Robustness Analysis of Time Petri Nets

Étienne André¹, Shweta Garg²

¹LIPN, Université Paris 13, Sorbonne Paris Cité, France
²Dept. of Computer Science, IIT Bombay, Mumbai, India
Context: Verifying Complex Timed Systems (1/2)

- Need for early bug detection
  - Bugs discovered when final testing: expensive
  - Need for thorough modeling and verification
Context: Verifying Complex Timed Systems (1/2)

- Need for early bug detection
  - Bugs discovered when final testing: expensive
  - Need for thorough modeling and verification

- Input

A timed concurrent system
Context: Verifying Complex Timed Systems (1/2)

- Need for early bug detection
  - Bugs discovered when final testing: expensive
  - Need for thorough modeling and verification

- Input

A timed concurrent system

A good behavior expected for the system
**Context: Verifying Complex Timed Systems (1/2)**

- **Need for early bug detection**
  - Bugs discovered when final testing: **expensive**
  - Need for thorough modeling and verification

- **Input**

  A timed concurrent system  
  A good behavior expected for the system

- **Question:** does the system behave well?
Context: Verifying Complex Timed Systems (2/2)

- Use formal methods

A finite model of the system

A formula to be satisfied
Use formal methods

A finite model of the system

A formula to be satisfied

Question: does the model of the system satisfy the formula?
Context: Verifying Complex Timed Systems (2/2)

- Use formal methods

\[ \text{A finite model of the system} \]

\[ \text{A formula to be satisfied} \]

- Question: does the model of the system satisfy the formula?

Yes

No

Counterexample
Motivation: Robustness Analysis

- Timed systems are characterized by a set of timing constants
  - “The packet transmission lasts for 50 ms”
  - “The sensor reads the value every 10 s”

- Verification for one set of constants does not guarantee the correctness for other values

- Challenge: Robustness [Markey, 2011]
  - What happens if 50 is implemented with 49.99?
  - Until which value can we increase or decrease 50 such that the system still behaves well?
Motivation: Robustness Analysis

- Timed systems are characterized by a set of timing constants
  - “The packet transmission lasts for 50 ms”
  - “The sensor reads the value every 10 s”

- Verification for one set of constants does not guarantee the correctness for other values

- Challenge: Robustness [Markey, 2011]
  - What happens if 50 is implemented with 49.99?
  - Until which value can we increase or decrease 50 such that the system still behaves well?

- Parametric analysis
  - Consider that timing constants are parameters
  - Find good values for the parameters, such that the system still behaves well
Outline

1. Parametric Inhibitor Time Petri Nets
2. Robustness Analysis Using the Inverse Method
3. Perspectives
Outline

1. Parametric Inhibitor Time Petri Nets
2. Robustness Analysis Using the Inverse Method
3. Perspectives
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

```
Customer’s coins

Earned coins ← □ ← DVDs available

DVDs on loan
```
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

![Diagram of DVD renting machine](image-url)
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

Customer’s coins

Earned coins

DVDs available

DVDs on loan
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

Customer’s coins

Earned coins

 DVDs available

 DVDs on loan
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

```
Customer’s coins

Earned coins   DVDs available

DVDs on loan
```
**Petri Nets [Petri, 1962]**

- **Advantages of Petri nets**
  - Detailed view of the process with an expressive **graphical representation** based on places and transitions
  - A **formal semantics**
  - Powerful model checking **tools**

- **Example: A DVD renting machine**

![Diagram of DVD renting machine]

- Customer’s coins
- Earned coins
- DVDs available
- DVDs on loan
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

```
Customer's coins

Earned coins ───>DVDs available

DVDs on loan
```
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive graphical representation based on places and transitions
  - A formal semantics
  - Powerful model checking tools

- Example: A DVD renting machine

Customer’s coins

Earned coins

DVDs available

DVDs on loan
Petri Nets [Petri, 1962]

- Advantages of Petri nets
  - Detailed view of the process with an expressive **graphical representation** based on places and transitions
  - A **formal semantics**
  - Powerful model checking **tools**

- Example: A DVD renting machine

Customer’s coins

Earned coins

DVDs available

DVDs on loan
Time Petri Nets With Inhibitor Arcs

- Powerful formalism for verifying real-time systems [Merlin, 1974]
- Transition $t_1$ can be fired from 5 to 6 units of time after it is enabled
- An enabled transition must fire before (or at) its upper bound
  - Except if another transition fires before
- An inhibitor arc ($t_2$) enables its transition once its predecessor place (A) is empty
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

{\text{AB}}
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[ AB \xrightarrow{3.4, t_3} AE \]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[
\begin{align*}
\text{AB} & \xrightarrow{3.4, t_3} \text{AE} & \xrightarrow{2, t_1} \text{CE}
\end{align*}
\]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[ \text{AB} \xrightarrow{3.4, t_3} \text{AE} \xrightarrow{2, t_1} \text{CE} \]

\[ \text{AB} \]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[ \text{AB} \xrightarrow{3,4, t_3} \text{AE} \xrightarrow{2, t_1} \text{CE} \]

\[ \text{AB} \xrightarrow{5, t_1} \text{CB} \]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[
\begin{align*}
\text{AB} & \xrightarrow{3.4, t_3} \text{AE}  & \xrightarrow{2, t_1} \text{CE} \\
\text{AB} & \xrightarrow{5, t_1} \text{CB}  & \xrightarrow{0, t_3} \text{CE}
\end{align*}
\]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[
\begin{align*}
\text{AB} & \quad 3.4, t_3 \quad \text{AE} \quad 2, t_1 \quad \text{CE} \\
\text{AB} & \quad 5, t_1 \quad \text{CB} \quad 0, t_3 \quad \text{CE} \\
\text{AB} & \quad \\
\end{align*}
\]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

\[ \text{AB} \xrightarrow{3.4, t_3} \text{AE} \xrightarrow{2, t_1} \text{CE} \]

\[ \text{AB} \xrightarrow{5, t_1} \text{CB} \xrightarrow{0, t_3} \text{CE} \]

\[ \text{AB} \xrightarrow{5, t_1} \text{CB} \]
Some possible runs

\[ \text{AB} \xrightarrow{3.4, t_3} \text{AE} \xrightarrow{2, t_1} \text{CE} \]

\[ \text{AB} \xrightarrow{5, t_1} \text{CB} \xrightarrow{0, t_3} \text{CE} \]

\[ \text{AB} \xrightarrow{5, t_1} \text{CB} \xrightarrow{0, t_2} \text{CD} \]
Time Petri Nets With Inhibitor Arcs: Example

Some possible runs

- AB → AE → CE, $t_3 = 3.4$
- AB → CB → CE, $t_1 = 5$
- AB → CB → CD, $t_1 = 5$

Set of traces

- AB → AE → CE, $t_3$
- AB → CB → CE, $t_1$
- AB → CB → CD, $t_1$

Trace: time-abstract behavior
Objectives

- We consider that the system behavior (good or bad) depends on the traces.

Questions

- Until which value can we minimize the upper bound of $t_3$ (5) so that the system behavior remains the same?
- Can we quantify the system robustness?
Objectives

- We consider that the system behavior (good or bad) depends on the traces.

Questions

- Until which value can we minimize the upper bound of $t_3$ (5) so that the system behavior remains the same?
- Can we quantify the system robustness?

Idea

- Reason with parametric time Petri nets
- Synthesize a constraint on the parameters that guarantees the same behavior
Parametric Time Petri Nets

- Constants in firing intervals replaced with parameters [Traonouez et al., 2009]
Parametric Inhibitor Time Petri Nets

- Constants in firing intervals replaced with parameters
  [Traonouez et al., 2009]
Outline

1. Parametric Inhibitor Time Petri Nets
2. Robustness Analysis Using the Inverse Method
3. Perspectives
The Inverse Method

- **Input**
  - A PITPN $\mathcal{P}$
  - A reference valuation $\pi_0$ of all the parameters of $\mathcal{P}$

\[ \pi_0 \]
The Inverse Method

- **Input**
  - A PITPN $\mathcal{P}$
  - A reference valuation $\pi_0$ of all the parameters of $\mathcal{P}$

- **Output**: $K_0$
  - Convex constraint on the parameters such that
    - $\pi_0 \models K_0$
    - For all points $\pi \models K_0$, $\mathcal{P}[\pi]$ and $\mathcal{P}[\pi_0]$ have the same trace sets
The Inverse Method: General Idea

Initially defined for timed automata [André et al., 2009]

Extended to PITPNs [André and Garg, 2012]

The idea

- Exploration of the parametric state space
- Instead of negating bad states (as in “CEGAR” approaches), remove $\pi_0$-incompatible states
- Return the intersection of all constraints on the parameters
Application to an Example

\[ a = 5 \quad b = 6 \]
\[ c = 0 \quad d = 2 \]
\[ e = 1 \quad f = 5 \]
\[ g = 6 \quad h = 7 \]

- Forward analysis
Application to an Example

\[ \begin{align*}
A & \xrightarrow{\pi_0} B \\
t_1[a; b] & \rightarrow t_2[c; d] & \rightarrow t_3[e; f] & \rightarrow t_4[g; h] \\
C & \rightarrow D & \rightarrow E & \rightarrow F
\end{align*} \]

### Forward analysis

\[ K : \]
- true

\[ \pi_0 \]
- \( a = 5 \)
- \( b = 6 \)
- \( c = 0 \)
- \( d = 2 \)
- \( e = 1 \)
- \( f = 5 \)
- \( g = 6 \)
- \( h = 7 \)

\[ AB \]
- \( a \leq b \)
- \( c \leq d \)
- \( e \leq f \)
- \( g \leq h \)
Application to an Example

Forward analysis

K: true

π₀

\[
\begin{align*}
a &= 5 & b &= 6 \\
c &= 0 & d &= 2 \\
e &= 1 & f &= 5 \\
g &= 6 & h &= 7
\end{align*}
\]
Application to an Example

For a robustness analysis, consider the following example:

\[ \pi_0 \]

\[ a = 5 \quad b = 6 \]
\[ c = 0 \quad d = 2 \]
\[ e = 1 \quad f = 5 \]
\[ g = 6 \quad h = 7 \]

**Forward analysis**

\[ K : \]

true

- **AB**
  - \( a \leq b \quad c \leq d \)
  - \( e \leq f \quad g \leq h \)

- **CB**
  - \( a \leq b \quad c \leq d \)
  - \( e \leq f \quad g \leq h \)
  - \( a \leq h \quad a \leq f \)

- **AE**
  - \( a \leq b \quad c \leq d \)
  - \( e \leq f \quad g \leq h \)
  - \( e \leq b \quad e \leq h \)
Application to an Example

Forward analysis

\[ A \leq B \]
\[ C \leq D \]
\[ E \leq F \]
\[ G \leq H \]

\[ \begin{align*}
    & A \leq b & c \leq d \\
    & e \leq f & g \leq h \\
    & a \leq h & a \leq f \\
\end{align*} \]

\[ \begin{align*}
    & a = 5 & b = 6 \\
    & c = 0 & d = 2 \\
    & e = 1 & f = 5 \\
    & g = 6 & h = 7 \\
\end{align*} \]
Application to an Example

Forward analysis

\[ K : \quad g > f \]

\[ \pi_0 \]
\[ a = 5 \quad b = 6 \]
\[ c = 0 \quad d = 2 \]
\[ e = 1 \quad f = 5 \]
\[ g = 6 \quad h = 7 \]
Application to an Example

Forward analysis

\[ K : \quad g > f \]
## Application to an Example

![Petri Net Diagram]

**Forward analysis**

\[ K : \ g > f \]

### Initial Marking \( \pi_0 \)
- \( a = 5 \) \( b = 6 \)
- \( c = 0 \) \( d = 2 \)
- \( e = 1 \) \( f = 5 \)
- \( g = 6 \) \( h = 7 \)

<table>
<thead>
<tr>
<th>Transition</th>
<th>Invariants</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>( a \leq b ) ( c \leq d ) ( e \leq f ) ( g \leq h )</td>
</tr>
<tr>
<td>CB</td>
<td>( a \leq b ) ( c \leq d ) ( e \leq f ) ( g \leq h ) ( a \leq h ) ( a \leq f )</td>
</tr>
<tr>
<td>CE</td>
<td>( a \leq b ) ( c \leq d ) ( e \leq f ) ( g \leq h ) ( e \leq h ) ( a \leq f ) ( e \leq b + d )</td>
</tr>
<tr>
<td>AE</td>
<td>( a \leq b ) ( c \leq d ) ( e \leq f ) ( g \leq h ) ( e \leq b ) ( e \leq h )</td>
</tr>
</tbody>
</table>

*Étienne André (Paris 13)*

Robustness of Time Petri Nets

31st October 2012
Application to an Example

Forward analysis

\[ K : \]
\[ g > f \]

\[ \pi_0 \]
\[ a = 5 \quad b = 6 \]
\[ c = 0 \quad d = 2 \]
\[ e = 1 \quad f = 5 \]
\[ g = 6 \quad h = 7 \]
Application to an Example

\[
\begin{align*}
A & \quad B \\
\tau_1[a;b] & \quad \tau_2[c;d] & \quad \tau_3[e;f] & \quad \tau_4[g;h] \\
C & \quad D & \quad E & \quad F
\end{align*}
\]

- **Forward analysis**

\[K:
\begin{align*}
g &> f
\end{align*}\]

\[\left\{\begin{array}{ll}
AB & \quad CB & \quad CD \\
K_0 & \quad AE & \quad CE
\end{array}\right\}
\]

\[
\begin{align*}
\pi_0
a & = 5 & \quad b & = 6 \\
c & = 0 & \quad d & = 2 \\
e & = 1 & \quad f & = 5 \\
g & = 6 & \quad h & = 7
\end{align*}
\]
Application to an Example: Interpretation

- **Resulting constraint** $K_0$
  
  $a \leq b \quad c \leq d \quad e \leq f \quad g \leq h$
  
  $g > f \quad f \geq a \quad b \geq e$

- **Interpretation**
  
  - For any $\pi \models K_0$, the trace set is the same as for $\pi_0$

- **Remark**
  
  - $c$ and $d$ do not appear within $K_0$ (except $c \leq d$): for any $\pi \models K_0$, the values of $c$ and $d$ do not influence the trace set
Application to an Example: Interpretation

- Resulting constraint $K_0$
  
  $a \leq b \quad c \leq d \quad e \leq f \quad g \leq h$
  
  $g > f \quad f \geq a \quad b \geq e$

- Interpretation
  
  For any $\pi \models K_0$, the trace set is the same as for $\pi_0$

- Remark
  
  c and d do not appear within $K_0$ (except $c \leq d$): for any $\pi \models K_0$, the values of c and d do not influence the trace set

- Application
  
  Until which value can we minimize the upper bound $f$ of $t_3$ (5) so that the system behavior remains the same?
Application to an Example: Interpretation

- Resulting constraint $K_0$
  
  \[
  a \leq b \quad c \leq d \quad e \leq f \quad g \leq h \\
  g > f \quad f \geq a \quad b \geq e
  \]

- Interpretation
  
  - For any $\pi \models K_0$, the trace set is the same as for $\pi_0$

- Remark
  
  - $c$ and $d$ do not appear within $K_0$ (except $c \leq d$): for any $\pi \models K_0$, the values of $c$ and $d$ do not influence the trace set

- Application
  
  - Until which value can we minimize the upper bound $f$ of $t_3$ (5) so that the system behavior remains the same?
  - Due to $f \geq a$ with $a = f = 5$, one cannot decrease $f$
  
  $\Rightarrow$ The system is not robust w.r.t. small variations of $f$ or $a$
Correctness

Theorem (Correctness)

Let $\mathcal{P}$ be a PITPN, and $\pi_0$ be a reference valuation. Let $K_0 = IM(\mathcal{P}, \pi_0)$. Then:

1. $\pi_0 \models K_0$ and
2. $\forall \pi \models K_0$, $\mathcal{P}[\pi]$ and $\mathcal{P}[\pi_0]$ have the same trace set.
Correctness

Theorem (Correctness)

Let $\mathcal{P}$ be a PITPN, and $\pi_0$ be a reference valuation. Let $K_0 = IM(\mathcal{P}, \pi_0)$. Then:

1. $\pi_0 \models K_0$ and
2. $\forall \pi \models K_0$, $\mathcal{P}[\pi]$ and $\mathcal{P}[\pi_0]$ have the same trace set.

Proof.

By induction on the length of the runs.
Advantages

- Quantification of the system robustness
- Allows timing optimizations
- Allows the replacement of a component with another one
  - As long as the new timings satisfy $K_0$
Outline

1. Parametric Inhibitor Time Petri Nets
2. Robustness Analysis Using the Inverse Method
3. Perspectives
Perspectives (1/2)

- Extension to colored Petri nets [Jensen and Kristensen, 2009]
  - Tokens and places have a type (“color set”)
  - Arcs are labeled with expressions
  - Transitions can have a guard
Perspectives (1/2)

- Extension to colored Petri nets [Jensen and Kristensen, 2009]
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard
  - Example: A more complex version of the DVD renting machine

Legend

- Customers
- Money earned
- DVDs available
- DVDs on loan
Perspectives (1/2)

- **Extension to colored Petri nets** [Jensen and Kristensen, 2009]
  - Tokens and places have a **type** ("color set")
  - Arcs are labeled with **expressions**
  - Transitions can have a **guard**
  - Example: A more complex version of the DVD renting machine

```
1'(Alice, 30€) + 1'(Bob, 20€)

1'(0€)

f, p

f, p

intersections
f, p

Legend

Customers

Money earned

DVDs available

DVDs on loan
```
**Perspectives (1/2)**

- **Extension to colored Petri nets** [Jensen and Kristensen, 2009]
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard

- Example: A more complex version of the DVD renting machine

---

**Legend**

- **Customers**
- **Money earned**
- **DVDs available**
- **DVDs on loan**
Perspectives (1/2)

- Extension to colored Petri nets [Jensen and Kristensen, 2009]
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard
  - Example: A more complex version of the DVD renting machine

```plaintext
NAMExINT
1'(Alice, 24€) ++ 1'(Bob, 20€)

INT
1'(6€)
e + p
e

FILMxINT
f, p
1'(Rashōmon, 6€)

FILMxINT
2'(Satan Tango, 12€) ++ 1'(Un retour, 10€)

Legend

- Customers
- Money earned
- DVDs available
- DVDs on loan
```
Perspectives (1/2)

- **Extension to colored Petri nets** [Jensen and Kristensen, 2009]
  - Tokens and places have a **type** ("color set")
  - Arcs are labeled with **expressions**
  - Transitions can have a **guard**

- Example: A more complex version of the DVD renting machine

```
INT
1'(18€) ∙ NAMExINT
1'(Alice, 24€) ++ 1'(Bob, 8€)

FILMxINT
1'(Rashōmon, 6€) ++ 1'(Satan Tango, 12€)
```

```
Legend

- Customers
- Money earned
- DVDs available
- DVDs on loan
```

```
INT
1'(18€) ∙ NAMExINT
1'(Alice, 24€) ++ 1'(Bob, 8€)

FILMxINT
1'(Satan Tango, 12€) ++ 1'(Un retour, 10€)
```

```
Legend

- Customers
- Money earned
- DVDs available
- DVDs on loan
```

```
INT
1'(18€) ∙ NAMExINT
1'(Alice, 24€) ++ 1'(Bob, 8€)

FILMxINT
1'(Rashōmon, 6€) ++ 1'(Satan Tango, 12€)
```

```
Legend

- Customers
- Money earned
- DVDs available
- DVDs on loan
```
Perspectives (1/2)

- Extension to colored Petri nets [Jensen and Kristensen, 2009]
  - Tokens and places have a type (“color set”)
  - Arcs are labeled with expressions
  - Transitions can have a guard
  - Example: A more complex version of the DVD renting machine

![Diagram of a colored Petri net model for a DVD rental system]
Perspectives (1/2)

- Extension to colored Petri nets [Jensen and Kristensen, 2009]
  - Tokens and places have a type ("color set")
  - Arcs are labeled with expressions
  - Transitions can have a guard
  - Example: A more complex version of the DVD renting machine

```
<table>
<thead>
<tr>
<th>NAMExINT</th>
<th>1' (Alice, 24€)</th>
<th>++ 1' (Bob, 8€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT</td>
<td>c, m</td>
<td>c, m - p</td>
</tr>
<tr>
<td>FILMxINT</td>
<td>2' (Satan Tango, 12€)</td>
<td>++ 1' (Un retour, 10€)</td>
</tr>
<tr>
<td>FILMxINT</td>
<td>f, p</td>
<td>f, p</td>
</tr>
<tr>
<td>FILMxINT</td>
<td>f, p</td>
<td>f, p</td>
</tr>
<tr>
<td>FILMxINT</td>
<td>f, p</td>
<td>f, p</td>
</tr>
</tbody>
</table>
```

Legend
- Customers
- Money earned
- DVDs available
- DVDs on loan
Perspectives (2/2)

- **Termination**
  - Probably does not terminate in the general case
  - … but no example exhibited so far

- **Implementation**
  - To do

- **Modular analysis**
  - Combine the inverse method with modular state space exploration for timed Petri nets [Lakos and Petrucci, 2007]
  - Idea: apply the inverse method to separate modules, then combine the result
  - Challenge: identify subclasses of time(d) Petri nets such that this applies
References I

An inverse method for parametric timed automata.

Robustness analysis of time Petri nets.
In *NWPT’12*, Bergen, Norway.

*Coloured Petri Nets – Modelling and Validation of Concurrent Systems*.
Springer.

Modular state space exploration for timed Petri nets.

Robustness in real-time systems.

*A study of the recoverability of computing systems*.
References II


The Algorithm

**Algorithm 1: IM(\(\mathcal{P}, \pi_0\))**

1. \(i \leftarrow 0; \ K \leftarrow K_{\text{init}}; \ C \leftarrow \{c_0\}\)
2. \(\text{while true do}\)
   3. \(\text{while } \exists \pi_0\text{-incompatible classes in } C \text{ do}\)
   4. \(\text{Select a } \pi_0\text{-incompatible class } (M, D) \text{ of } C\)
   5. \(\text{Select a } \pi_0\text{-incompatible } J \text{ in } D_{\downarrow \mathcal{P}}\)
   6. \(K \leftarrow K \land \neg J\)
   7. \(C \leftarrow \bigcup_{j=0}^{i} \text{Post}_{\mathcal{P}(K)}(\{c_0\})\)
   8. \(\text{if } \text{Post}_{\mathcal{P}(K)}(C) \subseteq C \text{ then}\)
   9. \(\text{return } K_0 \leftarrow \bigcap_{(M, D) \in C} D_{\downarrow \mathcal{P}}\)
10. \(i \leftarrow i + 1; \ C \leftarrow C \cup \text{Post}_{\mathcal{P}(K)}(C)\)
This document can be redistributed following the terms of license Creative Commons BY-NC-ND 3.0.

https://creativecommons.org/licenses/by-nc-nd/3.0/