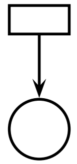
# Old And New Algorithms for Minimal Coverability Sets

**Petri Nets 2012, Hamburg**

**Antti Valmari**
Tampere University of Technology, FINLAND

# 1 Minimal Coverability Sets



The set of reachable markings of a Petri net is not necessarily finite

Karp & Miller 1969: Coverability set

- $M(p) = \omega$ denotes that the marking of place $p$ may grow without limit
- when $M\,[t_1 \cdots t_n\rangle\, M' > M$ and $M(p) < M'(p) < \omega$, replace $M'(p)$ by $\omega$

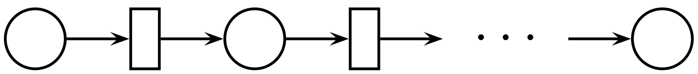
Properties

- for every reachable marking $M$, the set has an $\omega$-marking $M'$ s.t. $M \leq M'$
- for every $M'$ in the set and every $n \in \mathbb{N}$, there is a reachable $M$ such that $M(p) = M'(p) < \omega$ or $M(p) \geq n \wedge M'(p) = \omega$
- finite, but not unique

Finkel 1993: Minimal Coverability Set

- keep only maximal $\omega$-markings

! do so even if the marking is ordinary



Minimal coverability sets may be very big

- $n$ places, $n-1$ tokens in the leftmost: $\approx 2^{2n-2}/\sqrt{\pi(n-1)}$ $\omega$-markings

Geeraerts & Raskin & Van Begin 2010, Reynier & Servais 2011, ...: complicated algorithms

# 2 Mathematical Properties

Clarified proofs — not yet algorithms or even transitions

All $M$ are $\omega$-markings, unless otherwise stated

Every growing sequence $M_1 \leq M_2 \leq \ldots$ has a limit

- for each $p$, either $M(p) = M_i(p) = M_{i+1}(p) = \ldots$ from some $i$ on, or $M(p) = \omega$ and $M_i(p)$ grows without a limit

We define *a* limit of a set as any limit of a growing sequence of its elements

- every element of the set is a limit, because $M \leq M \leq \ldots$

**Lemma** The limit of any growing sequence of limits is a limit.

- the lemma can be applied $\leq |P|$ times in a row

$\Rightarrow$ each set is covered by its maximal limits

**Lemma** (follows from Dickson's, easier to prove directly)
Every infinite sequence of $\omega$-markings has an infinite growing subsequence.

- proof: construct one place at a time, by picking from previous sequence

Let $[\mathcal{M}]$ be the limits of $\mathcal{M}$ and $\lceil\mathcal{M}\rceil$ be the maximal elements of $[\mathcal{M}]$

**Theorem** $\lceil\mathcal{M}\rceil$ is finite and the only minimal coverability set.

# 3 Overview of New Algorithm

$F := \{\hat{M}\}$; $A := \{\hat{M}\}$; $W := \{\hat{M}\} \times T$; $\hat{M}.B :=$ nil
**while** $W \neq \emptyset$ **do**
  $(M, t) :=$ any element of $W$; $W := W \setminus \{(M, t)\}$
  **if** $\neg M[t\rangle$ **then** continue
  $M' :=$ the $\omega$-marking such that $M[t\rangle M'$
  **if** $M' \in F$ **then** continue
  Add-$\omega(M, M')$  // the $M_0\,[t_1 \cdots t\rangle\, M' > M_0$ test
  **if** $\omega$ was added **then if** $M' \in F$ **then** continue
  Cover-check$(M')$  // only keep maximal — may update $A$ and $W$
  **if** $M'$ is covered **then** continue
  $F := F \cup \{M'\}$; $A := A \cup \{M'\}$; $W := W \cup (\{M'\} \times T)$; $M'.B := M$

$F$ is a hash table of all constructed $\omega$-markings

- unnecessary for correctness, speeds up the algorithm, cheap

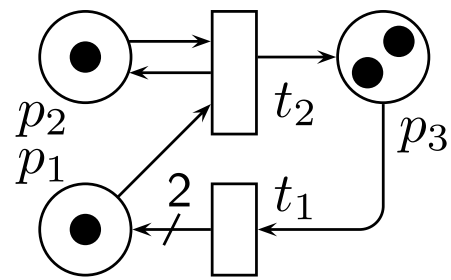
$A$ is all kept $\omega$-markings — expensive, touch as little as you can

$W$ is pending work, simpler than it seems, we come back

Simple and natural — how can this beat others?

# 4 Addition of $\omega$-symbols

Rule of thumb: the earlier they are added, the better for speed



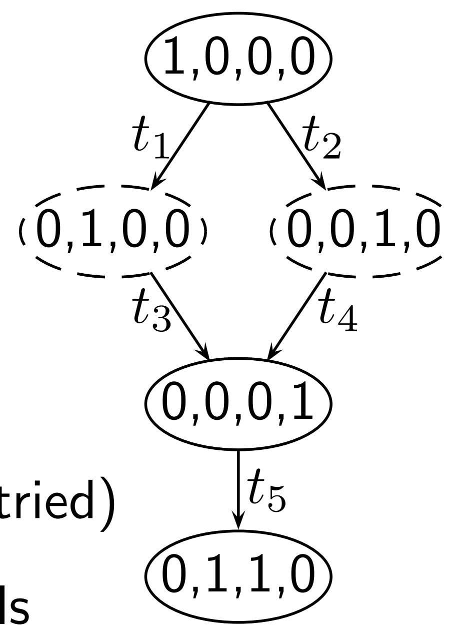

Traditional: scan linear history backwards (the $M.B$)

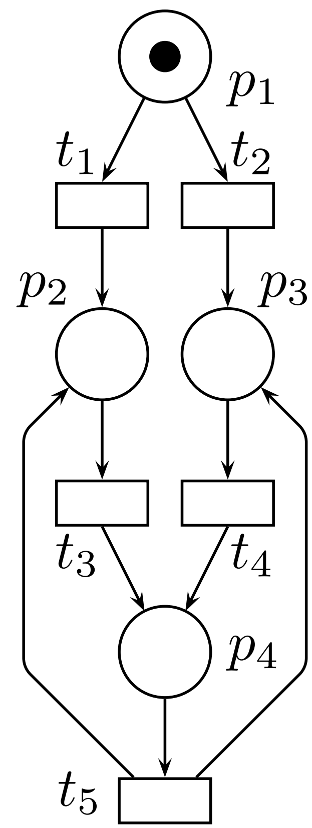
Repeated scanning of history

- $(1,1,2)\ [t_1\rangle\ (3,1,1)\ [t_2\rangle\ (2,1,2)$
- $(3,1,1)$ does not trigger $\omega$-addition to $(2,1,2)$
- $(1,1,2)$ triggers: $(2,1,2) \rightsquigarrow (\omega,1,2)$
- history is now fully scanned,
  but now $(3,1,1)$ triggers $(\omega,1,2) \rightsquigarrow (\omega,1,\omega)$
- cheap enough to be always used

History merging

- both (0,1,0,0) and (0,0,1,0) trigger $\omega$-addition in (0,1,1,0)
- history becomes a DAG $\Rightarrow$ expensive
- was not strong in our experiments
- $\omega$ in kept may match finite in other (not tried)




Let $M \xrightarrow{t} M'$ mean $M\ [t\rangle$ and add $\omega$-symbols

# 5 Correctness

**Lemma** After termination, all reachable markings are covered.

- invariant: for every $M \in F$, there is $M' \in A$ such that $M \leq M'$
- this issue is hard with some competing algorithms

**Lemma** Every element of $A$ is a limit of reachable markings.

- $\hat{M}$ is trivially a limit
- given limits, operations of the algorithm yield limits
- this lemma justifies non-standard addition of $\omega$-symbols, like history merging

**Lemma** $A$ only contains maximal $\omega$-markings.

- taken care of explicitly

**Lemma** The algorithm terminates.

- to avoid termination, infinitely many $\omega$-markings must be constructed

$\Rightarrow$ an infinite sequence of distinct $\omega$-markings, because $T$ is finite

$\Rightarrow$ an infinite strictly growing subsequence

$\Rightarrow$ Add-$\omega$ triggers repeatedly, adding $\omega$-symbols

- but there can be at most $|P|$ $\omega$-symbols in an $\omega$-marking

# 6 Construction Order

Necesssary to realize: for almost any algorithm there is a “cheating” easy input

- a transition that adds tokens to every place leads to immediate termination
- an otherwise bad algorithm may hit it much earlier than competing algorithms

Breadth-first

- simple and fast $W$: queue of $\omega$-markings, scan all transitions in a **for**-loop
- bad in experiments (big $|F|$)

Depth-first

- simple and fast $W$: a stack of $\omega$-markings, and a transition number in each
- **Lemma** If $M \xrightarrow{t} M'$ adds an $\omega$-symbol, the algorithm will not backtrack from $M'$ before it has investigated all its descendants.

Most tokens first

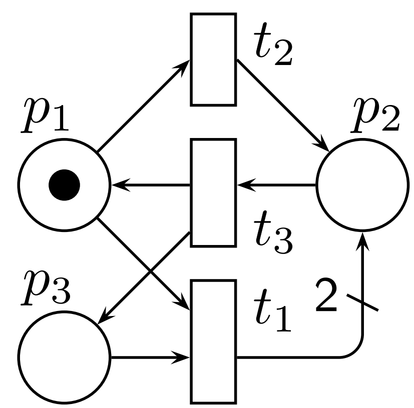
- let “$\prec$” sort first by the number of $\omega$-symbols, then by the number of tokens
- try $\omega$-markings in that order
- $W$ is a heap of $\omega$-markings with transition numbers, $O(\log |W|)$ operations
- intuitively promising and good in measurements

# 7 Pruning 1/2

Idea: when removing $M_0$ from $A$, also remove (part of) its constructed future

- applied in some competing algorithms
- motivation: if $M > M_0 \xrightarrow{t_1 \cdots t_n} M_n$, then $M_n$ would be removed eventually anyway, *if no $\xrightarrow{t_i}$ added $\omega$-symbols*

$\Rightarrow$ improved speed?




Even if the green assumption holds, total pruning of pumping cycles postpones $\omega$-addition

- $(1,0,0) \xrightarrow{t_2} (0,1,0) \xrightarrow{t_3} (1,0,\omega) \xrightarrow{t_1} (0,\omega,\omega) \xrightarrow{t_3} (\omega,\omega,\omega)$
- $(1,0,0) \xrightarrow{t_2} (0,1,0) \xrightarrow{t_3} (1,0,\omega) \xrightarrow{t_1} (0,2,\omega) \xrightarrow{t_3} (1,\omega,\omega) \xrightarrow{t_3} (\omega,\omega,\omega)$

Keeping all constructed in $F$ costs little, and protects against re-constructing

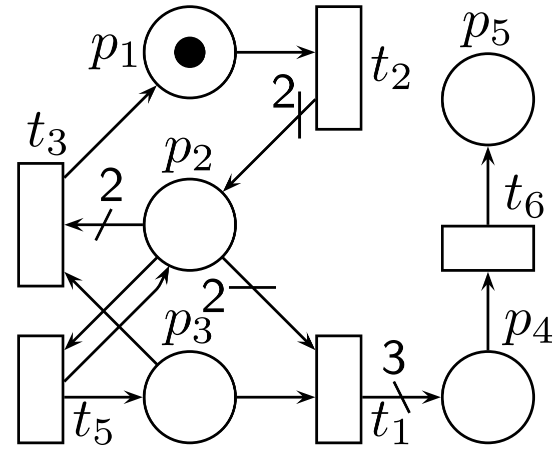
- if $M\,[t_1 t_2\rangle\, M_{12}$ and $M\,[t_2 t_1\rangle\, M_{21}$, then $M_{21} = M_{12}$

$\Rightarrow$ re-constructing would be common

[ReySer] does not totally prune, and cover-checks against $F$ instead of $A$

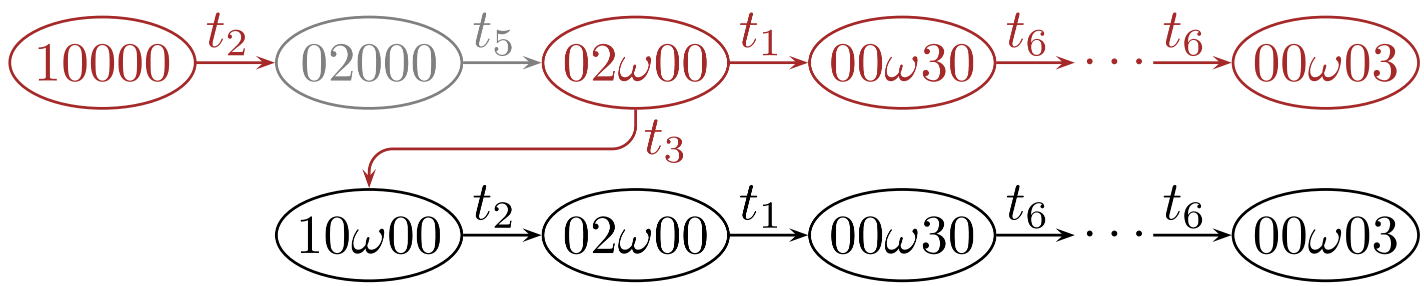
- bad for speed, because $F$ is much bigger and cover-checking is expensive

# 7 Pruning 2/2




Example violating the green assumption

- most tokens first
- transitions $t_1$ first, then $t_2$, then ...
- pruning algorithm:




$\Rightarrow$ lots of re-activation or re-construction

$\Rightarrow$ I do not believe that pruning is a good idea in this context

# 8 Good News

Overeager pruning

- assume $M_0 \xrightarrow{t_1 \cdots t_n} M_n$ has been constructed and $M_0' > M_0$ just been found
- we say that pruning $M_n$ is *overeager*, if $M_0' \, [t_1 \cdots t_n\rangle \, M_n$
- it is possible, if $\xrightarrow{t_1 \cdots t_n}$ adds $\omega$-symbols

**Theorem** and **Theorem** With depth-first and most tokens first, if history merging is applied, then the effect of non-overeager pruning occurs automatically.

- that is, if pruning would not be overeager, the algorithm will not any more fire transitions from $M_n$ (and if it would, there is no reason to not fire)
- "any more", because it may have fired many of them before finding $M_0'$
- proofs are not simple enough for the time that remains (enjoy the paper)
- does not give all possible pruning, but still gives a lot
- (also remember that almost any algorithm has a "cheating" winning example)

So theory suggests that depth-first and most tokens first be better than [ReySer]

- do measurements support this?

# 9 Measurements

| model | $\|A\|$ | most tokens f. | | depth-first | | breadth-first | | [ReySer] |
|---|---|---|---|---|---|---|---|---|
| fms | 24 | 63 | 53 | 110 | 56 | 421 | 139 | 809 |
| kanban | 1 | 12 | 12 | 12 | 12 | 12 | 12 | 114 |
| mesh2x2 | 256 | 479 | 465 | 774 | 455 | 10733 | 2977 | 6241 |
| mesh3x2 | 6400 | 11495 | 11485 | 8573 | 10394 | | | |
| multipoll | 220 | 245 | 234 | 244 | 244 | 507 | 507 | 2004 |
| pncsacover | 80 | 215 | 246 | 284 | 325 | 7122 | 5804 | 1604 |

- $|A|$: final number of $\omega$-markings in the coverability set
- other numbers: total numbers of constructed distinct $\omega$-markings
- transitions tried in two orders (numeric and reverse)
- running times cannot be compared to [ReySer]
- ours $< 0.1$ s except mesh3x2 and some breadth-first, all $\leq 30$ s

Observations

- transition order may have dramatic impact (winning "cheating" already here!)

$\Rightarrow$ I wish reviewers in general would be less measurement-oriented

- is most tokens first the best, see the largest case?
- we did not lose

# 10 Discussion

It seems that with this problem, a simple algorithm wins complicated ones

However, ours was not the most trivial possible

- hash table for $F$
- repeated scanning of history (cheap enough to be always on)
- history merging (although it had little effect in measurements)
- some thought given to data structures and other details

Construction order is important both at the overall and transition scanning level

- do not believe too much in measurements, in this paper or elsewhere, unless there is a huge meticulously chosen amount of them
- we hope to make some more measurements with bigger nets in the future

**THANK YOU FOR ATTENTION!**