Stubborn Set Intuition Explained

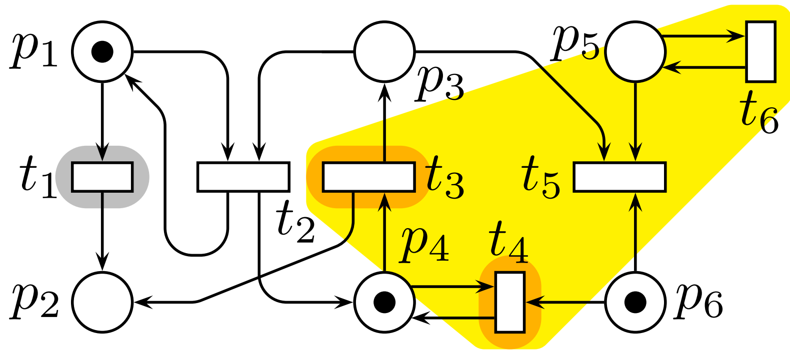
Antti Valmari

Tampere University of Technology
Department of Mathematics

1 Introduction

There are two classes of partial order methods

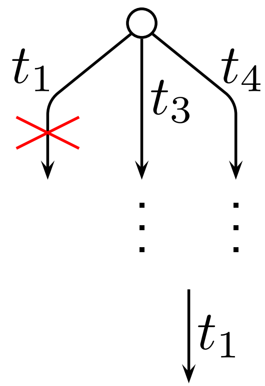
- based on partial order semantics
  - unfolding, step graphs, ...
- not based on partial order semantics
  - ample sets, persistent sets, stubborn sets
  - *aps sets*



Idea of aps sets

- in each state, only (try to) fire a subset of transitions
  - *aps set*
- choose the set so that the answer to the verification question does not change

⇒ choice of aps sets depends on the verified property
  - easiest property: deadlocks
  - safety, home markings, $\mathrm{LTL}_{\mathrm{X}}$, $\mathrm{CTL}^*_{\mathrm{X}}$, CSP-equivalence, ...

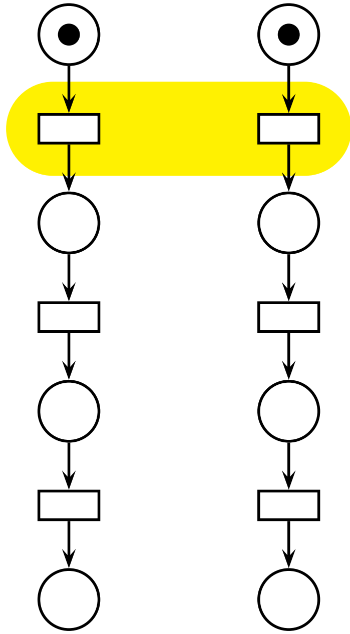


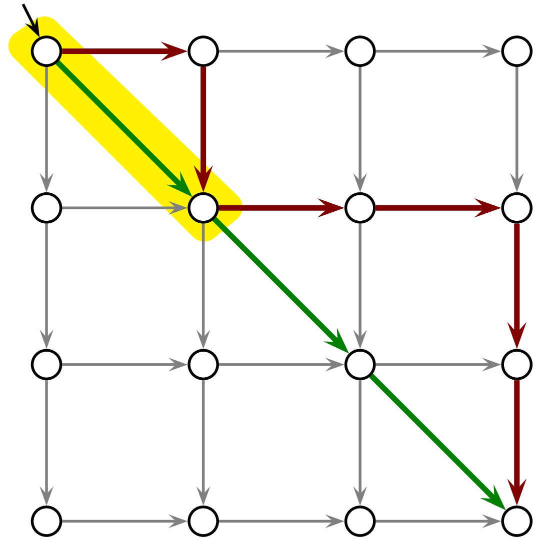
Goal of this publication:

why stubborn sets are like they are

- especially compared to ample and persistent sets

2 Why Not Steps?





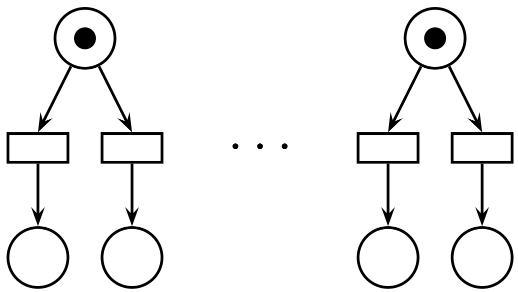
Idea: fire all transitions
of a step simultaneously

- intermediate states not stored
- order of firing not represented
- (aps sets choose, e.g., the brown path)

Elegant attractive idea, but . . .
fails in practice for more than one reason

- we discuss one reason

This net has 2^n deadlocks



- initially 2^n steps

⇒ too many steps with big n

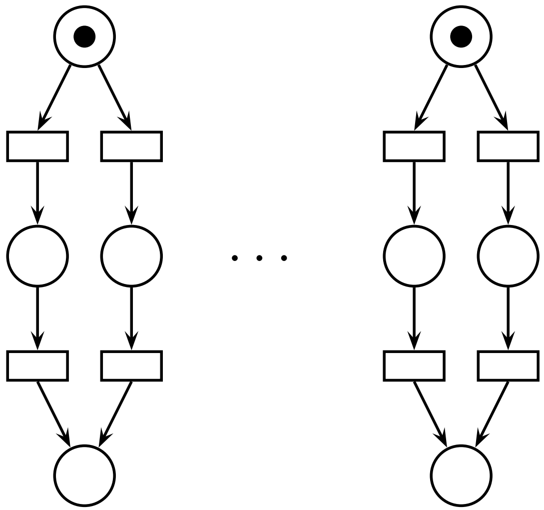
- 2^n deadlocks

⇒ any deadlock-preserving method suffers, so aps sets are not better

This net has one deadlock

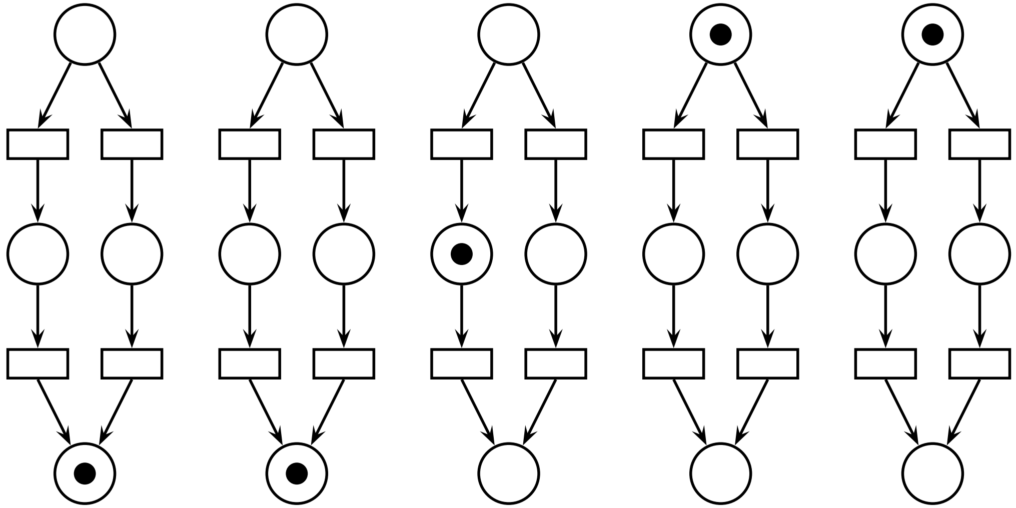
- initially the same steps

$\Rightarrow$ 2^n steps (plus 2^n second steps)



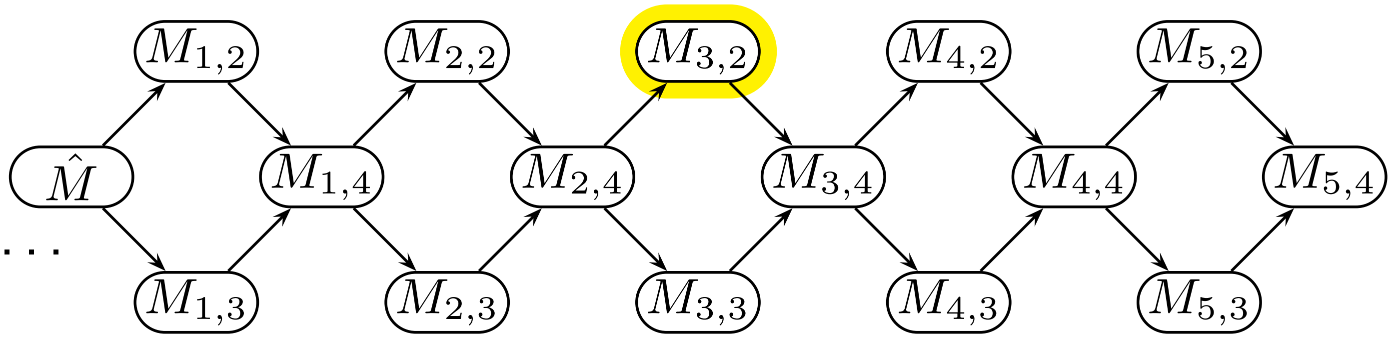
With a bit of luck, aps sets construct a small reduced state space

- e.g., always try leftmost transition first
  - $3n+1$ states
- e.g., always try topmost transition first
  - $3 \cdot 2^n - 2$ states
- aps sets *may* perform badly here
- steps *are guaranteed* to perform badly



Additional lesson

- we would like to treat input order as irrelevant . . .
- . . . but it may be crucial

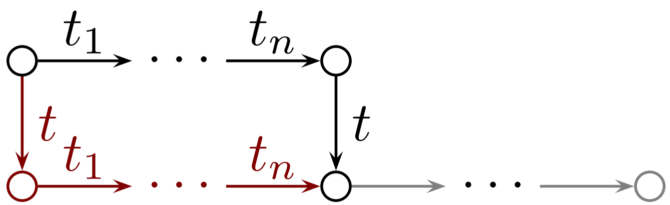


3 Deadlock-Preserving Strong Stubborn Sets

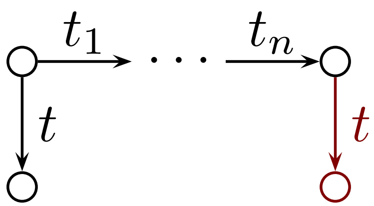
Build stubb(M) so that for every $t \in$ stubb(M) and $t_i \notin$ stubb(M):

D0 If en$(M) \neq \emptyset$, then stubb$(M) \cap$ en$(M) \neq \emptyset$.

D1 If $M\ [t_1 \cdots t_n t\rangle\ M''$, then $M\ [t t_1 \cdots t_n\rangle\ M''$.



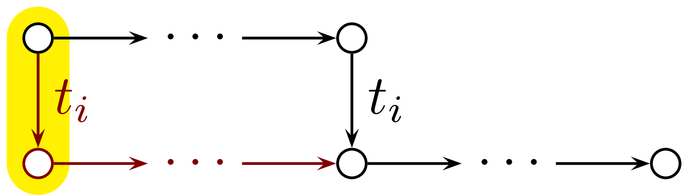
D2 If $M\ [t\rangle$ and $M\ [t_1 \cdots t_n\rangle\ M'$, then $M'\ [t\rangle$.



Facilitates an easy proof that the reduced state space contains all reachable deadlocks

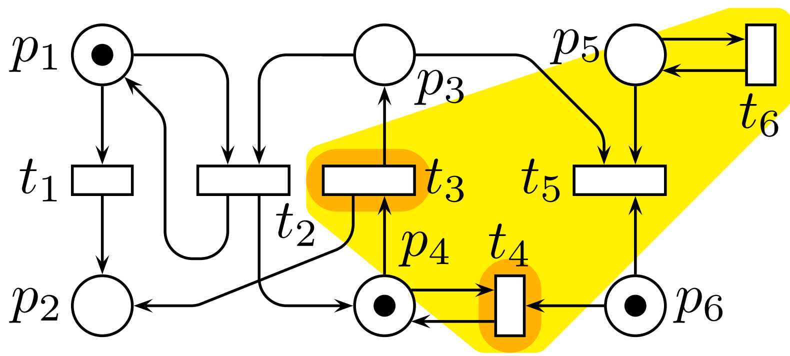
- assume $M \in$ reduced, $n > 0$, $M\ [t_1 \cdots t_n\rangle\ M_{\mathsf{d}}$, and M_{d} is a deadlock
- because $M\ [t_1\rangle$, **D0** implies that the stubborn set contains an enabled transition t
- if none of $t_1, \ldots, t_n \in$ stubb(M), then $M_{\mathsf{d}}\ [t\rangle$ by **D2** ↯
- by **D1**, the first t_i in stubb(M) moves to the front

$\Rightarrow$ a transition firing in the reduced state space leads towards the deadlock



4 Construction of Strong Stubborn Sets

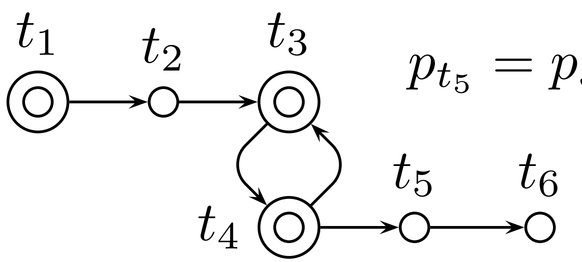
D1 and **D2** are ensured
via a suitable "$\leadsto_M$" $\subseteq T \times T$



- encodes knowledge about how transitions interfere with each other
- if $t \leadsto_M t'$ and $t \in \text{stubb}(M)$, then $t' \in \text{stubb}(M)$
- not necessarily vice versa
- not necessarily $t \in \text{stubb}(M)$

A simple (not good) example "$\leadsto_M$"

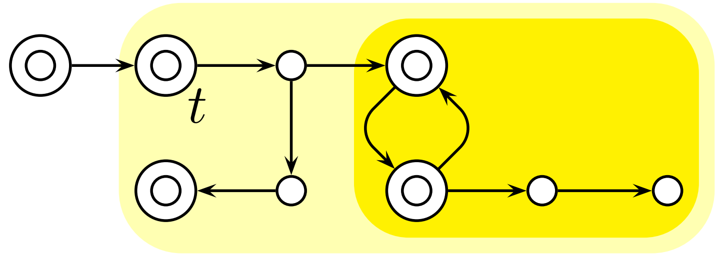
- if $\neg M\,[t\rangle$, then choose $p_t \in \bullet t$ such that $M(p_t) < W(p_t, t)$ and let $t \leadsto_M t' \Leftrightarrow t' \in \bullet p_t$
  - disabled inside transitions remain disabled while outside transitions occur
- if $M\,[t\rangle$, then let $t \leadsto_M t' \Leftrightarrow \bullet t \cap \bullet t' \neq \emptyset$
  - enabled inside transitions are $\approx$ concurrent with outside transitions



$p_{t_5} = p_5$

Two algorithms

- clsr$(t) = \{t' \mid t \rightsquigarrow^*_M t'\}$
  - bad sets in general, needed in Section 6
- esc(t) = a **minimal** closed subset of clsr(t) that contains an enabled transition, or indication that clsr(t) contains no enabled transitions
  - $O(|T| + |F|)$ time, **often** $o(|T|)$

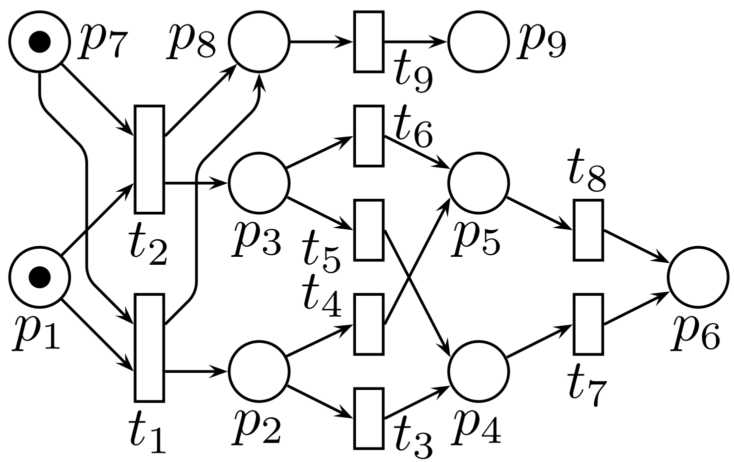


Old observations

- if T_1 and T_2 are stubborn and $T_1 \cap \text{en}(M) \subset T_2 \cap \text{en}(M)$, then T_1 yields better (or as good) reduction results
- favouring the smallest number of enabled transitions does not necessarily yield best reduction

New observation

- a stubborn set with one enabled transition is not always the best choice



The *non-subset choice problem*

- little is known how to choose, if $T_1 \cap \text{en}(M) \not\subseteq T_2 \cap \text{en}(M)$ and $T_2 \cap \text{en}(M) \not\subseteq T_1 \cap \text{en}(M)$

5 Comparison to Ample and Persistent Sets

Ample sets

- [Clarke, Grumberg, Peled 1999] Model Checking
- ample$(M) \subseteq$ en(M)

  C0 If en$(M) \neq \emptyset$, then ample$(M) \neq \emptyset$.

  C1 If $M\ [t_1 \cdots t_n\rangle$ and none of t_1, ..., t_n is in ample(M), then each of them is independent of all transitions in ample(M).

If transitions are deterministic

- **C0** $\wedge$ **C1** $\Rightarrow$ **D0** $\wedge$ **D1** $\wedge$ **D2**
- **D0** $\wedge$ **D1** $\wedge$ **D2** $\not\Rightarrow$ **C0** $\wedge$ **C1**
  - **D1** and **D2** only require independence in certain states

$\Rightarrow$ they are pretty much the same, although stubborn sets have a small advantage

If transitions (or actions) are not necessarily deterministic

- e.g., process algebras
- ample set formulation does not work
- stubborn set formulation does

No disabled transitions in ample sets

$\Rightarrow$ "$\leadsto_M$", clsr(t), and esc(t) cannot be formulated

- ample set algorithms try some obviously "$\leadsto_M$"-closed sets,
  and if that fails, revert to ample(M) = en(M)

Persistent sets

- [Godefroid 1996] LNCS 1032
- deterministic transitions:
  the same as stubborn sets without disabled transitions (except when en$(M) = \emptyset$)
- nondeterministic transitions:
  the formulation does not work

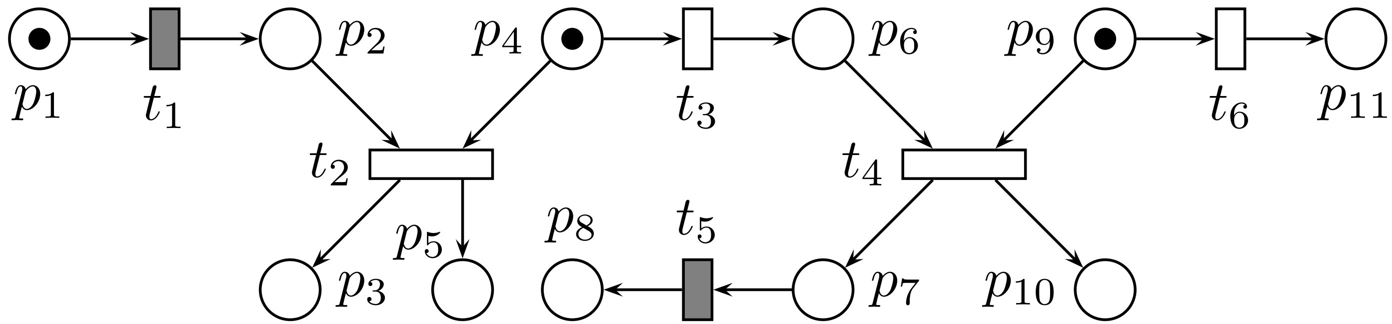
Weak stubborn sets

- **D0** and **D2** replaced by a weaker condition:
  one enabled transition satisfies what **D2** requires from all enabled transitions
- more reduction potential
- we largely lack good algorithms to exploit that potential

$\Rightarrow$ not in this talk

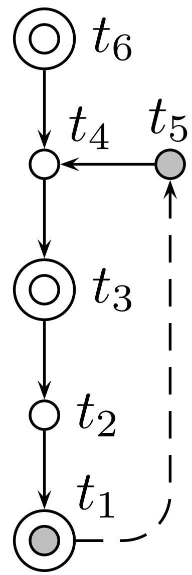
6 Visibility

Assume we want to (dis)prove $\Box(M(p_1) = 0 \vee M(p_8) = 0)$



- $t_3 t_4 t_5$ violates it
- **D0**, **D1**, and **D2** allow $\text{stubb}(\hat{M}) = \{t_1\}$
  $\Rightarrow$ all counterexamples may be lost

Solution



- *atomic propositions*: $M(p_1) = 0$ and $M(p_8) = 0$
- at least transitions that affect atomic propositions are *visible*
- the rest are *invisible*

V If $\text{stubb}(M)$ contains an enabled visible transition, then $\text{stubb}(M)$ contains all visible transitions (also disabled).

- **V** adds the dashed edge to the "$\leadsto_{\hat{M}}$"-graph
  $\Rightarrow$ also t_3 must be in $\text{stubb}(\hat{M})$

Implementation

- add $t \leadsto_M t'$ for every $t \in \mathsf{en}(M) \cap \mathsf{Vis}$ and $t' \in \mathsf{Vis}$
- easy!

Ample sets

> **C2** If $\mathsf{ample}(M)$ contains a visible transition, then $\mathsf{ample}(M) = \mathsf{en}(M)$.

- **C2** $\Rightarrow$ **V** and **V** $\not\Rightarrow$ **C2**
- taking initially an enabled visible transition t_1 cannot be avoided in the example
  $\Rightarrow$ **C2** unnecessarily forces to take t_6

V cannot be formulated without disabled transitions in the stubborn set

- e.g., $\mathsf{Vis} \cap \mathsf{en}(M) \subseteq \mathsf{stubb}(M)$ fails in the example
  - yields $\{t_1\}$

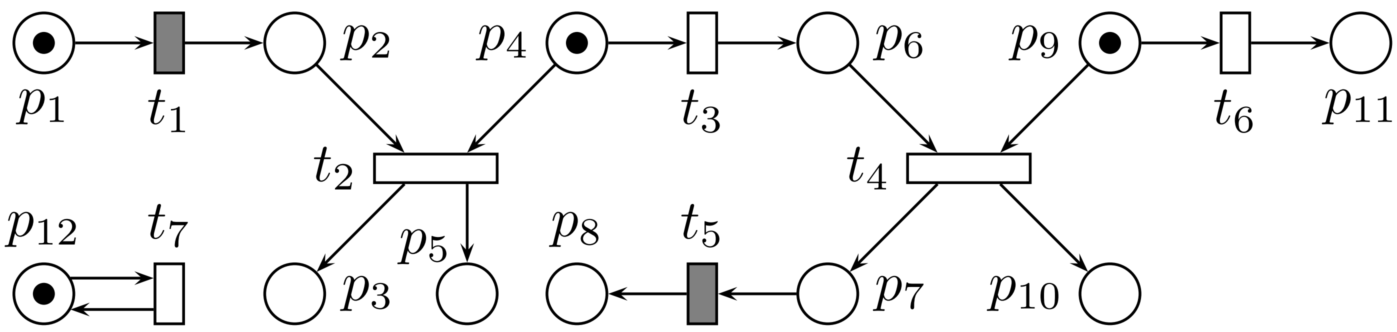
Future work

- a paper replacing a better condition for **V** has been submitted

7 A New Result on Safety Properties

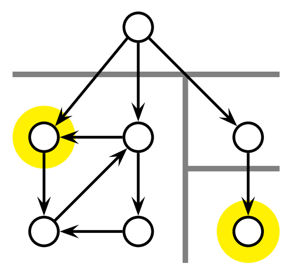
The *ignoring problem*

- $\{t_7\}$ satisfies **D0**, **D1**, **D2**, and **V**
- $\hat{M}\ [t_7\rangle\ \hat{M}$
  $\Rightarrow$ that is all ??



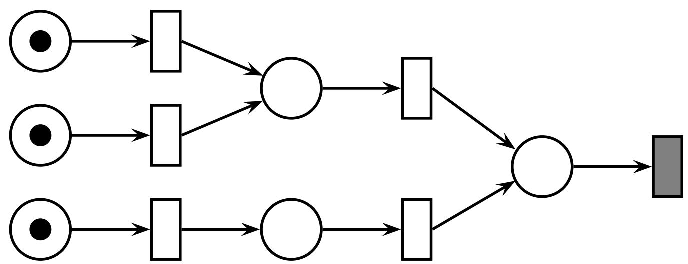
Old solution 1

- for every terminal strong component C of the reduced state space and every $t \in$ en(root(C)), there is $M_t \in C$ such that $t \in$ stubb(M_t)
- construct the reduced state space in depth-first order, apply Tarjan's strong component algorithm, and extend stubb(root(C)) as needed
- **may fire irrelevant transitions**
  - t_6 in the example



Old solution 2

- ... every $t \in$ Vis ...
- **too big stubborn sets**



Interesting transitions T_i

- e.g., all transitions, visible transitions, ...
- every (remaining) counterexample contains at least one interesting transition

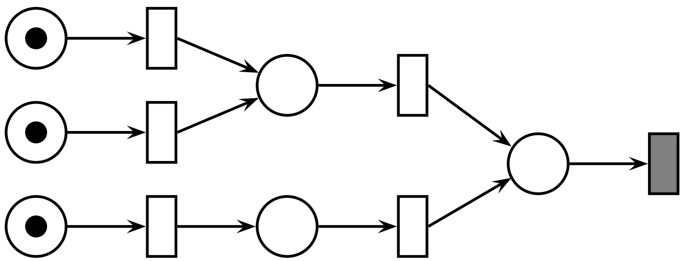
Semi-interesting transitions $T_{\mathsf{si}}(M)$

- at least all interesting transitions
- only semi-interesting transitions can enable disabled interesting transitions

⇒ every remaining counterexample contains
a currently enabled semi-interesting transition

- $T_{\mathsf{si}}(M)$ is computed as $\bigcup_{t \in T_{\mathsf{i}}} \mathsf{clsr}'(t)$, where $t' \leadsto'_M t''$ if and only if $\neg M\,[t'\rangle$ and ...
- for every terminal strong component C of the reduced state space and every $t \in \mathsf{en}(\ T_{\mathsf{si}}(\ \mathsf{root}(C)\ )\ )$, there is $M_t \in C$ such that $t \in \mathsf{stubb}(M_t)$

⇒ The transitions in en(T_{si}(root(C))) are interleaved instead of fired all in root(C)



8 Discussion

Comparison to ample and persistent sets

- same basic idea, different formulations
- advantages of stubborn set formulation:
  - nondeterministic transitions $\leadsto$ process algebras
  - disabled transitions in the set and $\leadsto_M$: better conditions and algorithms
  - (weak stubborn sets)

New results

- small improvement: singleton set not always best
- new **S** condition that combines advantages of two old ones
  - good algorithm is known, but has not been implemented
- (new **V**)

Liveness properties

- in the paper but not in the talk
- the performance of the well-known cycle condition deserves more research
- extending the new **S** to liveness is future work

The non-subset choice problem

- if one stubborn set is not a subset of another in either direction, which one to choose?
- important unstudied problem

Input order may be crucial

- do each measurement with more than one input order!

The how to stop Valmari talking problem:

Thank you for attention!
Questions?