The Saturation algorithm

Kronecker-consistent decomposition of a structured model

A decomposition of a discrete-state model is Kronecker-consistent if:

- $\mathcal{N}$ is disjunctively partitioned according to a set of events $\mathcal{E}$ $\quad$ $\mathcal{N}(\mathbf{i}) = \bigcup_{\alpha\in\mathcal{E}} \mathcal{N}_\alpha(\mathbf{i})$
- $\widehat{\mathcal{X}} = \times_{L\geq k\geq 1} \mathcal{X}_k$, a global state $\mathbf{i}$ consists of L local states $\quad$ $\mathbf{i} = (i_L, \ldots, i_1)$
- and, most importantly, we can write $\quad$ $\mathcal{N}_\alpha(\mathbf{i}) = \times_{L\geq k\geq 1} \mathcal{N}_{k,\alpha}(i_k)$

Define the (potential) incidence matrix $\mathbf{N}[\mathbf{i},\mathbf{j}] = 1 \;\Leftrightarrow\; \mathbf{j} \in \mathcal{N}(\mathbf{i})$

$$\mathbf{N} = \sum_{\alpha\in\mathcal{E}} \mathbf{N}_\alpha = \sum_{\alpha\in\mathcal{E}} \bigotimes_{L\geq k\geq 1} \mathbf{N}_{k,\alpha}$$

We encode the next state function with $L\cdot|\mathcal{E}|$ small matrices $\mathbf{N}_{k,\alpha} \in \mathbb{B}^{|\mathcal{X}_k\times\mathcal{X}_k|}$

for Petri nets, any partition of the places into L subsets will do!

(even with inhibitor, reset, or decision/probabilistic arcs)

Definition of Kronecker product

Given L matrices $\mathbf{A}_L, \ldots, \mathbf{A}_1$, where $\mathbf{A}_k$ is of size $n_k \times n_k$, their Kronecker product is

$$\mathbf{A} = \bigotimes_{L \geq k \geq 1} \mathbf{A}_k \quad \text{of size} \quad n_L \cdots n_1 \times n_L \cdots n_1$$

where

- $\mathbf{A}[\mathbf{i},\mathbf{j}] = \mathbf{A}_L[\mathbf{i}_L,\mathbf{j}_L] \cdot \mathbf{A}_{L-1}[\mathbf{i}_{L-1},\mathbf{j}_{L-1}] \cdots \mathbf{A}_1[\mathbf{i}_1,\mathbf{j}_1]$
- using the following mixed-base numbering schemes for rows and column (indices start at 0)

$$\mathbf{i} = (\ldots((\mathbf{i}_L) \cdot n_{L-1} + \mathbf{i}_{L-1}) \cdot n_{L-2} \cdots) \cdot n_1 + \mathbf{i}_1 = \sum_{L \geq k \geq 1} \mathbf{i}_k \cdot \prod_{l > h \geq 1} n_h$$

$$\mathbf{j} = (\ldots((\mathbf{j}_L) \cdot n_{L-1} + \mathbf{j}_{L-1}) \cdot n_{L-2} \cdots) \cdot n_1 + \mathbf{j}_1 = \sum_{L \geq k \geq 1} \mathbf{j}_k \cdot \prod_{l > h \geq 1} n_h$$

nonzeros: $\eta\left(\bigotimes_{L \geq k \geq 1} \mathbf{A}_k\right) = \prod_{L \geq k \geq 1} \eta(\mathbf{A}_k)$

Kronecker product by example

Given the matrices $\mathbf{A} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix}$

$$\mathbf{A} \otimes \mathbf{B} = \left[\begin{array}{c|c} a_{00}\mathbf{B} & a_{01}\mathbf{B} \\ \hline a_{10}\mathbf{B} & a_{11}\mathbf{B} \end{array}\right] = \left[\begin{array}{ccc|ccc} a_{00}b_{00} & a_{00}b_{01} & a_{00}b_{02} & a_{01}b_{00} & a_{01}b_{01} & a_{01}b_{02} \\ a_{00}b_{10} & a_{00}b_{11} & a_{00}b_{12} & a_{01}b_{10} & a_{01}b_{11} & a_{01}b_{12} \\ a_{00}b_{20} & a_{00}b_{21} & a_{00}b_{22} & a_{01}b_{20} & a_{01}b_{21} & a_{01}b_{22} \\ \hline a_{10}b_{00} & a_{10}b_{01} & a_{10}b_{02} & a_{11}b_{00} & a_{11}b_{01} & a_{11}b_{02} \\ a_{10}b_{10} & a_{10}b_{11} & a_{10}b_{12} & a_{11}b_{10} & a_{11}b_{11} & a_{11}b_{12} \\ a_{10}b_{20} & a_{10}b_{21} & a_{10}b_{22} & a_{11}b_{20} & a_{11}b_{21} & a_{11}b_{22} \end{array}\right]$$

Kronecker product expresses contemporaneity or synchronization

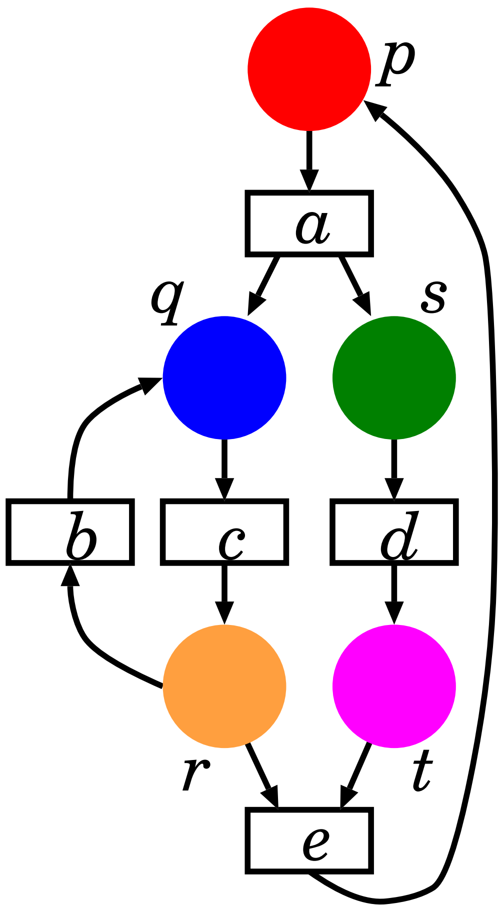
Using structural information to encode $\mathcal{N}$ $(L = 5)$

$\mathcal{X}_5 = ?$ $\mathcal{X}_4 = ?$ $\mathcal{X}_3 = ?$ $\mathcal{X}_2 = ?$ $\mathcal{X}_1 = ?$

EVENTS $\rightarrow$

LEVELS $\downarrow$

$\mathbf{N}_{5,a}$: ?	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{5,e}$: ?
$\mathbf{N}_{4,a}$: ?	$\mathbf{N}_{4,b}$: ?	$\mathbf{N}_{4,c}$: ?	$\mathbf{I}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{N}_{3,b}$: ?	$\mathbf{N}_{3,c}$: ?	$\mathbf{I}$	$\mathbf{N}_{3,e}$: ?
$\mathbf{N}_{2,a}$: ?	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{2,d}$: ?	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{1,d}$: ?	$\mathbf{N}_{1,e}$: ?
$Top(a):5$	$Top(b):4$	$Top(c):4$	$Top(d):2$	$Top(e):5$
$Bot(a):2$	$Bot(b):3$	$Bot(c):3$	$Bot(d):1$	$Bot(e):1$



we determine a priori from the model whether $\mathbf{N}_{k,\alpha} = \mathbf{I}$

Kronecker encoding of $\mathcal{N}$: $\mathbf{N} = \sum_{\alpha\in\{a,b,c,d,e\}} \bigotimes_{5\geq k\geq 1} \mathbf{N}_{k,\alpha}$

$\mathcal{X}_5 : \{p^1, p^0\} \equiv \{0,1\}$ $\mathcal{X}_4 : \{q^0, q^1\} \equiv \{0,1\}$ $\mathcal{X}_3 : \{r^0, r^1\} \equiv \{0,1\}$ $\mathcal{X}_2 : \{s^0, s^1\} \equiv \{0,1\}$ $\mathcal{X}_1 : \{t^0, t^1\} \equiv \{0,1\}$

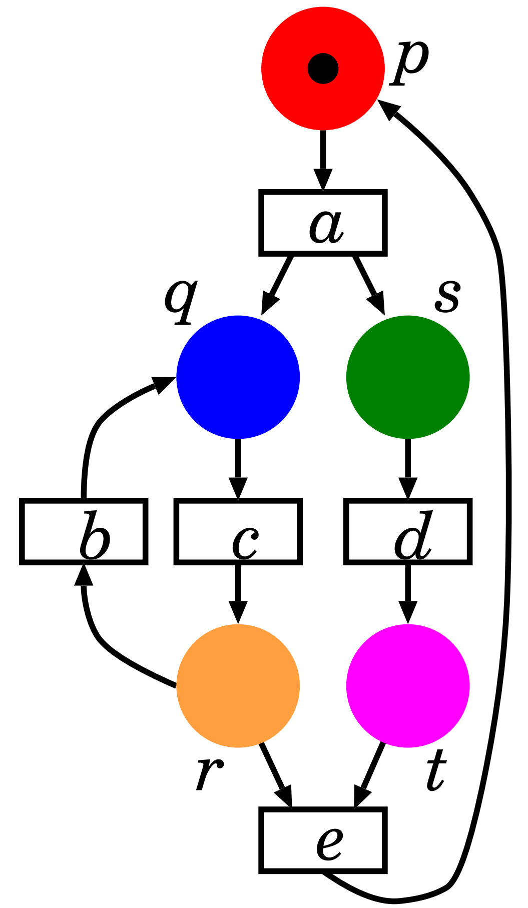
EVENTS $\rightarrow$

LEVELS $\downarrow$

$\mathbf{N}_{5,a}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{5,e}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$
$\mathbf{N}_{4,a}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{N}_{4,b}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{N}_{4,c}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{N}_{3,b}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$	$\mathbf{N}_{3,c}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{N}_{3,e}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$
$\mathbf{N}_{2,a}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{2,d}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{1,d}\colon \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{N}_{1,e}\colon \begin{bmatrix}0&0\\1&0\end{bmatrix}$

$Top(a)\!:\!5$ $Top(b)\!:\!4$ $Top(c)\!:\!4$ $Top(d)\!:\!2$ $Top(e)\!:\!5$

$Bot(a)\!:\!2$ $Bot(b)\!:\!3$ $Bot(c)\!:\!3$ $Bot(d)\!:\!1$ $Bot(e)\!:\!1$



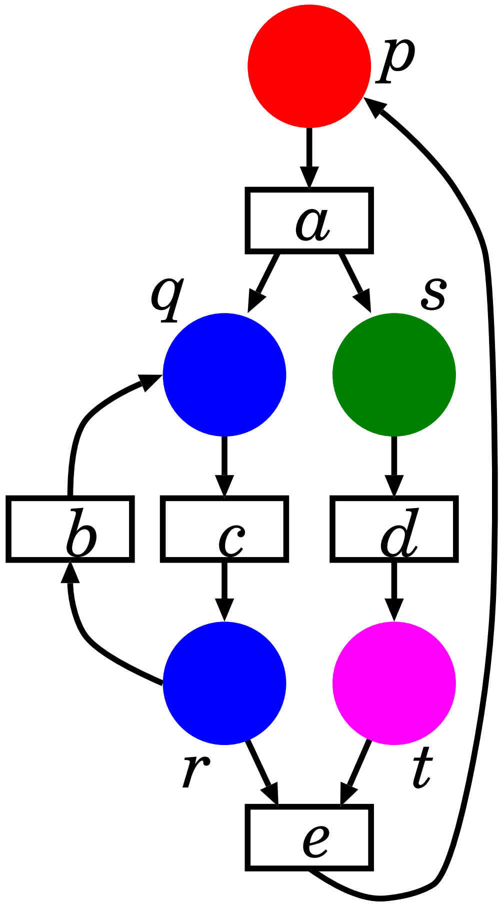
Using structural information to encode $\mathcal{N}$ $(K = 4)$

$\mathcal{X}_4 = ?$ $\mathcal{X}_3 = ?$ $\mathcal{X}_2 = ?$ $\mathcal{X}_1 = ?$

EVENTS →

LEVELS ↓

$\mathbf{N}_{4,a}$:?	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{4,e}$:?
$\mathbf{N}_{3,a}$:?	$\mathbf{N}_{3,b}$:?	$\mathbf{N}_{3,c}$:?	$\mathbf{I}$	$\mathbf{N}_{3,e}$:?
$\mathbf{N}_{2,a}$:?	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{2,d}$:?	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{1,d}$:?	$\mathbf{N}_{1,e}$:?
$Top(a):4$	$Top(b):3$	$Top(c):3$	$Top(d):2$	$Top(e):4$
$Bot(a):2$	$Bot(b):3$	$Bot(c):3$	$Bot(d):1$	$Bot(e):1$



we determine automatically from the model whether $\mathbf{N}_{k,\alpha} = \mathbf{I}$

Kronecker encoding of $\mathcal{N}$: $\mathbf{N} = \sum_{\alpha \in \{a,b,c,d,e\}} \bigotimes_{4 \geq k \geq 1} \mathbf{N}_{k,\alpha}$

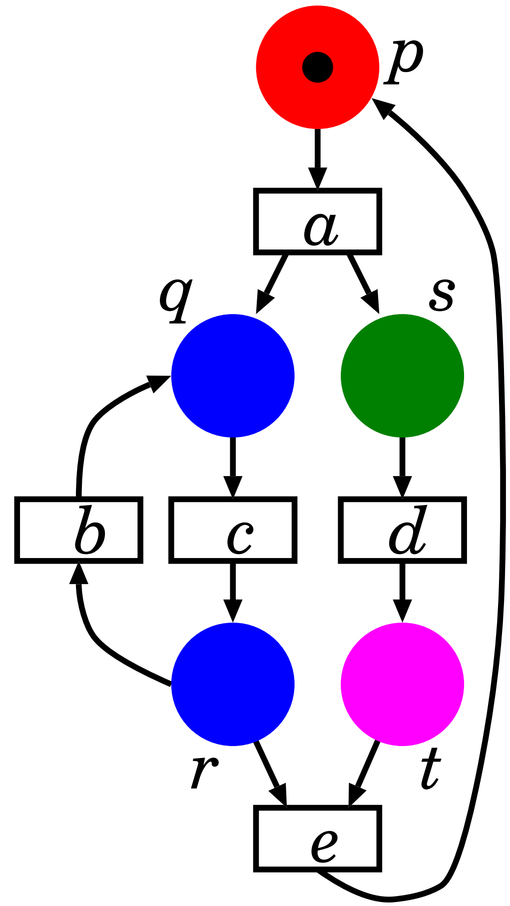
$\mathcal{X}_4 : \{p^1, p^0\} \equiv \{0,1\}$ $\mathcal{X}_3 : \{q^0r^0, q^1r^0, q^0r^1\} \equiv \{0,1,2\}$ $\mathcal{X}_2 : \{s^0, s^1\} \equiv \{0,1\}$ $\mathcal{X}_1 : \{t^0, t^1\} \equiv \{0,1\}$

EVENTS $\longrightarrow$

LEVELS $\downarrow$

$\mathbf{N}_{4,a}: \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{4,e}: \begin{bmatrix}0&0\\1&0\end{bmatrix}$
$\mathbf{N}_{3,a}: \begin{bmatrix}0&1&0\\0&0&0\\0&0&0\end{bmatrix}$	$\mathbf{N}_{3,b}: \begin{bmatrix}0&0&0\\0&0&0\\0&1&0\end{bmatrix}$	$\mathbf{N}_{3,c}: \begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{N}_{3,e}: \begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}$
$\mathbf{N}_{2,a}: \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{2,d}: \begin{bmatrix}0&0\\1&0\end{bmatrix}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{1,d}: \begin{bmatrix}0&1\\0&0\end{bmatrix}$	$\mathbf{N}_{1,e}: \begin{bmatrix}0&0\\1&0\end{bmatrix}$

$Top(a):4$	$Top(b):3$	$Top(c):3$	$Top(d):2$	$Top(e):4$
$Bot(a):2$	$Bot(b):3$	$Bot(c):3$	$Bot(d):1$	$Bot(e):1$

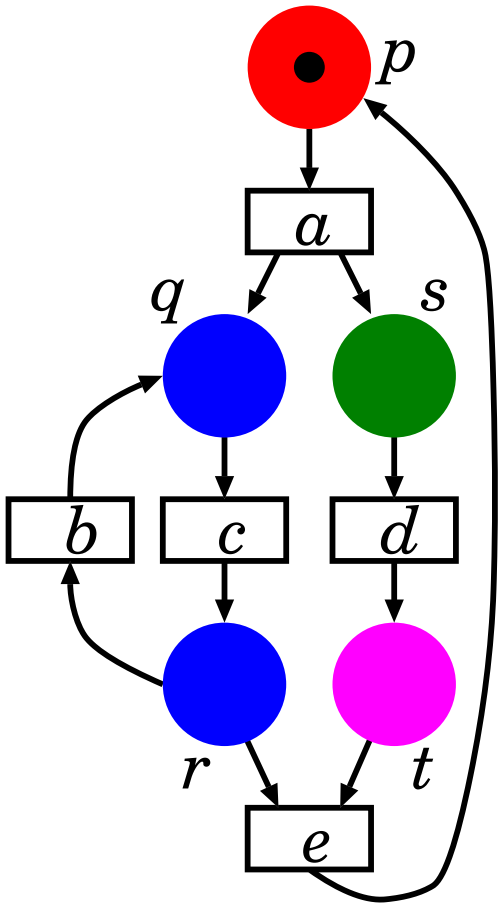


Kronecker encoding of $\mathcal{N}$: $\mathbf{N} = \sum_{\alpha\in\{a,bc,d,e\}} \bigotimes_{4\geq k\geq 1} \mathbf{N}_{k,\alpha}$

$\mathcal{X}_4 : \{p^1, p^0\} \equiv \{0,1\}$ $\mathcal{X}_3 : \{q^0r^0, q^1r^0, q^0r^1\} \equiv \{0,1,2\}$ $\mathcal{X}_2 : \{s^0, s^1\} \equiv \{0,1\}$ $\mathcal{X}_1 : \{t^0, t^1\} \equiv \{0,1\}$

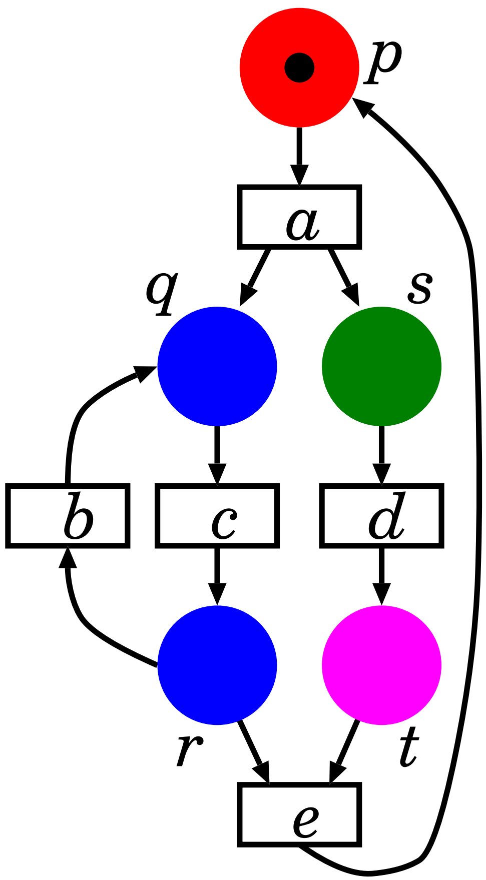
EVENTS →, LEVELS ↓

$\mathbf{N}_{4,a} : \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{4,e} : \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
$\mathbf{N}_{3,a} : \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\mathbf{N}_{3,bc} : \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\mathbf{I}$	$\mathbf{N}_{3,e} : \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$\mathbf{N}_{2,a} : \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$\mathbf{I}$	$\mathbf{N}_{2,d} : \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{N}_{1,d} : \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$\mathbf{N}_{1,e} : \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
$Top(a) : 4$, $Bot(a) : 2$	$Top(bc) : 3$, $Bot(bc) : 3$	$Top(d) : 2$, $Bot(d) : 1$	$Top(e) : 4$, $Bot(e) : 1$



$Top(b) = Bot(b) = Top(c) = Bot(c) = 3$: we have merged b and c into a single local event bc

The matrix $\mathbf{N}$ encoded by the Kronecker descriptor $(L = 4)$

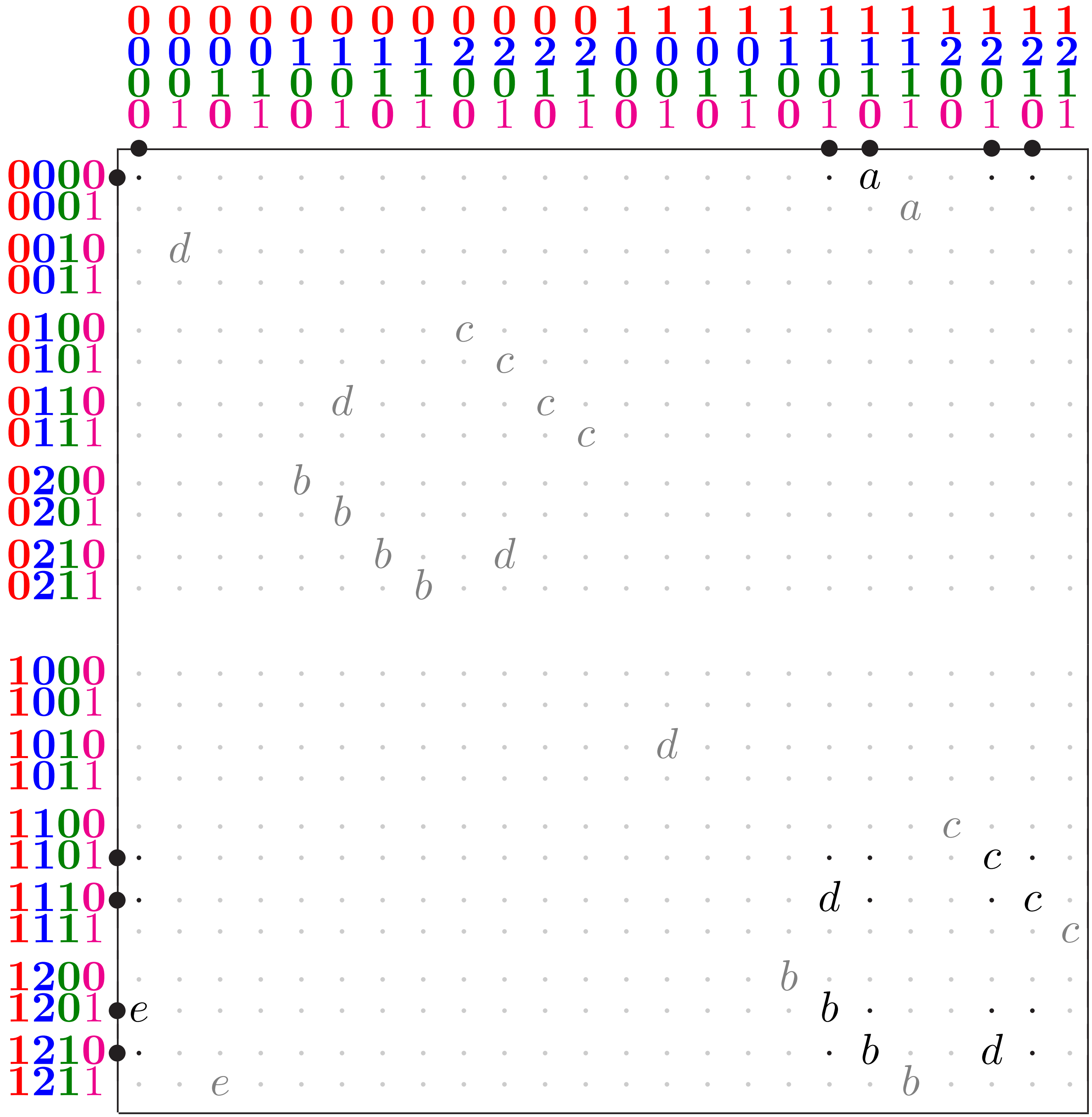


$\{p^1,p^0\}\equiv\{0,1\}$

$\{q^0r^0,q^1r^0,q^0r^1\}\equiv\{0,1,2\}$

$\{s^0,s^1\}\equiv\{0,1\}$

$\{t^0,t^1\}\equiv\{0,1\}$



Exploiting locality

Locality takes into account the span of levels to which $\mathcal{N}_\alpha$ must be applied:

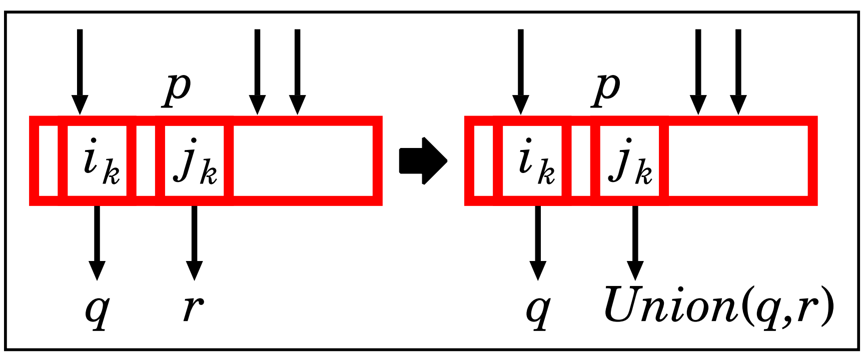
If $\mathbf{i} \in \mathcal{X}_{reach}, \mathbf{j} \in \mathcal{N}_\alpha(\mathbf{i})$, $Top(\alpha) = k \wedge Bot(\alpha) = h$: $\quad \mathbf{j} = (i_L, \ldots, i_{k+1}, j_k, \ldots, j_h, i_{h-1}, \ldots, i_1)$

In addition, it enables in-place updates of a node p at level k:

If $\mathbf{i}' = (i'_L, \ldots, i'_{k+1}, i_k, \ldots, i_1) \in \mathcal{X}_{reach}$: $\mathbf{j}' \in \mathcal{N}_\alpha(\mathbf{i}') \wedge \mathbf{j}' = (i'_L, \ldots, i'_{k+1}, j_k, \ldots, j_h, i_{h-1}, \ldots, i_1)$

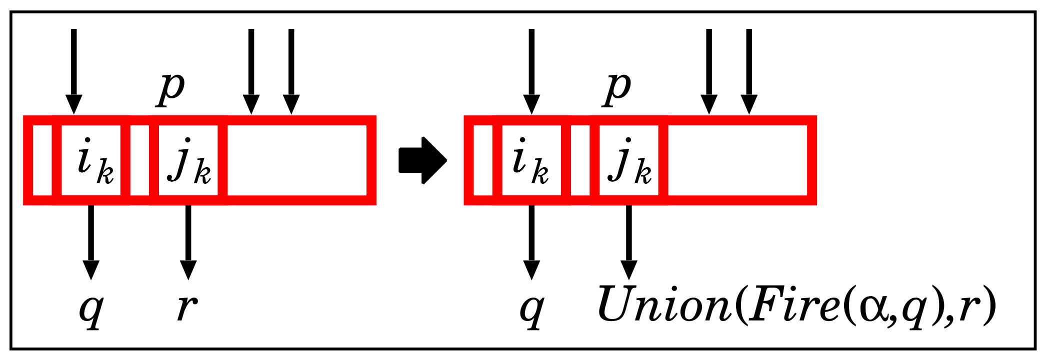
$Top(\alpha) = Bot(\alpha)$

Local event α: $i_k \xrightarrow{\alpha} j_k$



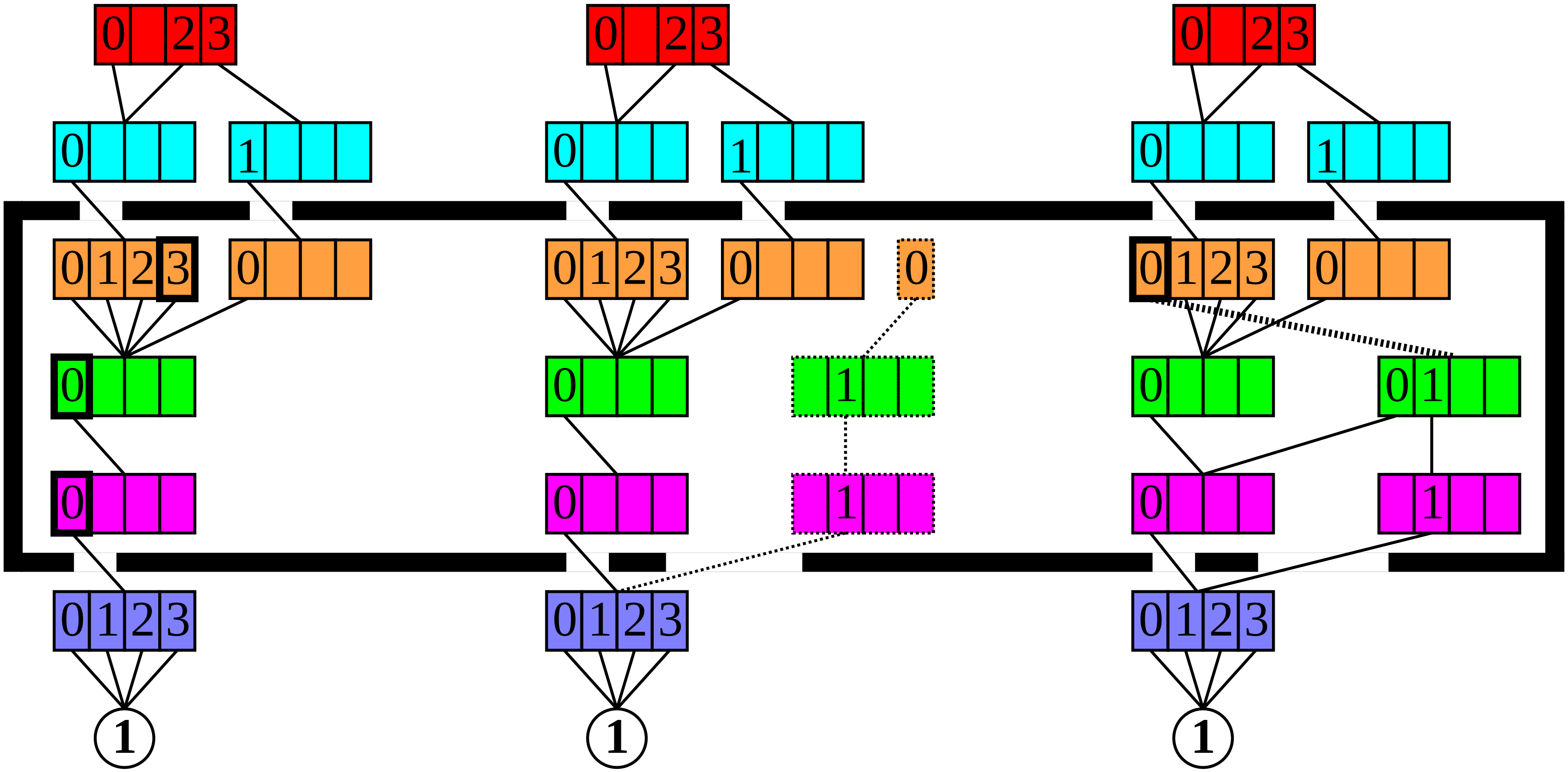
$Top(\alpha) > Bot(\alpha)$

Synchronizing event α: $(i_k, \ldots, i_h) \xrightarrow{\alpha} (j_k, \ldots, j_h)$



locality and **in-place-updates** save huge amounts of computation

How to exploit locality when firing an event



$$\mathcal{S}: \begin{matrix} [0,0,*,0,0,*] \\ [2,0,*,0,0,*] \\ [3,1,0,0,0,*] \end{matrix} \qquad [-,-,3,0,0,-] \xrightarrow{\alpha} [-,-,0,1,1,-] \qquad \mathcal{S}: \begin{matrix} [0,0,*,0,0,*] \\ {\color{red}[0,0,0,1,1,*]} \\ [2,0,*,0,0,*] \\ {\color{red}[2,0,0,1,1,*]} \\ [3,1,0,0,0,*] \end{matrix}$$

Improving the iteration strategy

Traditional iteration using a partitioned $\mathcal{N}$:

$$\begin{aligned} \mathcal{X}^{(\alpha)} &\leftarrow \mathcal{N}_\alpha(\mathcal{X}_{reach}) \\ &\cdots \\ \mathcal{X}^{(\omega)} &\leftarrow \mathcal{N}_\omega(\mathcal{X}_{reach}) \\ \mathcal{X}_{reach} &\leftarrow \mathcal{X}_{reach} \cup \mathcal{X}^{(\alpha)} \cup \cdots \cup \mathcal{X}^{(\omega)} \end{aligned}$$

We can improve by chaining [Roig95]:

$$\begin{aligned} \mathcal{X}_{reach} &\leftarrow \mathcal{X}_{reach} \cup \mathcal{N}_\alpha(\mathcal{X}_{reach}) \\ &\cdots \\ \mathcal{X}_{reach} &\leftarrow \mathcal{X}_{reach} \cup \mathcal{N}_\omega(\mathcal{X}_{reach}) \end{aligned}$$

And even more by exhaustive chaining:

$$\begin{aligned} \mathcal{X}_{reach} &\leftarrow \mathcal{X}_{reach} \cup \mathcal{N}_\alpha^*(\mathcal{X}_{reach}) \\ &\cdots \\ \mathcal{X}_{reach} &\leftarrow \mathcal{X}_{reach} \cup \mathcal{N}_\omega^*(\mathcal{X}_{reach}) \end{aligned}$$

But the best strategy by far is to saturate MDD nodes recursively bottom-up:

- a node at level k is saturated if it is a fixed point w.r.t. all events α s.t. $Top(\alpha) \leq k$
- traditional idea of a global fixed-point iteration for the overall MDD disappears

enormous savings in both time and (peak) memory

Saturation: an iteration strategy based on the model structure

MDD node p at level k is ***saturated*** if it encodes a fixed point w.r.t. any event α s.t. the highest MDD level it depends on, $Top(\alpha)$, is at most k $\Rightarrow$ all MDD nodes reachable from p are also saturated

- build the L-level MDD encoding of $\mathcal{X}_{init}$ — if $|\mathcal{X}_{init}| = 1$, there is one node per level
- saturate each node at level 1: fire in them all events α s.t. $Top(\alpha) = 1$
- saturate each node at level 2: fire in them all events α s.t. $Top(\alpha) = 2$
  (if this creates nodes at level 1, saturate them immediately upon creation)
- saturate each node at level 3: fire in them all events α s.t. $Top(\alpha) = 3$
  (if this creates nodes at levels 2 or 1, saturate them immediately upon creation)
- ...
- saturate the root node at level L: fire in it all events α s.t. $Top(\alpha) = L$
  (if this creates nodes at levels $L-1$, $L-2$, $\ldots$, 1, saturate them immediately upon creation)

states are **not** discovered in breadth-first order

enormous time and memory savings for asynchronous systems

Saturation pseudocode

$Generate$(in $\mathbf{s}$: array$[1..L]$ of lcl) : idx

- $p \leftarrow \mathbf{1}$;
- for $k = 1$ to L do
  - $r \leftarrow NewNode(k)$;
  - $r[\mathbf{s}[k]] \leftarrow p$;
  - $Saturate(k, r)$;
  - $UniqueTableInsert(k, r)$;
  - $p \leftarrow r$;
  - return r;

$Saturate$(in k : lvl, p : idx)

- repeat
  - $pCng \leftarrow false$;
  - foreach $\alpha \in \mathcal{E}_k$ do
    - $\mathcal{L} \leftarrow Locals(k, \alpha, p)$;
    - while $\mathcal{L} \neq \emptyset$ do
      - $i \leftarrow Pick(\mathcal{L})$;
      - $f \leftarrow RecFire(k-1, \alpha, p[i])$;
      - if $f \neq \mathbf{0}$ then
        - foreach $j \in \mathcal{N}_{k,\alpha}(i)$ do
          - $u \leftarrow Union(k-1, f, p[j])$;
          - if $u \neq p[j]$ then
            - $p[j] \leftarrow u$;
            - $pCng \leftarrow true$;
            - if $\mathcal{N}_{k,\alpha}(j) \neq \emptyset$ then
              - $\mathcal{L} \leftarrow \mathcal{L} \cup \{j\}$;
- until $pCng = false$;

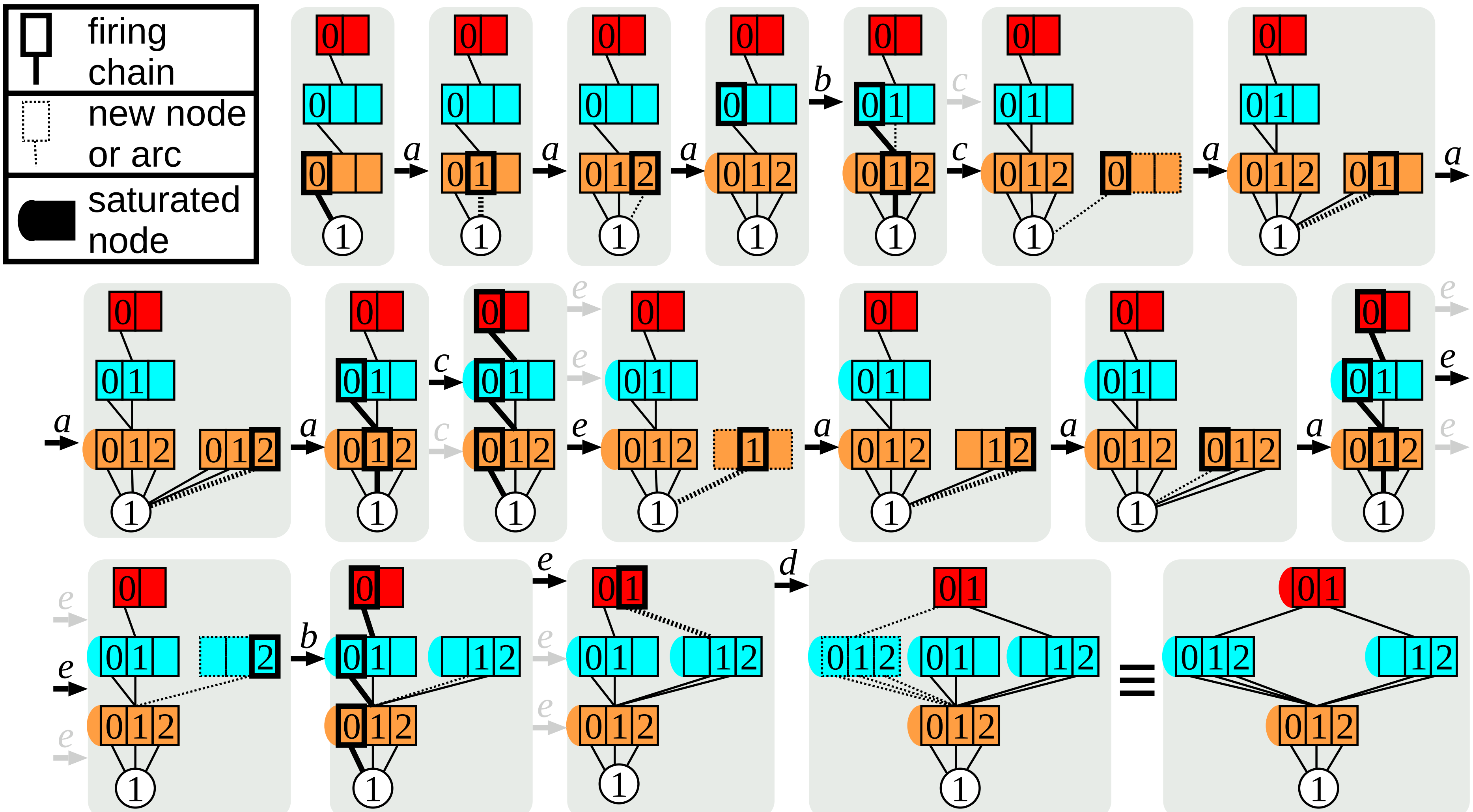
$RecFire$(in h : lvl, α : $evnt$, q : idx) : idx

- if $h < Bot(\alpha)$ then return q;
- if $Find(FC[h], (q, \alpha), s)$ then return s;
- $s \leftarrow NewNode(h)$;
- $sCng \leftarrow false$;
- $\mathcal{L} \leftarrow Locals(h, \alpha, q)$;
- while $\mathcal{L} \neq \emptyset$ do
  - $i \leftarrow Pick(\mathcal{L})$;
  - $f \leftarrow RecFire(h-1, \alpha, q[i])$;
  - if $f \neq \mathbf{0}$ then
    - foreach $j \in \mathcal{N}_{h,\alpha}(i)$ do
      - $u \leftarrow Union(h-1, f, s[j])$;
      - if $u \neq s[j]$ then
        - $s[j] \leftarrow u$;
        - $sCng \leftarrow true$;
- if $sCng$ then
  - $Saturate(h, s)$;
- $UniqueTableInsert(h, s)$;
- $Insert(FC[h], (q, \alpha), s)$;
- return s;

FC is the firing cache

Saturation by example

level	event: a	event: b	event: c	event: d	event: e
3	$*$	$*$	$*$	$1 \to 0$	$0 \to 1$
2	$*$	$0 \to 1, 2 \to 1$	$0 \to 1$	$*$	$0 \to 2$
1	$0 \to 1, 1 \to 2, 2 \to 0$	$*$	$1 \to 0$	$*$	$0 \to 1$



Saturation behavior and properties

Traditional approaches apply the global next-state function $\mathcal{N}$ once to each node at each iteration and make extensive use of the unique table and operation caches

- We exhaustively fire each event α in each node p at level $k = Top(\alpha)$, from $k = 1$ up to L
- We must consider redundant nodes as well, thus we prefer quasi-reduced MDDs
- Once node p at level k is saturated, we never fire an event α with $k = Top(\alpha)$ on p again
- The recursive $Fire$ calls stop at level $Bot(\alpha)$, although the $Union$ calls can go deeper
- Only saturated nodes are placed in the unique table and in the union and firing caches
- Many (most?) nodes we insert in the MDD will still be present in the final MDD
- Firing α in p benefits from having saturated the nodes below p (finds more states)

usually enormous memory and time savings

but Saturation is **not** optimal for all models

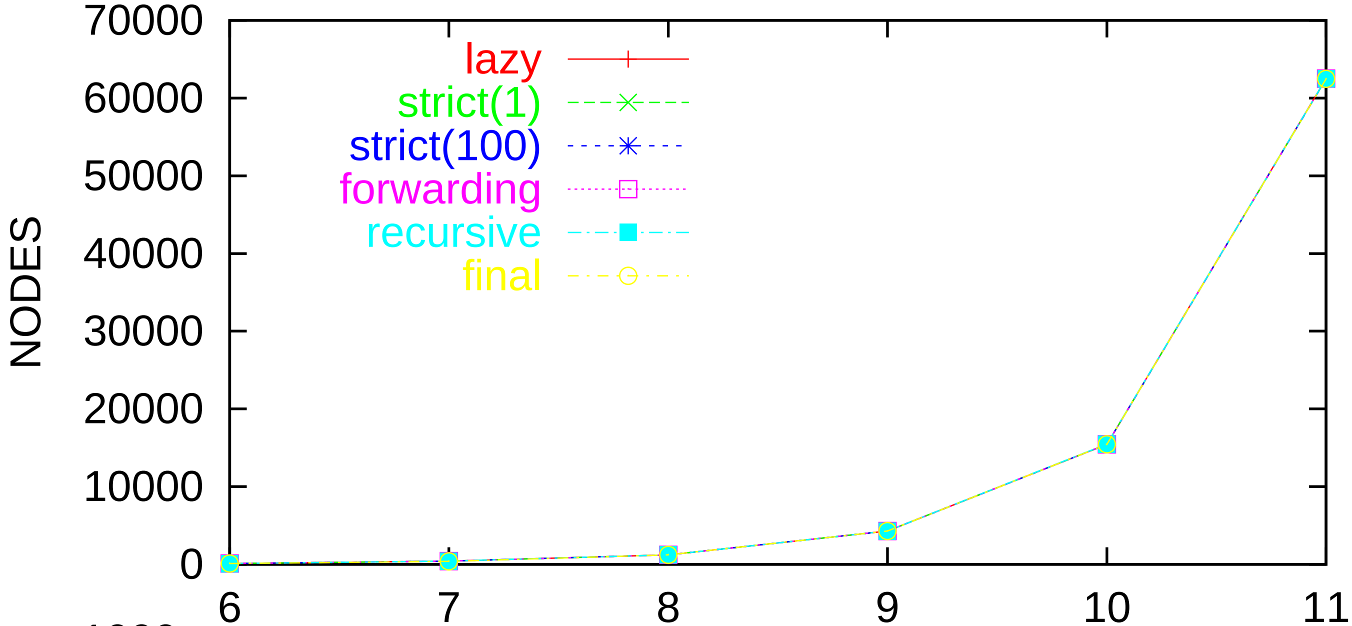
Implementation details

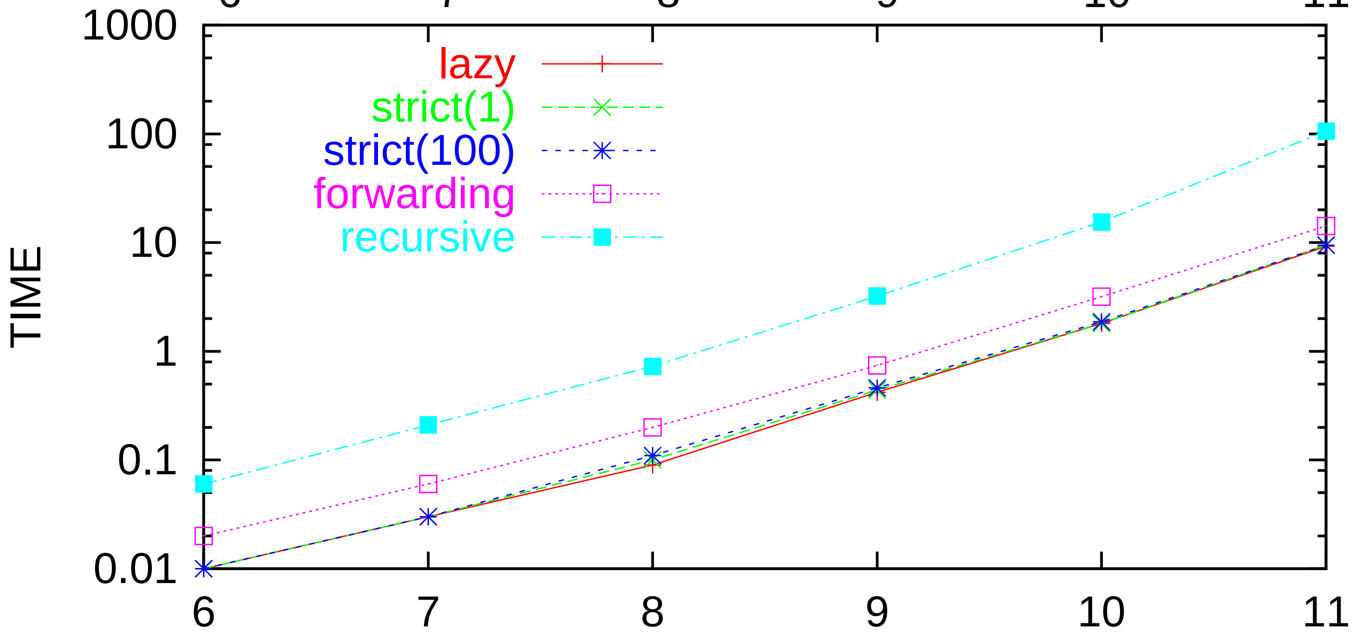
In traditional approaches, nodes can become disconnected and must be deleted and reused

- We use a standard per-node incoming–arc counter to detect disconnection
- LAZY policy has best runtime: recycle only at the end (this is feasible with our approach!)
- STRICT(1) minimizes memory: recycling after each deletion
- STRICT(r) is a good compromise: recycle when there are r deleted nodes

Experimental results: Queens

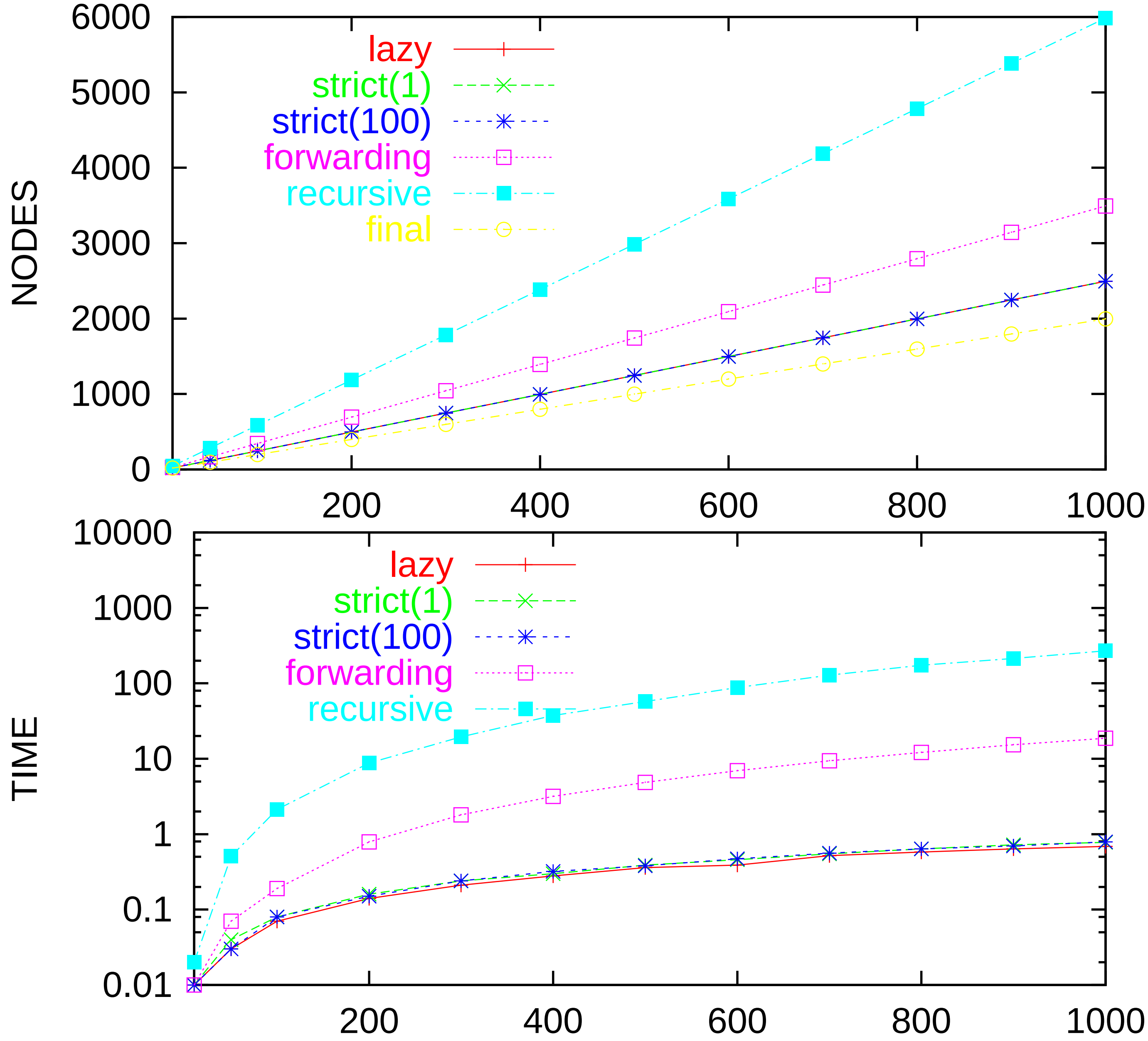
N	$\mathcal{S}$
6	1.53×10^2
7	5.52×10^2
8	2.06×10^3
9	8.39×10^3
10	3.55×10^4
11	1.67×10^5





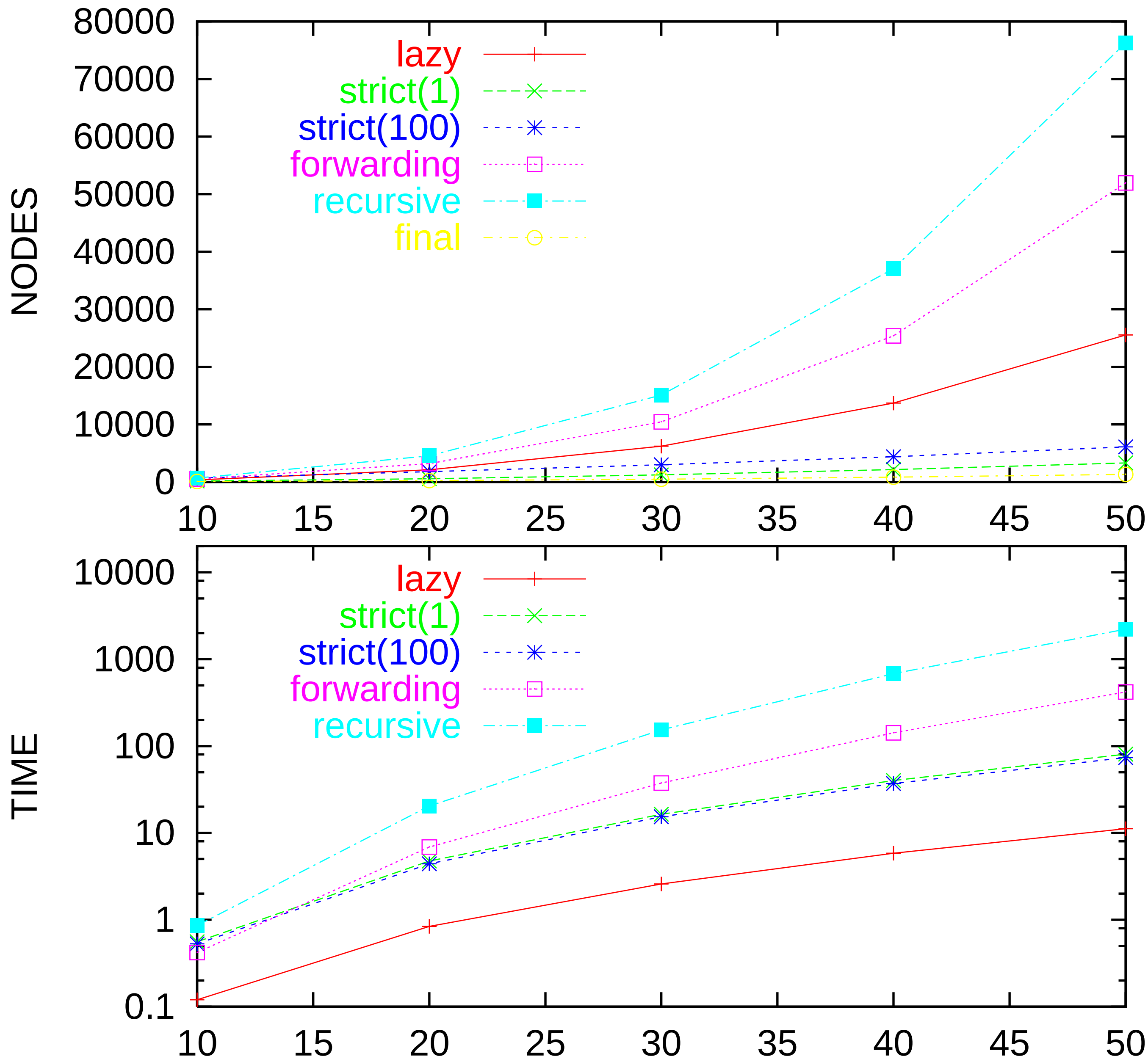
Experimental results: Dining philosophers

N	$\mathcal{S}$
100	4.97×10^{62}
200	2.47×10^{125}
400	6.10×10^{250}
600	1.51×10^{376}
800	3.72×10^{501}
1000	9.18×10^{626}



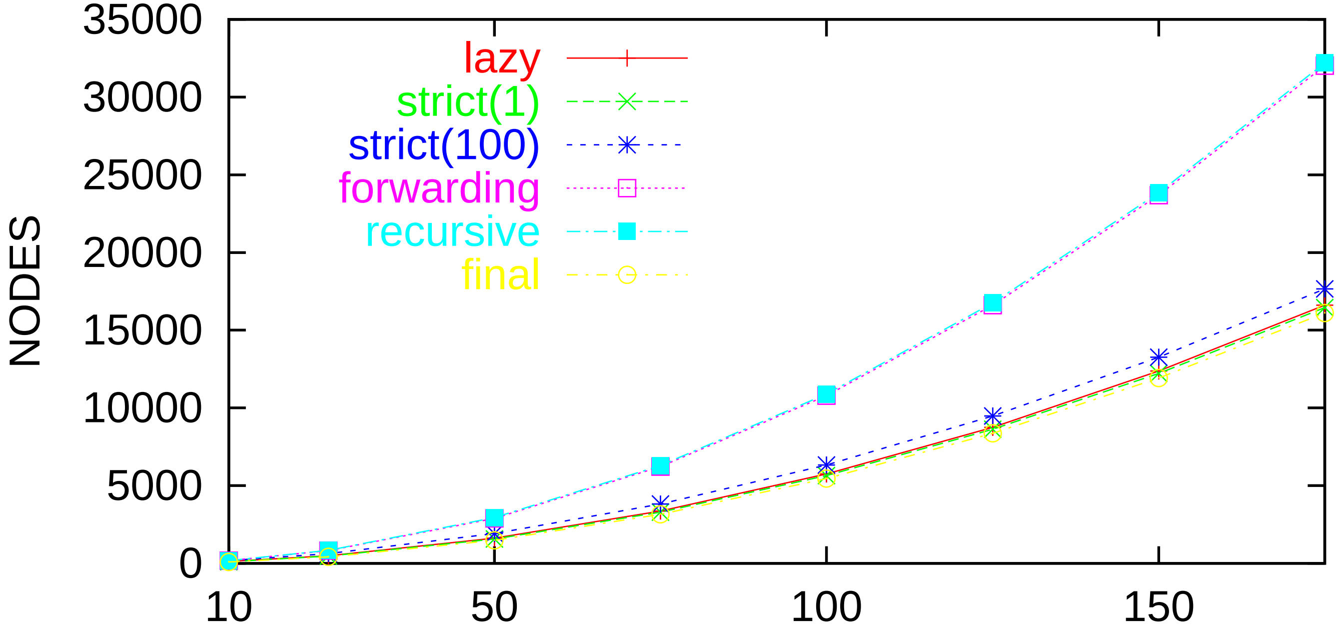
Experimental results: Slotted ring

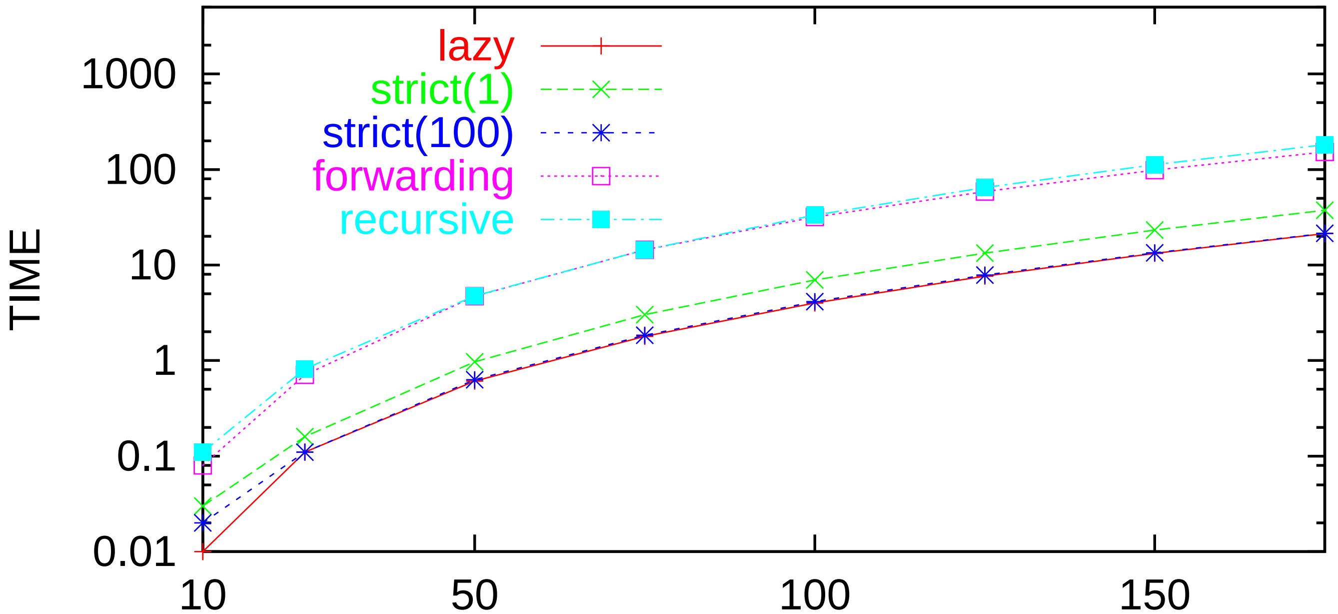
N	$\mathcal{S}$
10	8.29×10^{9}
20	2.73×10^{20}
30	1.04×10^{31}
40	4.16×10^{41}
50	1.72×10^{52}



Experimental results: Round robin

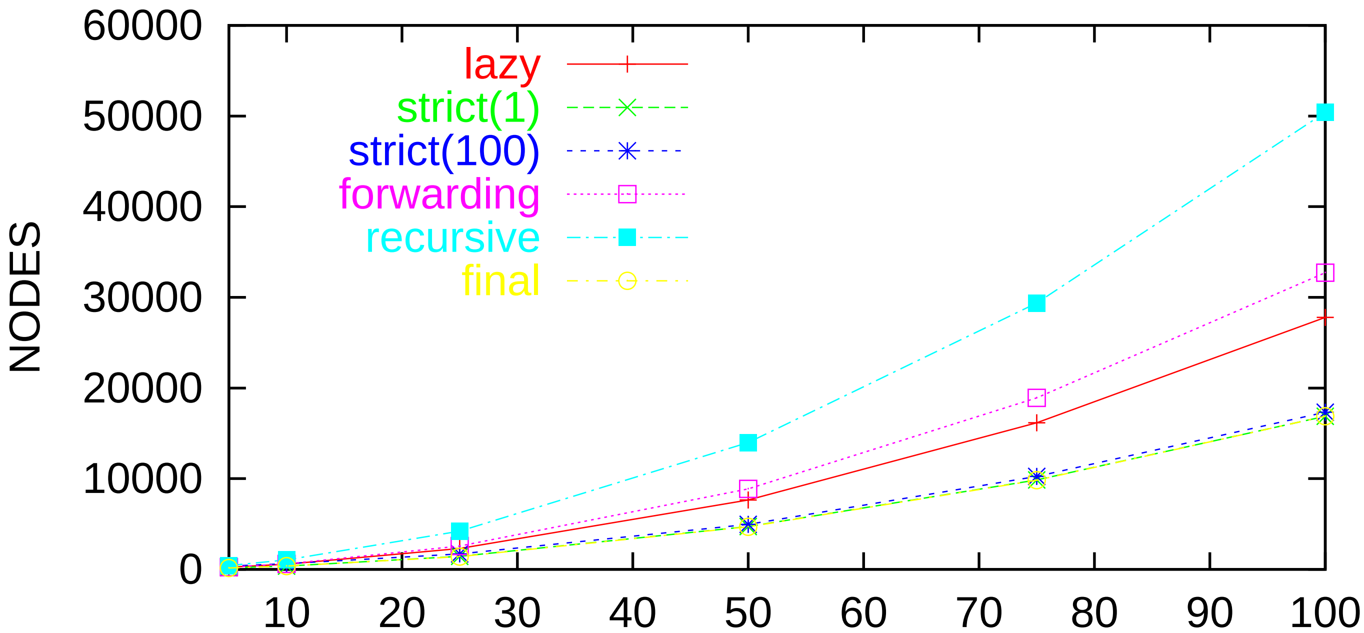
N	$\mathcal{S}$
10	2.30×10^{4}
25	1.89×10^{9}
50	1.27×10^{17}
100	2.85×10^{32}
150	4.82×10^{47}
200	7.23×10^{62}

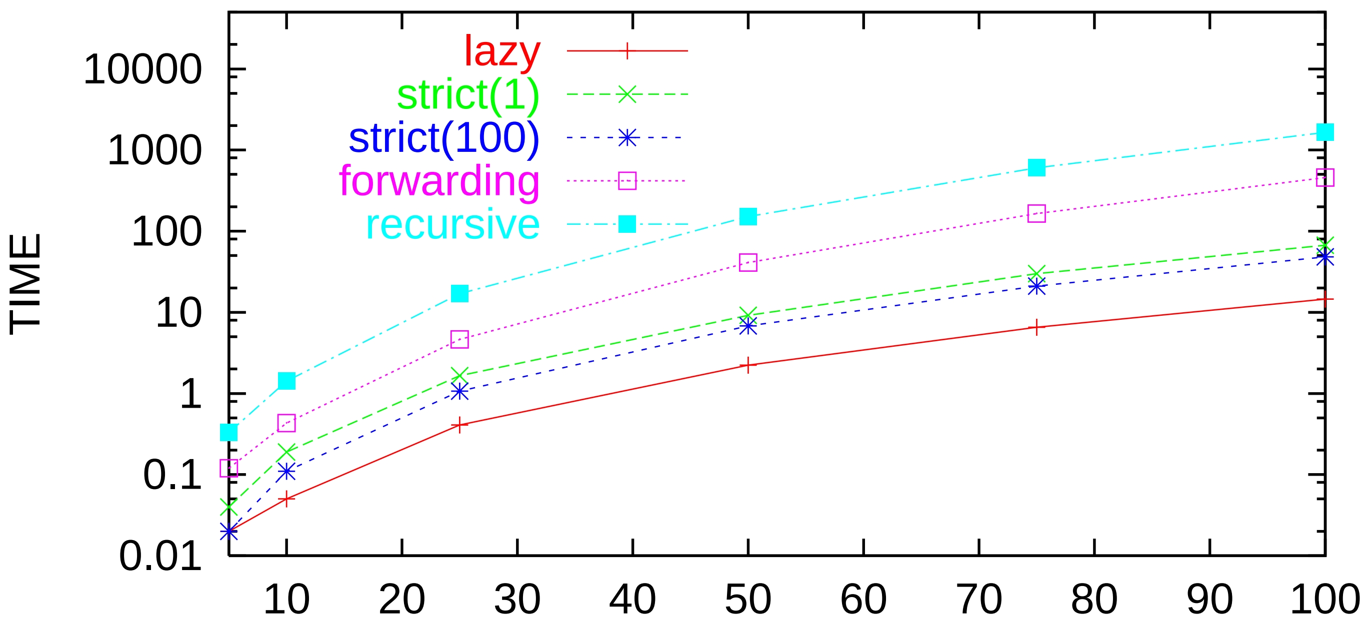




Experimental results: FMS

N	$\mathcal{S}$
5	2.90×10^{6}
10	2.50×10^{9}
25	8.54×10^{13}
50	4.24×10^{17}
75	6.98×10^{19}
100	2.70×10^{21}

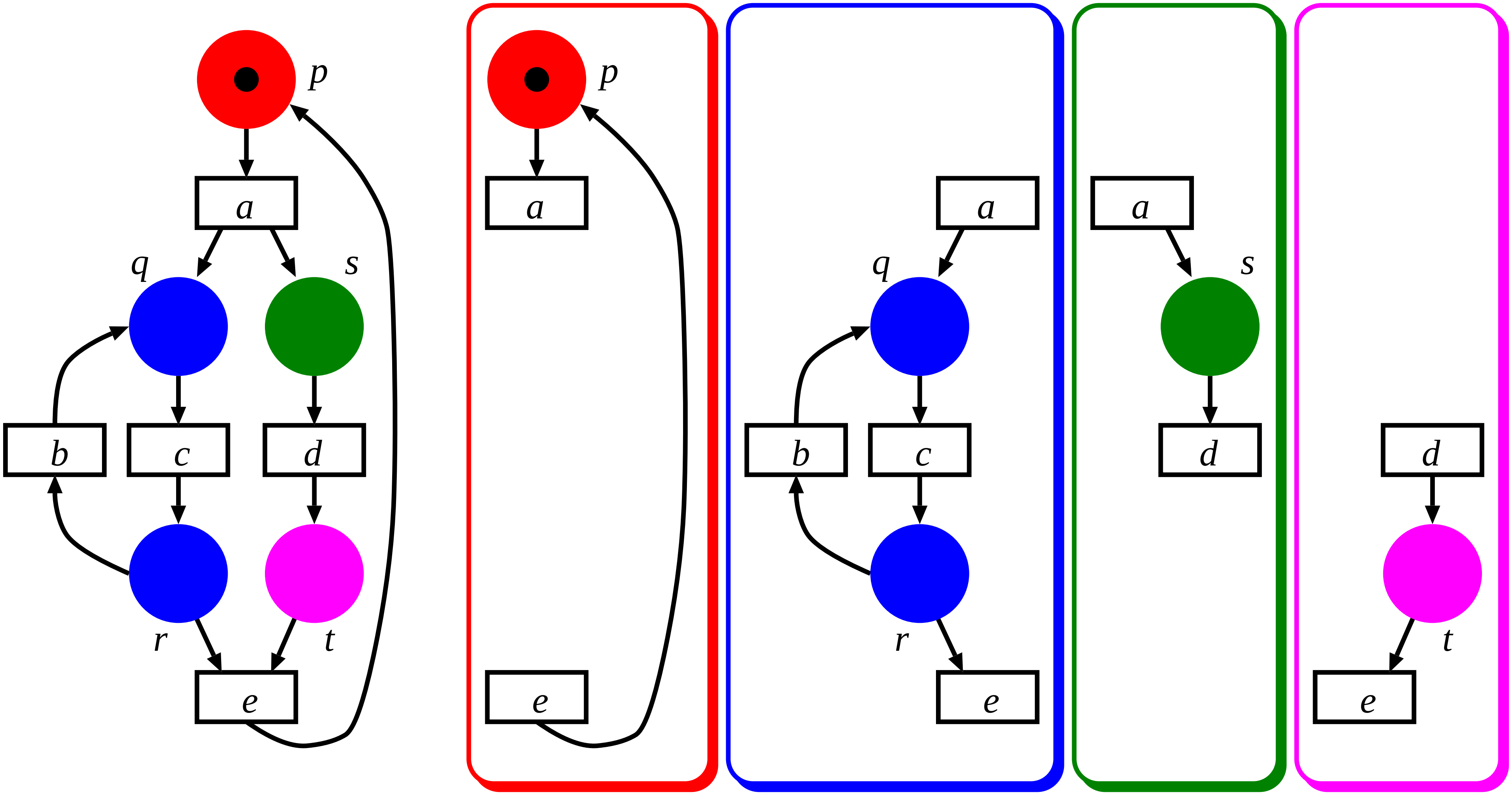




Saturation (SMART) vs. breadth-first search (NuSMV)

N	$\lvert\mathcal{X}_{reach}\rvert$	Peak memory (kB)		Time (sec)	
		SMART	NuSMV	SMART	NuSMV
Dining Philosophers: $L = N$					
50	2.23×10^{31}	22	10,819	0.15	5.9
200	2.47×10^{125}	93	72,199	0.68	12,905.7
10,000	4.26×10^{6269}	4,686	—	877.82	—
Slotted Ring Network: $L = N$					
10	8.29×10^{9}	28	10,819	0.13	5.5
15	1.46×10^{15}	80	13,573	0.39	2,039.5
200	8.38×10^{211}	120,316	—	902.11	—
Round Robin Mutual Exclusion: $L = N + 1$					
20	4.72×10^{7}	20	7,306	0.07	0.8
100	2.85×10^{32}	372	26,628	3.81	2,475.3
300	1.37×10^{93}	3,109	—	140.98	—
Flexible Manufacturing System: $L = 19$					
10	4.28×10^{6}	26	11,238	0.05	9.4
20	3.84×10^{9}	101	31,718	0.20	1,747.8
250	3.47×10^{26}	69,087	—	231.17	—

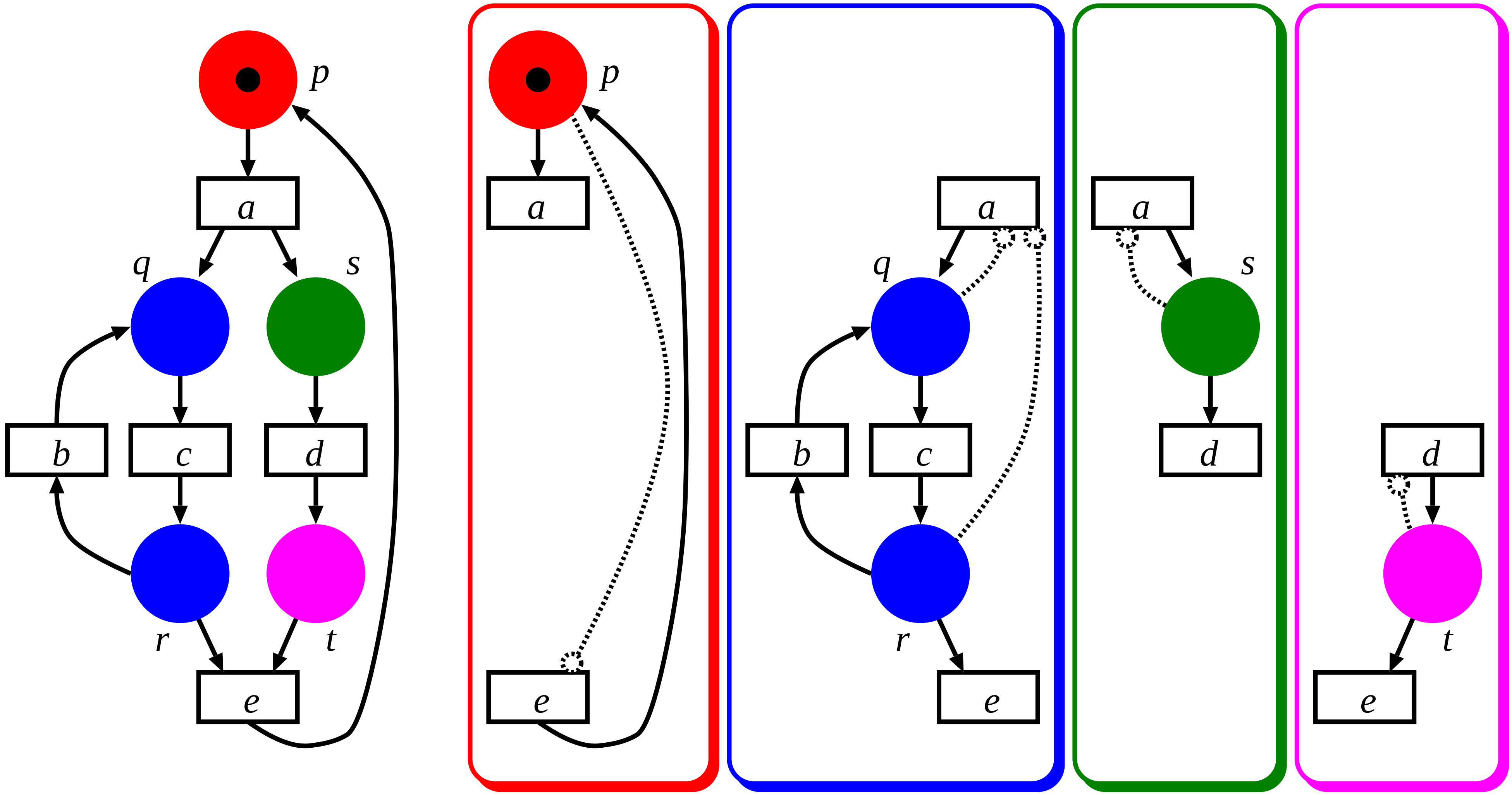
Problem with pre-generating the sets $\mathcal{X}_k$



Each one of these four submodels is unbounded (in isolation)

A poor solution to the problem

Modifying the model to enforce known bounds is difficult (especially if we want the smallest $\mathcal{X}_k$)



More importantly, it’s dangerous!

Explicit local and symbolic global state-space generation

Problem: local state spaces $\mathcal{X}_k$ are not known a priori

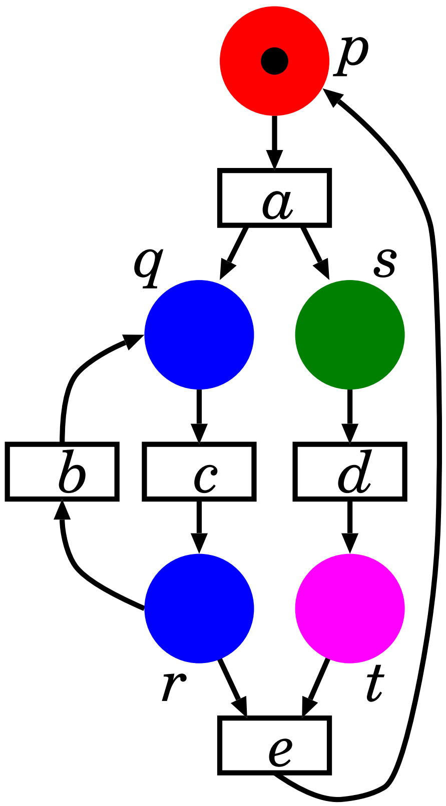
Solution: build $\mathcal{X}_k$ "on the fly" (explicitly) alongside the overall state space $\mathcal{X}_{reach}$ (symbolically)

for each component $\mathbf{i}_k$ of each initial state $(\mathbf{i}_L, \ldots, \mathbf{i}_1) \in \mathcal{X}_{init}$
  $\mathit{Confirm}(\mathbf{i}_k)$; *(explicitly) build row $\mathbf{N}_{k,\alpha}[\mathbf{i}_k, \cdot]$ for each $\alpha \in \mathcal{E}$ dependent on level k*
while the MDD encoding $\mathcal{X}_{reach}$ has not reached its fixed point w.r.t. $\mathcal{N}$ do
  symbolically explore global states reachable from the currently-known $\mathcal{X}_{reach}$;
  use the rows $\mathbf{N}_{k,\alpha}[\mathbf{i}_k, \cdot]$ of confirmed local states $\mathbf{i}_k$ only
  confirm $\mathbf{j}_k$ as soon as an entry $\mathbf{N}_{k,\alpha}[\mathbf{i}_k, \mathbf{j}_k]$ is used in a successful symbolic firing
end while

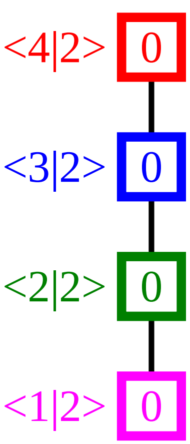
no need to know a priori the range of each state variable

at the end, we have the smallest confirmed $\mathcal{X}_k$ possible

Saturation-OTF in action: initial setup



a	bc	d	e
	I	I	
		I	
	I		I
I	I		



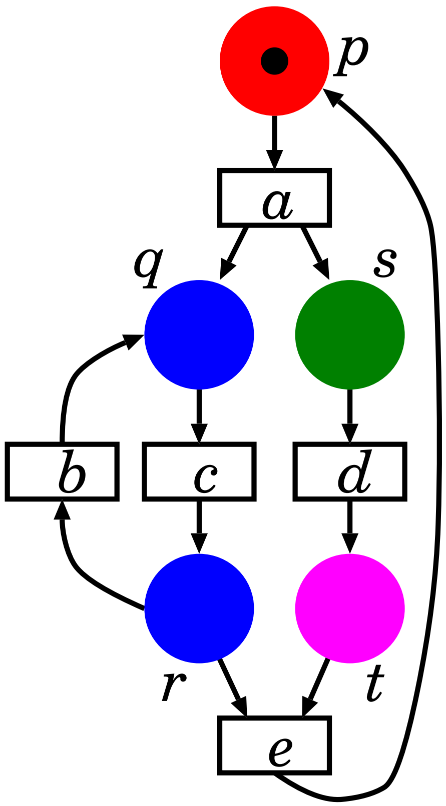
$\mathcal{X}_4 = \{p^1\} \equiv \{0\}$

$\mathcal{X}_3 = \{q^0 r^0\} \equiv \{0\}$

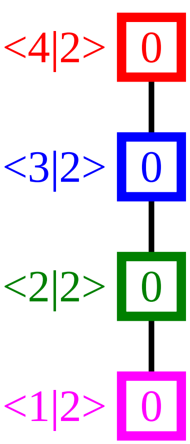
$\mathcal{X}_2 = \{s^0\} \equiv \{0\}$

$\mathcal{X}_1 = \{t^0\} \equiv \{0\}$

Saturation-OTF in action: confirm initial state



a	bc	d	e
$0:1$	**I**	**I**	$0:2$
$0:1$	$0:-$	**I**	$0:-$
$0:1$	**I**	$0:-$	**I**
I	**I**	$0:1$	$0:-$



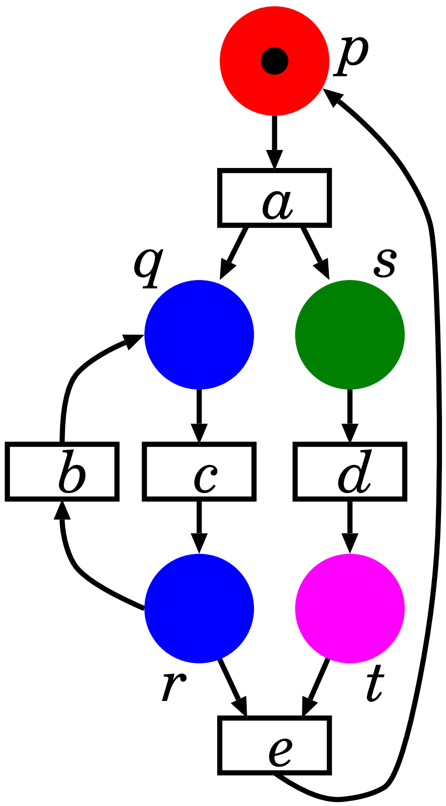
$$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$$

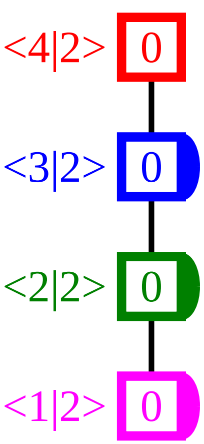
$$\mathcal{X}_2 = \{\underline{s^0}, s^1\} \equiv \{\underline{0}, 1\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, t^1\} \equiv \{\underline{0}, 1\}$$

Saturation-OTF in action: saturate $\langle 1|2\rangle$, $\langle 2|2\rangle$, $\langle 3|2\rangle$ (no firing)



a	bc	d	e
$0:1$	$\mathbf{I}$	$\mathbf{I}$	$0:2$
$0:1$	$0:-$	$\mathbf{I}$	$0:-$
$0:1$	$\mathbf{I}$	$0:-$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$0:1$	$0:-$



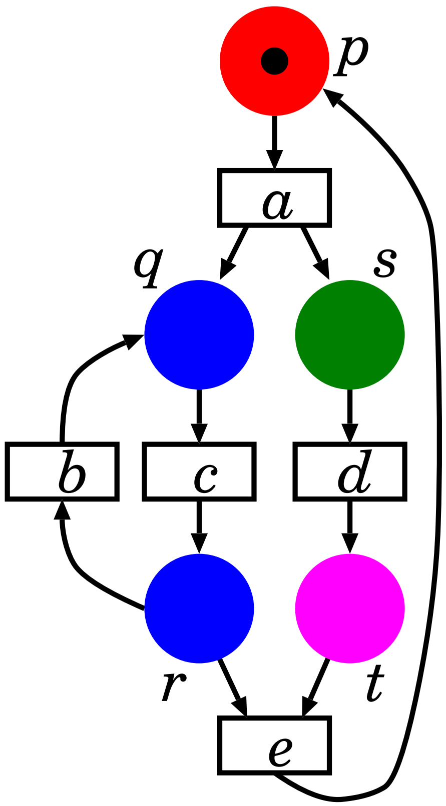
$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$

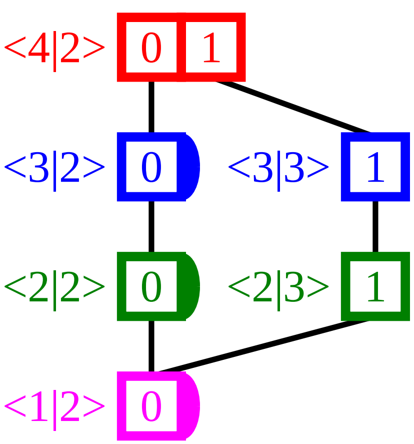
$\mathcal{X}_2 = \{\underline{s^0}, s^1\} \equiv \{\underline{0}, 1\}$

$\mathcal{X}_1 = \{\underline{t^0}, t^1\} \equiv \{\underline{0}, 1\}$

Saturation-OTF in action: saturate $\langle 4|2 \rangle$ (fire a)



a	bc	d	e
$0:1$	**I**	**I**	$0:2$
$0:1$	$0:-$	**I**	$0:-$
$0:1$	**I**	$0:-$	**I**
I	**I**	$0:1$	$0:-$



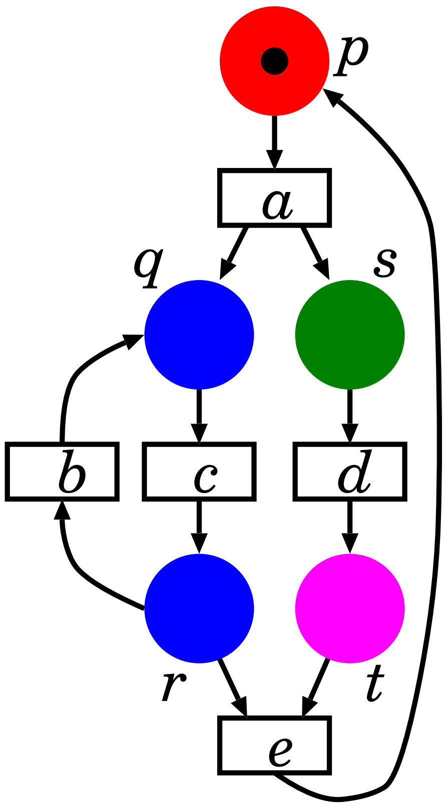
$$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$$

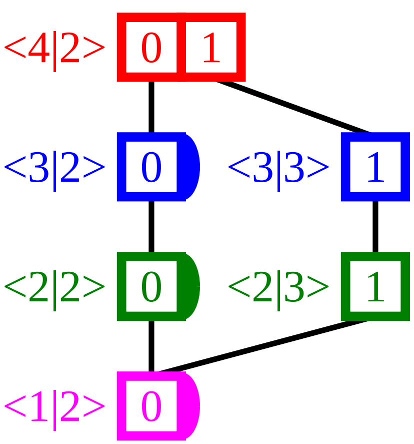
$$\mathcal{X}_2 = \{\underline{s^0}, s^1\} \equiv \{\underline{0}, 1\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, t^1\} \equiv \{\underline{0}, 1\}$$

Saturation-OTF in action: confirm $s^1 \equiv 1$



a	bc	d	e
0 : 1	I	I	0 : 2
0 : 1	0 : –	I	0 : –
0 : 1 1 : 2	I	0 : – 1 : 0	I
I	I	0 : 1	0 : –



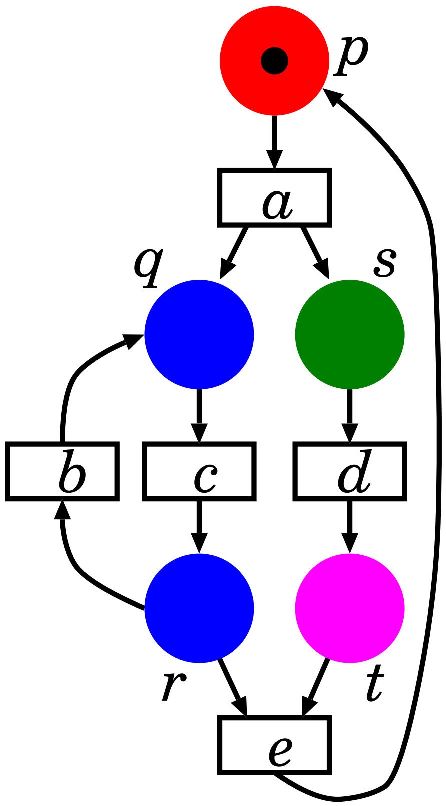
$$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$$

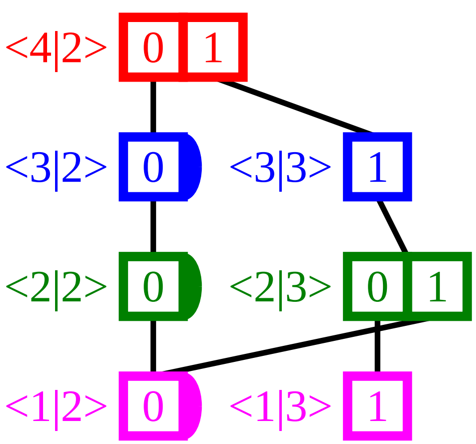
$$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, t^1\} \equiv \{\underline{0}, 1\}$$

Saturation-OTF in action: saturate $\langle 2|3 \rangle$ (fire d)



a	bc	d	e
0 : 1	**I**	**I**	0 : 2
0 : 1	0 : –	**I**	0 : –
0 : 1 1 : 2	**I**	0 : – 1 : 0	**I**
I	**I**	0 : 1	0 : –



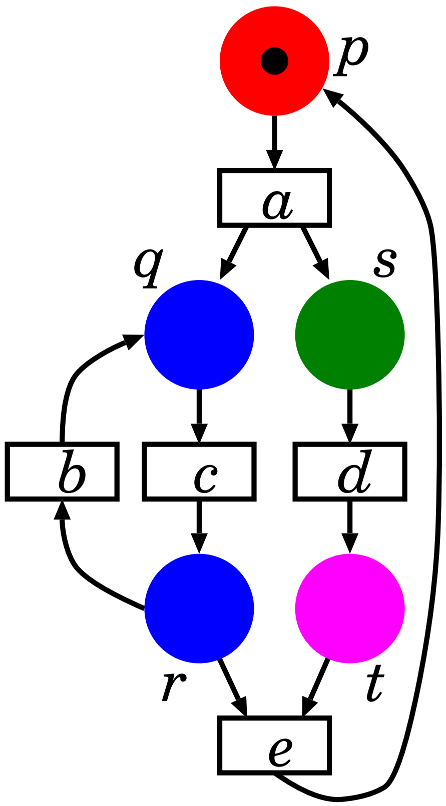
$$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$$

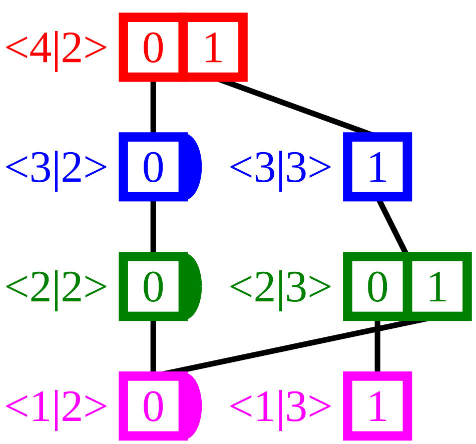
$$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, t^1\} \equiv \{\underline{0}, 1\}$$

Saturation-OTF in action: confirm $t^1 \equiv 1$



a	bc	d	e
$0:1$	**I**	**I**	$0:2$
$0:1$	$0:-$	**I**	$0:-$
$0:1$ $1:2$	**I**	$0:-$ $1:0$	**I**
I	**I**	$0:1$ $1:2$	$0:-$ $1:0$



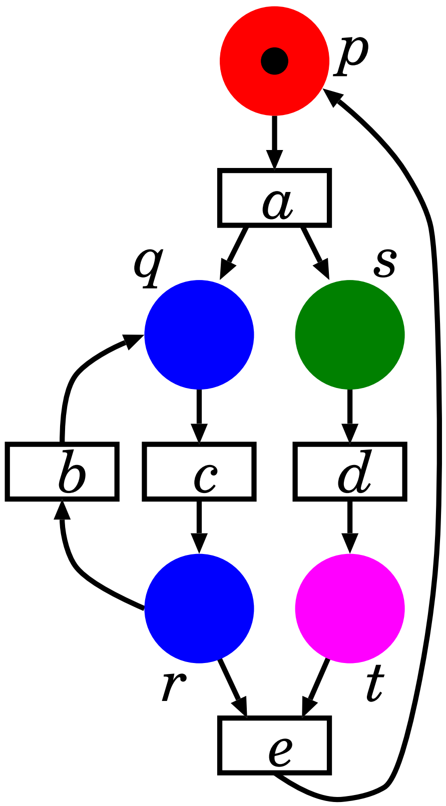
$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$

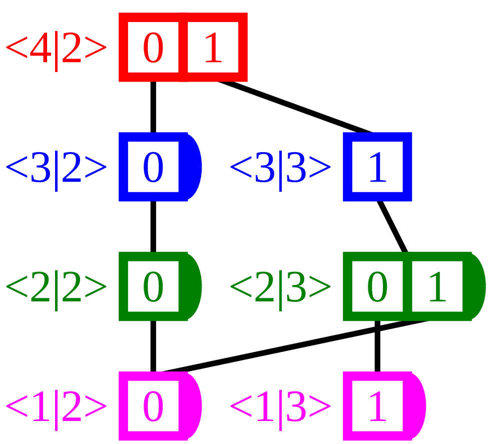
$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

Saturation-OTF in action: saturate $\langle 1|3 \rangle$ (no firing)



a	bc	d	e
$0:1$	$\mathbf{I}$	$\mathbf{I}$	$0:2$
$0:1$	$0:-$	$\mathbf{I}$	$0:-$
$0:1$ $1:2$	$\mathbf{I}$	$0:-$ $1:0$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$0:1$ $1:2$	$0:-$ $1:0$



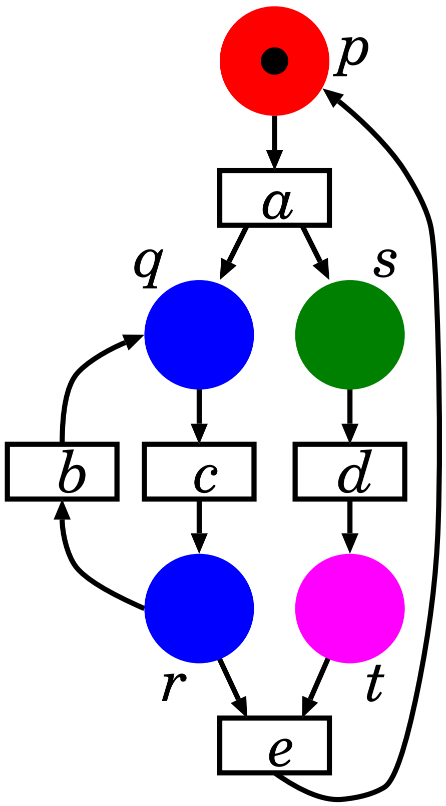
$$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, q^1 r^0\} \equiv \{\underline{0}, 1\}$$

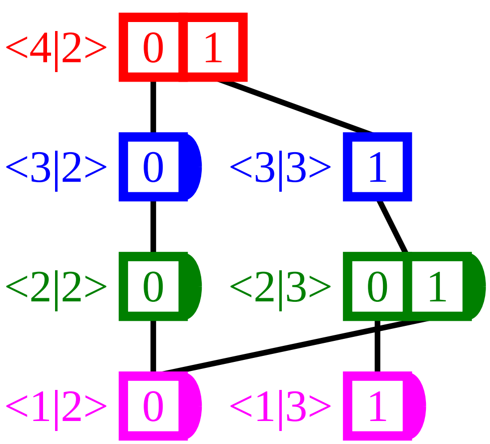
$$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

Saturation-OTF in action: confirm $q^1 r^0 \equiv 1$



a	bc	d	e
$0:1$	**I**	**I**	$0:2$
$0:1$ $1:2$	$0:-$ $1:3$	**I**	$0:-$ $1:-$
$0:1$ $1:2$	**I**	$0:-$ $1:0$	**I**
I	**I**	$0:1$ $1:2$	$0:-$ $1:0$



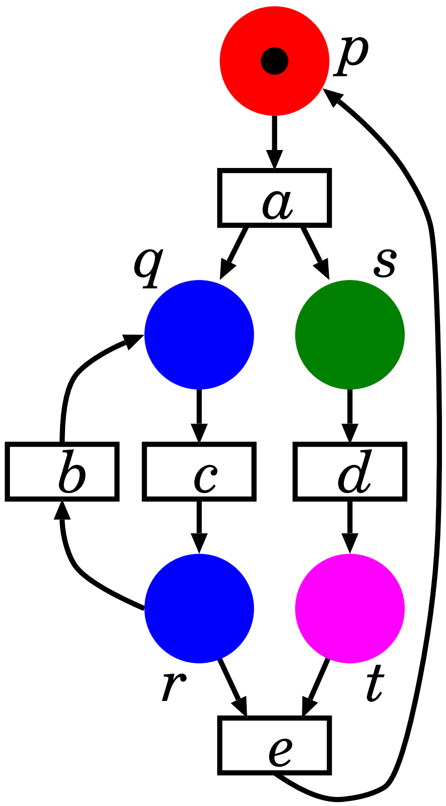
$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, q^2 r^0, q^0 r^1\} \equiv \{\underline{0}, \underline{1}, 2, 3\}$

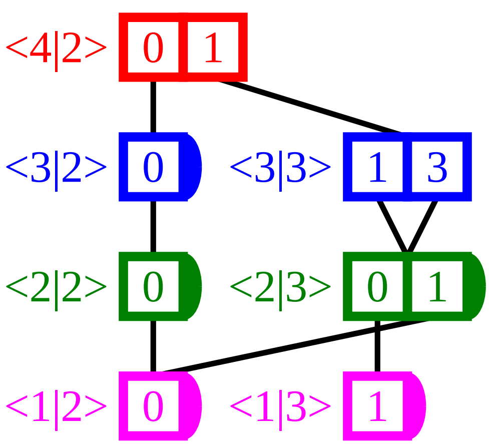
$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

Saturation-OTF in action: saturate $\langle 3|3 \rangle$ (fire bc)



	a	bc	d	e
	0 : 1	**I**	**I**	0 : 2
	0 : 1 1 : 2	0 : – 1 : 3	**I**	0 : – 1 : –
	0 : 1 1 : 2	**I**	0 : – 1 : 0	**I**
	I	**I**	0 : 1 1 : 2	0 : – 1 : 0



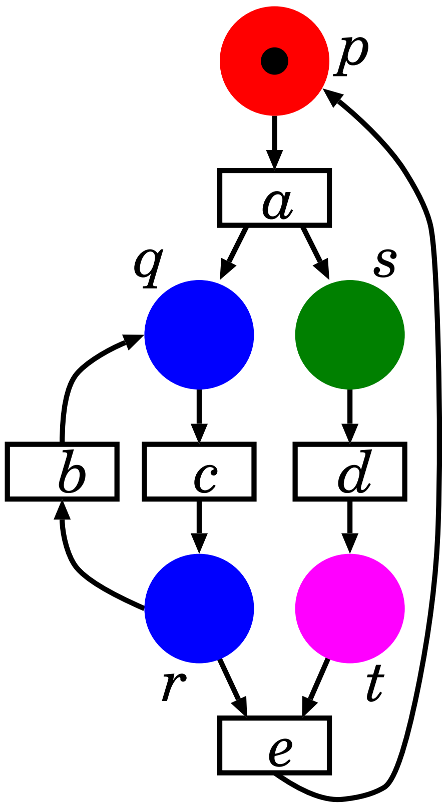
$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, q^2 r^0, q^0 r^1\} \equiv \{\underline{0}, \underline{1}, 2, 3\}$

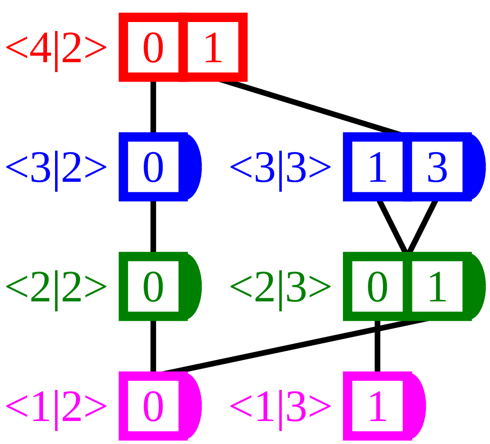
$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

Saturation-OTF in action: confirm $q^0 r^1 \equiv 3$



a	bc	d	e
0 : 1	I	I	0 : 2
0 : 1 1 : 2 3 : 4	0 : – 1 : 3 3 : 1	I	0 : – 1 : – 3 : 0
0 : 1 1 : 2	I	0 : – 1 : 0	I
I	I	0 : 1 1 : 2	0 : – 1 : 0



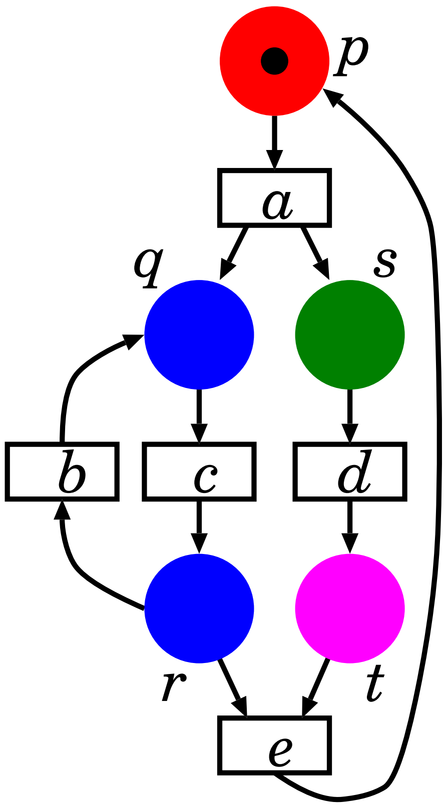
$\mathcal{X}_4 = \{\underline{p^1}, p^0, p^2\} \equiv \{\underline{0}, 1, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, q^2 r^0, \underline{q^0 r^1}, q^1 r^1\} \equiv \{\underline{0}, \underline{1}, 2, \underline{3}, 4\}$

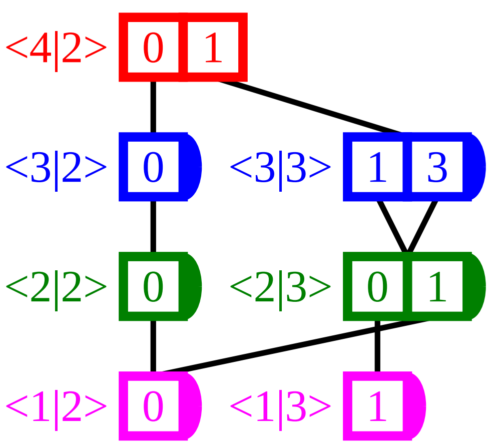
$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

Saturation-OTF in action: confirm $p^0 \equiv 1$



a	bc	d	e
$0:1$ $1:-$	**I**	**I**	$0:2$ $1:0$
$0:1$ $1:2$ $3:4$	$0:-$ $1:3$ $3:1$	**I**	$0:-$ $1:-$ $3:0$
$0:1$ $1:2$	**I**	$0:-$ $1:0$	**I**
I	**I**	$0:1$ $1:2$	$0:-$ $1:0$



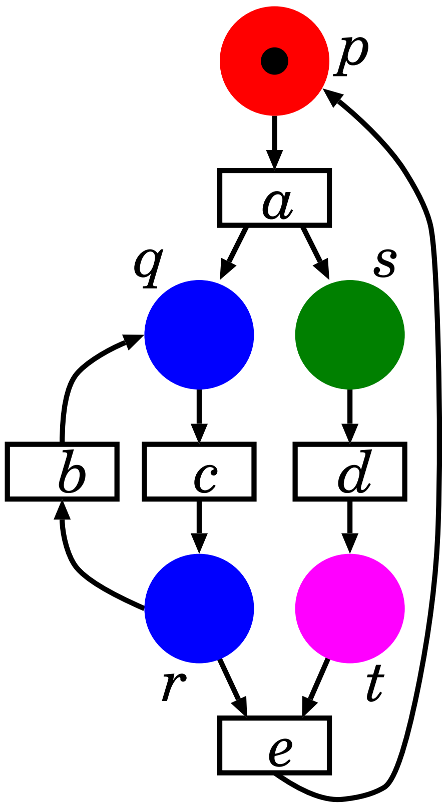
$$\mathcal{X}_4 = \{\underline{p^1}, \underline{p^0}, p^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, q^2 r^0, \underline{q^0 r^1}, q^1 r^1\} \equiv \{\underline{0}, \underline{1}, 2, \underline{3}, 4\}$$

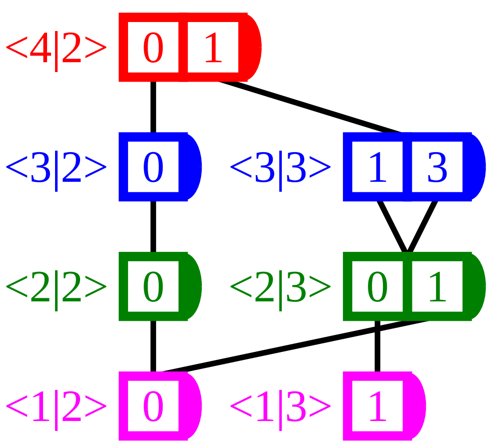
$$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$$

Saturation-OTF in action: saturate $\langle 4|2 \rangle$ (fire e)



	a	bc	d	e
	$0:1$ $1:-$	**I**	**I**	$0:2$ $1:0$
	$0:1$ $1:2$ $3:4$	$0:-$ $1:3$ $3:1$	**I**	$0:-$ $1:-$ $3:0$
	$0:1$ $1:2$	**I**	$0:-$ $1:0$	**I**
	I	**I**	$0:1$ $1:2$	$0:-$ $1:0$



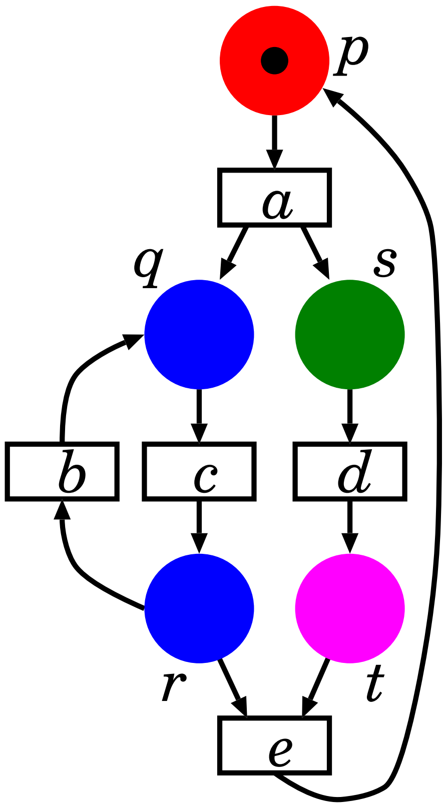
$\mathcal{X}_4 = \{\underline{p^1}, \underline{p^0}, p^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, q^2 r^0, \underline{q^0 r^1}, q^1 r^1\} \equiv \{\underline{0}, \underline{1}, 2, \underline{3}, 4\}$

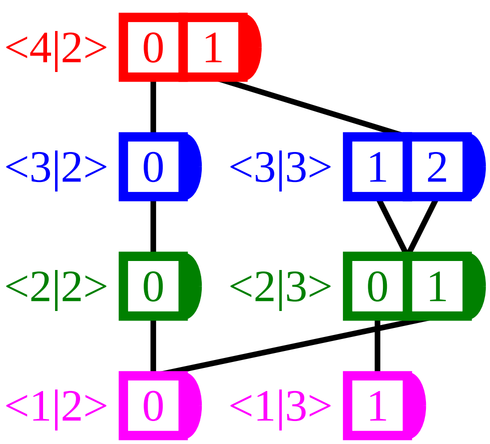
$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}, s^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}, t^2\} \equiv \{\underline{0}, \underline{1}, 2\}$

Saturation-OTF in action: remap confirmed indices



a	bc	d	e
0 : 1 1 : –	I	I	0 : – 1 : 0
0 : 1 1 : – 2 : –	0 : – 1 : 2 2 : 1	I	0 : – 1 : – 2 : 0
0 : 1 1 : –	I	0 : – 1 : 0	I
I	I	0 : 1 1 : –	0 : – 1 : 0



$$\mathcal{X}_4 = \{\underline{p^1}, \underline{p^0}\} \equiv \{\underline{0}, \underline{1}\}$$

$$\mathcal{X}_3 = \{\underline{q^0 r^0}, \underline{q^1 r^0}, \underline{q^0 r^1}\} \equiv \{\underline{0}, \underline{1}, \underline{2}\}$$

$$\mathcal{X}_2 = \{\underline{s^0}, \underline{s^1}\} \equiv \{\underline{0}, \underline{1}\}$$

$$\mathcal{X}_1 = \{\underline{t^0}, \underline{t^1}\} \equiv \{\underline{0}, \underline{1}\}$$

Results

N	Reachable states	Final memory (KB)			Peak memory (KB)			Time (sec)		
		OTF	PRE	NuSMV	OTF	PRE	NuSMV	OTF	PRE	NuSMV
Dining Philosophers: $L=N/2$, $\|\mathcal{X}_k\|=34$ for all k										
20	3.46×10^{12}	4	3	4,178	5	4	4,192	0.01	0.01	0.4
50	2.23×10^{31}	11	10	8,847	14	12	8,863	0.03	0.02	13.1
100	4.97×10^{62}	24	20	8,891	28	25	15,256	0.06	0.05	990.8
200	2.47×10^{125}	**48**	**40**	**21,618**	**57**	**50**	**59,423**	**0.15**	**0.11**	**18,129.3**
5,000	6.53×10^{3134}	1,210	1,015	—	1,445	1,269	—	65.55	51.29	—
Slotted Ring Network: $L=N$, $\|\mathcal{X}_k\|=15$ for all k										
5	5.39×10^{4}	1	1	502	5	5	507	0.01	0.01	0.1
10	8.29×10^{9}	5	5	4,332	28	27	8,863	0.06	0.04	6.1
15	1.46×10^{15}	10	9	771	80	77	11,054	0.18	0.13	2,853.1
100	2.60×10^{105}	434	398	—	15,753	14,486	—	41.72	25.78	—
Round Robin Mutual Exclusion: $L=N+1$, $\|\mathcal{X}_k\|=10$ for all k except $\|\mathcal{X}_1\|=N+1$										
10	2.30×10^{4}	5	5	917	6	7	932	0.01	0.01	0.2
20	4.72×10^{7}	18	17	5,980	20	21	5,985	0.04	0.03	1.4
30	7.25×10^{10}	37	36	2,222	41	41	8,716	0.09	0.07	5.6
100	2.85×10^{32}	357	355	13,789	372	372	21,814	2.11	1.55	2,836.5
150	4.82×10^{47}	784	781	—	807	807	—	7.04	5.07	—
FMS: $L=19$, $\|\mathcal{X}_k\|=N+1$ for all k except $\|\mathcal{X}_{17}\|=4$, $\|\mathcal{X}_{12}\|=3$, $\|\mathcal{X}_7\|=2$										
5	1.92×10^{4}	5	6	2,113	6	9	2,126	0.01	0.01	1.0
10	2.50×10^{9}	16	19	1,152	26	31	8,928	0.02	0.02	41.6
25	8.54×10^{13}	**86**	**135**	**17,045**	**163**	**239**	**152,253**	**0.16**	**0.11**	**17,321.9**
150	4.84×10^{23}	6,291	15,459	—	16,140	29,998	—	18.50	10.92	—

Locality

Given an event $\alpha \in \mathcal{E}$, consider the subset of the state variables $\{x_L, ..., x_1\}$ that:

- can be modified by α: $\mathcal{V}_M(t) = \{x_k : \exists \mathbf{i}, \mathbf{i}' \in \widehat{\mathcal{X}}, \mathbf{i}' \in \mathcal{N}_t(\mathbf{i}) \wedge \mathbf{i}[k] \neq \mathbf{i}'[k]\}$
- can disable α: $\mathcal{V}_D(t) = \{x_k : \exists \mathbf{i}, \mathbf{j} \in \widehat{\mathcal{X}}, \forall h \neq k, \mathbf{i}[h] = \mathbf{j}[h] \wedge \mathcal{N}_t(\mathbf{i}) \neq \emptyset \wedge \mathcal{N}_t(\mathbf{j}) = \emptyset\}$

If $x_k \notin \mathcal{V}_M \cup \mathcal{V}_D$, we say that event α and variable x_k, or level k, are independent

Most events in a globally-asynchronous locally-synchronous model are highly localized:

- Let $Top(\alpha) = \max\{k : x_k \in \mathcal{V}_M(\alpha) \cup \mathcal{V}_D(\alpha)\}$ be the highest level dependent on α
- Let $Bot(\alpha) = \min\{k : x_k \in \mathcal{V}_M(\alpha) \cup \mathcal{V}_D(\alpha)\}$ be the lowest level dependent on α
- The span (of levels) $\{Top(\alpha), ..., Bot(\alpha)\}$ for event α is often much smaller than $\{L, ..., 1\}$

fully-reduced $2L$-level MDD encoding of $\mathcal{N}$ does not exploit locality

need **Kronecker**, **identity-reduced** $2L$-level MDD, or **MxD** encoding

Locality in non-Kronecker consistent models

Another way to formalize the Kronecker encoding of $\mathcal{N}$:

$\mathcal{N} : \widehat{\mathcal{X}} \times \widehat{\mathcal{X}} \to \{0,1\}$ tells us which transitions between potential states are possible

$\mathcal{N} = \bigvee_{\alpha\in\mathcal{E}} \mathcal{N}_\alpha, \;\; \mathcal{N}_\alpha : \widehat{\mathcal{X}} \times \widehat{\mathcal{X}} \to \{0,1\}$ asynchronous, disjunctive, decomposition of $\mathcal{N}$

$\mathcal{N}_\alpha = \left(\bigwedge_{k\in\mathcal{D}_\alpha} \mathcal{N}_{k,\alpha}\right) \wedge \left(\bigwedge_{l\in\overline{\mathcal{D}_\alpha}} \mathcal{I}_l\right)$ synchronous, conjunctive, decomposition by level of $\mathcal{N}_\alpha$

$\mathcal{D}_\alpha \subseteq \{L, ..., 1\}$	set of levels on which event α depends
$\mathcal{I}_k$	identity transformation for the states of submodel k
$\mathcal{N}_{k,\alpha} : \mathcal{X}_k \times \mathcal{X}_k \to \{0,1\}$	next-state function restricted to level k only

A generalization of the Kronecker encoding:

$\mathcal{N}_\alpha = \left(\bigwedge_{c=1}^{m_\alpha} \mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha}\right) \wedge \left(\bigwedge_{k\in\overline{\mathcal{D}_\alpha}} \mathcal{I}_k\right)$ general synchronous, conjunctive, decomposition of $\mathcal{N}_\alpha$

$\bigcup_{c=1}^{m_\alpha} \mathcal{D}_{c,\alpha} = \mathcal{D}_\alpha \subseteq \{L, ..., 1\}$	$\mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha}$ depends on a set of levels
$\mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha} : \left(\times_{k\in\mathcal{D}_{c,\alpha}} \mathcal{X}_k\right) \times \left(\times_{k\in\mathcal{D}_{c,\alpha}} \mathcal{X}_k\right) \to \{0,1\}$	

$$Top(\alpha) = \max \mathcal{D}_\alpha \qquad Bot(\alpha) = \min \mathcal{D}_\alpha$$

Saturation without Kronecker consistency

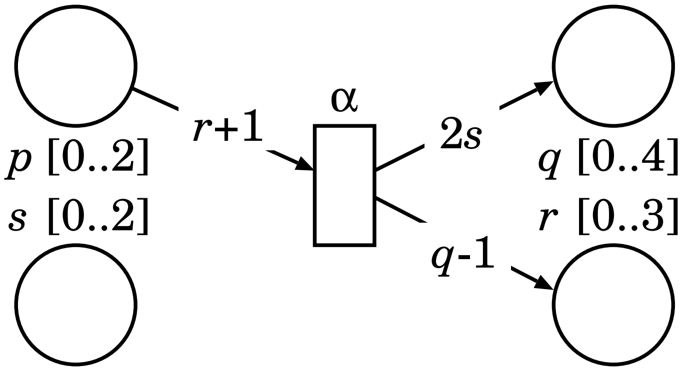
We define $\mathcal{N}_k = \bigvee_{\alpha:Top(\alpha)=k} \mathcal{N}_\alpha$ and use a general Saturation algorithm:

1. starting at level $k = 1$ and moving towards the root at level L...
2. saturate each node p at level k, by applying $\mathcal{N}_k$ to the node until it does not change...
3. any node q at level $h < k$ created in the process is saturated immediately by applying $\mathcal{N}_h$ to it...

Challenges:

- how do we define the domains $\mathcal{D}_{c,\alpha}$ to be "as small as possible"?
- how do we store $\mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha}$, $\mathcal{N}_\alpha$, and $\mathcal{N}_k$?
- how do we capture and exploit identity transformations?
- how do we do all of this "on-the-fly" (without knowing the local state spaces $\mathcal{X}_k$ a priori)?

Using MDDs for $\mathcal{N}$



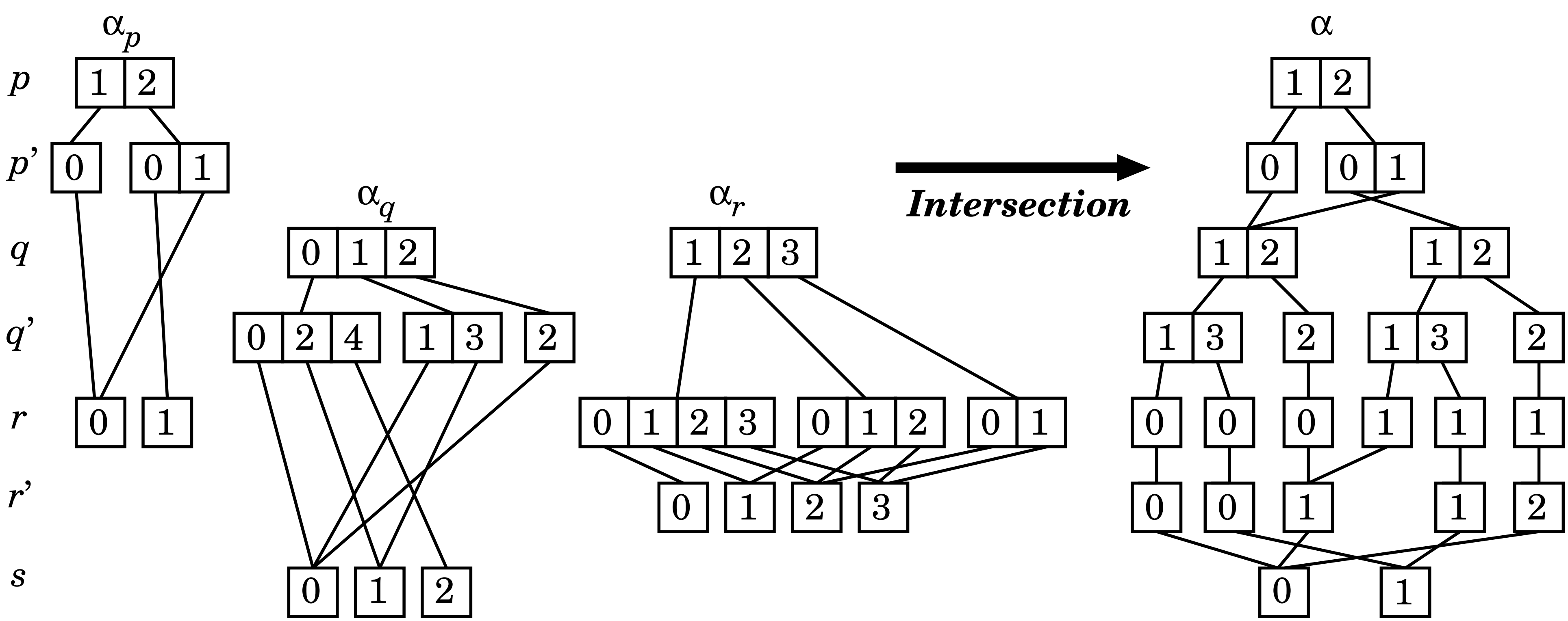
Equivalent pseudocode (simultaneous statements)

if $p > r \wedge q + 2s \leq 4 \wedge q > 0 \wedge r + q \leq 4$ then

$p \leftarrow p - (r + 1);$

$q \leftarrow q + 2s;$

$r \leftarrow r + q - 1;$



special reduction rule and interpretation of skipped levels

Building the MDDs encoding $\mathcal{N}$ on-the-fly

To confirm a new local state $\mathbf{i}_h \in \mathcal{X}_h$:

for $k = L$ down to h do
  for each α such that $Top(\alpha) = k$ and $h \in \mathcal{D}_\alpha$ do
    for each $\mathcal{D}_{c,\alpha}$ containing h do
      explicitly build the set $\mathcal{Y}$ of potential transitions from $\{\mathbf{i}_h\} \times \left(\times_{k \in \mathcal{D}_{c,\alpha} \setminus \{h\}} \mathcal{X}_k\right)$;
      $\mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha} \leftarrow \mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha} \cup \mathcal{Y}$;
    $\mathcal{N}_\alpha \leftarrow \left(\bigwedge_{c=1}^{m_\alpha} \mathcal{N}_{\mathcal{D}_{c,\alpha},\alpha}\right) \wedge \left(\bigwedge_{l \in \overline{\mathcal{D}_\alpha}} \mathcal{I}_l\right)$;
    $\mathcal{N}_k \leftarrow \mathcal{N}_k \cup \mathcal{N}_\alpha$;

the explicit enumeration of the elements of $\times_{h \in \mathcal{D}_{c,\alpha}} \mathcal{X}_h$
is the reason for seeking the smallest possible $\mathcal{D}_{c,\alpha}$

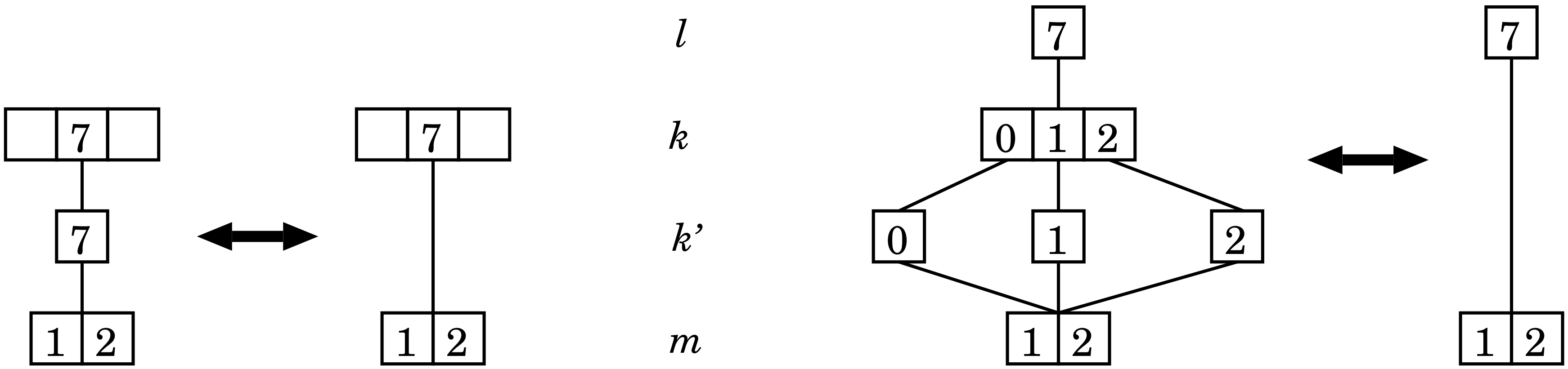
New reduction rule for MDDs

In addition to the

- quasi–reduced form
- reduced form

we need an

- identity–reduced form (for "to" levels only!)



Identity–reduced level k'

Identity–reduced level k' and fully–reduced level k

canonical even if we use different reduction rules for each level

Saturation pseudocode: $Saturate(L, \mathcal{X}_{init})$

mdd $Saturate$(level k, mdd p) is *quasi-reduced version*

local mdd $r, r_0, ..., r_{n_k-1}$;
if $k = 0$ then return p;
if $Cache$ contains entry $\langle SaturateCODE, p : r \rangle$ then return r;
foreach $i_k \in \mathcal{X}_k$ do
  $r_{i_k} \leftarrow Saturate(k-1, p[i_k])$; *first, be sure that the children are saturated*
repeat
  choose $\alpha \in \mathcal{E}$, $i_k, j_k \in \mathcal{X}_k$ s.t. $Top(\alpha) = k$ and $r_{i_k} \neq \mathbf{0}$ and $\mathcal{N}_\alpha[i_k][j_k] \neq \mathbf{0}$;
  $r_{j_k} \leftarrow Union(r_{j_k}, RelProdSat(k-1, r_{i_k}, \mathcal{N}_\alpha[i_k][j_k]))$;
until $r_0, ..., r_{n_k-1}$ do not change;
$r \leftarrow UniqueTableInsert(k, r_0, ..., r_{n_k-1})$;
enter $\langle SaturateCODE, p : r \rangle$ in $Cache$;
return r;

mdd $RelProdSat$(level k, mdd q, mdd2 f) is

local mdd $r, r_0, ..., r_{n_k-1}$;
if $k = 0$ then return $q \wedge f$;
if $Cache$ contains entry $\langle RelProdSatCODE, q, f : r \rangle$ then return r;
foreach $i_k, j_k \in \mathcal{X}_k$ s.t. $q[i_k] \neq \mathbf{0}$ and $f[i_k][j_k] \neq \mathbf{0}$ do
  $r_{j_k} \leftarrow Union(r_{j_k}, RelProdSat(k-1, q[i_k], f[i_k][j_k]))$;
$r \leftarrow Saturate(k, UniqueTableInsert(k, r_0, ..., r_{n_k-1}))$;
enter $\langle RelProdSatCODE, q, f : r \rangle$ in $Cache$;
return r.

Variable partitioning/ordering based on structural information

Variable ordering can have enormous effects on the size of the decision diagram

CUDD, used in NuSMV, attempts dynamic variable reordering when there are too many nodes

In the non-binary case, both partitioning and ordering affect the size

We are interested in heuristics to choose a good static variable partitioning and ordering

For variable ordering:

- obvious idea: minimize the sum of the "spans of events"
- An NP-complete problem in itself!
- 5×5 knights swapping game: a **speedup of 3** over our best "manual" ordering
- Asynchronous circuit verification problem from UPC (Barcelona): a **speedup of 40**

For variable partitioning:

- Use the finest Kronecker-consistent partition, which might require to group multiple variables...
- ...or use non-Kronecker algorithms and one variable per level
- The finest partition is often good, but not necessarily optimal
- Model invariants may suggest good coarsening

Variable ordering and variable partitioning are tightly coupled problems

Structural symbolic CTL model checking

Traditional symbolic CTL model checking (EF, EU, EG) uses a breadth-first fixed-point iteration

Just like for state-space generation, breadth-first can require huge peak memory, hence runtime

Using the model structure results in better algorithms for symbolic CTL model checking

- exploit locality through a Kronecker-based encoding of the next-state function
- employ a Saturation-based algorithm for EF (easy) and for EU (classify safe vs. unsafe events)
- greatly reduced memory and time requirements for asynchronous systems
- implemented in our tool SMART
- ○ can we apply Saturation to EG?
- ○ can we extend this to fair CTL?

Substantial time and memory improvements for EX and EG

Enormous time and memory improvements for EF and EU

Model checking results: SMART vs. NuSMV

$\|\mathcal{S}\|$ (depends on parameter N)	NuSMV				SMART						NuSMV				SMART	
	after SS		alone		$EUtrad$			$EUsat$			after SS		alone		$EGtrad$	
	sec	kB	sec	kB	iter	sec	kB	iter	sec	kB	sec	kB	sec	kB	sec	kB
Phils					E$[(phil_1 \neq eat)$ U $(phil_0 = eat)]$						EG$(phil_0 \neq eat)$ starvation					
2.23×10^{31}	1.2	46	39.7	46	100	0.17	1	4	0.06	1	0.9	46	132.3	50	0.02	1
4.96×10^{62}	7.9	316	1121.8	316	200	0.67	3	4	0.14	3	9.0	316	2525.3	358	0.05	3
3.03×10^{313}	—	—	—	—	1000	19.09	78	4	0.77	60	—	—	—	—	0.28	58
FMS					E$[(M_1 > 0)$ U $(P_1s = P_2s = P_3s = N)]$						EG$\neg(P_1s = P_2s = P_3s = N)$					
3.44×10^{3}	0.2	17	318.1	43	31	0.04	$<.5$	6	0.01	$<.5$	0.2	17	128.9	18	$<.005$	$<.5$
4.86×10^{4}	1.0	127	—	—	46	0.16	$<.5$	8	0.02	$<.5$	1.0	127	—	—	0.01	$<.5$
8.54×10^{13}	—	—	—	—	376	—	—	52	1010.85	293	—	—	—	—	50.38	251
Round robin					E$[(p_1 \neq load)$ U $(p_0 = send)]$						EG$(true)$ find all cycles					
2.30×10^{5}	0.2	11	85.0	11	39	0.01	$<.5$	11	0.01	$<.5$	0.3	11	78.5	13	$<.005$	$<.5$
1.10×10^{6}	0.6	40	4922.7	40	59	0.03	$<.5$	16	0.01	$<.5$	1.2	40	4739.5	44	0.01	$<.5$
2.85×10^{32}	—	—	—	—	399	13.32	32	101	4.67	19	—	—	—	—	1.29	20
Leader					E$[(pref_1 = 0)$ U $(status_0 = leader)]$						EG$(status_0 \neq leader)$					
1.15×10^{4}	2.3	11	8104.7	371	62	0.36	1	38	0.27	1	232.8	12	1189.1	235	0.11	2
1.50×10^{5}	52.0	33	—	—	81	3.74	7	52	3.09	7	18023.6	104	—	—	0.44	9
2.39×10^{7}	—	—	—	—	121	690.85	116	85	416.85	101	—	—	—	—	7.15	128
Slotted ring					E$[(slot_1 \neq bf)$ U $(slot_0 = ag)]$						EG$(slot_0 \neq hg)$					
8.29×10^{9}	0.2	10	0.4	3	63	0.01	$<.5$	9	0.01	$<.5$	0.6	10	0.1	1	0.01	$<.5$
1.46×10^{15}	1.8	15	2.0	10	93	0.37	1	9	0.02	$<.5$	4.7	15	0.2	2	0.01	$<.5$
3.03×10^{105}	—	—	—	—	603	—	—	9	1.60	62	—	—	—	—	0.62	62

Saturation history

MinCia ATPN'99 *Efficient reachability set generation and storage using decision diagrams*
Kronecker encoding, locality, exhaustive chaining for local events

CiaLueSim ATPN'00 *Efficient symbolic state-space construction for asynchronous systems*
In-place updates, exhaustive chaining for all events

CiaLueSim TACAS'01 *Saturation: An efficient iteration strategy for symbolic state space generation*
Saturation (Kronecker-consistent version)

CiaSim FMCAD'02 *Using edge-valued decision diagrams for symbolic generation of shortest paths*
Saturation (Kronecker-consistent version) with EV^+MDDs and ADDs

CiaMarSim TACAS'03 *Saturation unbound*
Saturation (Kronecker-consistent version) with a-priori unbounded local domains

CiaSim CAV'03 *Structural symbolic CTL model checking of asynchronous systems*
Saturation (Kronecker-consistent version) for the computation of the EF and EU operators

ChuCia QEST'04 *Saturation NOW*
Distributed Saturation (Kronecker-consistent version) for state-space generation: uses overall memory, no speedup

CiaYu CHARME'05 *Saturation-based symbolic reachability analysis using conjunctive and disjunctive partitioning*
General Saturation (arbitrary asynchronous models with a-priori unbounded local domains)

ChuCiaYu ATVA'06 *A fine-grained density-guided chaining heuristic for symbolic reachability analysis*
Speeding up General Saturation with a dynamic heuristic based on the decision diagram nodes

ChuCia IPDPS'06 *A dynamic firing speculation to speedup distributed symbolic state-space generation*
Distributed Saturation (Kronecker-consistent version) for state-space generation: speedup through speculation

YuCiaLue TACAS'07 *Bounded reachability checking of asynchronous systems using decision diagrams*
Bounded General Saturation using EV^+MDDs

WanCia SOFSEM'09 *Extensible decision diagrams for symbolic state-space generation of asynchronous systems*
A new variant of MDDs to speed-up general Saturation for state-space generation