Abstraction-based Incremental Inductive Coverability for Petri nets

Jiawen Kang YunJun Bai Li Jiao

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China

June 2021

Abstraction

- Check the coverability problem of Petri nets

- Combine IC3 with place-merge abstraction (IC3+PMA)

Outline

① Preliminaries

② IC3 algorithm for PN

③ Place-merge abstraction (PMA)

④ IC3+PMA algorithm

⑤ Experiments

Preliminaries

Definition

A Petri net is a tuple $N = \langle P, T, W, m_0 \rangle$ where:

- P is a finite set of places
- T is a finite set of transitions such that $P \cap T = \emptyset$
- W is an arc function: $(P \times T) \cup (T \times P) \to \mathbb{N}$ describing the relationship between places and transitions
- m_0 is the initial marking. A marking $m \in \mathbb{N}^{|P|}$ is a vector specifying a number $m(p)$ of tokens for each place $p \in P$.

for vector $m_1, m_2 \in \mathbb{N}^{|P|}$

$m_1 \preccurlyeq m_2$ iff for every $p \in P$: $m_1(p) \leq m_2(p)$

Preliminaries

Definition

Let N be a Petri net.

- $pre(m) = \{m' | \exists t \in T: m' \rightarrow m\}$
- $Reach_i$ contains all reachable markings from m_0 within i steps.
- $Reach = \bigcup_{i \geq 0} Reach_i$ contains all reachable markings from m_0.

Preliminaries

Coverability problem

Let N be a Petri net, m_t the target marking.

- The coverability problem is to prove whether there exists a reachable marking $m_r \in Reach$ such that $m_t \preccurlyeq m_r$.

Coverability problem

Let N be a Petri net, m_t the target marking.

- The coverability problem is to prove whether there exists a reachable marking $m_r \in Reach$ such that $m_t \preccurlyeq m_r$.

- The coverable set of N within i steps is $Cover_i = Reach_i^{\downarrow}$
- The coverable set of N is $Cover = Reach^{\downarrow}$

IC3 algorithm for Petri nets

IC3 is a state-of-art of model checking

Efficient implementation of IC3 to check the coverability problem of Petri nets without using SMT solvers

IC3 algorithm for Petri nets

IC3 maintains a sequence $F_0, F_1 \ldots F_k$

where F_i is a downward-closed set called $frame$ that over-approximates the coverable set within i steps.

The algorithm generally proceeds by alternating two phases: the blocking phase and the propagation phase.

IC3 algorithm for Petri nets

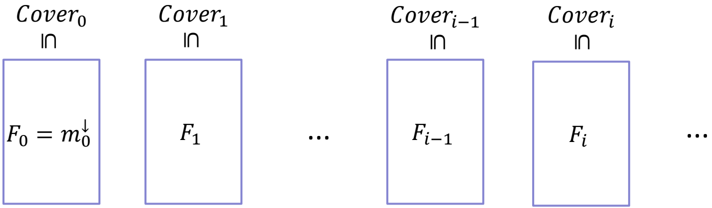
Blocking phase: $block(a, i)$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$ → try to prove $a^{\uparrow}$ is unreachable within i steps

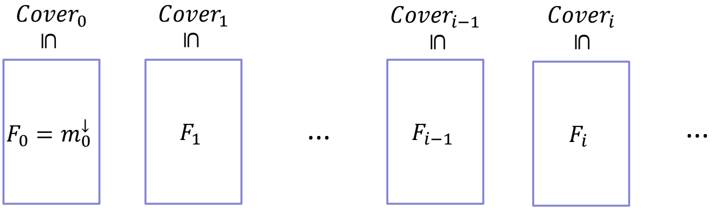
IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



IC3 algorithm for Petri nets

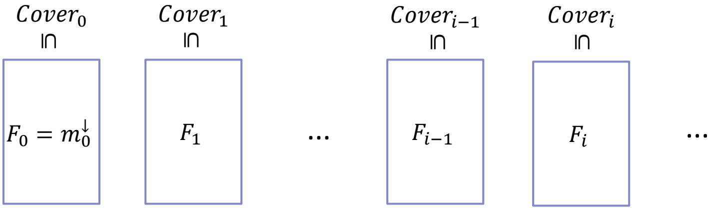
Blocking phase: $block(a, i)$



given a pair $(\boldsymbol{a}, \boldsymbol{i})$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$

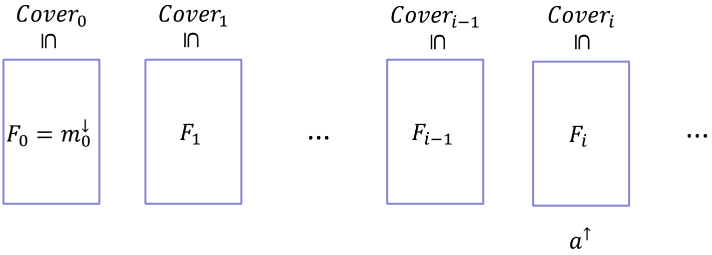


given a pair (a, i)

try to prove $a^{\uparrow}$ is unreachable within i steps

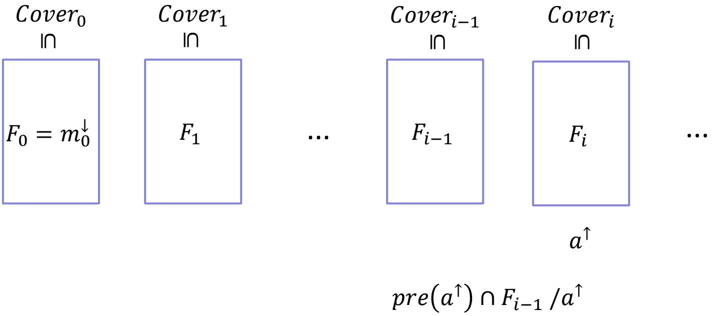
IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



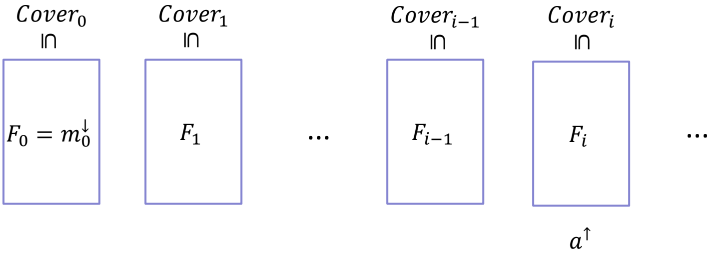
IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



IC3 algorithm for Petri nets

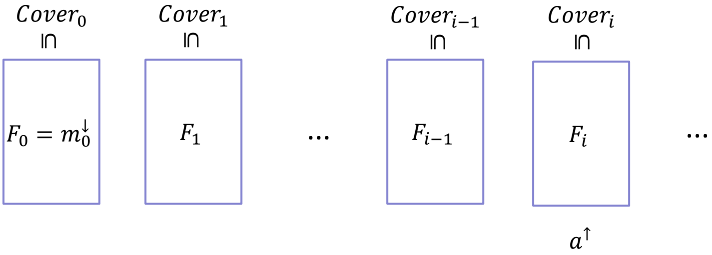
Blocking phase: $block(a, i)$



$$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$

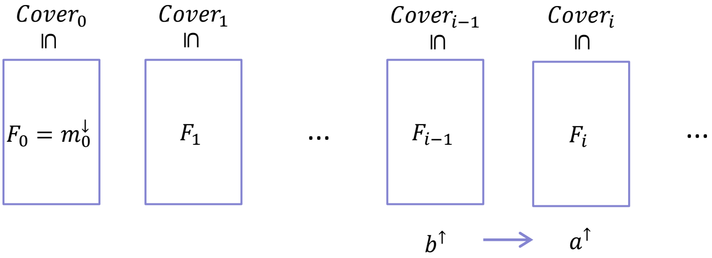


$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

extract an unselected marking b
from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$

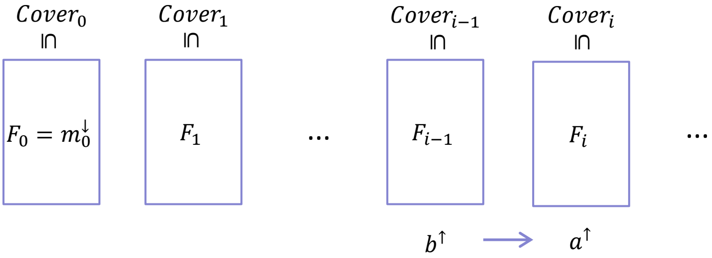


$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

extract an unselected marking b
from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



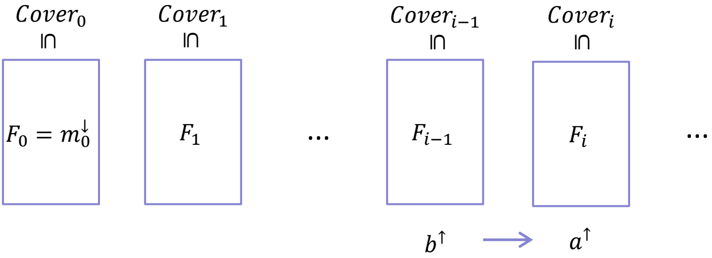
$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

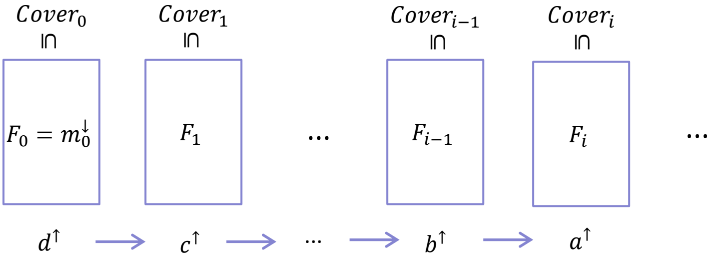
extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

try to prove $b^{\uparrow}$ is unreachable within $i-1$ steps

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

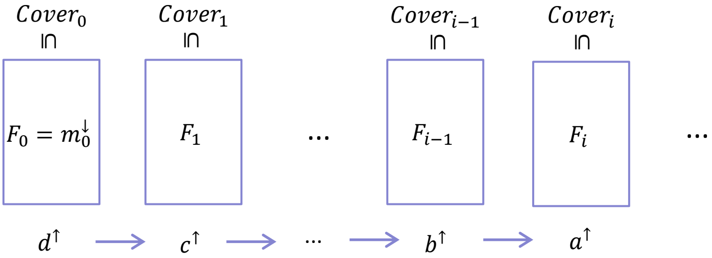
extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

try to prove $b^{\uparrow}$ is unreachable within $i-1$ steps

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



finally generate a new pair $(d, 0)$

$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

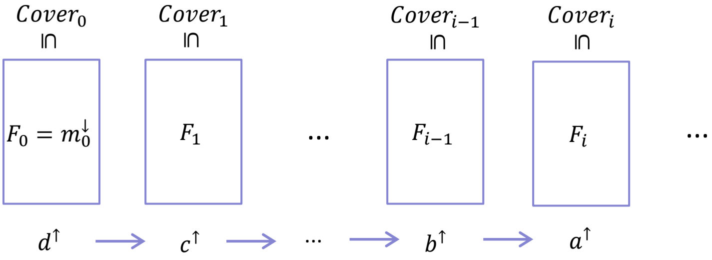
extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

try to prove $b^{\uparrow}$ is unreachable within $i-1$ steps

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



finally generate a new pair $(d, 0)$

find a path from $m_0^{\downarrow}$ to $a^{\uparrow}$

$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

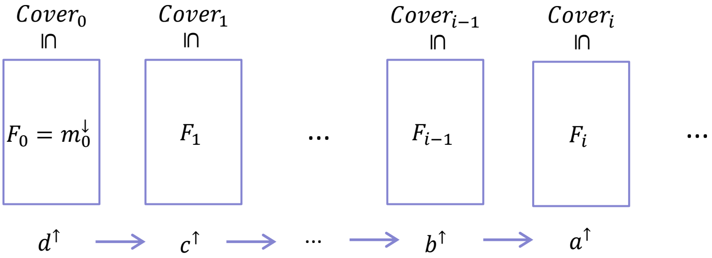
extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

try to prove $b^{\uparrow}$ is unreachable within $i-1$ steps

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



finally generate a new pair $(d, 0)$

find a path from $m_0^{\downarrow}$ to $a^{\uparrow}$

fail to block a at F_i
i.e. a is coverable

$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} \neq \emptyset$

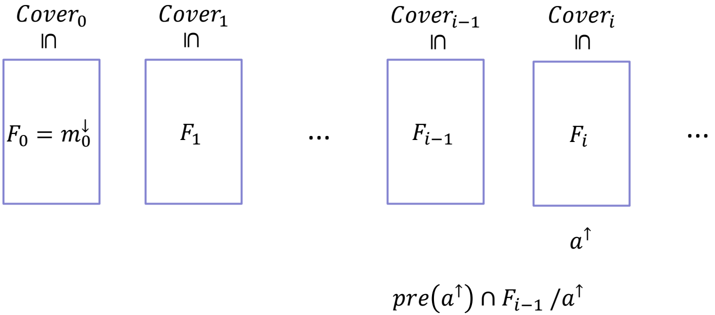
extract an unselected marking b from $pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow}$

generate a new pair $(b, i-1)$
$block(b, i-1)$

try to prove $b^{\uparrow}$ is unreachable within $i-1$ steps

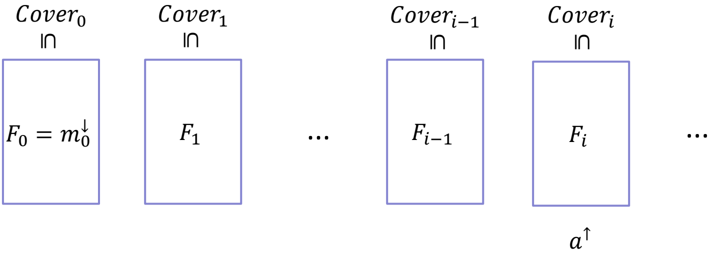
IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



IC3 algorithm for Petri nets

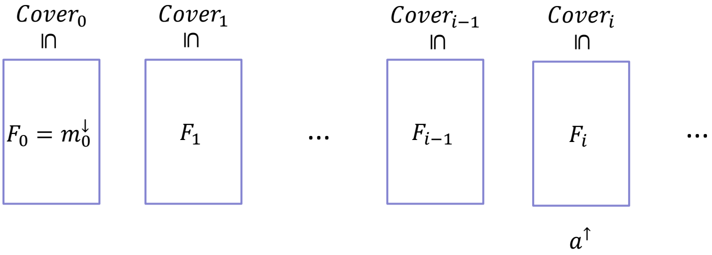
Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

$a^{\uparrow}$ cannot be reachable in 1 step from $Cover_{i-1}$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$

$Cover_0 \supseteq F_0 = m_0^{\downarrow}$ $\quad Cover_1 \supseteq F_1$ $\quad \cdots \quad Cover_{i-1} \supseteq F_{i-1}$ $\quad Cover_i \supseteq F_i$ $\quad \cdots$

$\emptyset \longrightarrow a^{\uparrow}$

$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

$a^{\uparrow}$ cannot be reachable in 1 step from $Cover_{i-1}$

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$

$Cover_0 \supseteq F_0 = m_0^{\downarrow}$ $\quad$ $Cover_1 \supseteq F_1$ $\quad \cdots \quad$ $Cover_{i-1} \supseteq F_{i-1}$ $\quad$ $Cover_i \supseteq F_i$ $\quad \cdots$

$\emptyset \longrightarrow a^{\uparrow}$

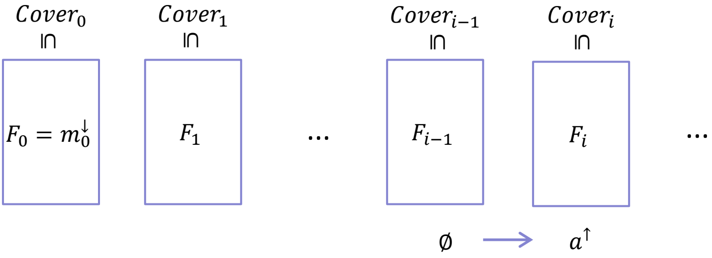
$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

$a^{\uparrow}$ cannot be reachable in 1 step from $Cover_{i-1}$

a is uncoverable within i steps

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

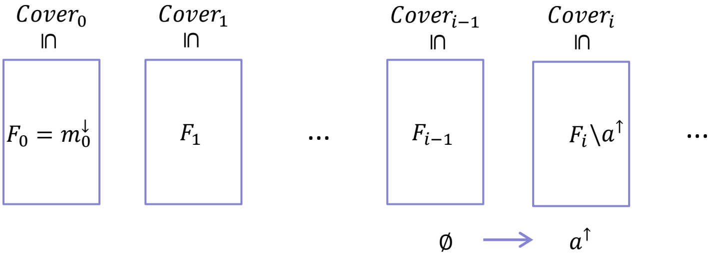
$a^{\uparrow}$ cannot be reachable in 1 step from $Cover_{i-1}$

a is uncoverable within i steps

$a^{\uparrow}$ can be removed from the coverable set F_i

IC3 algorithm for Petri nets

Blocking phase: $block(a, i)$



$pre(a^{\uparrow}) \cap F_{i-1} / a^{\uparrow} = \emptyset$

$a^{\uparrow}$ cannot be reachable in 1 step from $Cover_{i-1}$

a is uncoverable within i steps

$a^{\uparrow}$ can be removed from the coverable set F_i

IC3 algorithm for Petri nets

input $N = \langle P, T, W, m_0 \rangle$ and m_t
initialize $F_0 = m_0^{\downarrow}, F_1 = \mathbb{N}^{|P|}, k = 1$

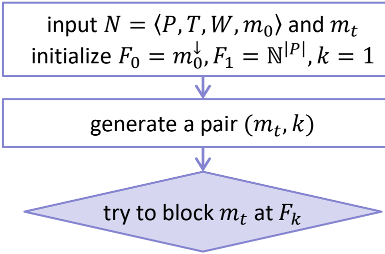
IC3 algorithm for Petri nets

input $N = \langle P, T, W, m_0 \rangle$ and m_t
initialize $F_0 = m_0^{\downarrow}, F_1 = \mathbb{N}^{|P|}, k = 1$

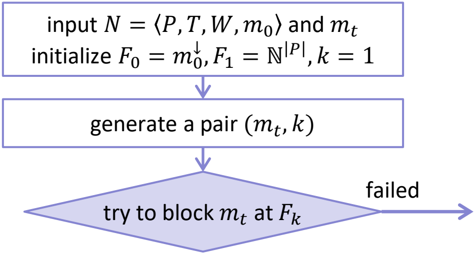
↓

generate a pair (m_t, k)

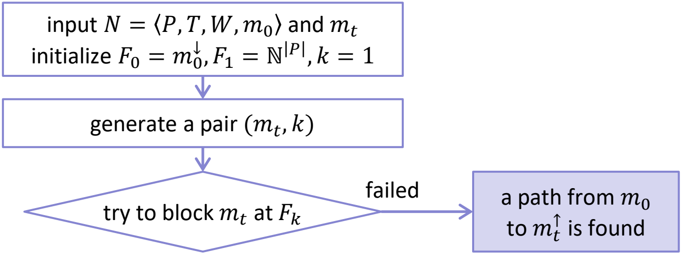
IC3 algorithm for Petri nets



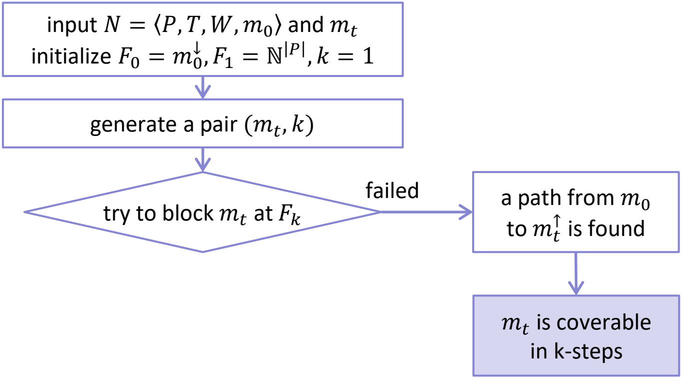
IC3 algorithm for Petri nets



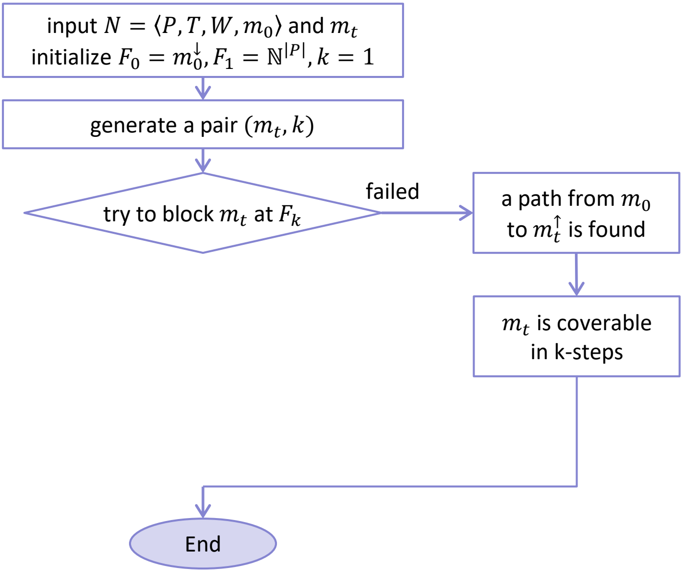
IC3 algorithm for Petri nets



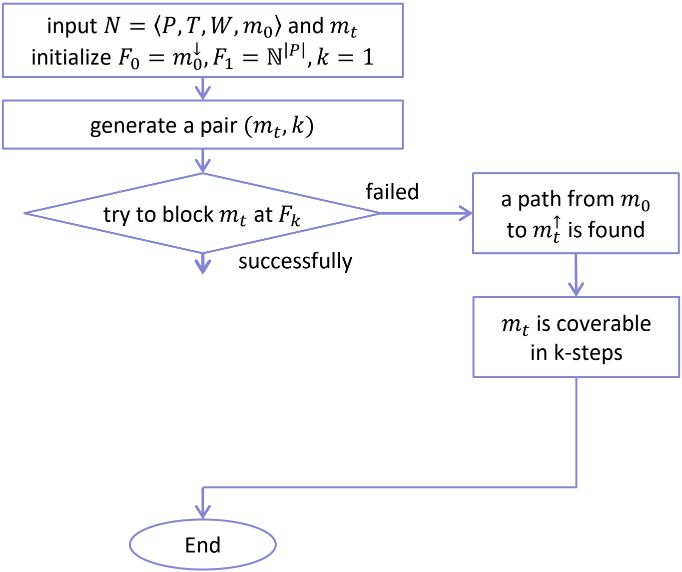
IC3 algorithm for Petri nets



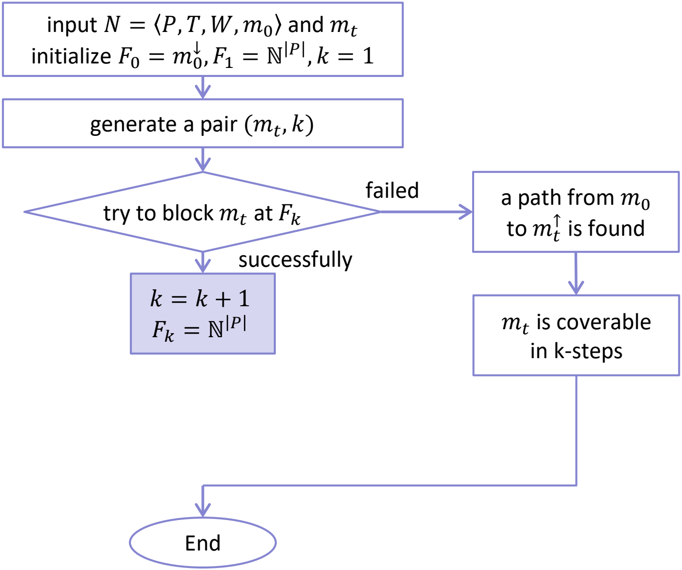
IC3 algorithm for Petri nets



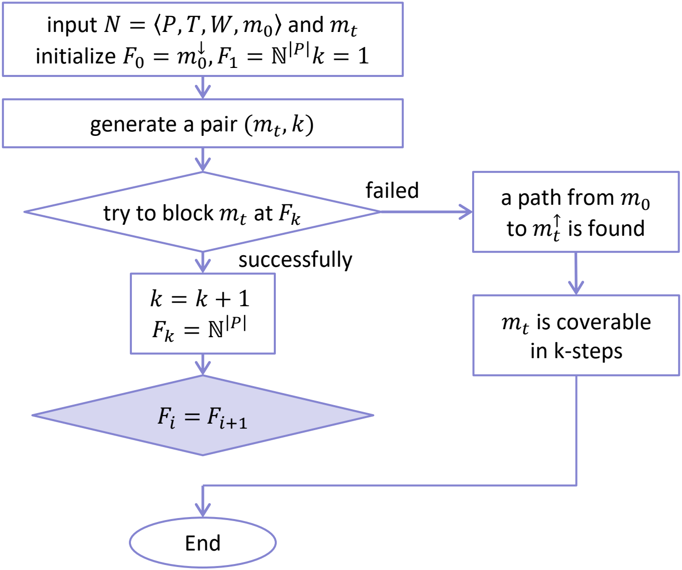
IC3 algorithm for Petri nets



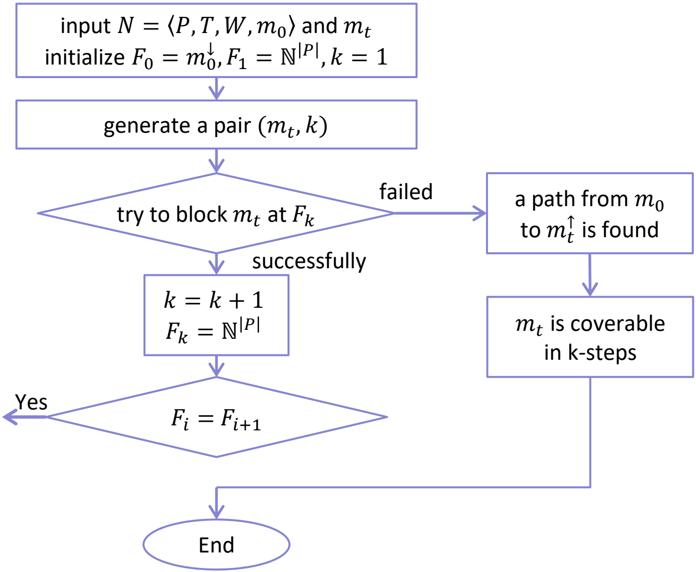
IC3 algorithm for Petri nets



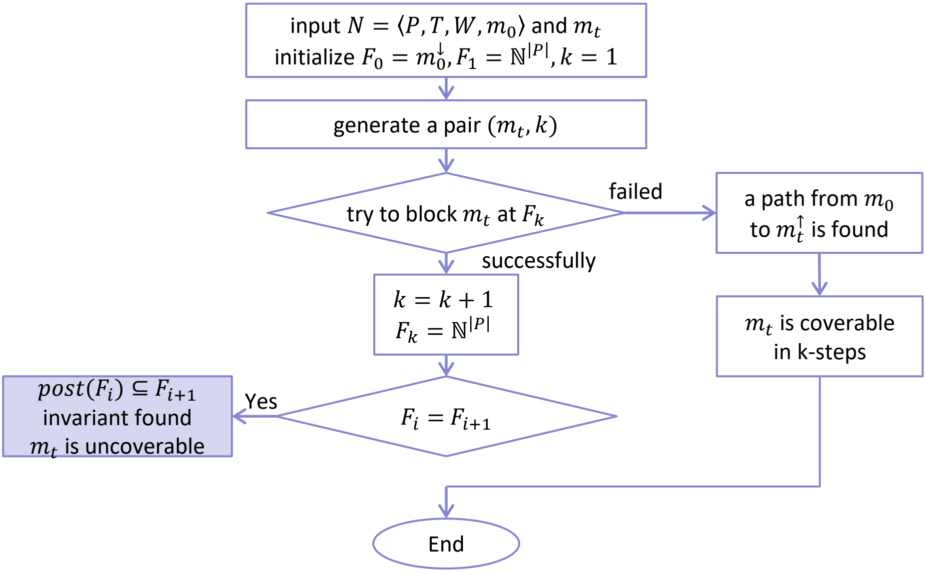
IC3 algorithm for Petri nets



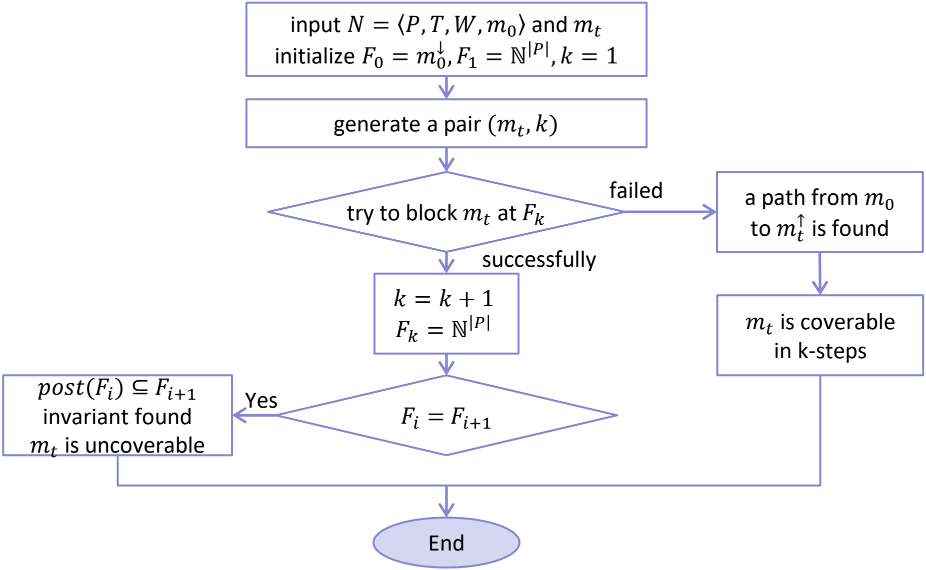
IC3 algorithm for Petri nets



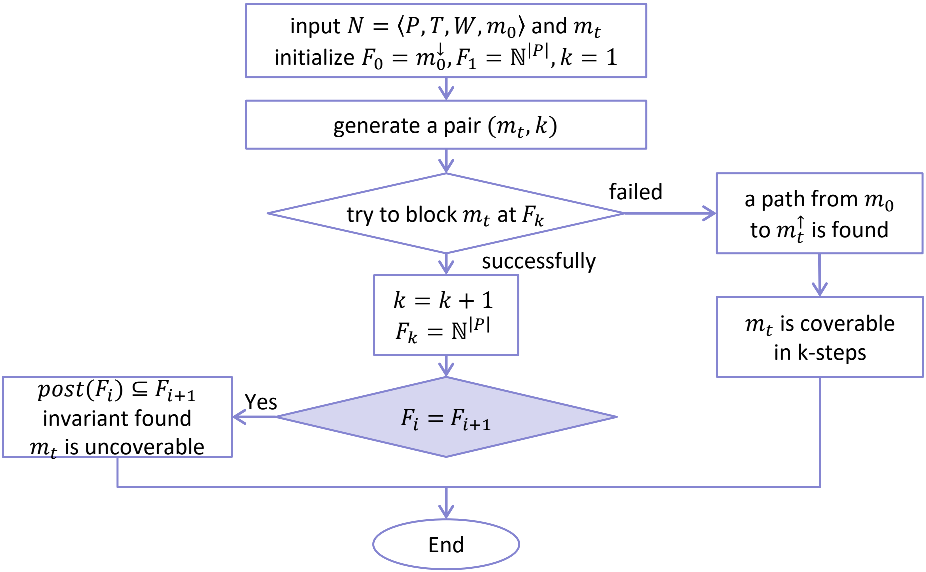
IC3 algorithm for Petri nets



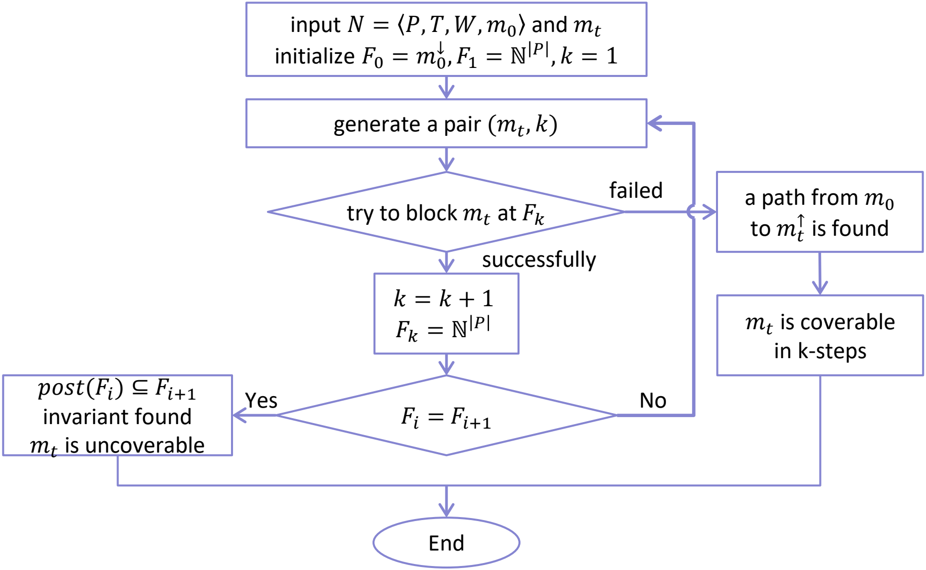
IC3 algorithm for Petri nets



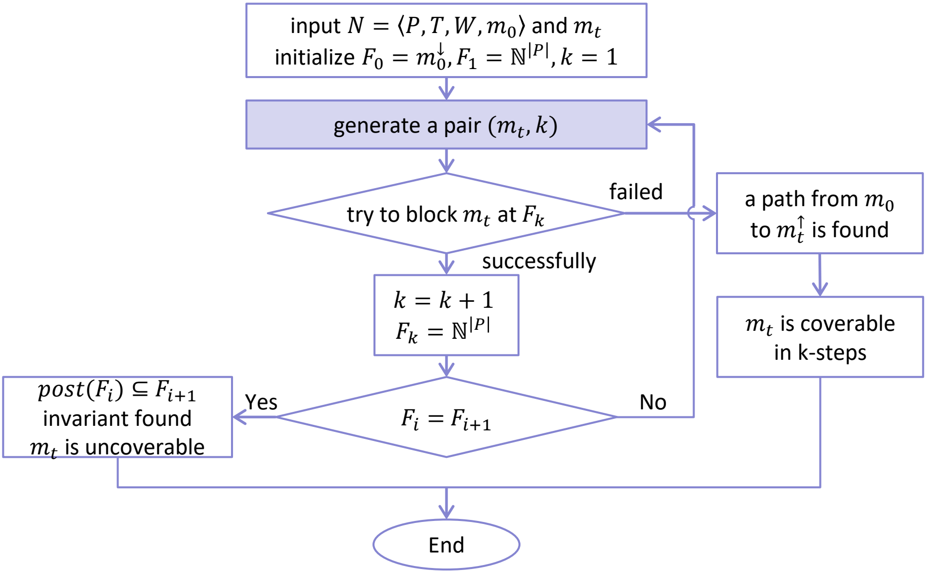
IC3 algorithm for Petri nets



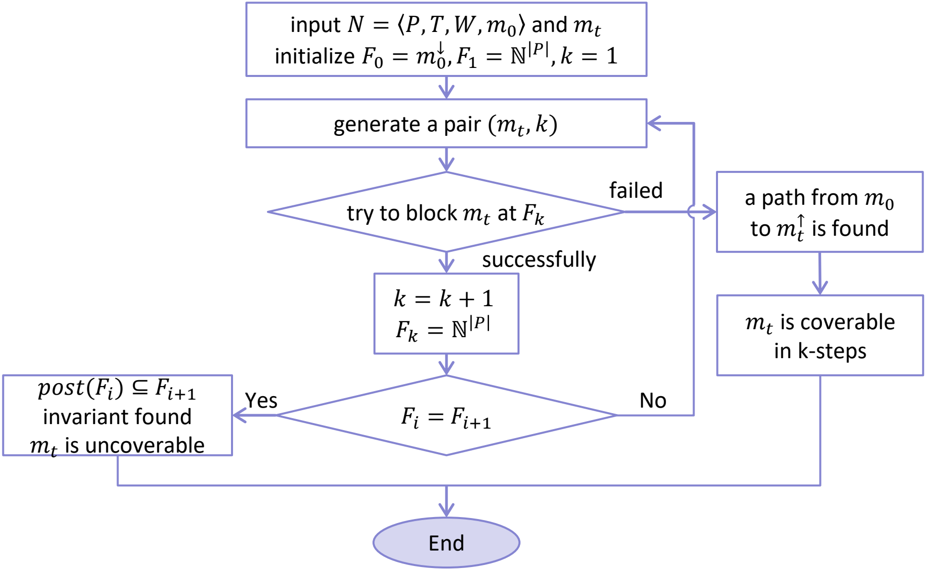
IC3 algorithm for Petri nets



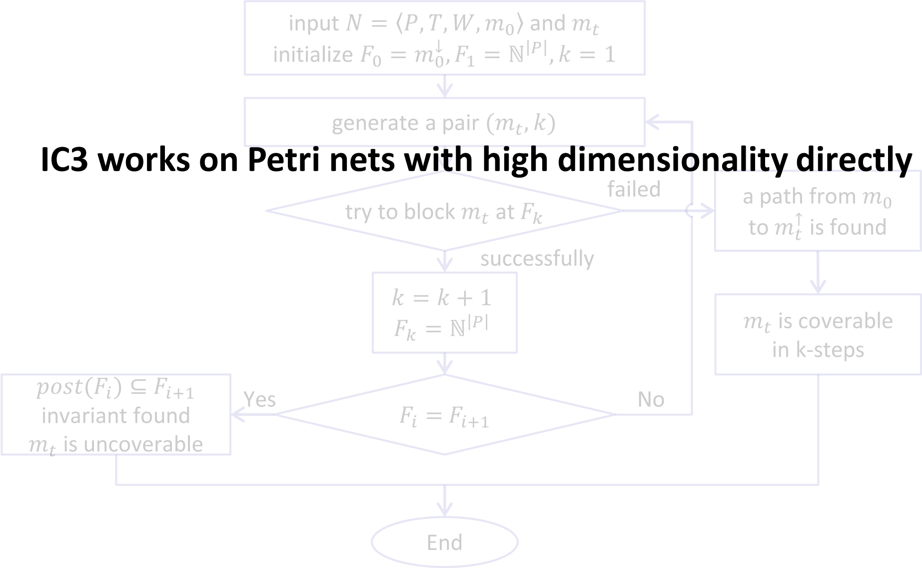
IC3 algorithm for Petri nets



IC3 algorithm for Petri nets

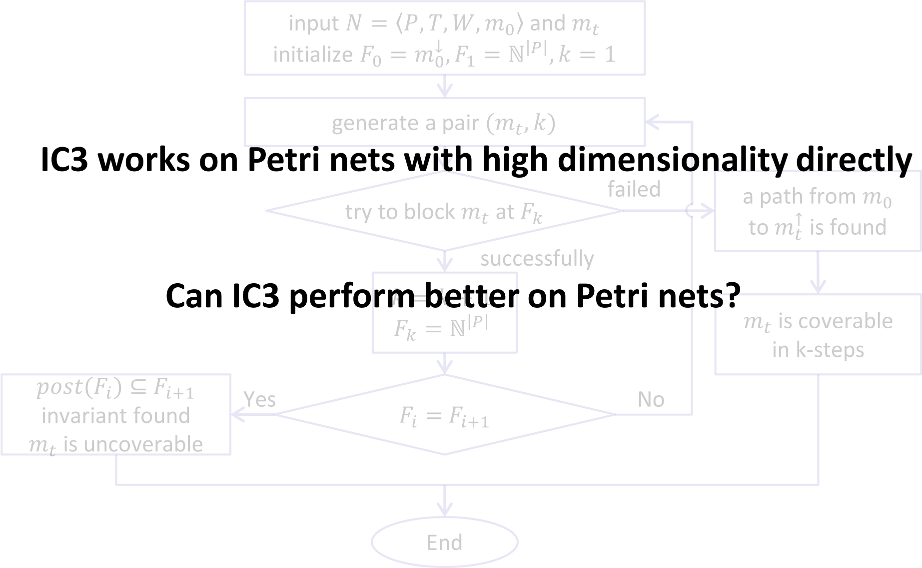


IC3 algorithm for Petri nets



IC3 works on Petri nets with high dimensionality directly

IC3 algorithm for Petri nets



IC3 works on Petri nets with high dimensionality directly

Can IC3 perform better on Petri nets?

Place-merge abstraction

Merge some places of original Petri net into **a single abstract place,** get an abstract Petri net with lower dimensionality.

Place-merge abstraction

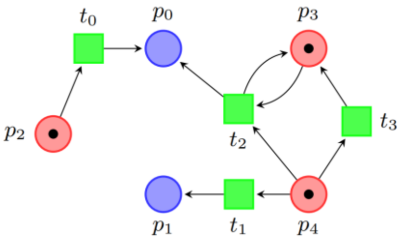
Merge some places of original Petri net into **a single abstract place,** get an abstract Petri net with lower dimensionality.

Definition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$, where $P = \{p_1, p_1 \dots p_k\}$
- The abstraction function is a surjective function $\alpha: P \to \hat{P}$,
where $\hat{P} = \{\hat{p}_1, \hat{p}_2 \dots \hat{p}_{\hat{k}}\}$ and $\hat{k} \leq k$.

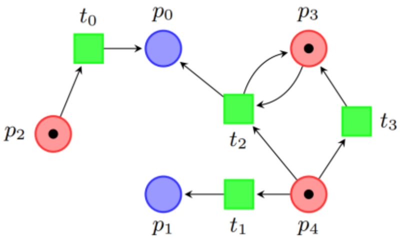
Place-merge abstraction

Merge some places of original Petri net into **a single abstract place,** get an abstract Petri net with lower dimensionality.



Place-merge abstraction

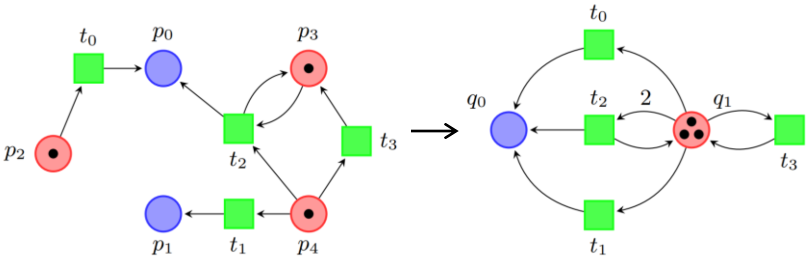
Merge some places of original Petri net into **a single abstract place,** get an abstract Petri net with lower dimensionality.



$$\alpha(p_0) = \alpha(p_1) = q_0$$
$$\alpha(p_2) = \alpha(p_3) = \alpha(p_4) = q_1$$

Place-merge abstraction

Merge some places of original Petri net into **a single abstract place,** get an abstract Petri net with lower dimensionality.



$\alpha(p_0) = \alpha(p_1) = q_0$
$\alpha(p_2) = \alpha(p_3) = \alpha(p_4) = q_1$

All weights of arcs are equal to 1 except for $W(q_1, t_2) = 2$.

Place-merge abstraction

Proposition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$ and one of its abstractions $\widehat{N} = \langle \widehat{P}, T, \widehat{W}, \widehat{m}_0 \rangle$, m_t and its abstract version $\widehat{m}_t$

- If m_t is coverable in N, then its abstract version $\widehat{m}_t$ is coverable in $\widehat{N}$. But the converse does not hold.

Place-merge abstraction

Proposition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$ and one of its abstractions $\widehat{N} = \langle \widehat{P}, T, \widehat{W}, \widehat{m}_0 \rangle$, m_t and its abstract version $\widehat{m}_t$

- If m_t is coverable in N, then its abstract version $\widehat{m}_t$ is coverable in $\widehat{N}$. But the converse does not hold.

$\widehat{m}_t$ is uncoverable in $\widehat{N}$

Place-merge abstraction

Proposition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$ and one of its abstractions $\widehat{N} = \langle \widehat{P}, T, \widehat{W}, \widehat{m}_0 \rangle$, m_t and its abstract version $\widehat{m}_t$

- If m_t is coverable in N, then its abstract version $\widehat{m}_t$ is coverable in $\widehat{N}$. But the converse does not hold.

$\widehat{m}_t$ is uncoverable in $\widehat{N}$ $\longrightarrow$ m_t is uncoverable in N

Place-merge abstraction

Proposition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$ and one of its abstractions $\widehat{N} = \langle \widehat{P}, T, \widehat{W}, \widehat{m}_0 \rangle$, m_t and its abstract version $\widehat{m}_t$

- If m_t is coverable in N, then its abstract version $\widehat{m}_t$ is coverable in $\widehat{N}$. But the converse does not hold.

$\widehat{m}_t$ is uncoverable in $\widehat{N}$ $\longrightarrow$ m_t is uncoverable in N

$\widehat{m}_t$ is coverable in $\widehat{N}$

Place-merge abstraction

Proposition

Given a Petri net $N = \langle P, T, W, m_0 \rangle$ and one of its abstractions $\widehat{N} = \langle \widehat{P}, T, \widehat{W}, \widehat{m}_0 \rangle$, m_t and its abstract version $\widehat{m}_t$

- If m_t is coverable in N, then its abstract version $\widehat{m}_t$ is coverable in $\widehat{N}$. But the converse does not hold.

$\widehat{m}_t$ is uncoverable in $\widehat{N}$ $\longrightarrow$ m_t is uncoverable in N

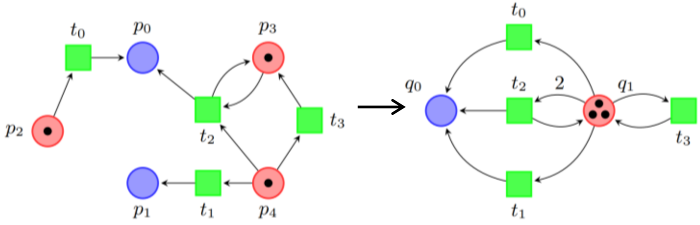
$\widehat{m}_t$ is coverable in $\widehat{N}$ $\not\longrightarrow$ m_t is coverable in N

Place-merge abstraction

Spurious counterexample

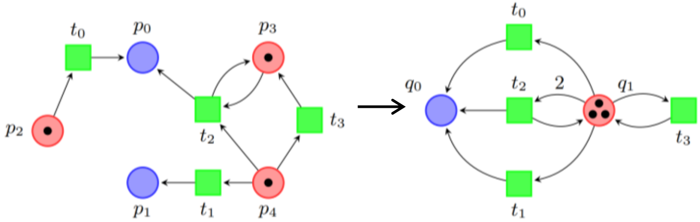
Place-merge abstraction

Spurious counterexample



Place-merge abstraction

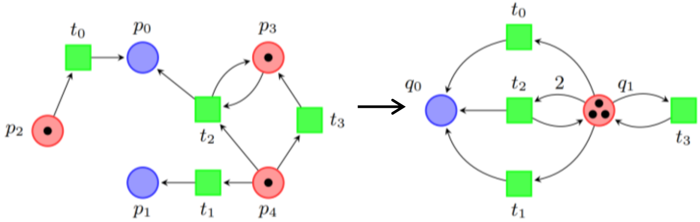
Spurious counterexample



Abstract PN $(0,3) \xrightarrow{t_0} (1,2) \xrightarrow{t_0} (2,1)$

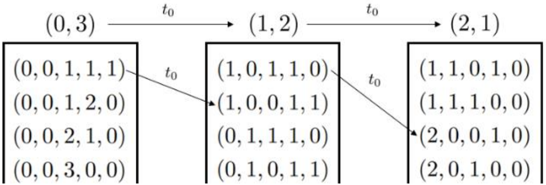
Place-merge abstraction

Spurious counterexample



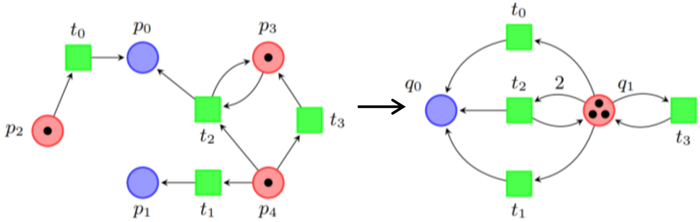
Abstract PN

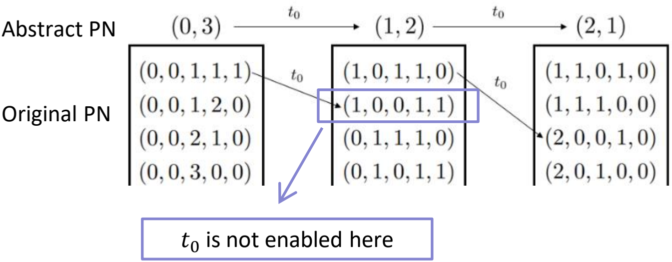
Original PN



Place-merge abstraction

Spurious counterexample





Place-merge abstraction

When a counterexample is spurious

Place-merge abstraction

When a counterexample is spurious

Counter-example $\pi = t_0 t_1 \dots t_{k-1}$ is not spurious iff

$$m_0 \xrightarrow{t_0} m_1 \xrightarrow{t_1} m_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{k-1}} m_k \wedge m_t \preccurlyeq m_k$$

Place-merge abstraction

When a counterexample is spurious

Counter-example $\pi = t_0 t_1 \ldots t_{k-1}$ is not spurious iff

$$m_0 \xrightarrow{t_0} m_1 \xrightarrow{t_1} m_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{k-1}} m_k \wedge m_t \preccurlyeq m_k$$

The path π is spurious:

① t_i is not enabled at m_i $(0 \le i < k)$, **or**

② t_i is enabled at m_i $(0 \le i < k)$, but $m_t \not\preccurlyeq m_k$

Place-merge abstraction

How to refine an abstraction?

Place-merge abstraction

How to refine an abstraction?

t_i is not enabled at m_i $(0 \le i < k)$

- extract places satisfying $m_i(p) < W(p, t_i)$
- merge these places into a new abstract place

Place-merge abstraction

How to refine an abstraction?

t_i is not enabled at m_i $(0 \leq i < k)$

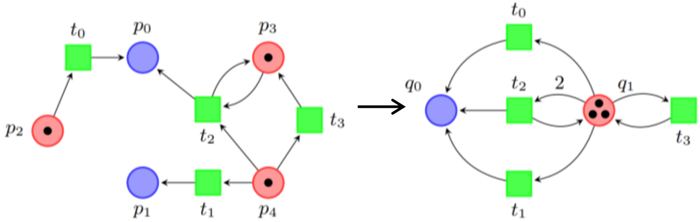
- extract places satisfying $m_i(p) < W(p, t_i)$
- merge these places into a new abstract place

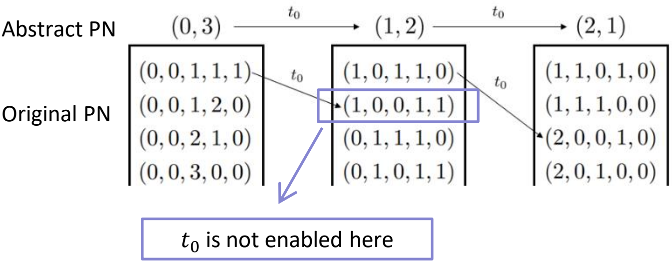
t_i is enabled at m_i $(0 \leq i < k)$, but $m_t \nleqslant m_k$

- extract places satisfying $m_t(p) > m_k(p)$
- merge these places into a new abstract place

Place-merge abstraction

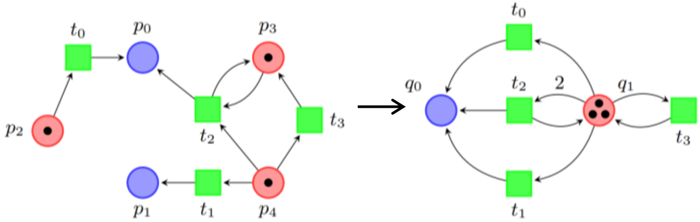
How to refine an abstraction?





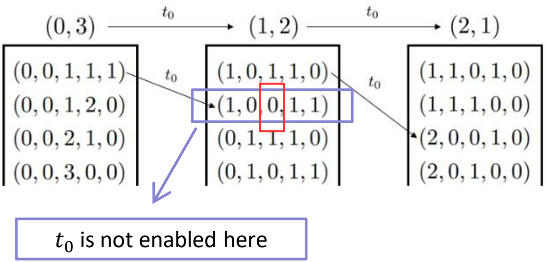
Place-merge abstraction

Abstraction refinement



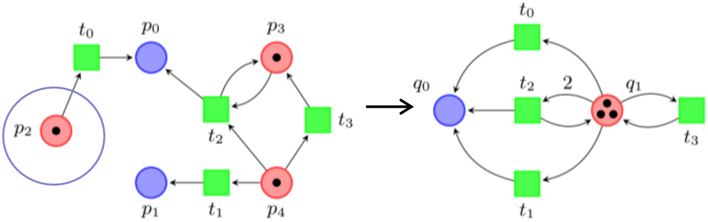
Abstract PN

Original PN



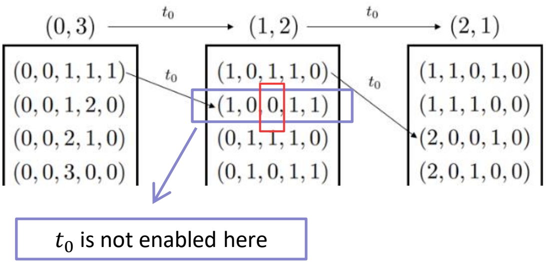
Place-merge abstraction

Abstraction refinement



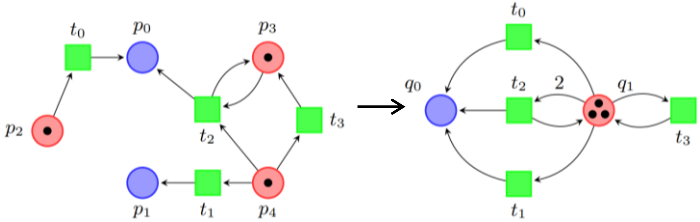
Abstract PN

Original PN



Place-merge abstraction

Abstraction refinement

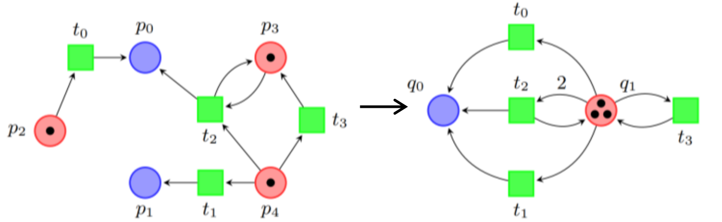


extract p_2 from q_1!

$$\alpha(p_0) = \alpha(p_1) = q_0$$
$$\alpha(p_2) = \alpha(p_3) = \alpha(p_4) = q_1$$

Place-merge abstraction

Abstraction refinement

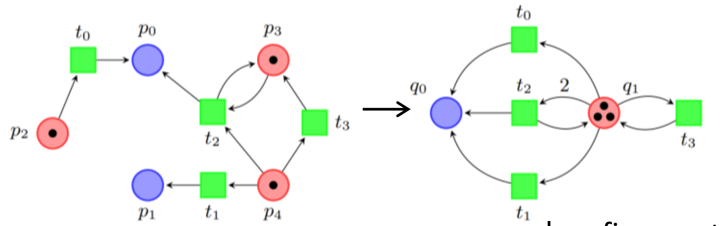


extract p_2 from q_1!

$$\alpha(p_0) = \alpha(p_1) = q_0$$
$$\alpha(p_3) = \alpha(p_4) = q_1$$
$$\alpha(p_2) = q_2$$

Place-merge abstraction

Abstraction refinement

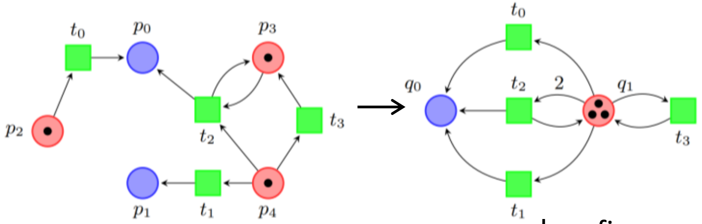


↓ refinement

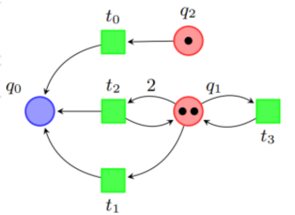
extract p_2 from q_1!

$\alpha(p_0) = \alpha(p_1) = q_0$
$\alpha(p_3) = \alpha(p_4) = q_1$
$\alpha(p_2) = q_2$

Abstraction refinement



↓ refinement



extract p_2 from q_1!

$\alpha(p_0) = \alpha(p_1) = q_0$
$\alpha(p_3) = \alpha(p_4) = q_1$
$\alpha(p_2) = q_2$

IC3+PMA algorithm

- Try to improve the outperformance of IC3

- IC3 is the core of IC3+PMA

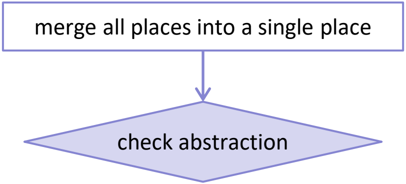
- Place-merge abstraction reduces the dimensionality of PN

- IC3 works on the abstract PN with lower dimensionality

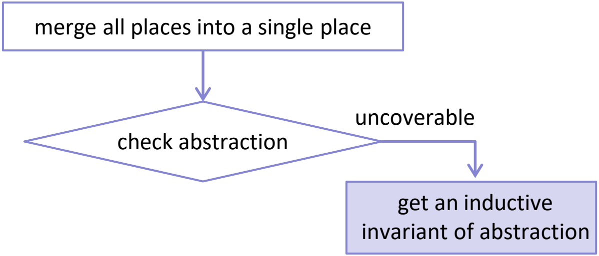
IC3+PMA algorithm

merge all places into a single place

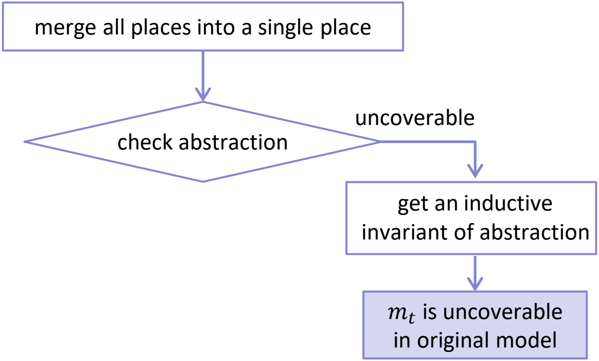
IC3+PMA algorithm



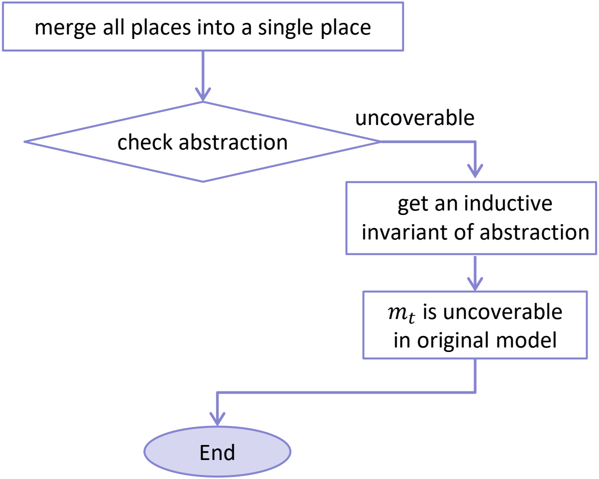
IC3+PMA algorithm



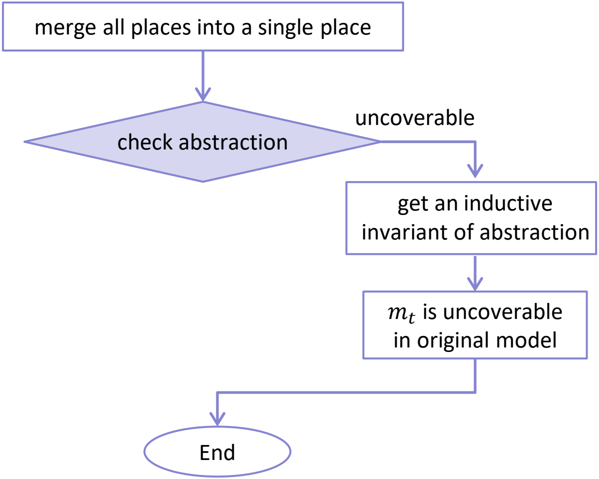
IC3+PMA algorithm



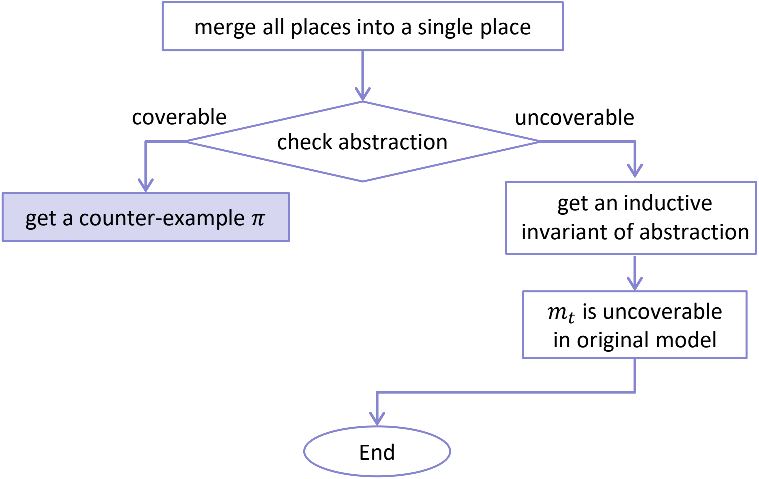
IC3+PMA algorithm



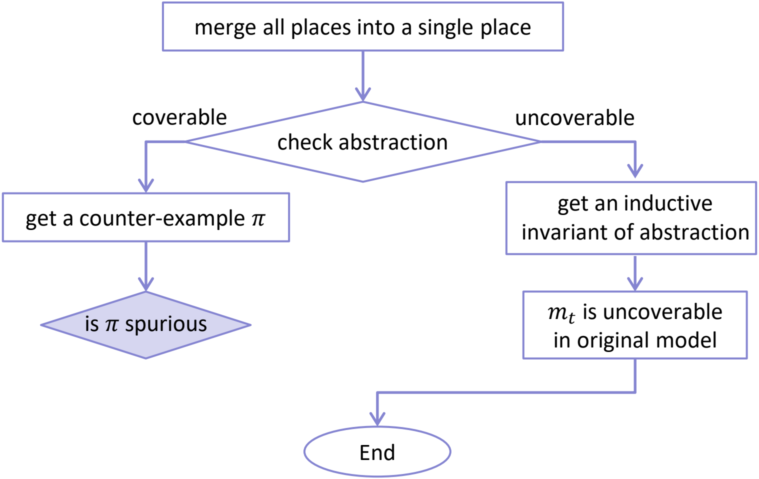
IC3+PMA algorithm



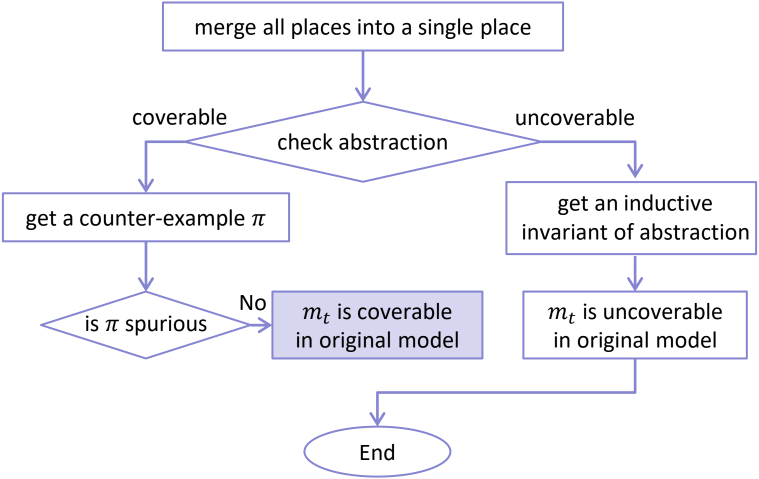
IC3+PMA algorithm



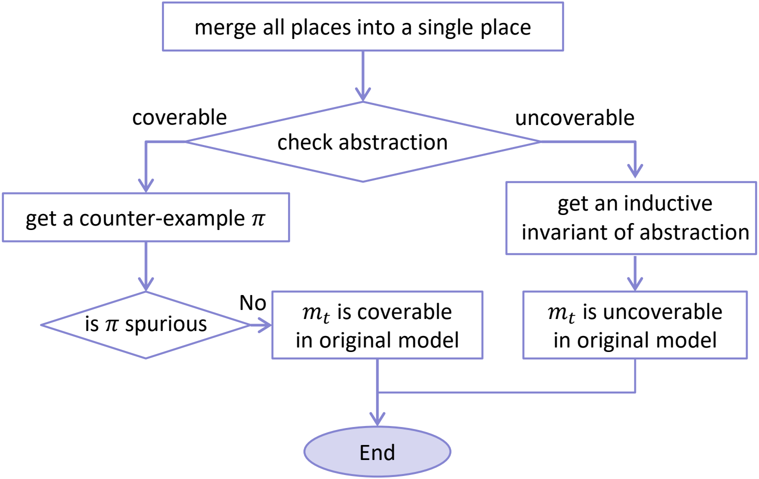
IC3+PMA algorithm



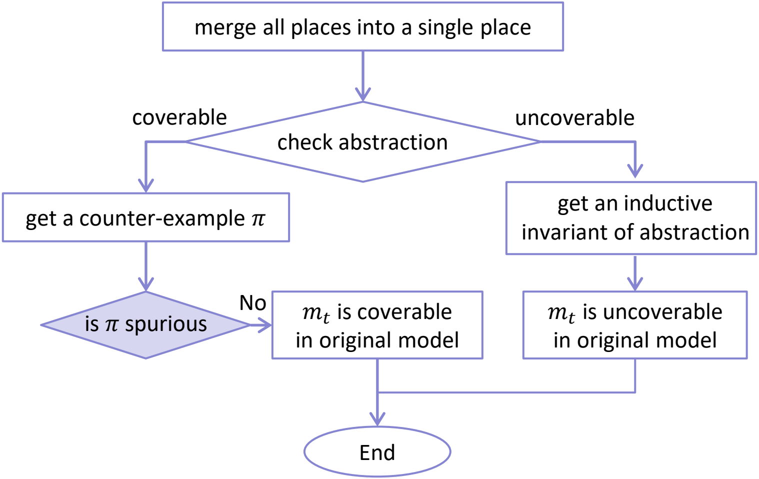
IC3+PMA algorithm



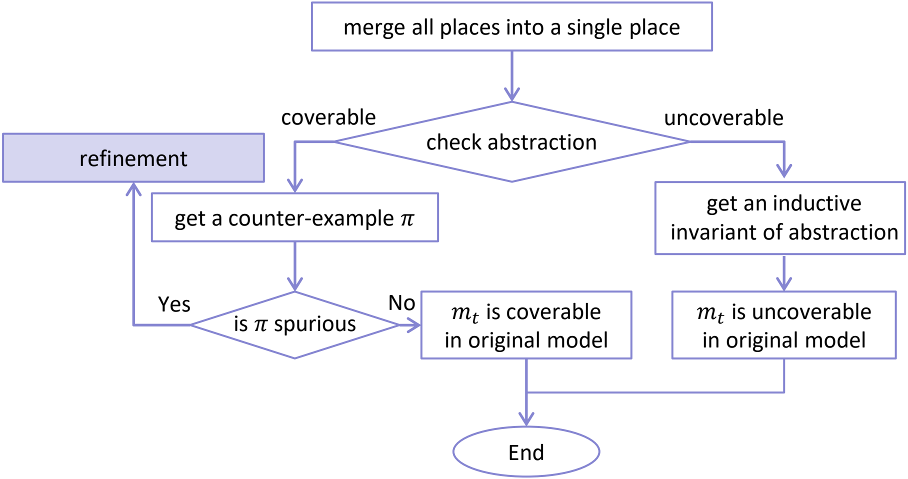
IC3+PMA algorithm



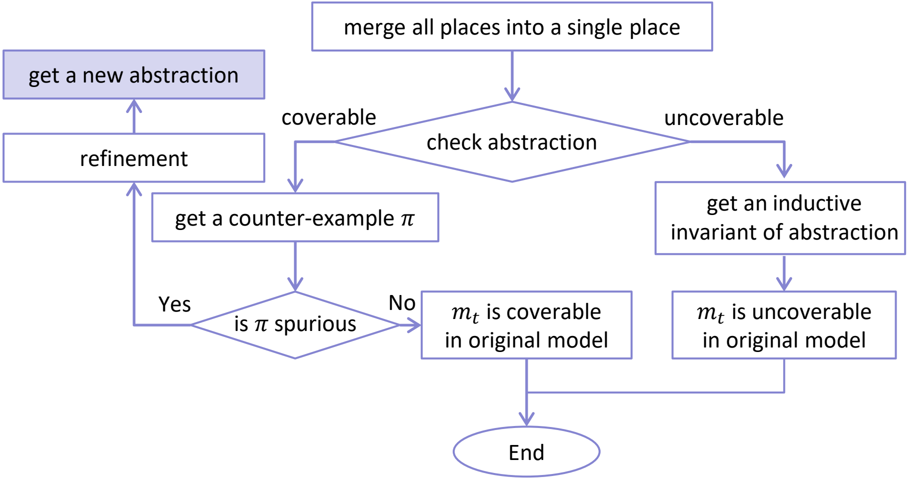
IC3+PMA algorithm



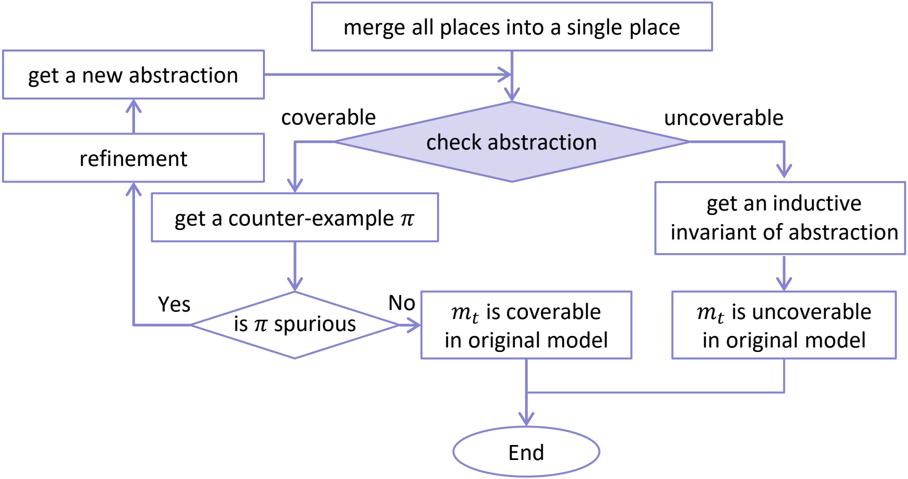
IC3+PMA algorithm



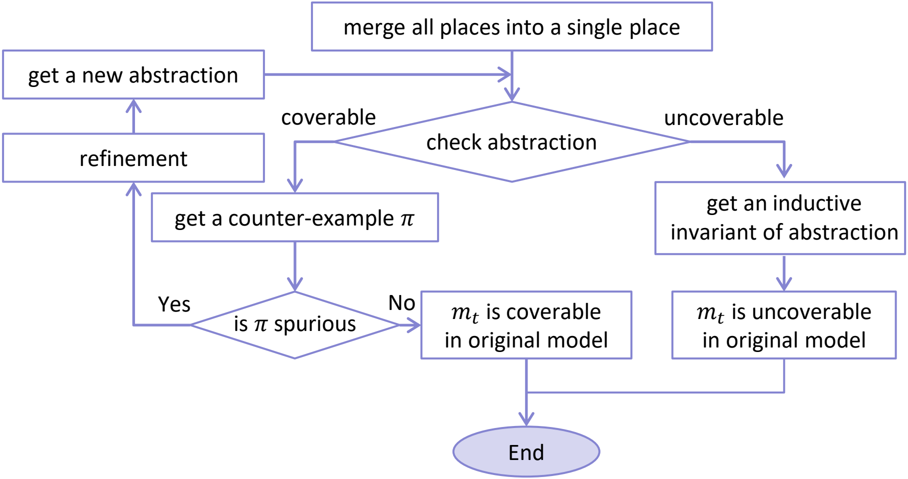
IC3+PMA algorithm



IC3+PMA algorithm



IC3+PMA algorithm



Experiments

- total 80 benchmarks
- compare running time between IC3 and IC3+PMA
- IC3+PMA outperforms IC3 on 53.75% of benchmarks
- dimensionality has decreased by 63.34% on average

Experiments

Benchmark	Places	IC3+PMA AbsPlaces	IC3+PMA Ref	IC3+PMA time(s)	IC3 time(s)
Uncoverable instances					
newrtp	9	**1**	0	<**0.01**	0.06
kanban (bounded)	16	**1**	0	<**0.01**	1.22
manufacturing	13	**1**	0	<**0.01**	0.16
fms	22	**4**	3	<**0.01**	<**0.01**
fms_attic	22	**4**	3	**0.01**	0.04
mesh2x2	32	**5**	4	**0.01**	0.03
mesh3x2	52	**5**	4	**0.02**	0.08
pingpong	6	**5**	4	<**0.01**	<**0.01**
RandCAS 2	110	**8**	7	**0.08**	0.44
Conditionals 2	214	**26**	25	**1.39**	5.79
Coverable instances					
leabasicapproach	16	**5**	4	<**0.01**	<**0.01**
Dekker 1	41	**27**	25	**2.08**	3.23
DoubleLock1 1	64	**35**	32	**11.26**	13.31
Pthread5 1	80	**47**	44	**97.28**	Timeout
RandLock0 2	110	**48**	46	**21.40**	24.89
Spin2003 2	56	**38**	35	**67.35**	Timeout
Szymanski 1	61	**46**	44	**19.62**	32.69
Constants 1	26	**14**	13	**0.03**	**0.03**
FuncPtr3 1	40	**16**	13	**0.19**	0.33

IC3+PMA performs better

Experiments

Benchmark	Places	IC3+PMA AbsPlaces	IC3+PMA Ref	IC3+PMA time(s)	IC3 time(s)
Uncoverable instances					
newrtp	9	**1**	0	<**0.01**	0.06
kanban (bounded)	16	**1**	0	<**0.01**	1.22
manufacturing	13	**1**	0	<**0.01**	0.16
fms	22	**4**	3	<**0.01**	<**0.01**
fms_attic	22	**4**	3	**0.01**	0.04
mesh2x2	32	**5**	4	**0.01**	0.03
mesh3x2	52	**5**	4	**0.02**	0.08
pingpong	6	**5**	4	<**0.01**	<**0.01**
RandCAS 2	110	**8**	7	**0.08**	0.44
Conditionals 2	214	**26**	25	**1.39**	5.79
Coverable instances					
leabasicapproach	16	**5**	4	<**0.01**	<**0.01**
Dekker 1	41	**27**	25	**2.08**	3.23
DoubleLock1 1	64	**35**	32	**11.26**	13.31
Pthread5 1	80	**47**	44	**97.28**	Timeout
RandLock0 2	110	**48**	46	**21.40**	24.89
Spin2003 2	56	**38**	35	**67.35**	Timeout
Szymanski 1	61	**46**	44	**19.62**	32.69
Constants 1	26	**14**	13	**0.03**	**0.03**
FuncPtr3 1	40	**16**	13	**0.19**	0.33

IC3+PMA performs better

Experiments

Benchmark	Places	IC3+PMA AbsPlaces	IC3+PMA Ref	IC3+PMA time(s)	IC3 time(s)
Uncoverable instances					
Peterson	14	**10**	8	0.35	**0.13**
Lamport	11	**7**	6	0.06	**0.02**
Ext. ReadWrite (small consts)	24	**14**	13	1.23	**0.28**
x0_AA_q1	312	#	#	Timeout	**70.28**
csm	14	**9**	8	0.19	**0.02**
Coverable instances					
RandCAS 1	48	**34**	33	0.85	**0.67**
StackCAS0 1	41	**30**	29	3.72	**2.14**
StackLock0 1	37	**26**	25	2.33	**1.06**
Lu-fig2 1	39	**20**	19	0.22	**0.12**
Lu-fig2 2	61	**35**	32	43.06	**9.05**

IC3+PMA performs worse

Experiments

Benchmark	Places	IC3+PMA AbsPlaces	IC3+PMA Ref	IC3+PMA time(s)	IC3 time(s)
Uncoverable instances					
Peterson	14	**10**	8	0.35	**0.13**
Lamport	11	**7**	6	0.06	**0.02**
Ext. ReadWrite (small consts)	24	**14**	13	1.23	**0.28**
x0_AA_q1	312	#	#	Timeout	**70.28**
csm	14	**9**	8	0.19	**0.02**
Coverable instances					
RandCAS 1	48	**34**	33	0.85	**0.67**
StackCAS0 1	41	**30**	29	3.72	**2.14**
StackLock0 1	37	**26**	25	2.33	**1.06**
Lu-fig2 1	39	**20**	19	0.22	**0.12**
Lu-fig2 2	61	**35**	32	43.06	**9.05**

- the efficiency of refinement method is not so high
- the way to deal with frames after refinement is not efficient

future work

- optimize the implementation to achieve better results
- apply the approach to analyze more properties and models

Thank You For Your Attention