Minimal Coverability Tree Construction Made Complete and Efficient

Alain Finkel[1,3], Serge Haddad[1,2], and **Igor Khmelnitsky**[1,2]

June 17, 2021

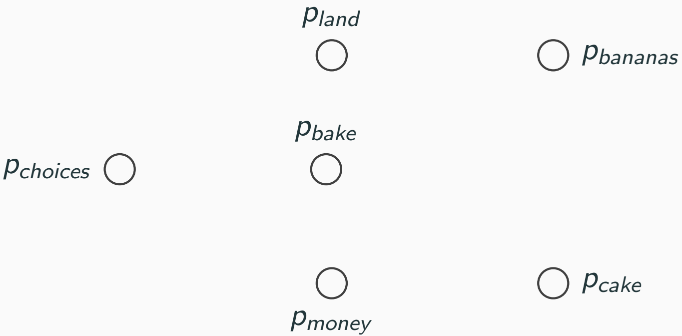
[1] LMF, ENS Paris-Saclay, CNRS, Universite Paris-Saclay, Cachan, France
[2] Inria, France
[3] IUF, France

Table of contents

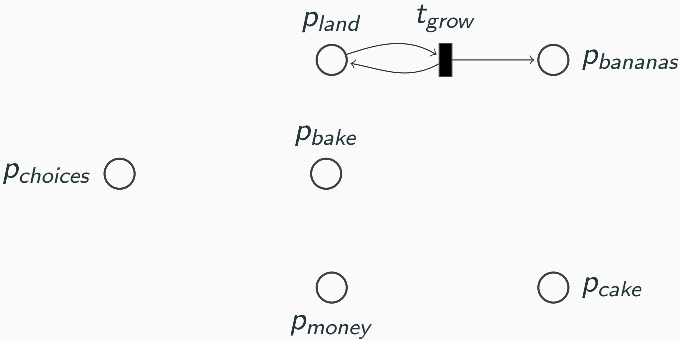
Petri Nets

Banana Land

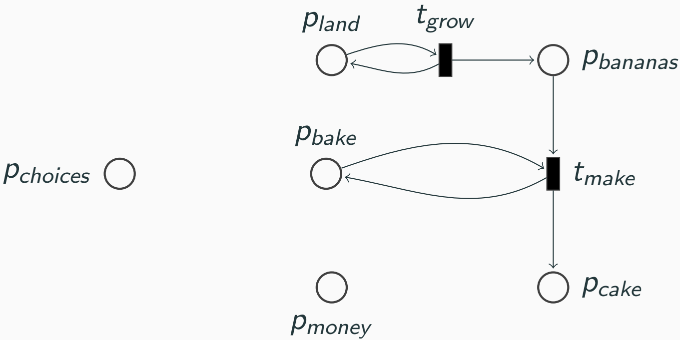
Banana Land



Banana Land

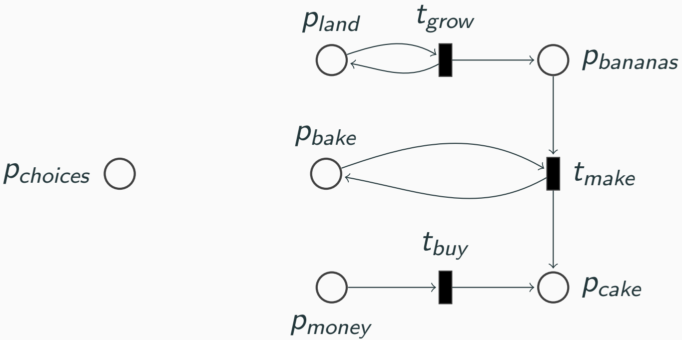


Banana Land



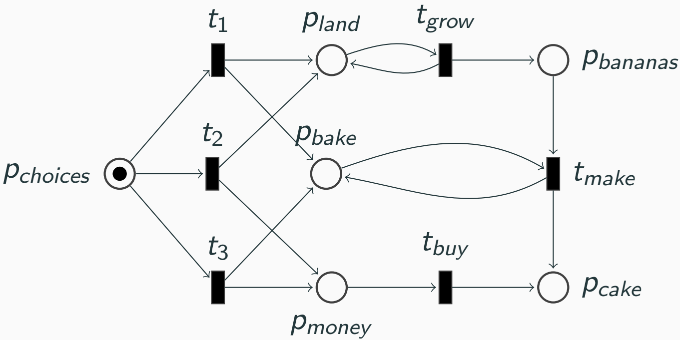
$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}} \mathbf{m}_2$$

Banana Land



$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}} \mathbf{m}_2$$

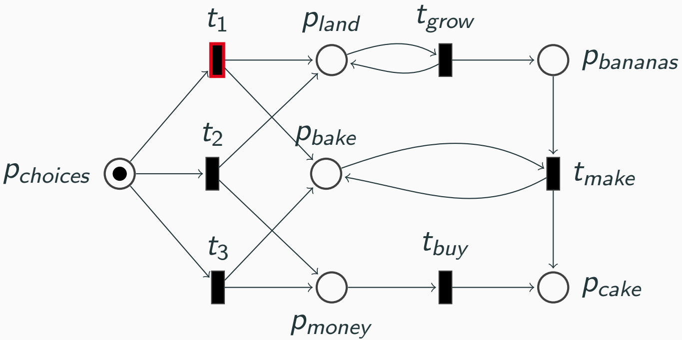
Banana Land



$\mathbf{m}_0$

$\mathbf{m}_0 = p_{choices}$

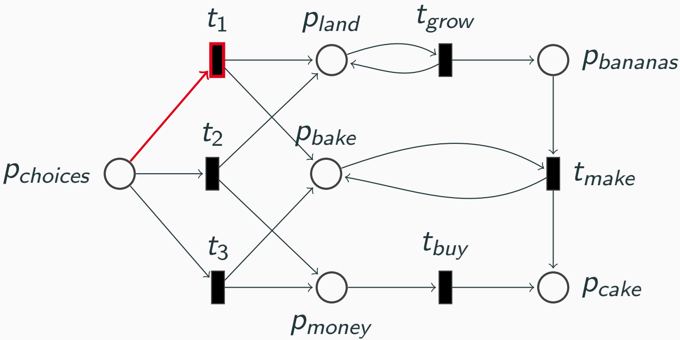
Banana Land



$$\mathbf{m}_0 \xrightarrow{t_1}$$

$\mathbf{m}_0 = p_{choices}$

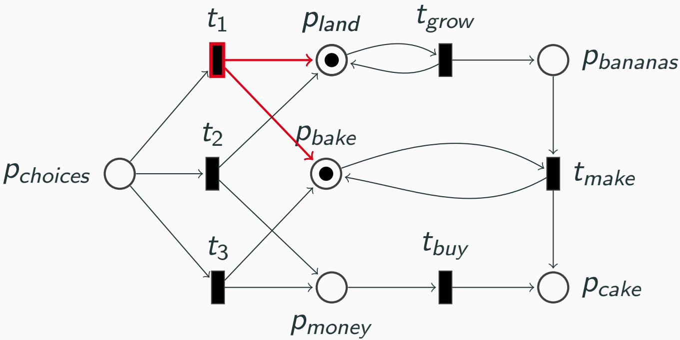
Banana Land



$$\mathbf{m}_0 \xrightarrow{t_1}$$

$\mathbf{m}_0 = p_{choices}$

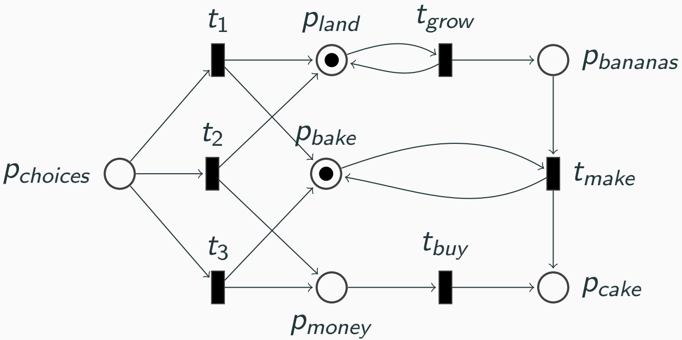
Banana Land



$$\mathbf{m}_0 \xrightarrow{t_1}$$

$\mathbf{m}_0 = p_{choices}$

Banana Land

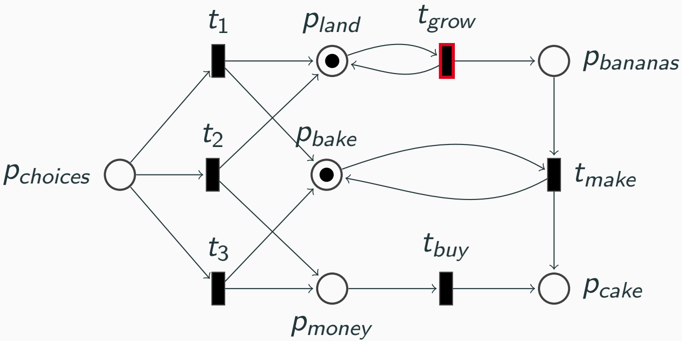


$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1$$

$\mathbf{m}_0 = p_{choices}$

$\mathbf{m}_1 = p_{land} + p_{bake}$

Banana Land

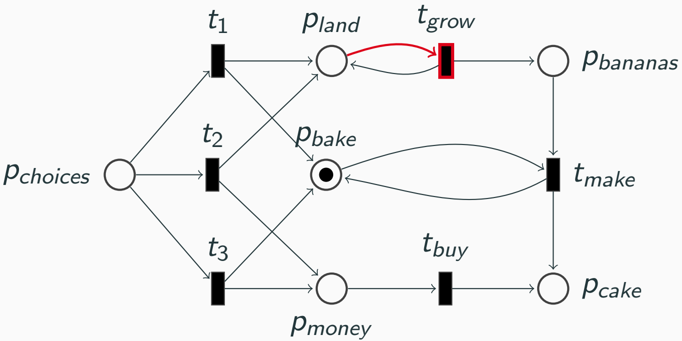


$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}}$$

$\mathbf{m}_0 = p_{choices}$

$\mathbf{m}_1 = p_{land} + p_{bake}$

Banana Land

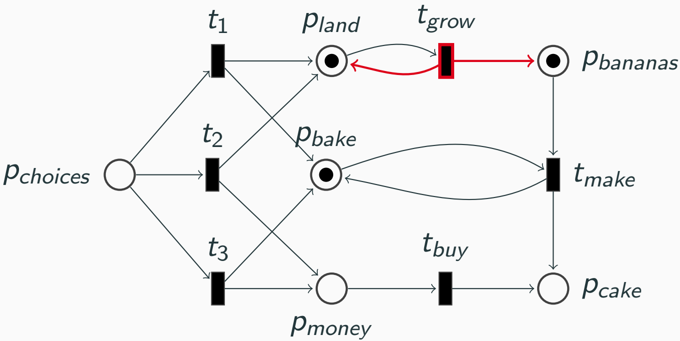


$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}}$$

$\mathbf{m}_0 = p_{choices}$

$\mathbf{m}_1 = p_{land} + p_{bake}$

Banana Land

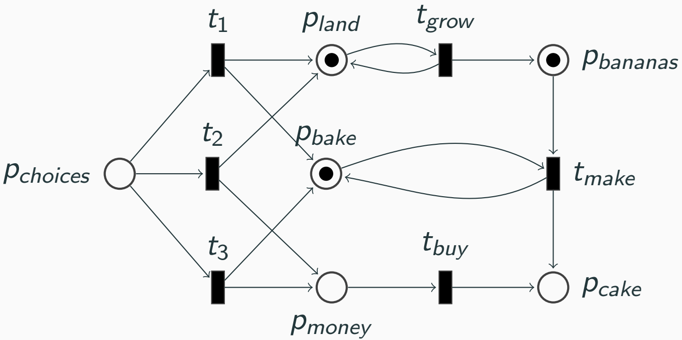


$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}}$$

$\mathbf{m}_0 = p_{choices}$

$\mathbf{m}_1 = p_{land} + p_{bake}$

Banana Land



$$\mathbf{m}_0 \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_{grow}} \mathbf{m}_2$$

$\mathbf{m}_0 = p_{choices}$

$\mathbf{m}_1 = p_{land} + p_{bake}$

$\mathbf{m}_2 = p_{land} + p_{bake} + p_{banana}$

Petri Nets & Coverability

Petri Nets & Coverability

Reachability:

Input: $(\mathcal{N}, \mathbf{m}_0, \mathbf{m})$; Output: $\exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ?\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ?\mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state $\rightarrow$ Coverability.

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m})\text{; Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state $\rightarrow$ Coverability.

Coverability set: Downward closure of the reachability set

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state $\rightarrow$ Coverability.

Coverability set: Downward closure of the reachability set

- Parameterized coverability

Petri Nets & Coverability

Reachability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}$$

Decidable but hard (Not Primitive Recursive [Leroux-21]), but still widely used.

Coverability:

$$\text{Input: } (\mathcal{N}, \mathbf{m}_0, \mathbf{m}); \text{ Output: } \exists ? \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$$

Easier then reachability (EXPSPACE [Rackoff-78]), and can still be useful:

- Over-approximation
- Safety$\rightarrow$ Control state $\rightarrow$ Coverability.

Coverability set: Downward closure of the reachability set

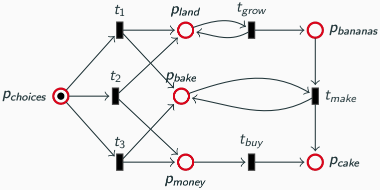
- Parameterized coverability
- Boundedness

Reachability and Coverability in Petri Nets

Reachability and Coverability in Petri Nets

Petri Net:

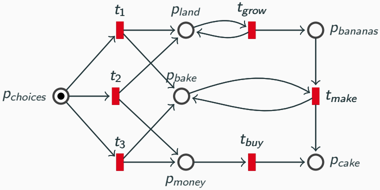
$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability and Coverability in Petri Nets

Petri Net:

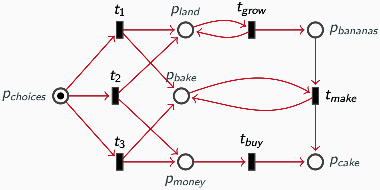
$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability and Coverability in Petri Nets

Petri Net:

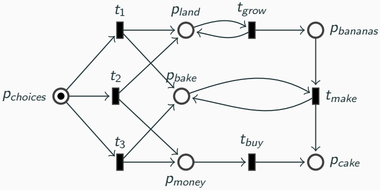
$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability and Coverability in Petri Nets

Petri Net:

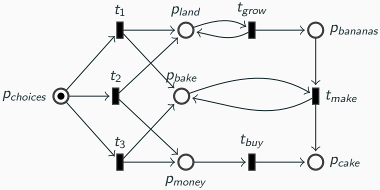
$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability and Coverability in Petri Nets

Petri Net:

$$\mathcal{N} = \langle P, T, Pre, C \rangle$$

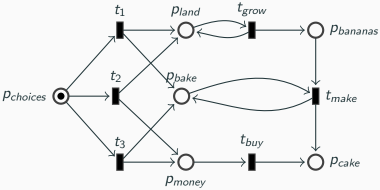


Reachability set: $Reach(\mathcal{N}, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}\}$

Reachability and Coverability in Petri Nets

Petri Net:

$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



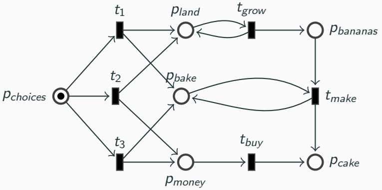
Reachability set: $Reach(\mathcal{N}, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}\}$

Order on markings: $\mathbf{m} \geq \mathbf{m}' \iff \forall p \in P\ \mathbf{m}(p) \geq \mathbf{m}'(p)$

Reachability and Coverability in Petri Nets

Petri Net:

$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability set: $Reach(\mathcal{N}, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}\}$

Order on markings: $\mathbf{m} \geq \mathbf{m}' \iff \forall p \in P\ \mathbf{m}(p) \geq \mathbf{m}'(p)$

Coverability set:

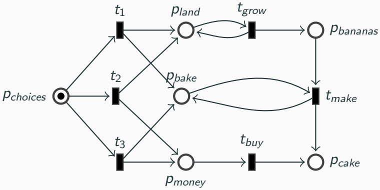
$$Cover(\mathcal{N}, \mathbf{m}_0) \stackrel{def}{=} \downarrow Reach(\mathcal{N}, \mathbf{m}_0)$$

$$= \{\mathbf{m} \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}\}$$

Reachability and Coverability in Petri Nets

Petri Net:

$$\mathcal{N} = \langle P, T, Pre, C \rangle$$



Reachability set: $Reach(\mathcal{N}, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}\}$

Order on markings: $\mathbf{m} \geq \mathbf{m}' \iff \forall p \in P\ \mathbf{m}(p) \geq \mathbf{m}'(p)$

Coverability set:

$$Cover(\mathcal{N}, \mathbf{m}_0) \stackrel{def}{=} \downarrow Reach(\mathcal{N}, \mathbf{m}_0)$$
$$= \{\mathbf{m} \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}\}$$

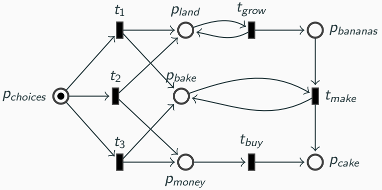
Questions:

1. Does there exist a finite representation of $Cover(\mathcal{N}, \mathbf{m}_0)$?

Reachability and Coverability in Petri Nets

Petri Net:

$$\mathcal{N} = \langle P, T, Pre, C\rangle$$



Reachability set: $Reach(\mathcal{N}, \mathbf{m}_0) = \{\mathbf{m} \in \mathbb{N}^P \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}\}$

Order on markings: $\mathbf{m} \geq \mathbf{m}' \iff \forall p \in P\ \mathbf{m}(p) \geq \mathbf{m}'(p)$

Coverability set:

$$Cover(\mathcal{N}, \mathbf{m}_0) \stackrel{def}{=} \downarrow Reach(\mathcal{N}, \mathbf{m}_0)$$
$$= \{\mathbf{m} \mid \exists \sigma \in T^*, \mathbf{m}_0 \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}\}$$

Questions:

1. Does there exist a finite representation of $Cover(\mathcal{N}, \mathbf{m}_0)$?
2. How to compute it?

A Finite representation of the Coverability Set

A Finite representation of the Coverability Set

- $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, $\mathbb{Z}_\omega = \mathbb{Z} \cup \{\omega\}$

A Finite representation of the Coverability Set

- $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, $\mathbb{Z}_\omega = \mathbb{Z} \cup \{\omega\}$
- Let $\mathbf{m} \in \mathbb{N}_\omega^P$ (an ω-marking):

$$[\![\mathbf{m}]\!] = \{\mathbf{m}' \in \mathbb{N}^P \mid \mathbf{m}' \leq \mathbf{m}\}$$

A Finite representation of the Coverability Set

- $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, $\mathbb{Z}_\omega = \mathbb{Z} \cup \{\omega\}$
- Let $\mathbf{m} \in \mathbb{N}_\omega^P$ (an ω-marking):

$$[\![\mathbf{m}]\!] = \{\mathbf{m}' \in \mathbb{N}^p \mid \mathbf{m}' \leq \mathbf{m}\}$$

Theorem (Erdős)
There exists a finite (minimal) set of ω-markings $Clover(\mathcal{N}, \mathbf{m}_0)$ s.t.

$$Cover(\mathcal{N}, \mathbf{m}_0) = \bigcup_{\mathbf{m} \in Clover(\mathcal{N}, \mathbf{m}_0)} [\![\mathbf{m}]\!]$$

A Finite representation of the Coverability Set

- $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, $\mathbb{Z}_\omega = \mathbb{Z} \cup \{\omega\}$
- Let $\mathbf{m} \in \mathbb{N}_\omega^P$ (an ω-marking):

$$[\![\mathbf{m}]\!] = \{\mathbf{m}' \in \mathbb{N}^p \mid \mathbf{m}' \leq \mathbf{m}\}$$

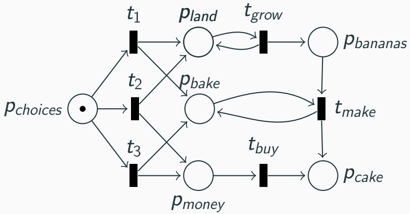
Theorem (Erdős)
There exists a finite (minimal) set of ω-markings $Clover(\mathcal{N}, \mathbf{m}_0)$ s.t.

$$Cover(\mathcal{N}, \mathbf{m}_0) = \bigcup_{\mathbf{m} \in Clover(\mathcal{N}, \mathbf{m}_0)} [\![\mathbf{m}]\!]$$

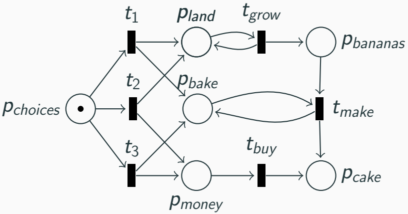
Revisited question: Can one build $Clover(\mathcal{N}, \mathbf{m}_0)$?

First Steps

Reachability Tree

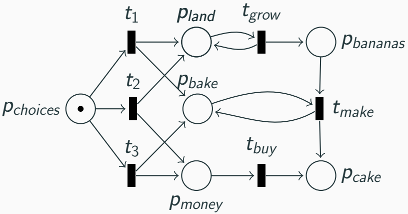


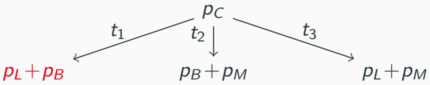
Reachability Tree



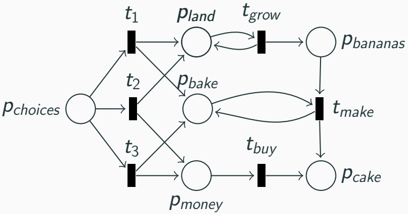
p_C

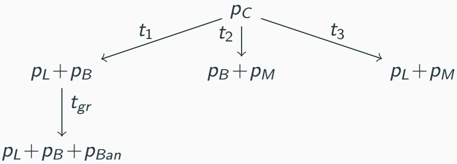
Reachability Tree



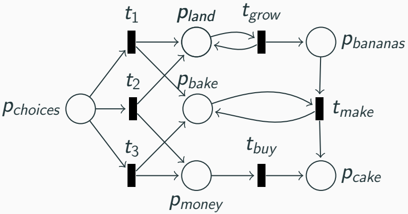


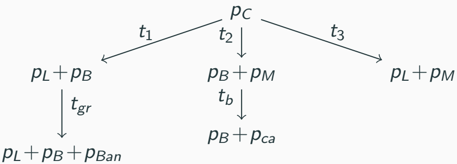
Reachability Tree



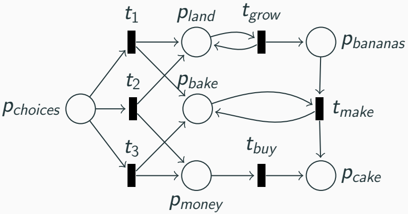


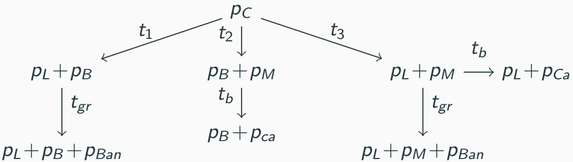
Reachability Tree



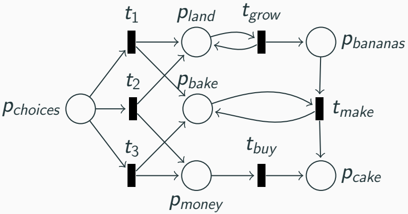


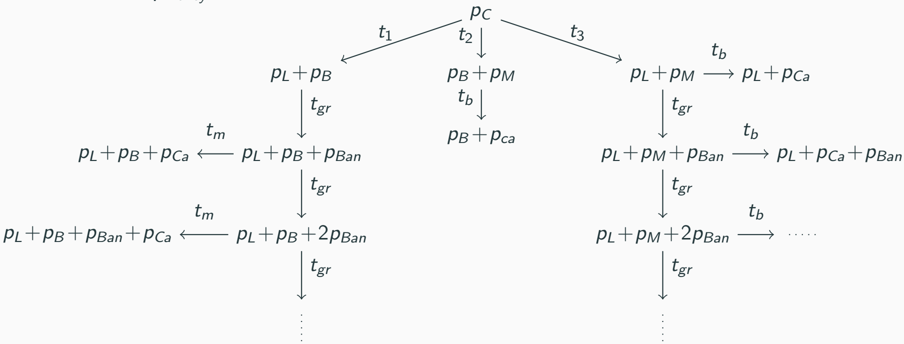
Reachability Tree





Reachability Tree





Reachability Tree

Reachability enumeration algorithm:

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; *Front* $\subseteq V$;

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; *Front* $\subseteq V$;

Main loop:

While *Front* $\neq \emptyset$

Fairly pop *v* from *Front*

Explore(v)

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $\mathit{Front} \subseteq V$;

Main loop:

While $\mathit{Front} \neq \emptyset$

Fairly pop v from Front

Explore(v)

Correctness proof sketch:

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Fairly pop v from $Front$

Explore(v)

Correctness proof sketch:

- Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Fairly pop v from $Front$

Explore(v)

Correctness proof sketch:

- Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:

  $\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; *Front* $\subseteq V$;

Main loop:

While *Front* $\neq \emptyset$

Fairly pop v from *Front*

Explore(v)

Correctness proof sketch:

- Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:

  $\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.

- Completeness - $Reach(\mathcal{N}, \mathbf{m}_0) \subseteq \lambda(V)$:

  $\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Fairly pop v from $Front$

Explore(v)

Correctness proof sketch:

- Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:

  $\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.

- Completeness - $Reach(\mathcal{N}, \mathbf{m}_0) \subseteq \lambda(V)$:

  $\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

  - $u \notin$ Front and $\lambda(u) = \mathbf{m}$

Reachability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Fairly pop v from $Front$

Explore(v)

Correctness proof sketch:

- Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:

  $\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.

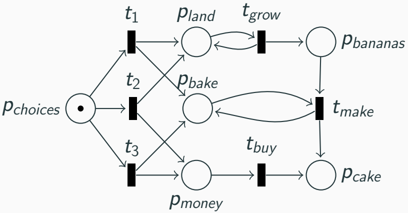
- Completeness - $Reach(\mathcal{N}, \mathbf{m}_0) \subseteq \lambda(V)$:

  $\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

  - $u \notin$ Front and $\lambda(u) = \mathbf{m}$
  - $u \in$ Front , $\exists \sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

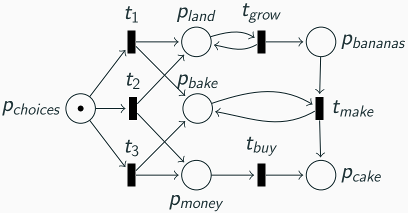
  and applying fairness.

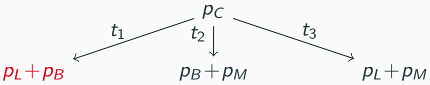
K&M Coverability Tree



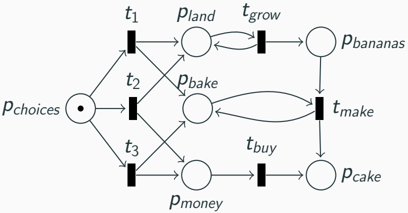
p_C

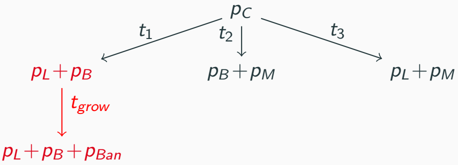
K&M Coverability Tree



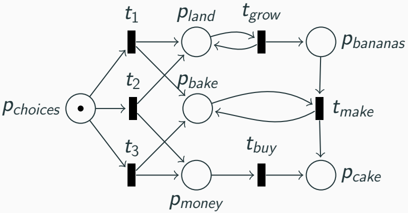


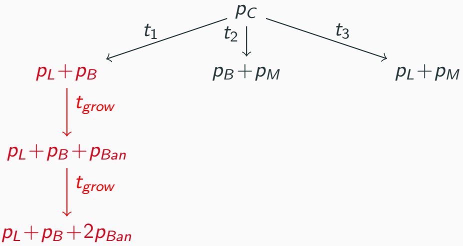
K&M Coverability Tree



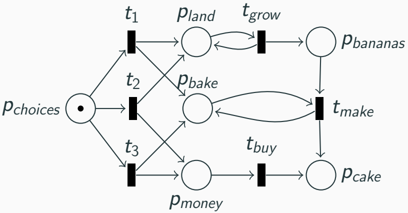


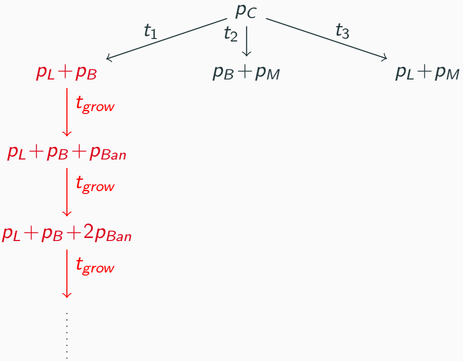
K&M Coverability Tree



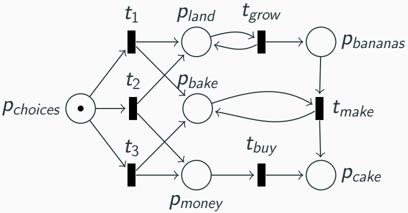


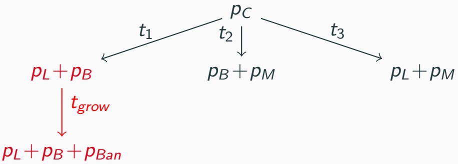
K&M Coverability Tree



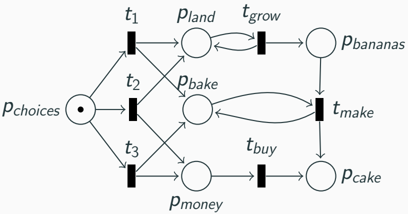


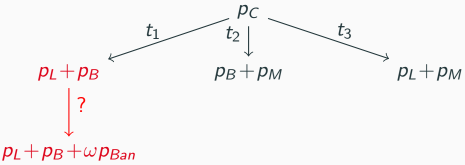
K&M Coverability Tree



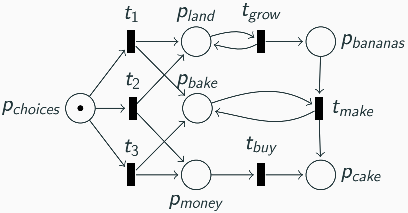


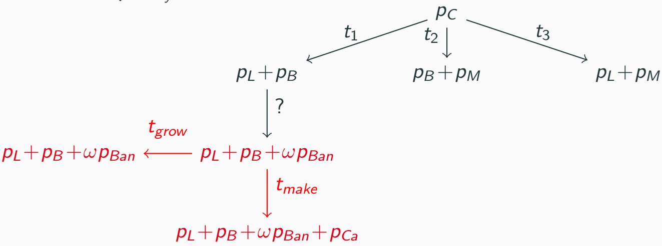
K&M Coverability Tree



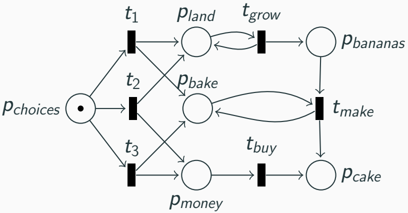


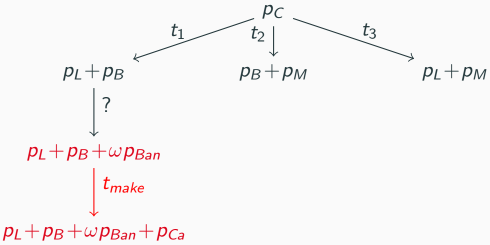
K&M Coverability Tree



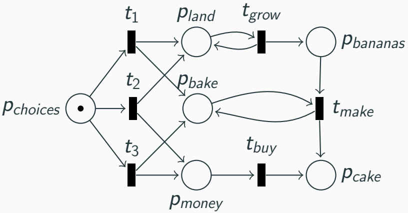


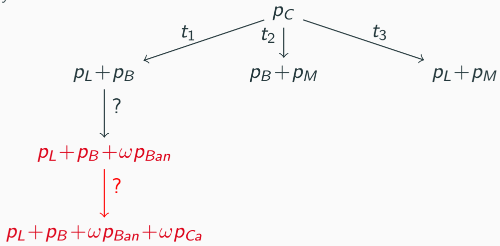
K&M Coverability Tree



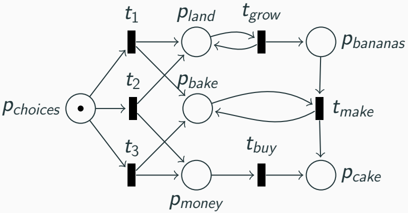


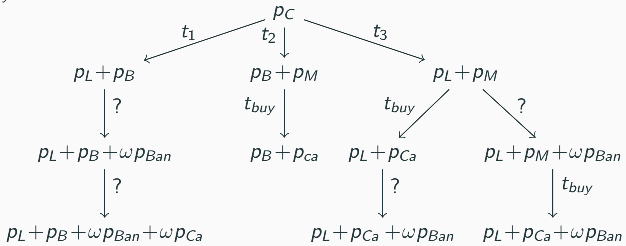
K&M Coverability Tree





K&M Coverability Tree





Reachability enumeration algorithm:

K&M Coverability Tree

Reachability enumeration algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

Explore(v)

K&M Coverability Tree

K&M algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then** Accelerate(u,v)

Explore(v)

K&M Coverability Tree

K&M algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then** Accelerate(u,v)

Explore(v)

K&M Coverability Tree

K&M algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then** Accelerate(u,v)

Explore(v)

Correctness proof

- The original proof of K&M-algorithm is incomplete [Hack-74].
- Formal COQ proof of K&M-algorithm [Yamamoto-17].
- Hard to generalize the proof to variants.

Can We Find a Simple Proof?

How to adapt the reachability proof?

Can We Find a Simple Proof?

How to adapt the reachability proof?

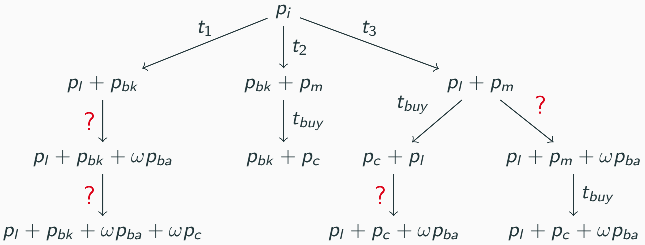
Consistency $\lambda(V) \subseteq \mathit{Reach}(\mathcal{N}, \mathbf{m}_0)$:
$\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.

Can We Find a Simple Proof?

How to adapt the reachability proof?

Consistency $\lambda(V) \subseteq \mathit{Reach}(\mathcal{N}, \mathbf{m}_0)$:

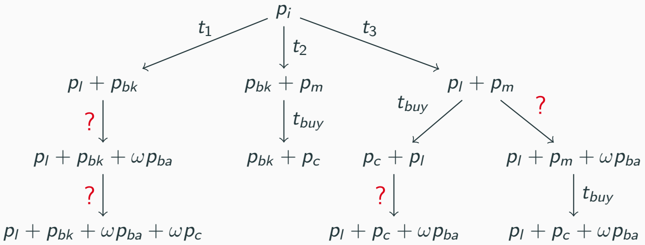
$\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.



Can We Find a Simple Proof?

How to adapt the reachability proof?

Consistency $\lambda(V) \subseteq Reach(\mathcal{N}, \mathbf{m}_0)$:
$\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.



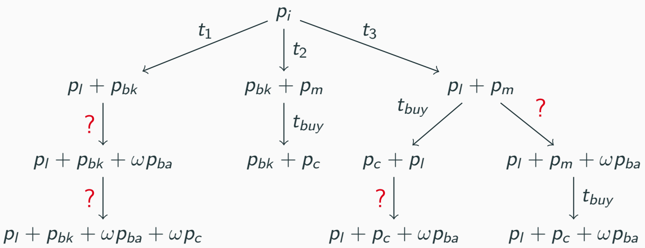
For all edges $u \xrightarrow{?} v$, there does not exist $t \in T$ s.t. $\lambda(u) \xrightarrow{t} \lambda(v)$.

Can We Find a Simple Proof?

How to adapt the reachability proof?

Consistency $\lambda(V) \subseteq \mathit{Reach}(\mathcal{N}, \mathbf{m}_0)$:

$\lambda(r) = \mathbf{m}_0$ and for all edge $u \xrightarrow{t} v$, one has $\lambda(u) \xrightarrow{t} \lambda(v)$.



For all edges $u \xrightarrow{?} v$, there does not exist $\sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Abstractions and Accelerations

$\omega-$Transitions

$\omega-$Transitions

Syntax An *ω-transition* **a** is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

$\omega-$Transitions

Syntax An *ω-transition* **a** is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

Semantic

$$\mathbf{m} \xrightarrow{\mathbf{a}} \text{ if } \mathbf{Pre}(\mathbf{a}) \leq \mathbf{m}, \text{ and } \mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m} + \mathbf{C}(\mathbf{a}).$$

$\omega-$Transitions

Syntax An ω-*transition* $\mathbf{a}$ is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

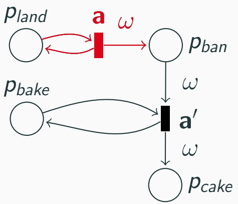
$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

Semantic

$$\mathbf{m} \xrightarrow{\mathbf{a}} \text{ if } \mathbf{Pre}(\mathbf{a}) \leq \mathbf{m}, \text{ and } \mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m} + \mathbf{C}(\mathbf{a}).$$

Concatenation (by example)

- $\mathbf{Pre}(\mathbf{a}) = p_{land}, \mathbf{C}(\mathbf{a}) = \omega p_{ban}$



Syntax An ω-*transition* $\mathbf{a}$ is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

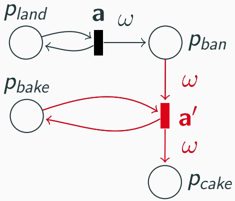
$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

Semantic

$$\mathbf{m} \xrightarrow{\mathbf{a}} \text{ if } \mathbf{Pre}(\mathbf{a}) \leq \mathbf{m}, \text{ and } \mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m} + \mathbf{C}(\mathbf{a}).$$

Concatenation (by example)

- $\mathbf{Pre}(\mathbf{a}) = p_{land}$, $\mathbf{C}(\mathbf{a}) = \omega p_{ban}$
- $\mathbf{Pre}(\mathbf{a}') = \omega p_{ban} + p_{bake}$, $\mathbf{C}(\mathbf{a}') = \omega p_{ban} + \omega p_{cake}$



$\omega-$Transitions

Syntax An ω-*transition* **a** is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

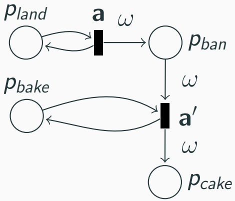
$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

Semantic

$$\mathbf{m} \xrightarrow{\mathbf{a}} \text{ if } \mathbf{Pre}(\mathbf{a}) \leq \mathbf{m}, \text{ and } \mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m} + \mathbf{C}(\mathbf{a}).$$

Concatenation (by example)

- $\mathbf{Pre}(\mathbf{a}) = p_{land}$, $\mathbf{C}(\mathbf{a}) = \omega p_{ban}$
- $\mathbf{Pre}(\mathbf{a}') = \omega p_{ban} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}') = \omega p_{ban} + \omega p_{cake}$
- $\mathbf{Pre}(\mathbf{a} \cdot \mathbf{a}') = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a} \cdot \mathbf{a}') = \omega p_{ban} + \omega p_{cake}$



ω−Transitions

Syntax An ω-*transition* **a** is defined by $\mathbf{Pre}(\mathbf{a}) \in \mathbb{N}_\omega^P, \mathbf{C}(\mathbf{a}) \in \mathbb{Z}_\omega^P$ where:

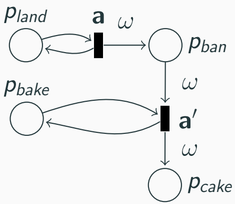
$$\mathbf{Pre}(\mathbf{a}) + \mathbf{C}(\mathbf{a}) \geq 0 \text{ and } \forall p, \text{ s.t. } \mathbf{Pre}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \mathbf{a}) = \omega.$$

Semantic

$$\mathbf{m} \xrightarrow{\mathbf{a}} \text{ if } \mathbf{Pre}(\mathbf{a}) \leq \mathbf{m}, \text{ and } \mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m} + \mathbf{C}(\mathbf{a}).$$

Concatenation (by example)

- $\mathbf{Pre}(\mathbf{a}) = p_{land}$, $\mathbf{C}(\mathbf{a}) = \omega p_{ban}$
- $\mathbf{Pre}(\mathbf{a}') = \omega p_{ban} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}') = \omega p_{ban} + \omega p_{cake}$
- $\mathbf{Pre}(\mathbf{a} \cdot \mathbf{a}') = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a} \cdot \mathbf{a}') = \omega p_{ban} + \omega p_{cake}$



$$\mathbf{m} \xrightarrow{\mathbf{aa}'} \mathbf{m}' \text{ if and only if } \mathbf{m} \xrightarrow{\mathbf{a}\cdot\mathbf{a}'} \mathbf{m}'.$$

Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;

Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_{\boldsymbol{n}} \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;

Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

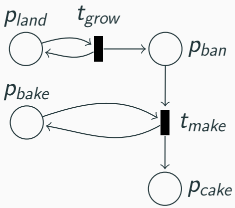
Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

Illustration

- $\mathbf{Pre}(\mathbf{a}) = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}) = \omega p_{ban} + \omega p_{cake}$



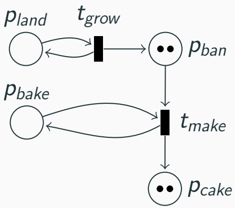
Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

Illustration

- $\mathbf{Pre}(\mathbf{a}) = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}) = \omega p_{ban} + \omega p_{cake}$
- $\sigma_2 =$

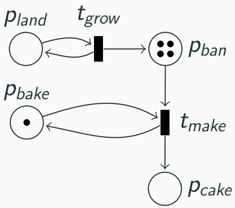


An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

Illustration

- $\mathbf{Pre}(\mathbf{a}) = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}) = \omega p_{ban} + \omega p_{cake}$
- $\sigma_2 = \quad t_{make}^2$



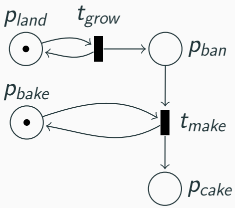
Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

Illustration

- $\mathbf{Pre}(\mathbf{a}) = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}) = \omega p_{ban} + \omega p_{cake}$
- $\sigma_2 = t_{grow}^4 t_{make}^2$



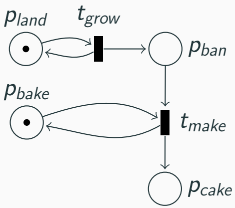
Coverability Abstraction

An ω-transition $\mathbf{a}$ is an *abstraction* if for all n, there exists $\sigma_n \in T^*$ s.t. for all p where $\mathbf{Pre}(p, \mathbf{a}) \neq \omega$:

- $\mathbf{Pre}(p, \sigma_n) \leq \mathbf{Pre}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) \neq \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq \mathbf{C}(p, \mathbf{a})$;
- If $\mathbf{C}(p, \mathbf{a}) = \omega \Rightarrow \mathbf{C}(p, \sigma_n) \geq n$.

Illustration

- $\mathbf{Pre}(\mathbf{a}) = p_{land} + p_{bake}$,
  $\mathbf{C}(\mathbf{a}) = \omega p_{ban} + \omega p_{cake}$
- $\sigma_2 = t_{grow}^4 t_{make}^2$
- $\sigma_n = t_{grow}^{2n} t_{make}^n$
  $\mathbf{Pre}(\sigma_n) = p_{land} + p_{bake}$,
  $\mathbf{C}(\sigma_n) = np_{ban} + np_{cake}$



Properties of Abstractions

The coverability set is closed by abstraction firing.

Properties of Abstractions

The coverability set is closed by abstraction firing.

Let **a** be an abstraction and $\mathbf{m} \in \mathbb{N}_\omega^P$.

Properties of Abstractions

The coverability set is closed by abstraction firing.

Let $\mathbf{a}$ be an abstraction and $\mathbf{m} \in \mathbb{N}_\omega^P$.

If:

$$[\![\mathbf{m}]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$$

and $\mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m}'$

Properties of Abstractions

The coverability set is closed by abstraction firing.

Let **a** be an abstraction and $\mathbf{m} \in \mathbb{N}_\omega^P$.

If:

$[\![\mathbf{m}]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$

and $\mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m}'$

Then:

$$[\![\mathbf{m}']\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$$

Properties of Abstractions

The coverability set is closed by abstraction firing.

Let **a** be an abstraction and $\mathbf{m} \in \mathbb{N}_\omega^P$.

If:

$$[\![\mathbf{m}]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$$

and $\mathbf{m} \xrightarrow{\mathbf{a}} \mathbf{m}'$

Then:

$$[\![\mathbf{m}']\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$$

The set of abstractions is closed by concatenation.

Acceleration

Acceleration

An abstraction **a** with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

Acceleration

An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction $\mathbf{a}$:

Acceleration

An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction $\mathbf{a}$:

If	$\mathbf{C}(p, \mathbf{a}) < 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$
If	$\mathbf{C}(p, \mathbf{a}) = 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$
If	$\mathbf{C}(p, \mathbf{a}) > 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Acceleration

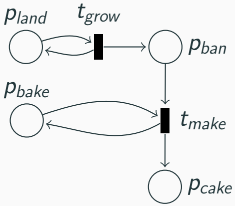
An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction $\mathbf{a}$:

If	$\mathbf{C}(p, \mathbf{a}) < 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$
If	$\mathbf{C}(p, \mathbf{a}) = 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$
If	$\mathbf{C}(p, \mathbf{a}) > 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Example

- $\mathbf{a} = t_{make}$
- $\mathbf{Pre}(\widehat{\mathbf{a}}) = \omega p_{ban} + p_{bake}$
  $\mathbf{C}(\widehat{\mathbf{a}}) = \omega p_{ban} + \omega p_{cake}$



Acceleration

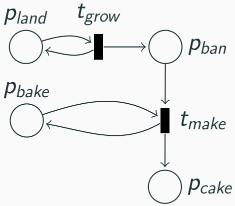
An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction $\mathbf{a}$:

If	$\mathbf{C}(p, \mathbf{a}) < 0$	*Then*	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$
If	$\mathbf{C}(p, \mathbf{a}) = 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$
If	$\mathbf{C}(p, \mathbf{a}) > 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Example

- $\mathbf{a} = t_{make}$
- $\mathbf{Pre}(\widehat{\mathbf{a}}) = \omega \mathbf{p_{ban}} + p_{bake}$
  $\mathbf{C}(\widehat{\mathbf{a}}) = \omega \mathbf{p_{ban}} + \omega p_{cake}$



Acceleration

An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction $\mathbf{a}$:

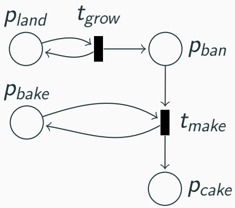
If $\mathbf{C}(p, \mathbf{a}) < 0$ Then $\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$

If $\mathbf{C}(p, \mathbf{a}) = 0$ *Then* $\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$

If $\mathbf{C}(p, \mathbf{a}) > 0$ Then $\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Example

- $\mathbf{a} = t_{make}$
- $\mathbf{Pre}(\widehat{\mathbf{a}}) = \omega p_{ban} + \mathbf{p_{bake}}$
  $\mathbf{C}(\widehat{\mathbf{a}}) = \omega p_{ban} + \omega p_{cake}$



Acceleration

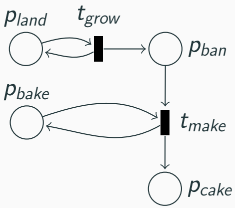
An abstraction **a** with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ from an abstraction **a** *:*

If	$\mathbf{C}(p, \mathbf{a}) < 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$
If	$\mathbf{C}(p, \mathbf{a}) = 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$
If	$\mathbf{C}(p, \mathbf{a}) > 0$	*Then*	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Example

- $\mathbf{a} = t_{make}$
- $\mathbf{Pre}(\widehat{\mathbf{a}}) = \omega p_{ban} + p_{bake}$
  $\mathbf{C}(\widehat{\mathbf{a}}) = \omega p_{ban} + \omega \mathbf{p_{cake}}$



Acceleration

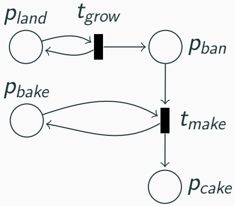
An abstraction $\mathbf{a}$ with $\mathbf{C}(\mathbf{a}) \in \{0, \omega\}^P$ is an acceleration.

How to get an acceleration $\widehat{\mathbf{a}}$ *from an abstraction* $\mathbf{a}$ *:*

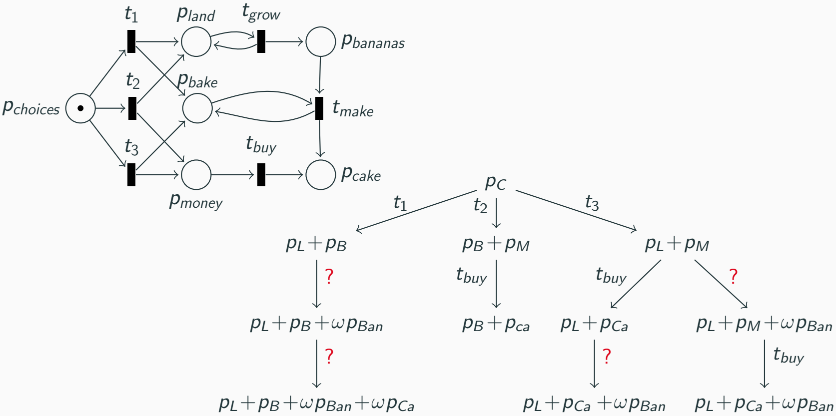
If	$\mathbf{C}(p, \mathbf{a}) < 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{C}(p, \widehat{\mathbf{a}}) = \omega$
If	$\mathbf{C}(p, \mathbf{a}) = 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = 0$
If	$\mathbf{C}(p, \mathbf{a}) > 0$	Then	$\mathbf{Pre}(p, \widehat{\mathbf{a}}) = \mathbf{Pre}(p, \mathbf{a}), \mathbf{C}(\widehat{\mathbf{a}}) = \omega$

Example

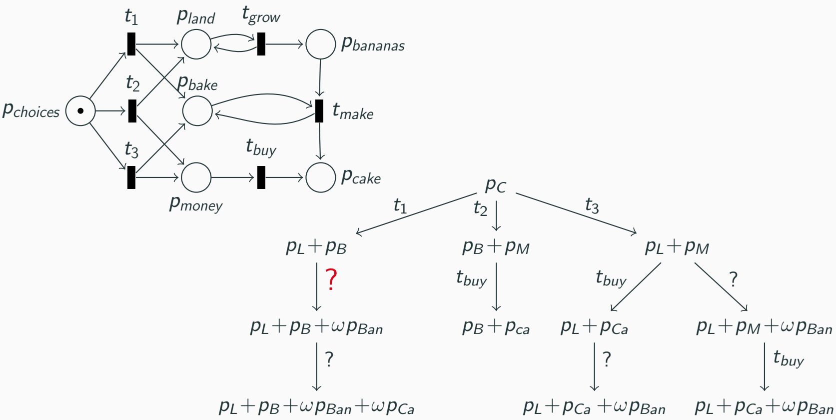
- $\mathbf{a} = t_{make}$
- $\mathbf{Pre}(\widehat{\mathbf{a}}) = \omega p_{ban} + p_{bake}$
  $\mathbf{C}(\widehat{\mathbf{a}}) = \omega p_{ban} + \omega p_{cake}$



Completing the K&M tree

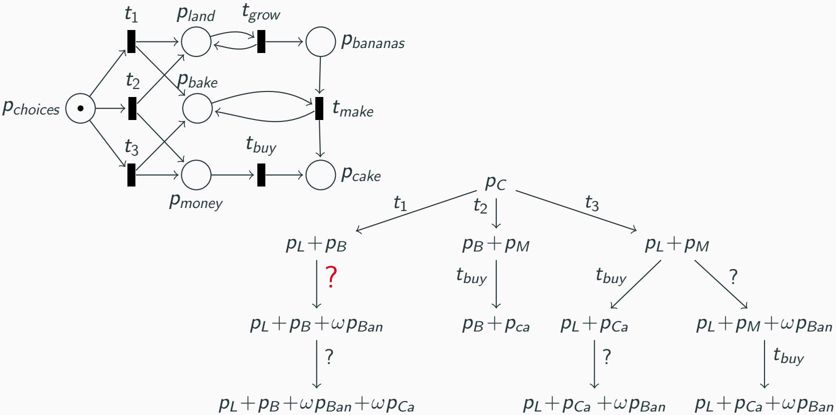


Completing the K&M tree



$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban}$$

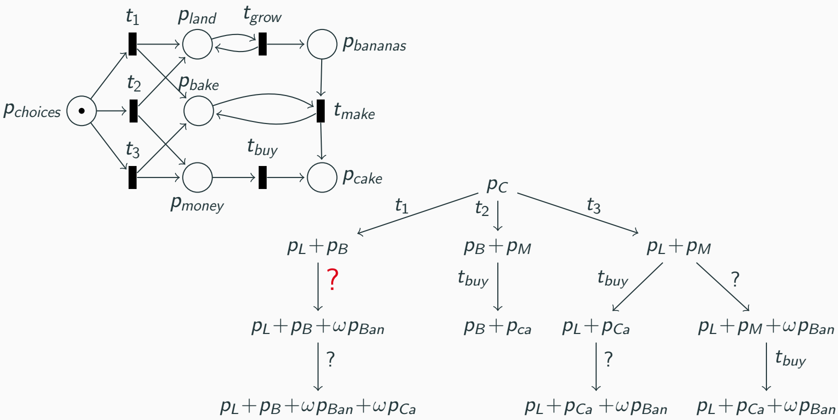
Completing the K&M tree



$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban}$$

$\mathbf{a_1} = \widehat{t_{grow}}$, $\mathbf{Pre(a_1)} = p_L$; $\mathbf{C(a_1)} = \omega p_{Ban}$

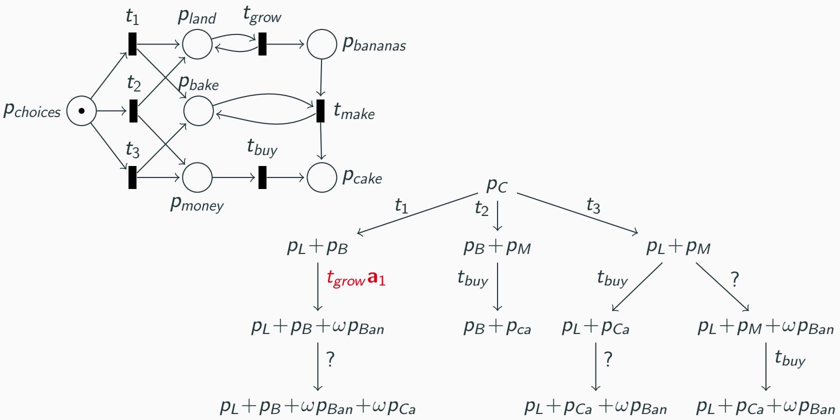
Completing the K&M tree



$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban} \xrightarrow{\mathbf{a}_1} p_L + p_B + \omega p_{ban}$$

$\mathbf{a}_1 = \widehat{t_{grow}}$, $\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{Ban}$

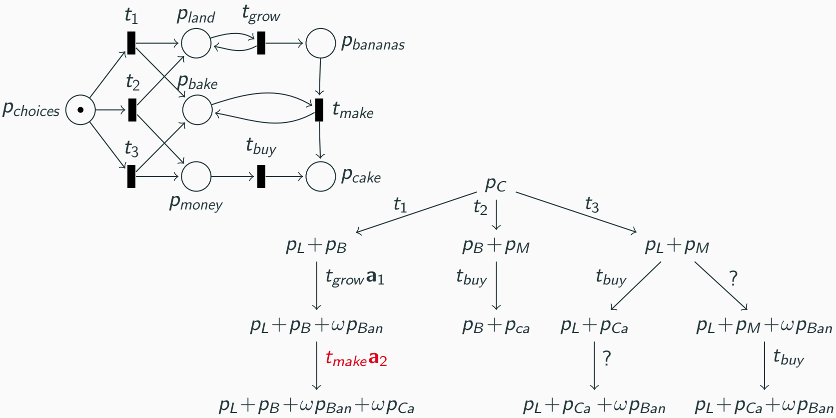
Completing the K&M tree



$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban} \xrightarrow{\mathbf{a}_1} p_L + p_B + \omega p_{ban}$$

$\mathbf{a}_1 = \widehat{t_{grow}}$, $\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\qquad$ $\mathbf{C}(\mathbf{a}_1) = \omega p_{Ban}$

Completing the K&M tree

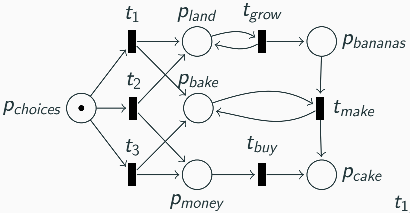


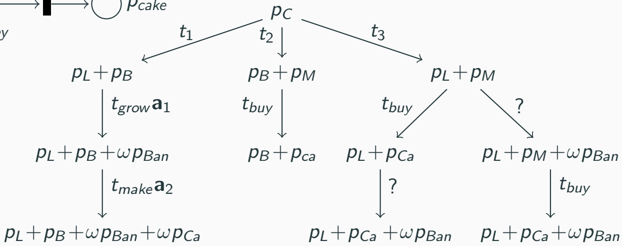
$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban} \xrightarrow{\mathbf{a}_1} p_L + p_B + \omega p_{ban}$$

$\mathbf{a}_1 = \widehat{t_{grow}}$, $\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{Ban}$

$\mathbf{a}_2 = \widehat{t_{make}}$, $\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{Ban}$

Completing the K&M tree



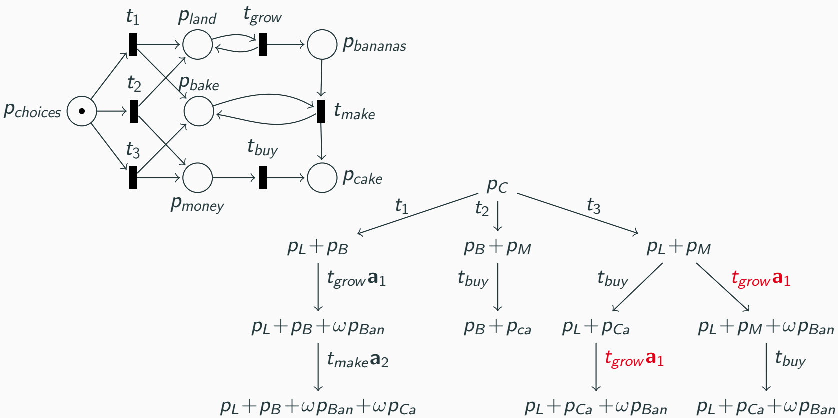


$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban} \xrightarrow{\mathbf{a}_1} p_L + p_B + \omega p_{ban}$$

$\mathbf{a}_1 = \widehat{t_{grow}}$, $\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\quad \mathbf{C}(\mathbf{a}_1) = \omega p_{Ban}$

$\mathbf{a}_2 = \widehat{t_{make}}$, $\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{Ban}$

Completing the K&M tree



$$p_L + p_B \xrightarrow{\sigma} p_L + p_B + p_{ban} \xrightarrow{\mathbf{a}_1} p_L + p_B + \omega p_{ban}$$

$\mathbf{a}_1 = \widehat{t_{grow}}$, $\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{Ban}$

$\mathbf{a}_2 = \widehat{t_{make}}$, $\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{Ban}$

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$:

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $\mathit{Cover}(\mathcal{N}, \mathbf{m}_0) \subseteq [\![\lambda(V)]\!]$:

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $\mathit{Cover}(\mathcal{N}, \mathbf{m}_0) \subseteq [\![\lambda(V)]\!]$:

Reachability tree:

$\forall \mathbf{m} \in \mathit{Reach}(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) = \mathbf{m}$
- $u \in$ Front , $\exists \sigma \in T^*$ s.t.
  $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

Correctness Proof

Consistency $\llbracket\lambda(V)\rrbracket \subseteq \mathit{Cover}(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $\mathit{Cover}(\mathcal{N}, \mathbf{m}_0) \subseteq \llbracket\lambda(V)\rrbracket$:

Reachability tree:

$\forall \mathbf{m} \in \mathit{Reach}(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) = \mathbf{m}$
- $u \in$ Front , $\exists \sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

K&M:

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq Cover(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $Cover(\mathcal{N}, \mathbf{m}_0) \subseteq [\![\lambda(V)]\!]$:

Reachability tree:

$\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) = \mathbf{m}$
- $u \in$ Front , $\exists \sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

K&M:

$\forall \mathbf{m} \in Cover(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq Cover(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $Cover(\mathcal{N}, \mathbf{m}_0) \subseteq [\![\lambda(V)]\!]$:

Reachability tree:

$\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) = \mathbf{m}$
- $u \in$ Front , $\exists \sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

K&M:

$\forall \mathbf{m} \in Cover(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) \geq \mathbf{m}$

Correctness Proof

Consistency $[\![\lambda(V)]\!] \subseteq Cover(\mathcal{N}, \mathbf{m}_0)$:

- For every edges $u \xrightarrow{\sigma} v$, one has $\lambda(u) \xrightarrow{\sigma} \lambda(v)$.

Completeness $Cover(\mathcal{N}, \mathbf{m}_0) \subseteq [\![\lambda(V)]\!]$:

Reachability tree:

$\forall \mathbf{m} \in Reach(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

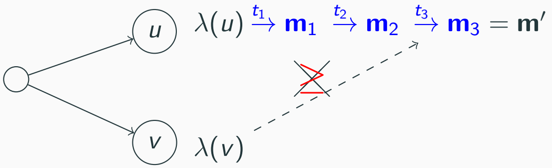
- $u \notin$ Front and $\lambda(u) = \mathbf{m}$
- $u \in$ Front , $\exists \sigma \in T^*$ s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}$

K&M:

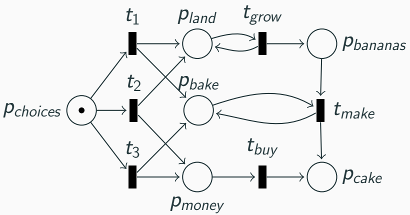
$\forall \mathbf{m} \in Cover(\mathcal{N}, \mathbf{m}_0)$, $\exists u \in V$ s.t.:

- $u \notin$ Front and $\lambda(u) \geq \mathbf{m}$
- $u \in$ Front, $\exists \sigma$ and exploring sequence s.t. $\lambda(u) \xrightarrow{\sigma} \mathbf{m}' \geq \mathbf{m}$

Exploring sequence(Illustration):

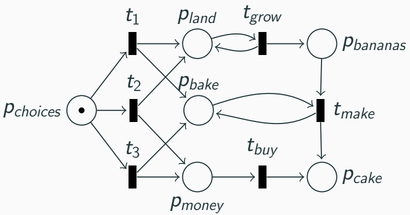


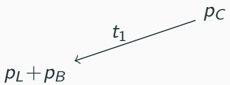
K&M With Accelerations



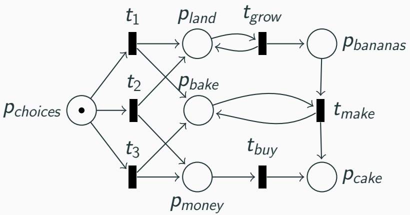
p_C

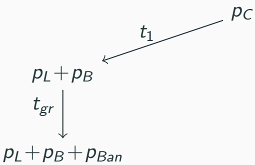
K&M With Accelerations



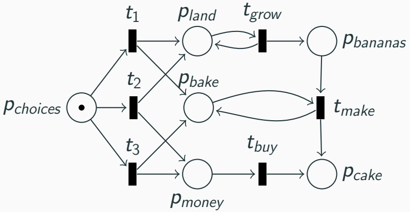


K&M With Accelerations

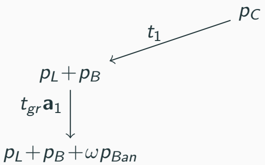




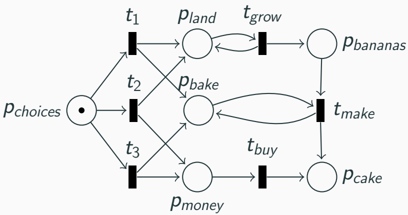
K&M With Accelerations



$\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{ban}$

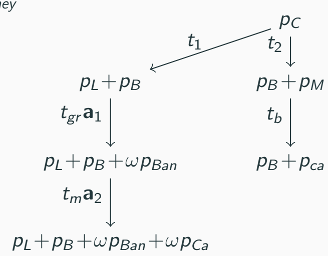


K&M With Accelerations

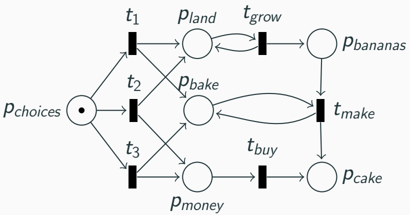


$\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{ban}$

$\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{ban}$

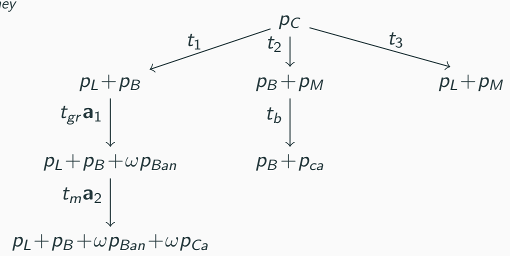


K&M With Accelerations

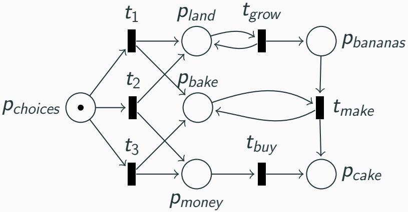


$\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{ban}$

$\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{ban}$

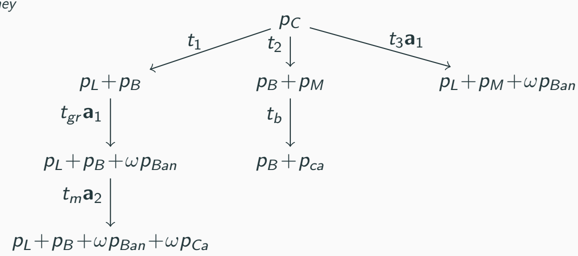


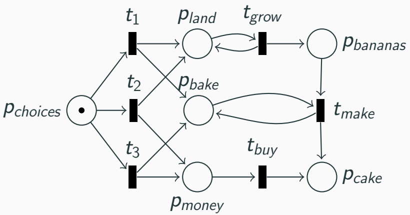
K&M With Accelerations



$\mathbf{Pre}(\mathbf{a}_1) = p_L$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{ban}$

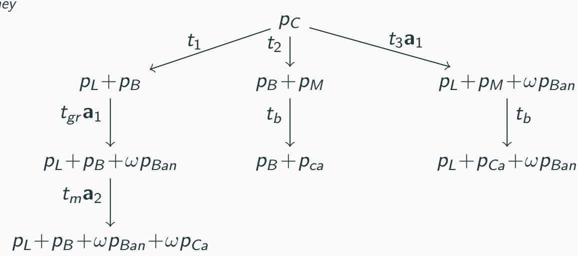
$\mathbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{ban}$





$$\textbf{Pre}(\mathbf{a}_1) = p_L;\ \textbf{C}(\mathbf{a}_1) = \omega p_{ban}$$

$$\textbf{Pre}(\mathbf{a}_2) = p_B + \omega p_{Ban};\ \textbf{C}(\mathbf{a}_2) = \omega p_{Ca} + \omega p_{ban}$$



K&M With Accelerations

K&M algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then** Accelerate(u, v)

Explore(v)

K&M With Accelerations

K&M with accelerations algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

Use Acc on $\lambda(v)$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**

$\mathbf{a} = \text{Accelerate}(u \rightarrow v)$

$\text{Acc} = \text{Acc} \cup \{\mathbf{a}\}$

Explore(v)

K&M With Accelerations

K&M with accelerations algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;
Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$
Pop $v \in Front$
Use Acc on $\lambda(v)$
if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue
if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**
$\mathbf{a} =$ Accelerate$(u \to v)$
Acc $=$ Acc $\cup \{\mathbf{a}\}$
Explore(v)

Correctness proof

Similar to our K&M proof!

Can we do better?

Can we do better?

Maximal number of nodes:

Can we do better?

Maximal number of nodes:

- *Reachability enumeration*: ∞

Can we do better?

Maximal number of nodes:

- *Reachability enumeration*: ∞
- *K&M* : 11

Can we do better?

Maximal number of nodes:

- *Reachability enumeration*: ∞
- *K&M* : 11
- *K&M* + *Accelerations* : 8 (+2 acc)

Can we do better?

Maximal number of nodes:

- *Reachability enumeration*: ∞
- *K&M* : 11
- *K&M* + *Accelerations* : 8 (+2 acc)

Size of Clover is 4

Can we do better?

Maximal number of nodes:

- *Reachability enumeration*: ∞
- *K&M* : 11
- *K&M + Accelerations* : 8 (+2 acc)

Size of Clover is 4

Can we do better?

Minimal Coverability Tree

Minimal Coverability Tree

MCT[Finkel 93]:

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

. . . but incomplete [Finkel 05].

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

. . . but incomplete [Finkel 05].

Bug very subtle.

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

. . . but incomplete [Finkel 05].

Bug very subtle.

Attempts to fix:

MP[Reynier 13/19]: Deactivates nodes.

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

. . . but incomplete [Finkel 05].

Bug very subtle.

Attempts to fix:

MP[Reynier 13/19]: Deactivates nodes.

VH[Valmari 14/16]: Deletes less nodes, and provides other heuristic optimization.

Minimal Coverability Tree

MCT[Finkel 93]:

Improved on K&M algorithm, by keeping V an antichain at any step of the algorithm.

... but incomplete [Finkel 05].

Bug very subtle.

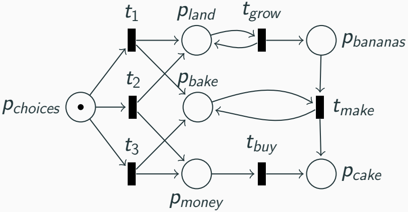
Attempts to fix:

MP[Reynier 13/19]: Deactivates nodes.

VH[Valmari 14/16]: Deletes less nodes, and provides other heuristic optimization.

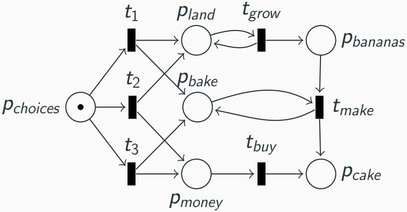
`CovProc`[Geeraerts 10] Alternative construction

Minimal Coverability Tree - MinCov



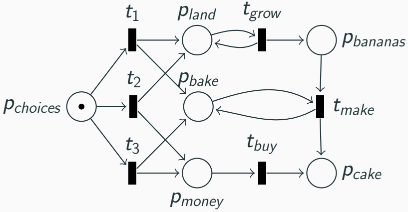
p_C

Minimal Coverability Tree - MinCov



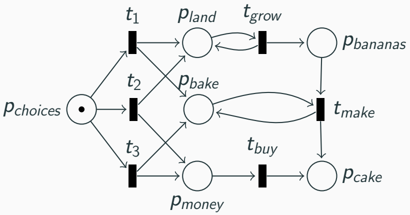
$$p_C \xrightarrow{t_2} p_B + p_M$$

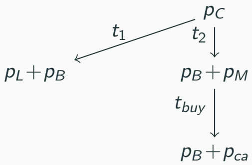
Minimal Coverability Tree - MinCov



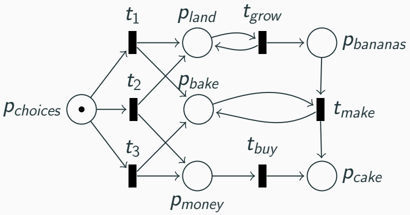
$$
\begin{array}{c}
p_C \\
\downarrow t_2 \\
p_B + p_M \\
\downarrow t_{buy} \\
p_B + p_{ca}
\end{array}
$$

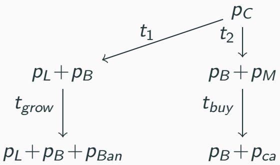
Minimal Coverability Tree - MinCov



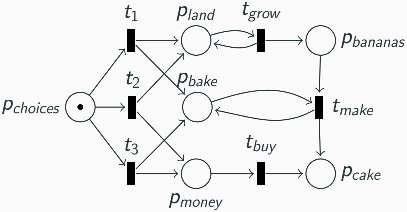


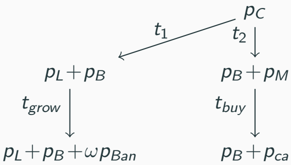
Minimal Coverability Tree - MinCov



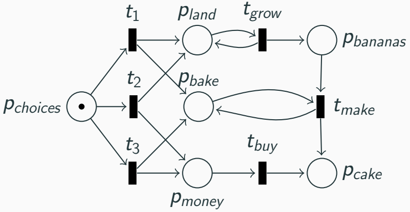


Minimal Coverability Tree - MinCov

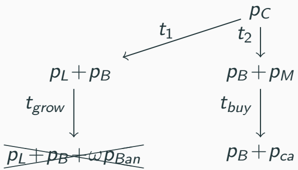




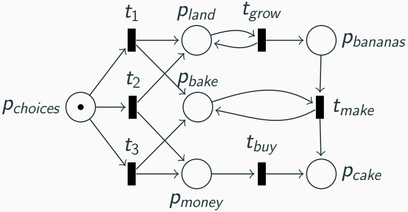
Minimal Coverability Tree - MinCov



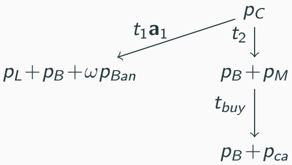
$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$



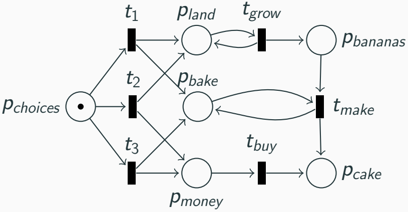
Minimal Coverability Tree - MinCov



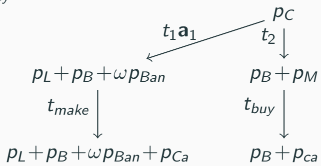
$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$



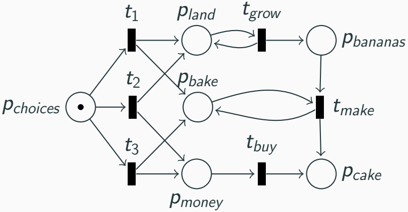
Minimal Coverability Tree - MinCov



$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

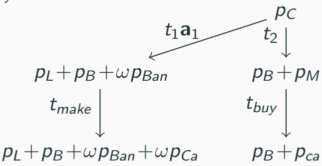


Minimal Coverability Tree - MinCov

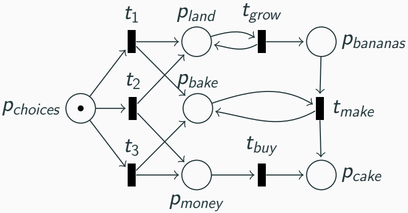


$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\mathbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{cake}$

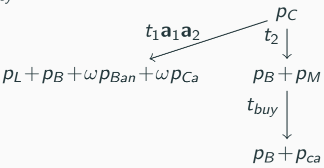


Minimal Coverability Tree - MinCov

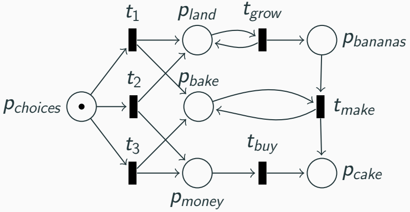


$\textbf{Pre}(\mathbf{a}_1) = P_{land}$; $\textbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\textbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\textbf{C}(\mathbf{a}_2) = \omega p_{cake}$

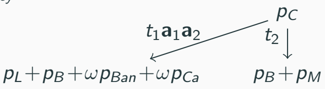


Minimal Coverability Tree - MinCov

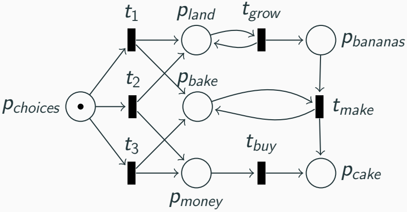


$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\mathbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{cake}$

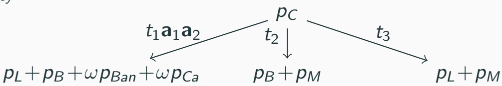


Minimal Coverability Tree - MinCov

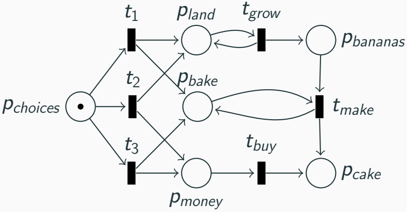


$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\mathbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{cake}$

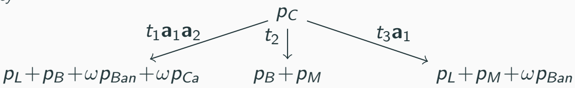


Minimal Coverability Tree - MinCov

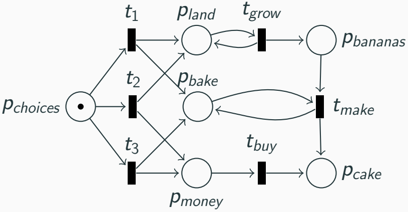


$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\mathbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{cake}$

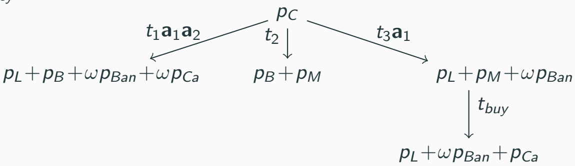


Minimal Coverability Tree - MinCov

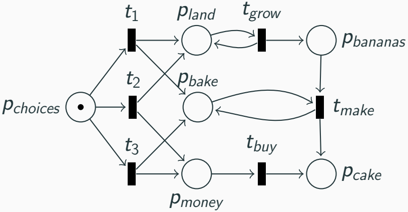


$\mathbf{Pre}(\mathbf{a}_1) = P_{land}$; $\mathbf{C}(\mathbf{a}_1) = \omega p_{bananas}$

$\mathbf{Pre}(\mathbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\mathbf{C}(\mathbf{a}_2) = \omega p_{cake}$

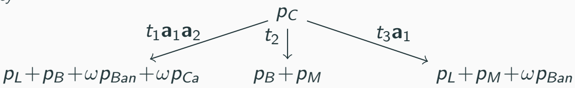


Minimal Coverability Tree - MinCov

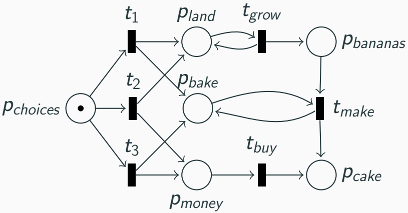


$\textbf{Pre}(\textbf{a}_1) = P_{land}$; $\textbf{C}(\textbf{a}_1) = \omega p_{bananas}$

$\textbf{Pre}(\textbf{a}_2) = P_{bake} + \omega p_{bananas}$; $\textbf{C}(\textbf{a}_2) = \omega p_{cake}$

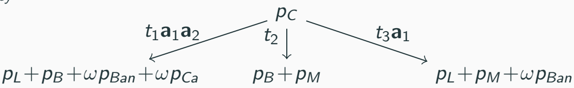


Minimal Coverability Tree - MinCov



$\textbf{Pre}(\textbf{a}_1) = P_{land};\ \textbf{C}(\textbf{a}_1) = \omega p_{bananas}$

$\textbf{Pre}(\textbf{a}_2) = P_{bake} + \omega p_{bananas};\ \textbf{C}(\textbf{a}_2) = \omega p_{cake}$



We have the Clover!

K&M With Accelerations

K&M with accelerations algorithm:

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;

Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

Try use Acc on $\lambda(v)$

if $\lambda(v) \leq \lambda(u)$ for $u \in Anc(v)$, **then** Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**

$\mathbf{a} = \text{Accelerate}(u \rightarrow v)$

$\text{Acc} = \text{Acc} \cup \{\mathbf{a}\}$

Explore(v)

K&M With Accelerations

MinCov

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;
Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$
Pop $v \in Front$
Try use Acc on $\lambda(v)$
if $\lambda(v) \leq \lambda(u)$ for $u \in V$, **then** $delete(v)$; Continue
if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**
$\mathbf{a}$ = Accelerate($u \rightarrow v$)
Acc = Acc $\cup \{\mathbf{a}\}$
$prune(u)$; Continue
for any $u \in V$, if $\lambda(v) \geq \lambda(u)$ **then** $prune(u)$, $delete(u)$;
Explore(v)

K&M With Accelerations

MinCov

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;
Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$

Pop $v \in Front$

Try use Acc on $\lambda(v)$

if $\lambda(v) \leq \lambda(u)$ for $u \in V$, **then** $delete(v)$; Continue

if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**

$\mathbf{a}$ = Accelerate($u \to v$)

Acc = Acc $\cup \{\mathbf{a}\}$

$prune(u)$; Continue

for any $u \in V$, if $\lambda(v) \geq \lambda(u)$ **then** $prune(u)$, $delete(u)$;

Explore(v)

Correctness proof

K&M With Accelerations

`MinCov`

Variables: (V, E, λ)- a labeled tree; $Front \subseteq V$;
Acc- a set of ω-transitions;

Main loop:

While $Front \neq \emptyset$
Pop $v \in Front$
Try use Acc on $\lambda(v)$
if $\lambda(v) \leq \lambda(u)$ for $u \in V$, **then** $delete(v)$; Continue
if $\lambda(v) > \lambda(u)$ for $u \in Anc(v)$, **then**
$\mathbf{a}$ = Accelerate($u \rightarrow v$)
Acc = Acc $\cup \{\mathbf{a}\}$
$prune(u)$; Continue
for any $u \in V$, if $\lambda(v) \geq \lambda(u)$ **then** $prune(u)$, $delete(u)$;
Explore(v)

Correctness proof

Similar to our K&M proof! (with minor modifications)

MinCov

Goals:

Goals:

- Computing the coverability set

MinCov

Goals:

- Computing the coverability set
- Solving the coverability problem

MinCov

Goals:

- Computing the coverability set
- Solving the coverability problem

Implementation features

Goals:

- Computing the coverability set
- Solving the coverability problem

Implementation features

- Written in Python3, using the Numpy and Z3-solver libraries.
- $\approx$ 2000 lines.
- Imports Petri nets in ".spec" format from Mist.
- Can be found in https://github.com/IgorKhm/MinCov

Benchmarks

123 benchmarks (literature)

Benchmarks

123 benchmarks (literature)

1078 benchmarks (random)

Benchmarks

123 benchmarks (literature)

	T/O	Time	#Nodes
MinCov	16	18127	48218
VH	15	14873	75225
MP	24	23904	478681
CovProc	49	47089	N/A
AF	19	19223	45660

1078 benchmarks (random)

Benchmarks

123 benchmarks (literature)

	T/O	Time	#Nodes
MinCov	16	18127	48218
VH	15	14873	75225
MP	24	23904	478681
CovProc	49	47089	N/A
AF	19	19223	45660

1078 benchmarks (random)

	T/O	Time	#Nodes
MinCov	146	159940	1291066
VH	111	110431	2454490
MP	231	260608	31354531
CovProc	N/A	N/A	N/A
AF	163	178322	1267076

Benchmarks

123 benchmarks (literature)

	T/O	Time	#Nodes
MinCov	16	18127	48218
VH	15	14873	75225
MP	24	23904	478681
CovProc	49	47089	N/A
AF	19	19223	45660

1078 benchmarks (random)

	T/O	Time	#Nodes
MinCov	146	159940	1291066
VH	111	110431	2454490
MP	231	260608	31354531
CovProc	N/A	N/A	N/A
AF	163	178322	1267076

- MinCov is twice as economical space-wise compared to the other tools

Benchmarks

123 benchmarks (literature)

	T/O	Time	#Nodes
MinCov	16	18127	48218
VH	15	14873	75225
MP	24	23904	478681
CovProc	49	47089	N/A
AF	19	19223	45660

1078 benchmarks (random)

	T/O	Time	#Nodes
MinCov	146	159940	1291066
VH	111	110431	2454490
MP	231	260608	31354531
CovProc	N/A	N/A	N/A
AF	163	178322	1267076

- MinCov is twice as economical space-wise compared to the other tools
- MinCov only 1.2 slower then the fastest tool.

Comparing MinCov and qCover for Coverability

Comparing `MinCov` and `qCover` for Coverability

Blondin et al. (`qCover`) (2016)

Combining backward exploration with forward over-approximation

Comparing MinCov and qCover for Coverability

Blondin et al. (qCover) (2016)

Combining backward exploration with forward over-approximation

MinCov

Partial forward construction of the coverability set

Comparing MinCov and qCover for Coverability

Blondin et al. (qCover) (2016)

Combining backward exploration with forward over-approximation

MinCov

Partial forward construction of the coverability set

	Covered (60)		Not covered(115)		Total	
	Time	T/O	Time	T/O	T/O	Time
MinCov	1754	1	51323	53	54	53077
qCover	26467	26	11865	11	37	38332

Comparing MinCov and qCover for Coverability

Blondin et al. (qCover) (2016)

Combining backward exploration with forward over-approximation

MinCov

Partial forward construction of the coverability set

	Covered (60)		Not covered(115)		Total	
	Time	T/O	Time	T/O	T/O	Time
MinCov	1754	1	51323	53	54	53077
qCover	26467	26	11865	11	37	38332

Complementary tools!

Comparing MinCov and qCover for Coverability

Blondin et al. (qCover) (2016)

Combining backward exploration with forward over-approximation

MinCov

Partial forward construction of the coverability set

	Covered (60)		Not covered(115)		Total	
	Time	T/O	Time	T/O	T/O	Time
MinCov	1754	1	51323	53	54	53077
qCover	26467	26	11865	11	37	38332
MinCov ∥ qCover[1]	1841	2	13493	11	13	15334

1. $\text{Time}(\texttt{MinCov} \parallel \texttt{qCover}) = 2\min(\text{Time}(\texttt{MinCov}), \text{Time}(\texttt{qCover}))$.

Conclusion and Future Work

Contributions

Conclusion and Future Work

Contributions

- Commodification of accelerations

Conclusion and Future Work

Contributions

- Commodification of accelerations
- Fixing the minimal coverability tree algorithm

Conclusion and Future Work

Contributions

- Commodification of accelerations
- Fixing the minimal coverability tree algorithm
- Implantation of `MinCov`, which out preforms all other tools space-wise.

Future Work

Conclusion and Future Work

Contributions

- Commodification of accelerations
- Fixing the minimal coverability tree algorithm
- Implantation of `MinCov`, which out preforms all other tools space-wise.

Future Work

- Combining the power of `qCover` and `MinCov`

Conclusion and Future Work

Contributions

- Commodification of accelerations
- Fixing the minimal coverability tree algorithm
- Implantation of `MinCov`, which out preforms all other tools space-wise.

Future Work

- Combining the power of `qCover` and `MinCov`
- Further development of `MinCov`

Conclusion and Future Work

Contributions

- Commodification of accelerations
- Fixing the minimal coverability tree algorithm
- Implantation of `MinCov`, which out preforms all other tools space-wise.

Future Work

- Combining the power of `qCover` and `MinCov`
- Further development of `MinCov`

Minimal Acceleration

Minimal Acceleration

Minimal Acceleration

Order: Given two abstractions $\mathbf{a}, \mathbf{a}'$:

$$\mathbf{a} \preceq \mathbf{a}' \overset{def}{\iff} \mathbf{Pre}(\mathbf{a}) \leq \mathbf{Pre}(\mathbf{a}') \wedge \mathbf{C}(\mathbf{a}) \geq \mathbf{C}(\mathbf{a}')$$

Minimal Acceleration

Order: Given two abstractions $\mathbf{a}, \mathbf{a}'$:

$$\mathbf{a} \preceq \mathbf{a}' \overset{def}{\iff} \mathbf{Pre}(\mathbf{a}) \leq \mathbf{Pre}(\mathbf{a}') \wedge \mathbf{C}(\mathbf{a}) \geq \mathbf{C}(\mathbf{a}')$$

Proposition

Given a Petri net $\mathcal{N}$ and Acc the set of all $\mathcal{N}$'s accelerations, then:

Minimal Acceleration

Order: Given two abstractions $\mathbf{a}, \mathbf{a}'$:

$$\mathbf{a} \preceq \mathbf{a}' \overset{def}{\iff} \mathbf{Pre}(\mathbf{a}) \leq \mathbf{Pre}(\mathbf{a}') \wedge \mathbf{C}(\mathbf{a}) \geq \mathbf{C}(\mathbf{a}')$$

Proposition

Given a Petri net $\mathcal{N}$ and Acc the set of all $\mathcal{N}$'s accelerations, then:

- (Acc, $\preceq$) is a well-ordered with

Minimal Acceleration

Order: Given two abstractions $\mathbf{a}, \mathbf{a}'$:

$$\mathbf{a} \preceq \mathbf{a}' \overset{def}{\iff} \mathbf{Pre}(\mathbf{a}) \leq \mathbf{Pre}(\mathbf{a}') \wedge \mathbf{C}(\mathbf{a}) \geq \mathbf{C}(\mathbf{a}')$$

Proposition

Given a Petri net $\mathcal{N}$ and Acc the set of all $\mathcal{N}$'s accelerations, then:

- $(\mathrm{Acc}, \preceq)$ is a well-ordered with
- Given a minimal acceleration $\mathbf{a}$, then $\mathbf{Pre}(\mathbf{a}) < 2{-}EXP(\mathcal{N})$ - using [Leroux-19].

Minimal Acceleration

Order: Given two abstractions $\mathbf{a}, \mathbf{a}'$:

$$\mathbf{a} \preceq \mathbf{a}' \overset{def}{\iff} \mathbf{Pre}(\mathbf{a}) \leq \mathbf{Pre}(\mathbf{a}') \wedge \mathbf{C}(\mathbf{a}) \geq \mathbf{C}(\mathbf{a}')$$

Proposition

Given a Petri net $\mathcal{N}$ and Acc the set of all $\mathcal{N}$'s accelerations, then:

- $(\text{Acc}, \preceq)$ is a well-ordered with
- Given a minimal acceleration $\mathbf{a}$, then $\mathbf{Pre}(\mathbf{a}) < 2{-}EXP(\mathcal{N})$ - using [Leroux-19].
- $\exists A \subset \text{Acc}$ such that $\uparrow A = \text{Acc}$ and $|A| \leq 3{-}EXP(\mathcal{N})$