Question:	Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

Coverability problem

Problem

Input: Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$

Question: Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

How to solve it?

- Forward: build reachability tree from initial marking
- Backward: find predecessors of markings covering target
- EXPSPACE-complete

Coverability problem

Problem

Input: Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$

Question: Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

How to solve it? **Karp & Miller '69**

- Forward: build reachability tree from initial marking
- Backward: find predecessors of markings covering target
- EXPSPACE-complete

Coverability problem

Problem

Input:	Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$
Question:	Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

How to solve it? **Arnold & Latteux '78, Abdulla *et al.* '96**

- Forward: build reachability tree from initial marking
- Backward: find predecessors of markings covering target
- EXPSPACE-complete

Coverability problem

Problem

Input: Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$

Question: Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

How to solve it? **Lipton '76, Rackoff '78**

- Forward: build reachability tree from initial marking
- Backward: find predecessors of markings covering target
- EXPSPACE-complete

Coverability problem

Problem

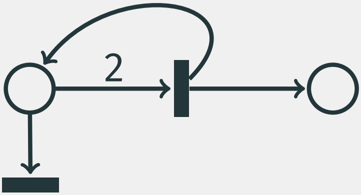
Input: Petri net $\mathcal{N}$, initial marking $\boldsymbol{m}_0$, target marking $\boldsymbol{m}$

Question: Is some $\boldsymbol{m}' \geq \boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$?

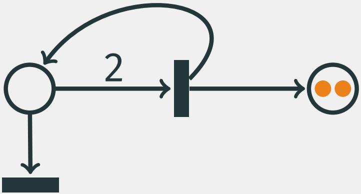
How to solve it?

- Forward: build reachability tree from initial marking
- Backward: find predecessors of markings covering target
- EXPSPACE-complete

Backward algorithm

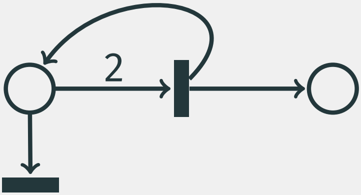


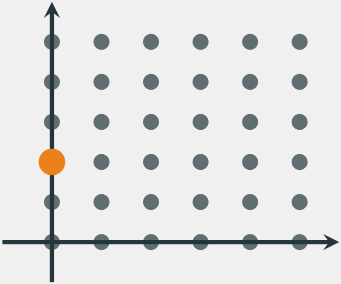
Backward algorithm



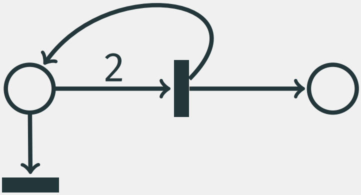
What initial markings may cover $(0, 2)$?

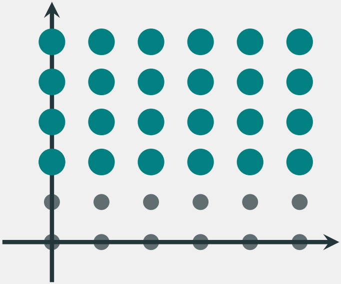
Backward algorithm



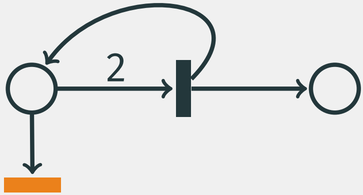


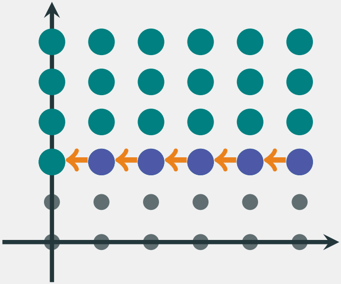
Backward algorithm



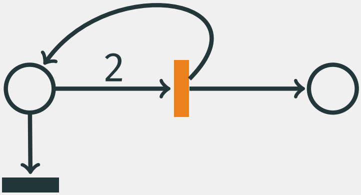


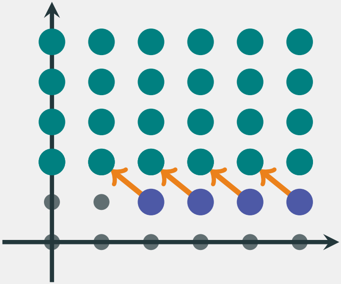
Backward algorithm



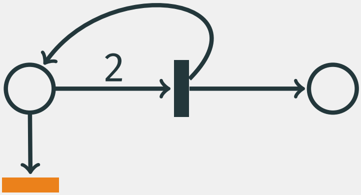


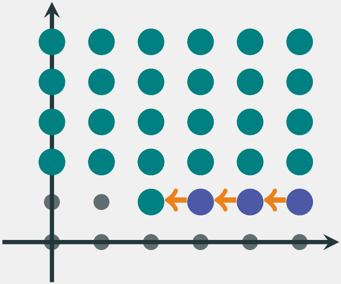
Backward algorithm



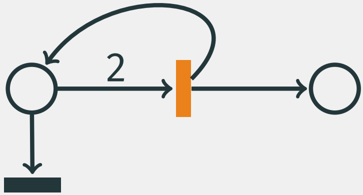


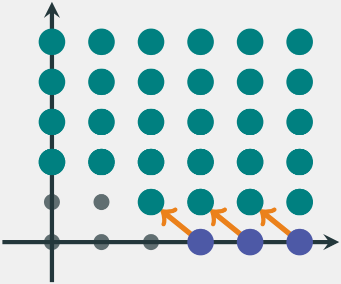
Backward algorithm



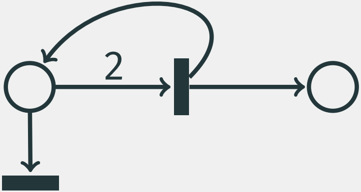


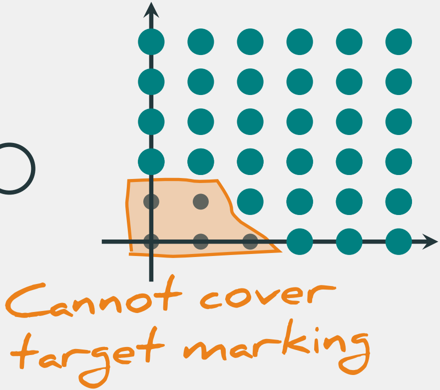
Backward algorithm



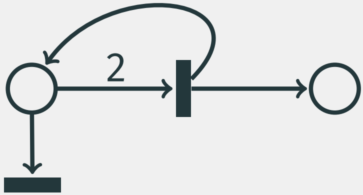


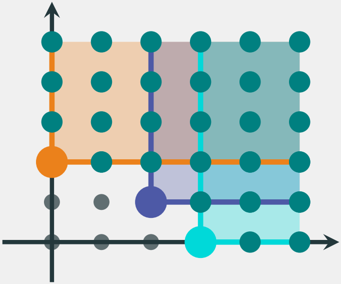
Backward algorithm





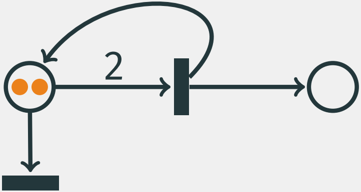
Backward algorithm

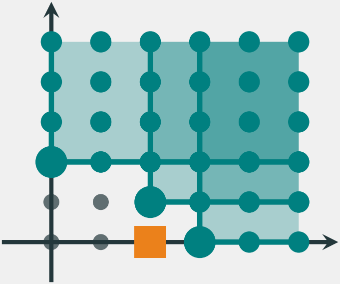




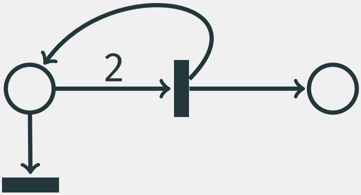
Basis size may become doubly exponential

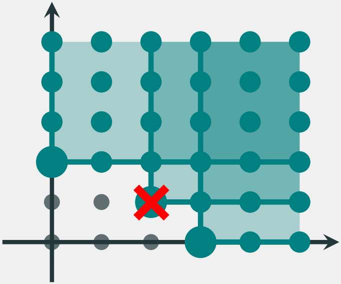
(Bozzelli & Ganty '11)





We only care about some initial marking...

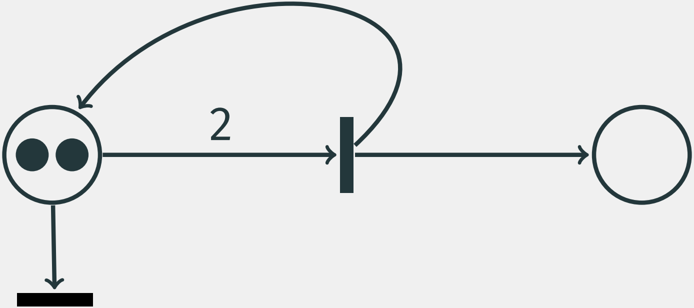




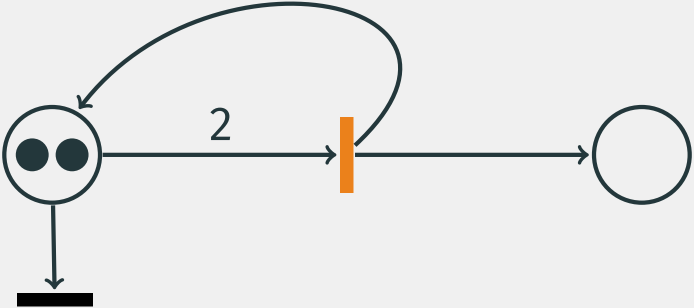
We only care about some initial marking...

Speedup by pruning basis!

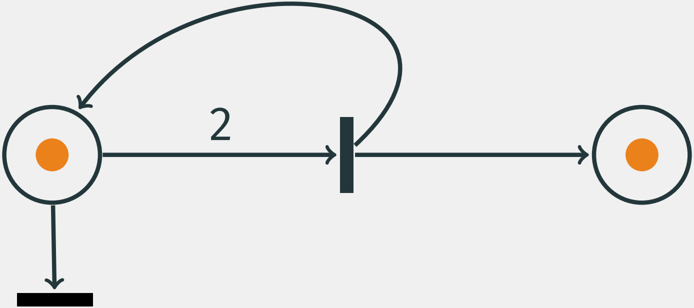
(Discrete) Petri nets



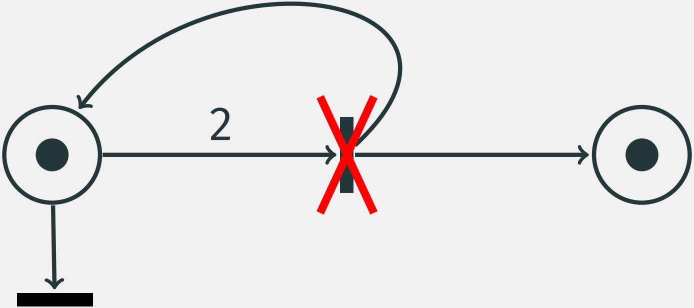
(Discrete) Petri nets



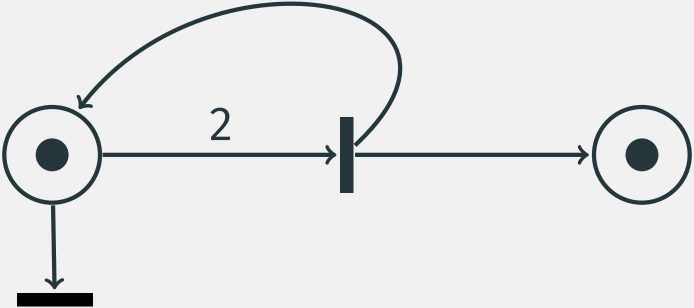
(Discrete) Petri nets



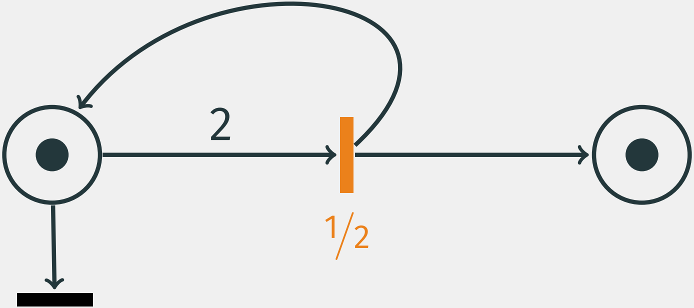
(Discrete) Petri nets



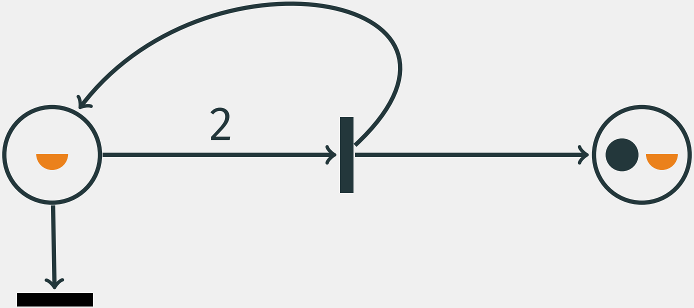
~~(Discrete)~~ Continuous Petri nets



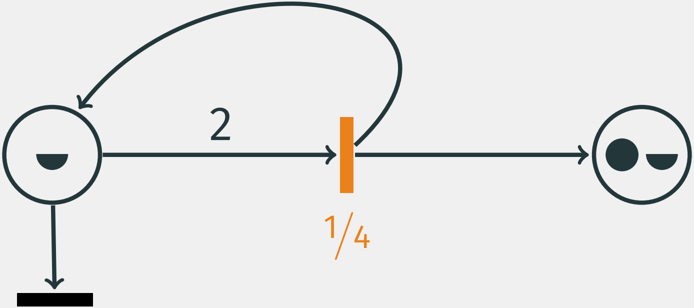
~~(Discrete)~~ Continuous Petri nets



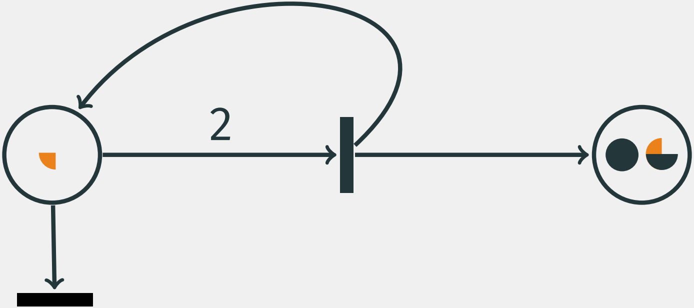
~~(Discrete)~~ Continuous Petri nets



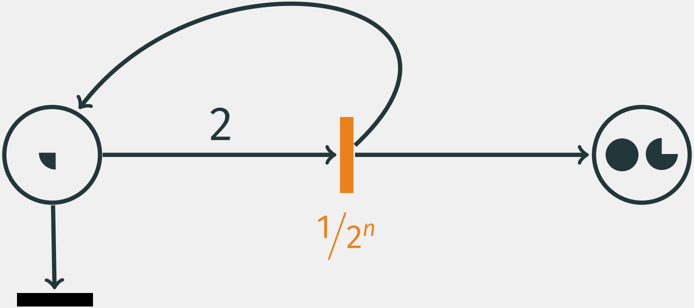
~~(Discrete)~~ Continuous Petri nets



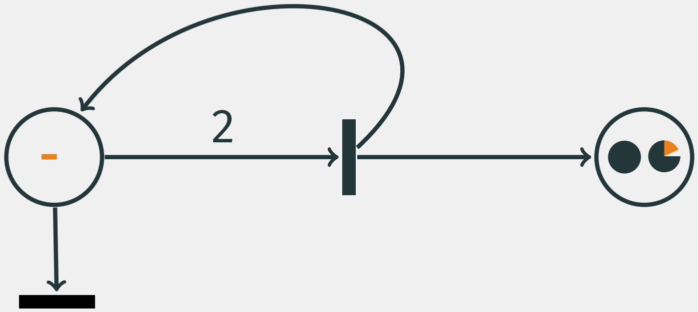
~~(Discrete)~~ Continuous Petri nets



~~(Discrete)~~ Continuous Petri nets



~~(Discrete)~~ Continuous Petri nets



Continuity to over-approximate coverability

$$\boldsymbol{m} \text{ is coverable from } \boldsymbol{m}_0$$

$$\Downarrow$$

$$\boldsymbol{m} \text{ is } \mathbb{Q}\text{-coverable from } \boldsymbol{m}_0$$

$\boldsymbol{m}$ is coverable from $\boldsymbol{m}_0$

EXPSPACE

$$\Downarrow$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$

PTIME

$$\Downarrow \quad \not\Uparrow$$

$\boldsymbol{m}_0$ and $\boldsymbol{m}$ satisfy conditions of
Esparza, Ledesma-Garza, Majumdar, Meyer & Niksic '14

NP / EXPTIME

Continuity to over-approximate coverability

$\boldsymbol{m}$ is not coverable from $\boldsymbol{m}_0$

Safety

$$\Uparrow$$

$\boldsymbol{m}$ is not $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$

Coverability in continuous Petri nets

Fix some continuous Petri net $(P, T, \mathbf{Pre}, \mathbf{Post})$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

Coverability in continuous Petri nets

Fix some continuous Petri net $(P, T, \mathbf{Pre}, \mathbf{Post})$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$

Coverability in continuous Petri nets

Fix some continuous Petri net $(P, T, \mathbf{Pre}, \mathbf{Post})$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m}_0 + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets

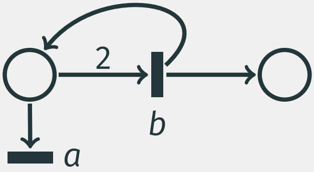
Fix some continuous Petri net $(P, T, \mathbf{Pre}, \mathbf{Post})$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets



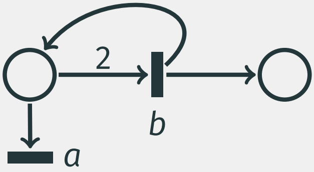
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} \;\; = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets



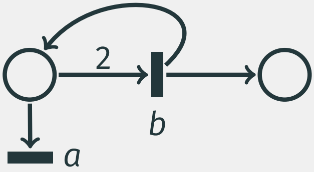
$\boldsymbol{m}_0 = (2, 0)$
$\boldsymbol{m} \;\; = (0, 2)$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2$
  $2 \leq \boldsymbol{v}_b$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets



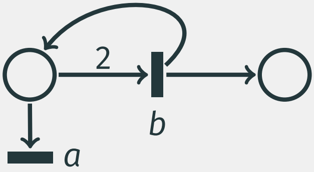
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} \;\; = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets

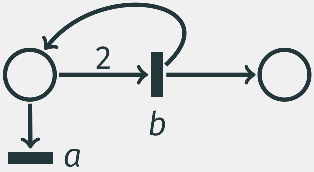


$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$



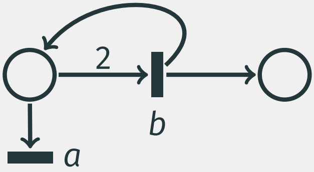
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} \; = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in \{a, b\} : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets



$\boldsymbol{m}_0 = (2, 0)$

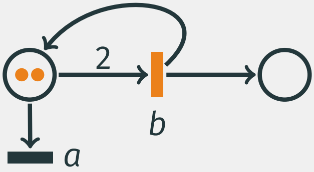
$\boldsymbol{m} \;\; = (0, 2)$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$

Coverability in continuous Petri nets



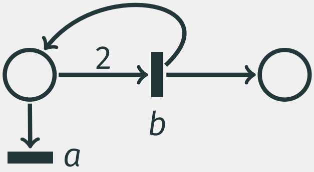
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{matrix} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{matrix} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$

Coverability in continuous Petri nets



$\boldsymbol{m}_0 = (2, 0)$

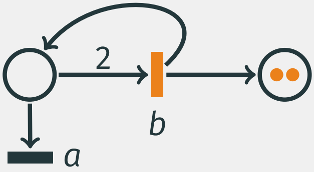
$\boldsymbol{m} \ \ = (0, 2)$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$ ✓
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$

Coverability in continuous Petri nets



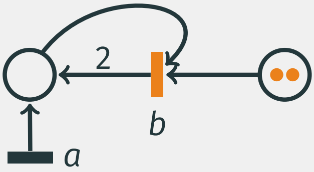
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$ ✓
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$

Coverability in continuous Petri nets



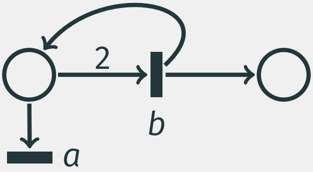
$$\boldsymbol{m}_0 = (2, 0)$$
$$\boldsymbol{m} = (0, 2)$$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{array}{l} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{array} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$ ✓
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$

Coverability in continuous Petri nets



$\boldsymbol{m}_0 = (2, 0)$

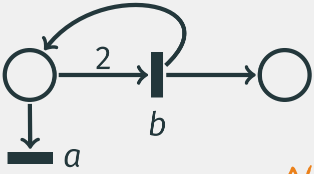
$\boldsymbol{m} \;\; = (0, 2)$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{matrix} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{matrix} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$ ✓
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$ ✗

Coverability in continuous Petri nets



$\boldsymbol{m}_0 = (2, 0)$

$\boxed{\boldsymbol{m} = (0, 2)}$

Not $\mathbb{Q}$-coverable from

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v}_a, \boldsymbol{v}_b \in \mathbb{Q}_{\geq 0}$ such that

- $\begin{matrix} 0 \leq \boldsymbol{v}_b + \boldsymbol{v}_a \leq 2 \\ 2 \leq \boldsymbol{v}_b \end{matrix} \implies \boldsymbol{v}_a = 0,\ \boldsymbol{v}_b = 2,\ \boldsymbol{m}' = \boldsymbol{m}$ ✓
- some execution from $\boldsymbol{m}_0$ fires exactly $\{b\}$ ✓
- some execution to $\boldsymbol{m}'$ fires exactly $\{b\}$ ✗

Coverability in continuous Petri nets

Polynomial time!

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets

Logical characterization **TACAS'16**

$\mathbb{Q}$-coverability can be encoded in a linear size formula of

existential $\text{FO}(\mathbb{Q}_{\geq 0}, +, <)$

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... **Fraca & Haddad '13**

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets

Logical characterization TACAS'16

$\mathbb{Q}$-coverability can be encoded in a linear size formula of

$$\text{existential FO}(\mathbb{N}, \quad +, <)$$

Even better approximation

$\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff... Fraca & Haddad '13

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}^T_{\geq 0}$ such that

- $\boldsymbol{m}' = \boldsymbol{m}_0 + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Logical characterization TACAS'16

$\mathbb{Q}$-coverability can be encoded in a linear size formula of

$$\text{existential FO}(\mathbb{Q}_{\geq 0}, +, <)$$

m **is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff...** Fraca & Haddad '13

there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}^T_{\geq 0}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\textbf{Post} - \textbf{Pre}) \cdot \boldsymbol{v}$ Straightforward
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Coverability in continuous Petri nets

Logical characterization **TACAS'16**

$\mathbb{Q}$-coverability can be encoded in a linear size formula of

$$\text{existential FO}(\mathbb{Q}_{\geq 0}, +, <)$$

m **is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ iff...** **Fraca & Haddad '13**

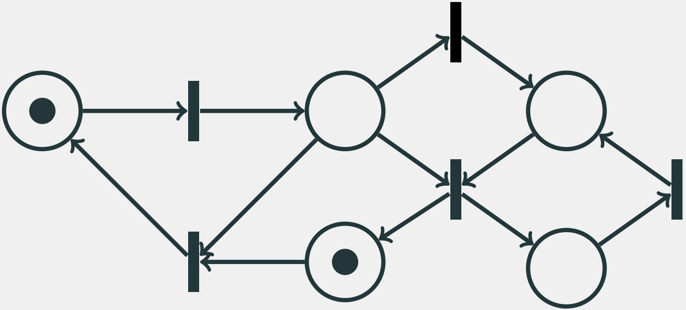
there exist $\boldsymbol{m}' \geq \boldsymbol{m}$ and $\boldsymbol{v} \in \mathbb{Q}_{\geq 0}^{T}$ such that

- $\boldsymbol{m}' = \boldsymbol{m_0} + (\mathbf{Post} - \mathbf{Pre}) \cdot \boldsymbol{v}$

More subtle

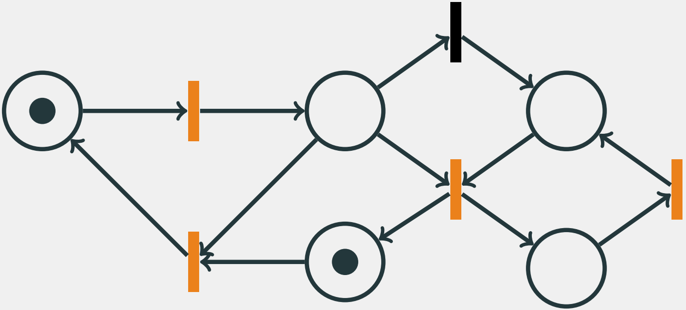
- some execution from $\boldsymbol{m}_0$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$
- some execution to $\boldsymbol{m}'$ fires exactly $\{t \in T : \boldsymbol{v}_t > 0\}$

Encoding the firing set conditions



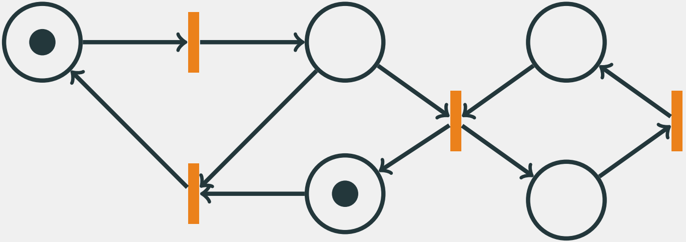
Testing whether some transitions can be fired from initial marking

Encoding the firing set conditions



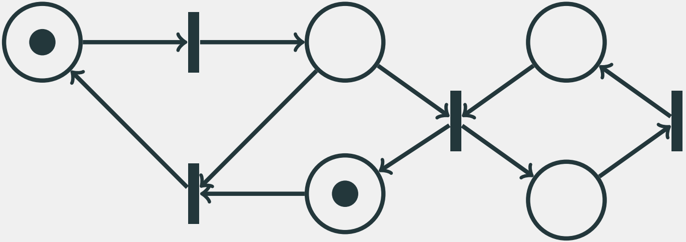
Testing whether some transitions can be fired from initial marking

Encoding the firing set conditions



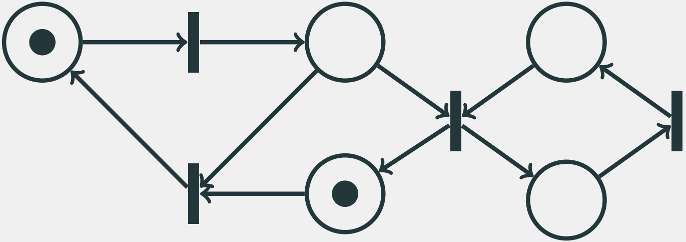
Testing whether some transitions can be fired from initial marking

Encoding the firing set conditions



Simulate a "breadth-first" transitions firing

Encoding the firing set conditions

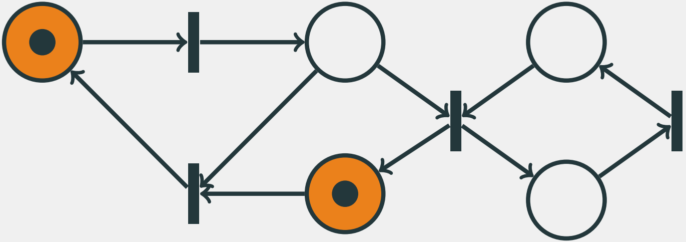


Simulate a "breadth-first" transitions firing

by numbering places/transitions

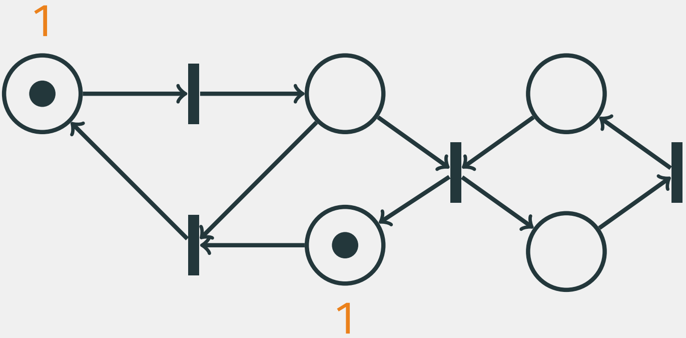
(Verma, Seidl & Schwentick '05)

Encoding the firing set conditions



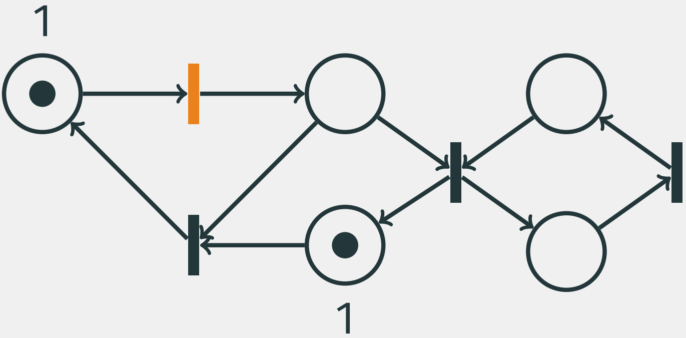
Simulate a "breadth-first" transitions firing
by numbering places/transitions
(Verma, Seidl & Schwentick '05)

Encoding the firing set conditions



Simulate a "breadth-first" transitions firing
by numbering places/transitions
(Verma, Seidl & Schwentick '05)

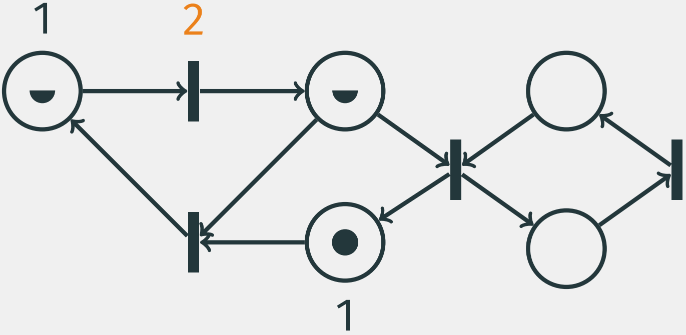
Encoding the firing set conditions



Simulate a "breadth-first" transitions firing by numbering places/transitions

(Verma, Seidl & Schwentick '05)

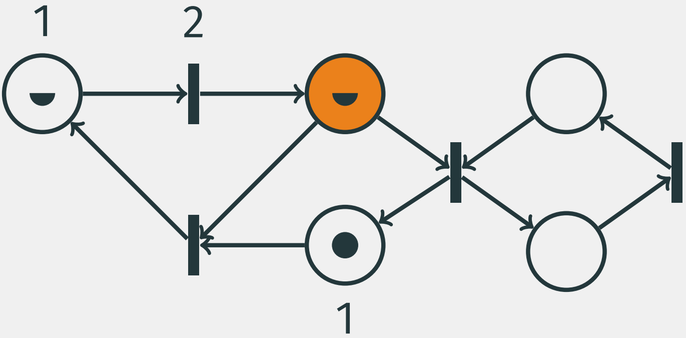
Encoding the firing set conditions



Simulate a "breadth-first" transitions firing by numbering places/transitions

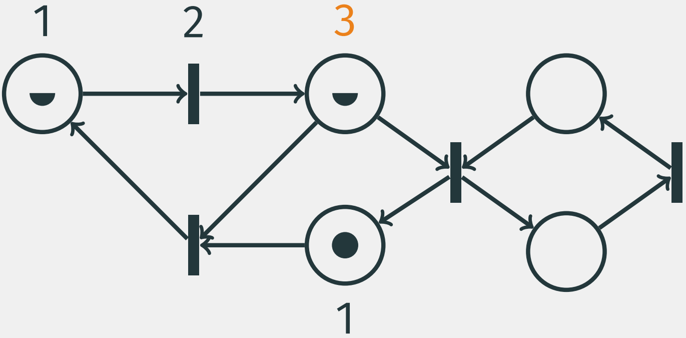
(Verma, Seidl & Schwentick '05)

Encoding the firing set conditions



Simulate a "breadth-first" transitions firing by numbering places/transitions

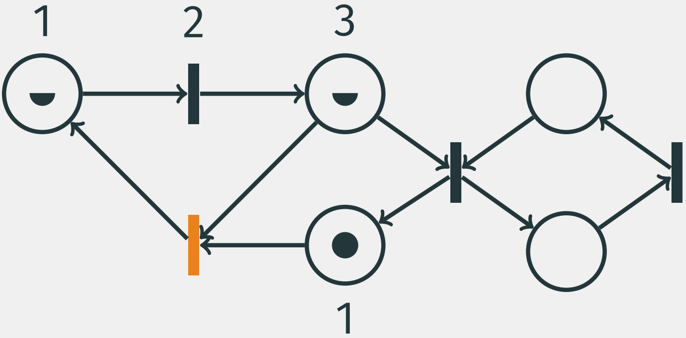
(Verma, Seidl & Schwentick '05)



Simulate a "breadth-first" transitions firing by numbering places/transitions

(Verma, Seidl & Schwentick '05)

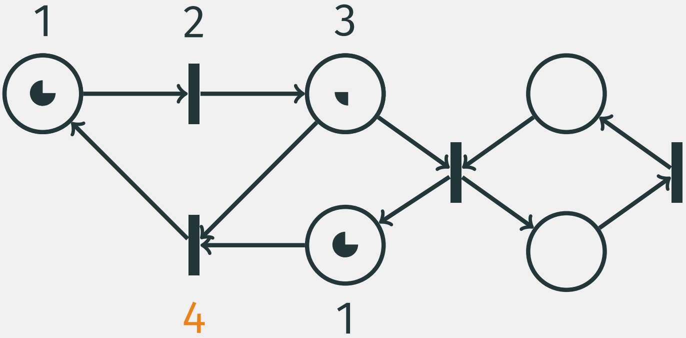
Encoding the firing set conditions



Simulate a "breadth-first" transitions firing by numbering places/transitions

(Verma, Seidl & Schwentick '05)

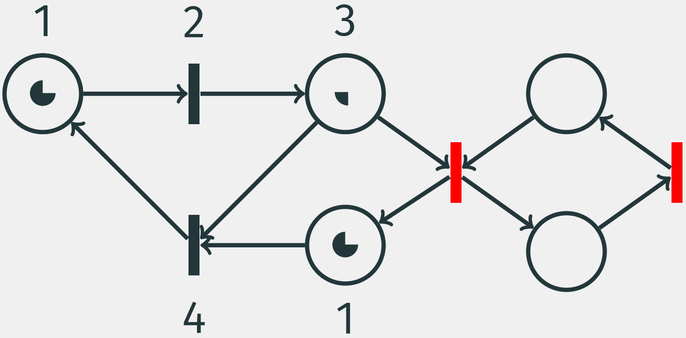
Encoding the firing set conditions



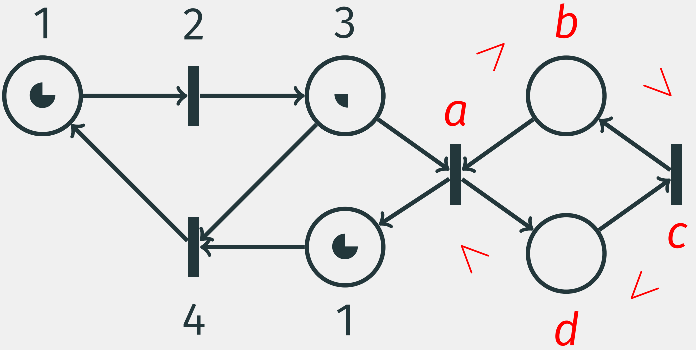
Simulate a "breadth-first" transitions firing by numbering places/transitions

(Verma, Seidl & Schwentick '05)

Encoding the firing set conditions



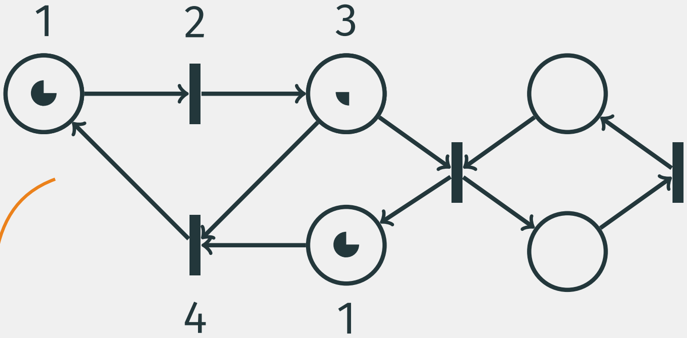
Simulate a "breadth-first" transitions firing
by numbering places/transitions
(Verma, Seidl & Schwentick '05)



Simulate a "breadth-first" transitions firing by numbering places/transitions

(Verma, Seidl & Schwentick '05)

Encoding the firing set conditions



$$\varphi(\boldsymbol{x}) = \exists \boldsymbol{y} : \bigwedge_{p \in P} \boldsymbol{y}(p) > 0 \rightarrow \bigwedge_{t \in {}^{\bullet}p} \boldsymbol{y}(t) < \boldsymbol{y}(p) \cdots$$

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
```

Polynomial time

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
```

$X = \{\text{target marking } \boldsymbol{m}\}$

`while` (initial marking $\boldsymbol{m}_0$ not covered by X):

  $B =$ markings obtained from X one step backward

  $B = B \setminus \{\boldsymbol{b} \in B : \neg\varphi(\boldsymbol{b})\}$

  `if` $B = \emptyset$: `return False`

  $\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{x}) \ \wedge \ \bigwedge_{\text{pruned } \boldsymbol{b}} \boldsymbol{x} \not\geq \boldsymbol{b}$

  $X = X \cup B$

`return True`

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

Backward coverability modulo $\mathbb{Q}$-coverability

`if` target marking $\boldsymbol{m}$ is not $\mathbb{Q}$-coverable`:`

  `return False`

$X = \{\text{target marking } \boldsymbol{m}\}$

`while` (initial marking $\boldsymbol{m}_0$ not covered by X)`:`

  $B =$ markings obtained from X one step backward

  $B = B \setminus \{\boldsymbol{b} \in B : \neg\varphi(\boldsymbol{b})\}$ $\mathbb{Q}$-coverability pruning (better than poly. time)

  `if` $B = \emptyset$`: return False`

  $\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{x}) \ \wedge \ \bigwedge_{\text{pruned } \boldsymbol{b}} \boldsymbol{x} \not\geq \boldsymbol{b}$

  $X = X \cup B$

`return True`

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
```

$X = \{\text{target marking } \boldsymbol{m}\}$

`while` (initial marking $\boldsymbol{m}_0$ not covered by X):

$B =$ markings obtained from X one step backward

$B = B \setminus \{\boldsymbol{b} \in B : \neg\varphi(\boldsymbol{b})\}$

`if` $B = \emptyset$: `return False`

$\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{x}) \ \wedge \ \bigwedge_{\text{pruned } \boldsymbol{b}} \boldsymbol{x} \not\geq \boldsymbol{b}$

$X = X \cup B$

`return True`

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

SMT solver guidance

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

Backward coverability modulo $\mathbb{Q}$-coverability

```
if target marking m is not ℚ-coverable:
  return False
X = {target marking m}
while (initial marking m₀ not covered by X):
  B = markings obtained from X one step backward
  B = B \ {b ∈ B : ¬φ(b)}
  if B = ∅: return False
  φ(x) = φ(x) ∧ ⋀_{pruned b} x ≱ b
  X = X ∪ B
return True
```

An implementation: QCover

python™ + SMT solver Z3 (Microsoft Research)

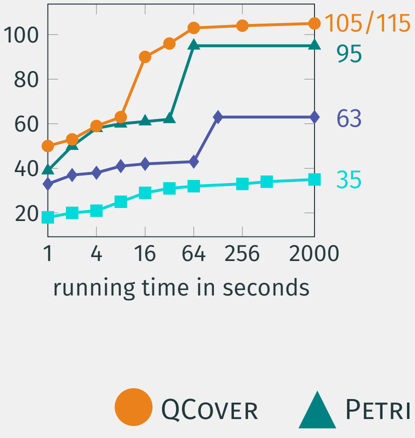
`https://github.com/blondimi/qcover`

Tested on...

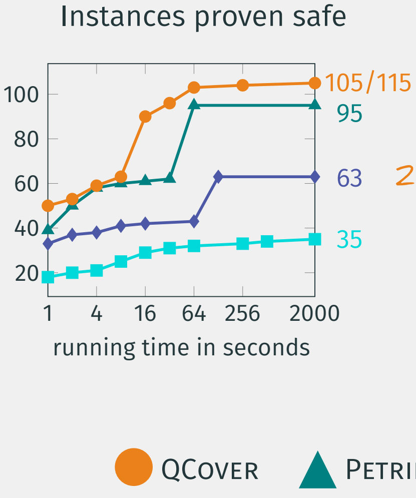
- 176 Petri nets (avg. 1054 places, 8458 transitions)
- C/Erlang programs with threads
- Mutual exclusion protocols, communication protocols, etc.
- Message analysis of a medical and a bug tracking system

An implementation: QCover

Instances proven safe



An implementation: QCover



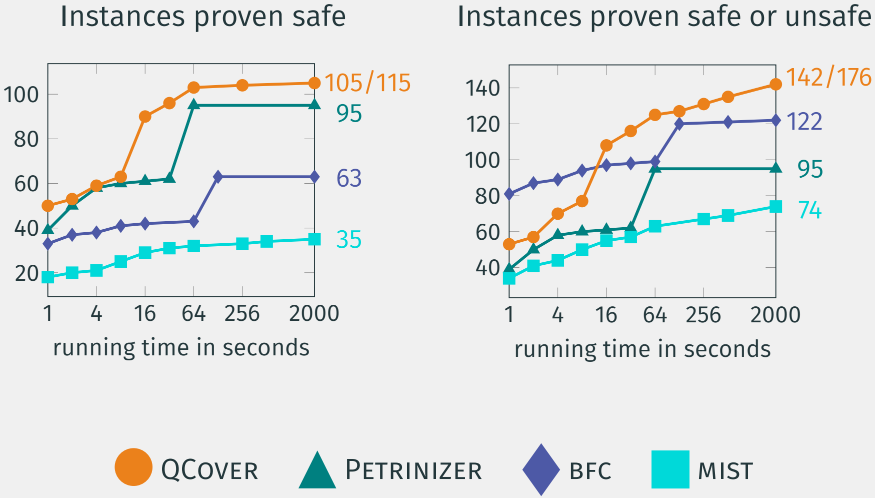
Largest nets proved safe:

21143 places
7150 trans.
42 secs.

6690 places
11934 trans.
21 secs.

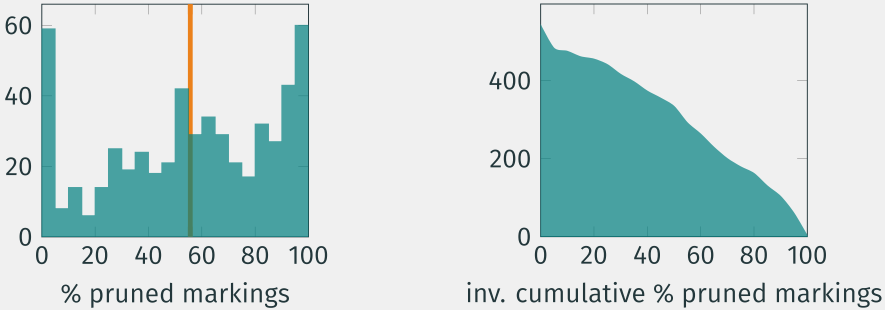
754 places
27370 trans.
3 secs.

An implementation: QCover



An implementation: QCover

Markings pruning efficiency across all iterations

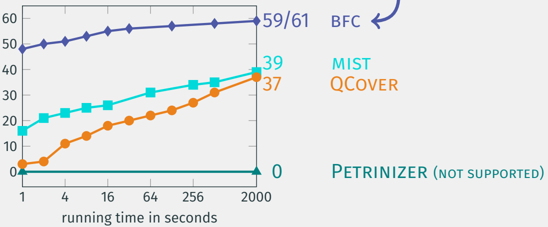


Possible extensions

- Combine our approach with a forward algorithm to better handle unsafe instances

Possible extensions

- Combine our approach with a forward algorithm to better handle unsafe instances



Possible extensions

- Combine our approach with a forward algorithm to better handle unsafe instances
- Use more efficient data structures, *e.g.* sharing trees

(Delzanno, Raskin & Van Begin '04)

Possible extensions

- Combine our approach with a forward algorithm to better handle unsafe instances

- Use more efficient data structures, *e.g.* sharing trees

  (Delzanno, Raskin & Van Begin '04)

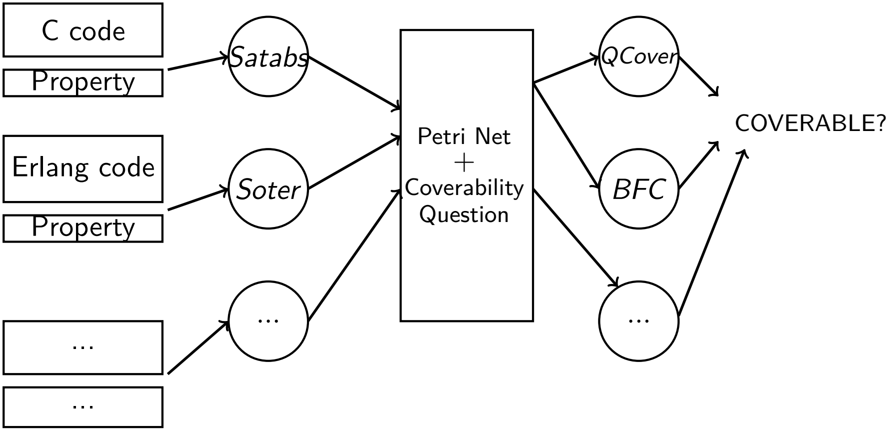
- Support Petri nets extensions

Part II: ICover

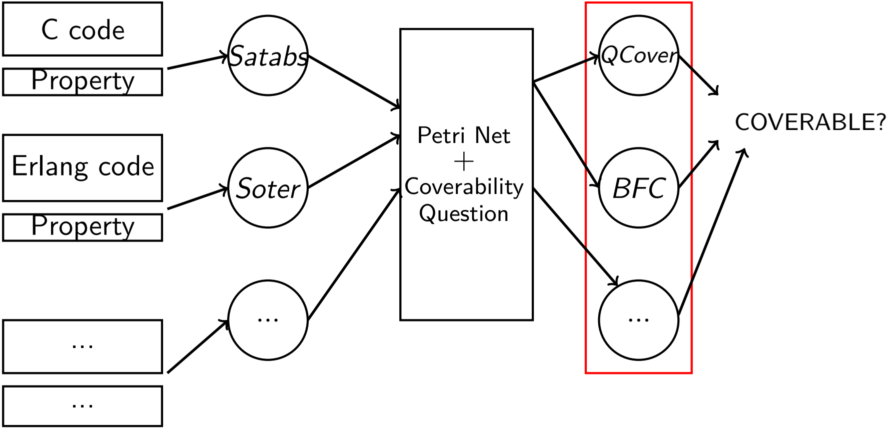
Grégoire Sutre

Joint work with Thomas Geffroy and Jérôme Leroux

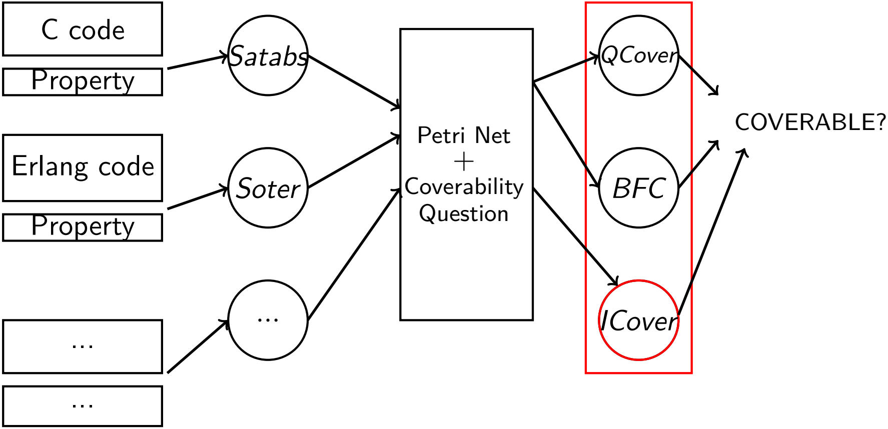
Verifying Systems with Petri Nets



Verifying Systems with Petri Nets



Verifying Systems with Petri Nets



Coverability in Petri nets

$$\mathit{init} \xrightarrow{*} m \geq \mathit{target}?$$

Decidability - Complexity

- Decidable (Karp and Miller - 1969)
- $\textsc{ExpSpace}$-complete (Lipton - 1976, Rackoff - 1978)

Coverability in Petri nets

$$init \xrightarrow{*} m \geq target?$$

Tools

Mist (Ganty, Geeraerts, Raskin, Van Begin, . . .)

- interval sharing trees
- backward search + place invariants
- abstraction refinement

BFC (Kaiser, Kroening, Wahl)

Target set widening + forward Karp-Miller

Petrinizer (Esparza, Ledesma-Garza, Majumdar, Meyer, Niksic)

SMT, state equation + traps

QCover (Blondin, Finkel, Haase, Haddad)

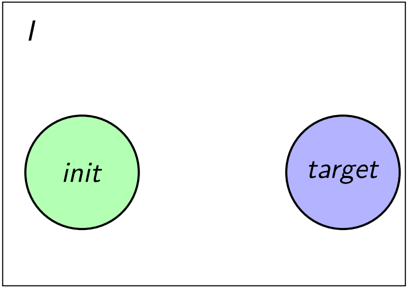
SMT, continuous reachability + backward search

ICover: Generalisation of *QCover* with Invariants

Assumption:

1. I is an invariant (I contains all reachable markings)
2. I is a downward closed set

$U_0 := \uparrow(target \cap I)$

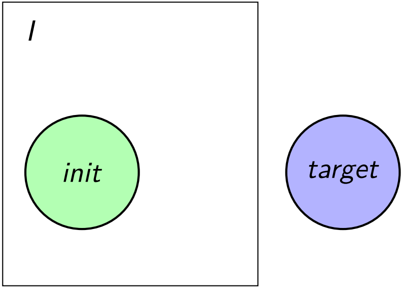


ICover: Generalisation of *QCover* with Invariants

Assumption:

1. I is an invariant (I contains all reachable markings)
2. I is a downward closed set

$U_0 := \emptyset$: Safe !



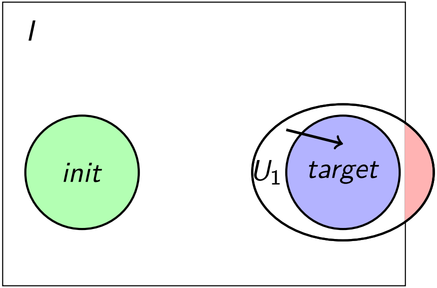
ICover: Generalisation of *QCover* with Invariants

Assumption:

1. I is an invariant (I contains all reachable markings)
2. I is a downward closed set

$U_0 := \uparrow(target \cap I)$

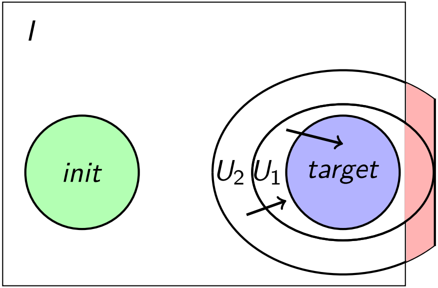
$U_1 := U_0 \cup \uparrow(pre(U_0) \cap I)$



ICover: Generalisation of *QCover* with Invariants

Assumption:

1. I is an invariant (I contains all reachable markings)
2. I is a downward closed set



$U_0 := \uparrow(target \cap I)$

$U_1 := U_0 \cup \uparrow(pre(U_0) \cap I)$
$U_2 := U_1 \cup \uparrow(pre(U_1) \cap I)$

ICover: Generalisation of *QCover* with Invariants

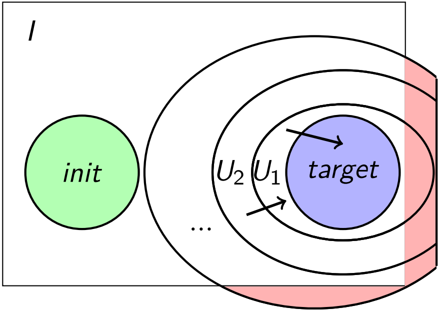
Assumption:

1. I is an invariant (I contains all reachable markings)
2. I is a downward closed set

$U_0 := \uparrow(\mathit{target} \cap I)$

$U_1 := U_0 \cup \uparrow(\mathit{pre}(U_0) \cap I)$
$U_2 := U_1 \cup \uparrow(\mathit{pre}(U_1) \cap I)$
$\ldots$
$U_{k+1} := U_k \cup \uparrow(\mathit{pre}(U_k) \cap I)$

Always terminates
(Dickson's lemma)



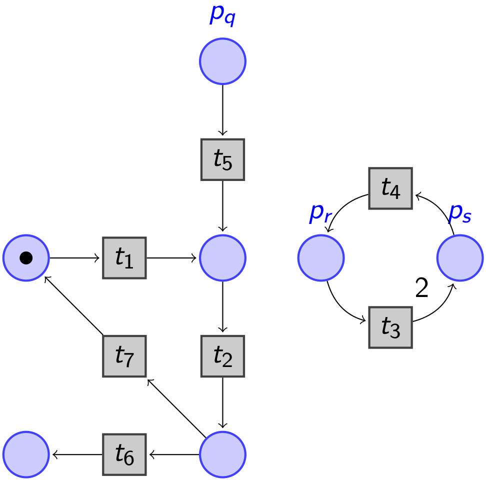
$U_{n+1} = U_n$

Backward Algorithm with Invariant-Based Pruning

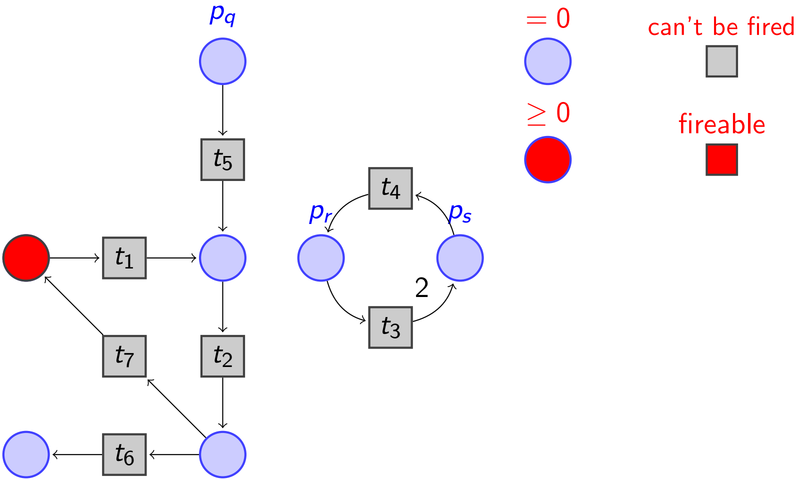
if $target \in I$ **then**
  $B \leftarrow \{target\}$;
else
  return `False`;
end
while $m_{init} \notin \uparrow B$ **do**
  $N \leftarrow min(pre(\uparrow B)) \setminus \uparrow B$
  $P \leftarrow N \cap I$
  if $P = \emptyset$ **then**
    return `False`;
  end
  $B \leftarrow \min(B \cup P)$;
end
return `True`;

- I is an invariant
- I is a downward closed set

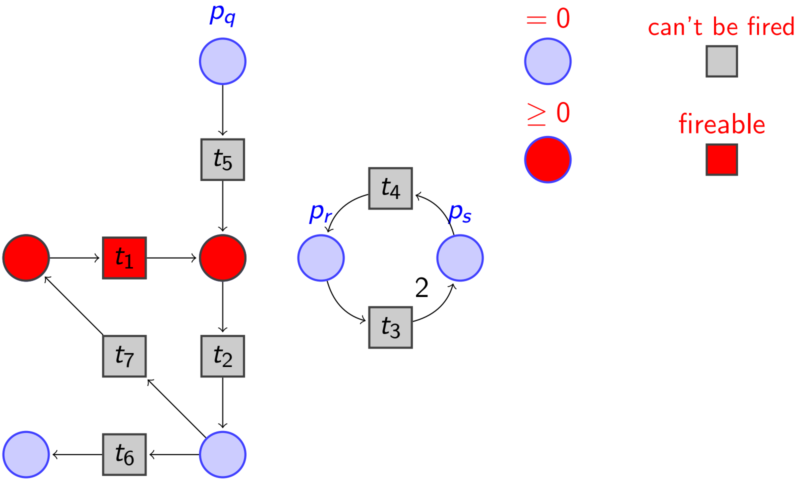
Invariant: Sign Analysis



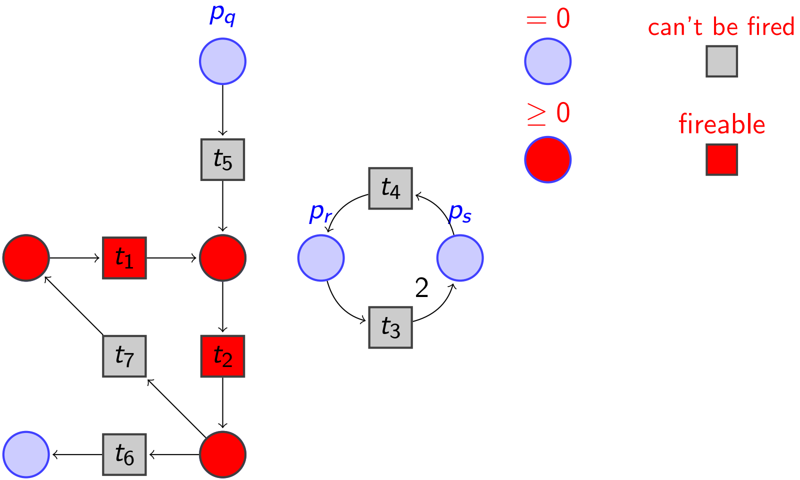
Invariant: Sign Analysis



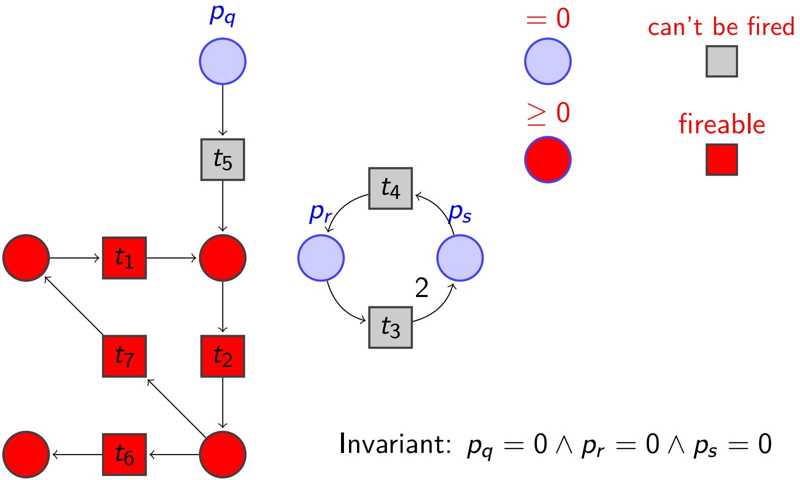
Invariant: Sign Analysis



Invariant: Sign Analysis

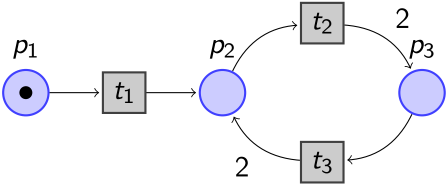


Invariant: Sign Analysis



Invariant: $p_q = 0 \wedge p_r = 0 \wedge p_s = 0$

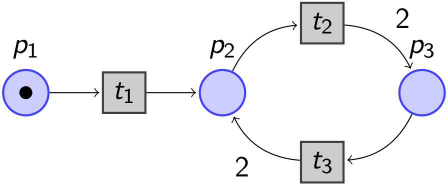
Invariant: State Inequation



$$p_1 \xrightarrow{t_1} \xrightarrow{t_2} \xrightarrow{t_3} \dots \xrightarrow{t_2} \xrightarrow{t_3} r$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

Invariant: State Inequation



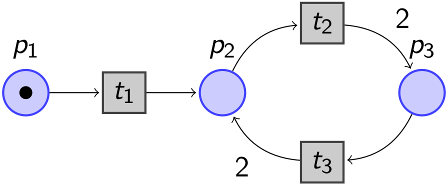
$$p_1 \xrightarrow{t_1}\xrightarrow{t_2}\xrightarrow{t_3} \dots \xrightarrow{t_2}\xrightarrow{t_3} r$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

x_i: number of occurrences of t_i

$$r = init + x_1\Delta(t_1) + x_2\Delta(t_2) + x_3\Delta(t_3)$$

Invariant: State Inequation



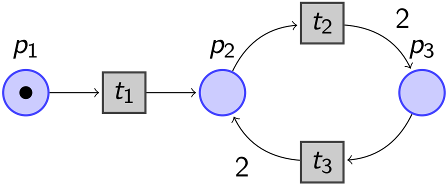
$$p_1 \xrightarrow{t_1}\xrightarrow{t_2}\xrightarrow{t_3} \dots \xrightarrow{t_2}\xrightarrow{t_3} r \geq m$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

x_i: number of occurrences of t_i

$$m \leq r = init + x_1\Delta(t_1) + x_2\Delta(t_2) + x_3\Delta(t_3)$$

Invariant: State Inequation



$$p_1 \xrightarrow{t_1}\xrightarrow{t_2}\xrightarrow{t_3} \dots \xrightarrow{t_2}\xrightarrow{t_3} r \geq m$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

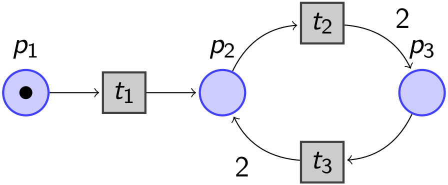
x_i: number of occurrences of t_i

$$m \leq r = init + x_1\Delta(t_1) + x_2\Delta(t_2) + x_3\Delta(t_3)$$

Invariant

$I := \{m \mid \exists x, init + \sum_{t \in T} x(t)\Delta(t) \geq m\}$

Invariant: State Inequation



$$p_1 \xrightarrow{t_1} \xrightarrow{t_2} \xrightarrow{t_3} \ldots \xrightarrow{t_2} \xrightarrow{t_3} r \geq m$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

x_i: number of occurrences of t_i

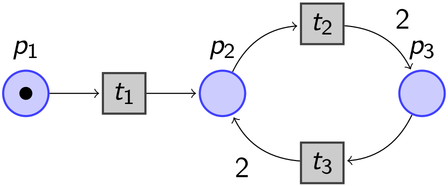
$$m \leq r = init + x_1\Delta(t_1) + x_2\Delta(t_2) + x_3\Delta(t_3)$$

$$\begin{cases} x_1, x_2, x_3 \geq 0 \\ m(p_1) \leq 1 - x_1 \\ m(p_2) \leq x_1 - x_2 + 2x_3 \\ m(p_3) \leq 2x_2 - x_3 \end{cases}$$

Invariant

$I := \{m \mid \exists x, init + \sum_{t \in T} x(t)\Delta(t) \geq m\}$

Invariant: State Inequation



$$p_1 \xrightarrow{t_1}\xrightarrow{t_2}\xrightarrow{t_3} \dots \xrightarrow{t_2}\xrightarrow{t_3} r \geq m$$

$$\Delta(t_1) = p_2 - p_1$$
$$\Delta(t_2) = 2p_3 - p_2$$
$$\Delta(t_3) = 2p_2 - p_3$$

x_i: number of occurrences of t_i

$$p_1 \leq 1 \qquad m \leq r = init + x_1\Delta(t_1) + x_2\Delta(t_2) + x_3\Delta(t_3)$$

Invariant

$I := \{m \mid \exists x, init + \sum_{t \in T} x(t)\Delta(t) \geq m\}$

Experimentations

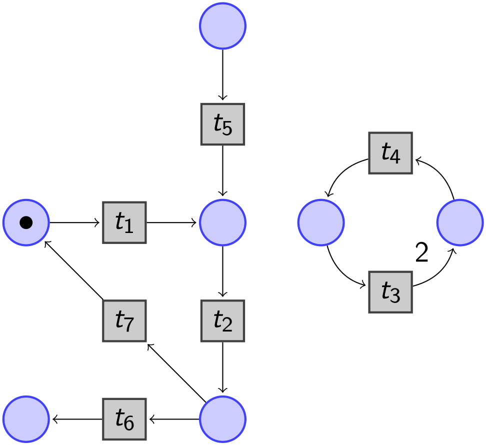
New Tool: *ICover*

- Based on *QCover* written in Python (~900 lines of codes)
- Both use the SMT-Solver *z3* (Bjorner et al. - 2007)
- *ICover* available as a patch of *QCover* (~400 lines of codes)
- `dept-info.labri.u-bordeaux.fr/~tgeffroy/icover`

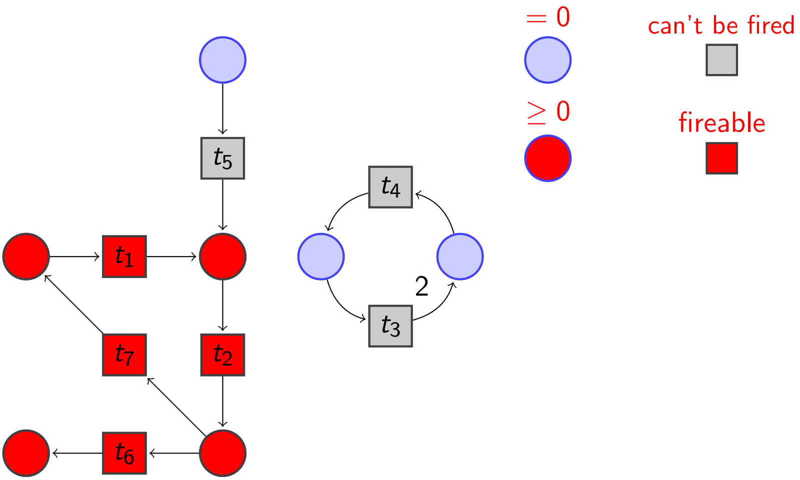
Results

- Benchmarks (176 instances) used by *QCover* and others
- *QCover* solved 106 / 115 safe instances (2000 seconds per instance)
- *QCover* solved 37 / 61 unsafe instances (idem)
- *ICover* solved as much safe instances and one more unsafe
- It works ! 10 000 seconds (*QCover*) to 5 000 seconds (*ICover*)

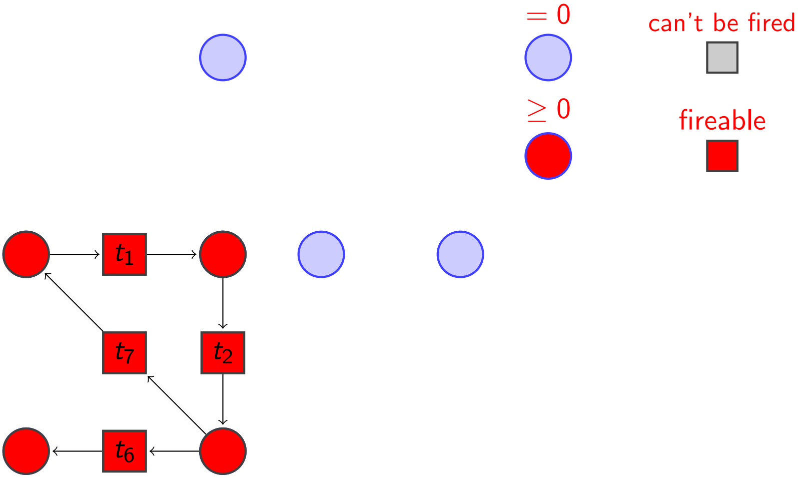
Experimentations: Sign Analysis As A Pre-processing



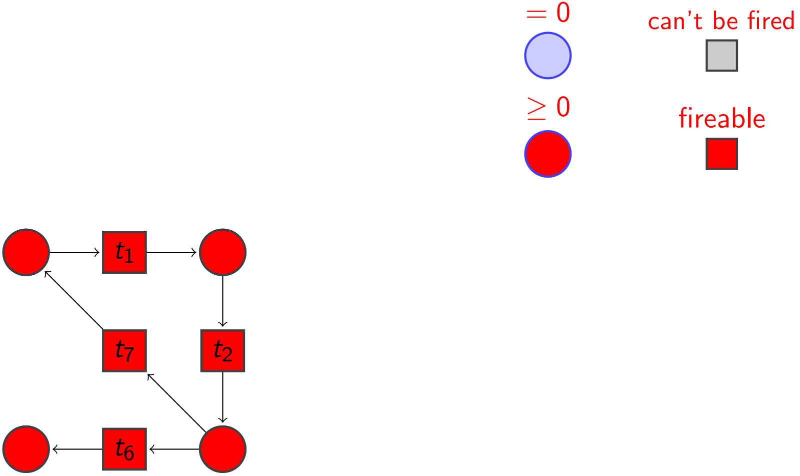
Experimentations: Sign Analysis As A Pre-processing



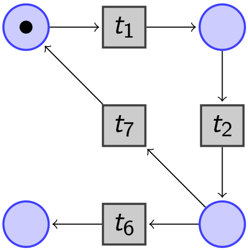
Experimentations: Sign Analysis As A Pre-processing



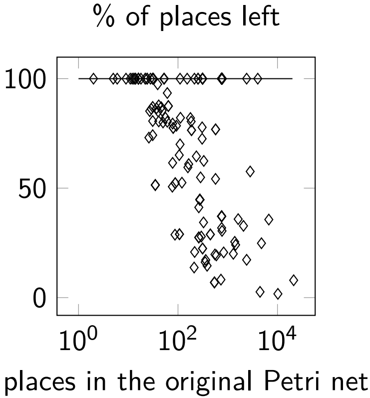
Experimentations: Sign Analysis As A Pre-processing

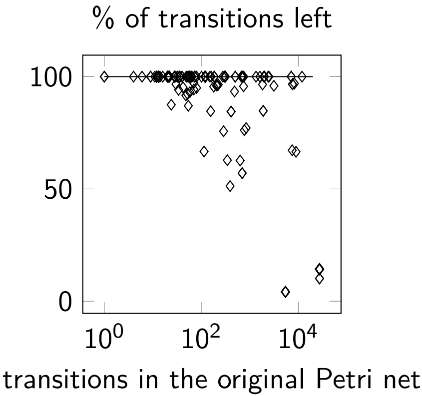


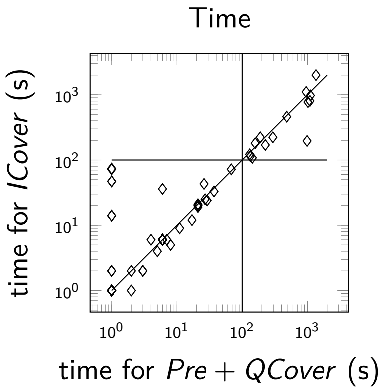
Experimentations: Sign Analysis As A Pre-processing

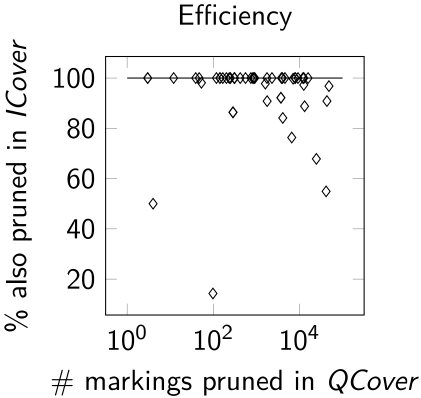


Results of Pre-processing

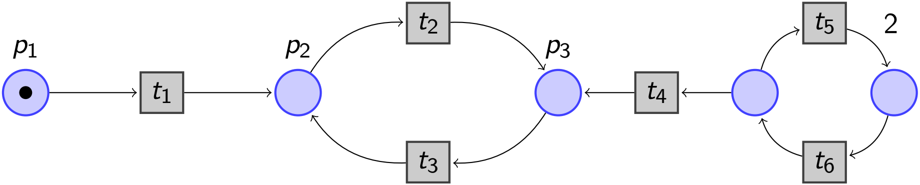





Time
time for ICover (s)
10^0
10^1
10^2
10^3
time for $Pre + QCover$ (s)

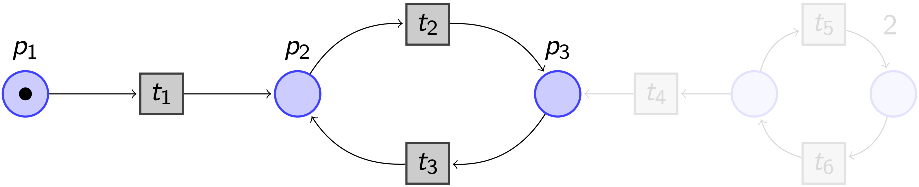

Efficiency
% also pruned in ICover
100
80
60
40
20
10^0
10^2
10^4
markings pruned in QCover

State Inequation More Precise with Pre-Processing



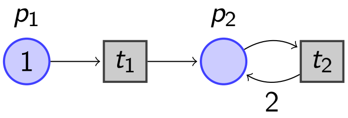
- Can't cover $p_1 + p_2 + p_3$ from p_1
- State inequation: $p_1 \leq 1$ not precise enough
- State inequation: $p_1 + p_2 + p_3 \leq 1$ precise enough

State Inequation More Precise with Pre-Processing



- Can't cover $p_1 + p_2 + p_3$ from p_1
- State inequation: $p_1 \leq 1$ not precise enough
- State inequation: $p_1 + p_2 + p_3 \leq 1$ precise enough

State Inequation vs $\dashrightarrow$

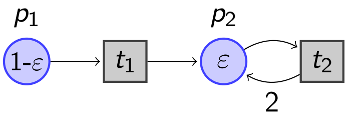


- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \ldots$ such that init $\overset{*}{\dashrightarrow} m_k$ for every k and such that $m_0, m_1, \ldots$ converges toward m'.

State Inequation vs $\dashrightarrow$

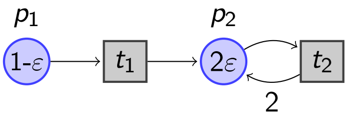


- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \ldots$ such that init $\overset{*}{\dashrightarrow}$ m_k for every k and such that $m_0, m_1, \ldots$ converges toward m'.

State Inequation vs $\dashrightarrow$

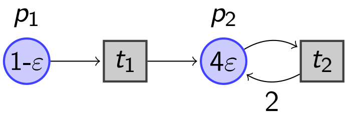


- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \ldots$ such that init $\overset{*}{\dashrightarrow}$ m_k for every k and such that $m_0, m_1, \ldots$ converges toward m'.

State Inequation vs $\dashrightarrow$

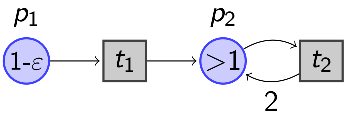


- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \ldots$ such that init $\overset{*}{\dashrightarrow}$ m_k for every k and such that $m_0, m_1, \ldots$ converges toward m'.

State Inequation vs $\dashrightarrow$

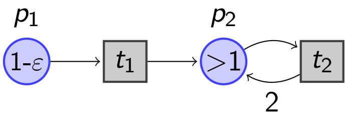


- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \dots$ such that init $\overset{*}{\dashrightarrow}$ m_k for every k and such that $m_0, m_1, \dots$ converges toward m'.

State Inequation vs $\dashrightarrow$



- $p_1 + p_2$ not coverable from p_1 with $\dashrightarrow$
- $p_1 + p_2$ satisfy the state inequation: $p_1 \leq 1$

Theorem (Recalde, Teruel and Silva - 1999)

In a pre-processed Petri net, m satisfies the state inequation iff there exists $m' \geq m$ and a sequence $m_0, m_1, \ldots$ such that $init \overset{*}{\dashrightarrow} m_k$ for every k and such that $m_0, m_1, \ldots$ converges toward m'.

Conclusion

New

- Backward coverability algorithm with invariant-based pruning
- Pre-processing is a cheap way to accelerate verification
- In practice, in a pre-processed Petri net, state inequation is almost as good as $\dashrightarrow$ coverability

Future work

- Find other cheap pre-processings and invariants
- Apply to other classes of well-structured transition systems

Part III: Best practices

Christoph Haase

Tools...

- increase visibility outside your peer group
- help understanding what is relevant to other people
- generate feedback for theoretical work
- can convince reviewers
- attract students

Before you start

- Choice of language
  - interpreted vs. compiled
  - statically vs. dynamically typed
- Bindings for SMT solver
- Performance of memory operations

Software engineering aspects

- Object oriented programming
- Unit tests
- Documentation
- Use profilers to find bottlenecks

Benchmarks

- One of the most important aspects
- Use other people's benchmarks
- Contact authors if necessary
- Pitfalls:
    - Parsing can entail large costs
    - Avoid unfair treatment of competitors
    - Choose evaluation metrics wisely

Availability

- Obtain institutional clearance $\in \mathbf{F}_\omega$
- Choose license: BSD preferred by industry
- Use public code repositories, e.g. GitHub

Future

- Identify relevant Petri net subclasses and extensions, e.g.
  - business processes
  - process mining
  - population protocols
  - thread transition systems
- Submit to and integrate into existing software competitions

Final words

The SMT solver is always faster than you!

Thank you!   Diolch!