Exploiting model structure to encode transition relations and transition rate matrices

Gianfranco Ciardo

Department of Computer Science and Engineering

University of California, Riverside

Supported in part by the National Science Foundation

under Grants CSR-0546041, CCF-0848463, and CCF-1018057

Background

Structured discrete-state models

A structured discrete-state model is specified by:

- a potential state space $\mathcal{X}_{pot} = \mathcal{X}_L \times \cdots \times \mathcal{X}_1$
  - a (global) state is of the form $\mathbf{i} = (i_L, ..., i_1)$
  - $\mathcal{X}_k$ is the (discrete) local state space for submodel k, or local domain for state variable x_k
  - if $\mathcal{X}_k$ is finite, we can map it to $\{0, 1, \ldots, n_k - 1\}$ — n_k might be unknown a priori

- a set of initial states $\mathcal{X}_{init} \subseteq \mathcal{X}_{pot}$
  - often there is a single initial state, $\mathcal{X}_{init} = \{\mathbf{i}_{init}\}$

- a set of events $\mathcal{E}$ defining a disjunctively-partitioned next-state function or transition relation
  - $\mathcal{N}_e : \mathcal{X}_{pot} \to 2^{\mathcal{X}_{pot}}$ — $\mathbf{j} \in \mathcal{N}_e(\mathbf{i})$ iff state $\mathbf{j}$ can be reached by firing event e in state $\mathbf{i}$
  - $\mathcal{N} : \mathcal{X}_{pot} \to 2^{\mathcal{X}_{pot}}$ — $\mathcal{N}(\mathbf{i}) = \bigcup_{e \in \mathcal{E}} \mathcal{N}_e(\mathbf{i})$
  - naturally extended to sets of states $\mathcal{N}_e(\mathcal{Y}) = \bigcup_{\mathbf{i} \in \mathcal{Y}} \mathcal{N}_e(\mathbf{i})$ and $\mathcal{N}(\mathcal{Y}) = \bigcup_{\mathbf{i} \in \mathcal{Y}} \mathcal{N}(\mathbf{i})$
  - e is enabled in $\mathbf{i}$ iff $\mathcal{N}_e(\mathbf{i}) \neq \emptyset$, otherwise it is disabled
  - $\mathbf{i}$ is absorbing, or dead, or terminal, or a sink if $\mathcal{N}(\mathbf{i}) = \emptyset$

The (reachable) state space of a discrete-state model

The state space $\mathcal{X}_{rch}$ of the model is the smallest set $\mathcal{X} \subseteq \mathcal{X}_{pot}$ containing $\mathcal{X}_{init}$ and satisfying:

- the recursive definition $\mathbf{i} \in \mathcal{X} \wedge \mathbf{j} \in \mathcal{N}(\mathbf{i}) \Rightarrow \mathbf{j} \in \mathcal{X}$ (base for explicit methods)

or

- the fixpoint equation $\mathcal{X} = \mathcal{X} \cup \mathcal{N}(\mathcal{X})$ (base for symbolic methods)

$$\mathcal{X}_{rch} = \mathcal{X}_{init} \cup \mathcal{N}(\mathcal{X}_{init}) \cup \mathcal{N}^2(\mathcal{X}_{init}) \cup \mathcal{N}^3(\mathcal{X}_{init}) \cup \cdots = \mathcal{N}^*(\mathcal{X}_{init})$$

Definition of Kronecker product (assuming square matrices)

Given L square 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)

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

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

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 can express contemporaneity or synchronization

Definition of Kronecker sum

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

$$\bigoplus_{L \geq k \geq 1} \mathbf{A}_k = \sum_{L \geq k \geq 1} \mathbf{I}_{n_L \cdots n_{k+1}} \otimes \mathbf{A}_k \otimes \mathbf{I}_{n_k - 1 \cdots n_1} \in \mathbb{R}^{n_L \cdots n_1 \times n_L \cdots n_1}$$

where $\mathbf{I}_m$ is the identity matrix of size $m \times m$ and

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

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

nonzeros: $$\eta\left(\bigoplus_{L \geq k \geq 1} \mathbf{A}_k\right) \leq \sum_{L \geq k \geq 1} \frac{\eta(\mathbf{A}_k)}{n_k} \cdot \prod_{L \geq h \geq 1} n_h$$

Given the matrices $\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} \\ b_{1,0} & b_{1,1} & b_{1,2} \\ b_{2,0} & b_{2,1} & b_{2,2} \end{bmatrix},$

$$\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_3 + \mathbf{I}_2 \otimes \mathbf{B} =$$

$$\left[\begin{array}{ccc|ccc} a_{0,0} & & & a_{0,1} & & \\ & a_{0,0} & & & a_{0,1} & \\ & & a_{0,0} & & & a_{0,1} \\ \hline a_{1,0} & & & a_{1,1} & & \\ & a_{1,0} & & & a_{1,1} & \\ & & a_{1,0} & & & a_{1,1} \end{array}\right] + \left[\begin{array}{ccc|ccc} b_{0,0} & b_{0,1} & b_{0,2} & & & \\ b_{1,0} & b_{1,1} & b_{1,2} & & & \\ b_{2,0} & b_{2,1} & b_{2,2} & & & \\ \hline & & & b_{0,0} & b_{0,1} & b_{0,2} \\ & & & b_{1,0} & b_{1,1} & b_{1,2} \\ & & & b_{2,0} & b_{2,1} & b_{2,2} \end{array}\right] =$$

$$\left[\begin{array}{ccc|ccc} a_{0,0}{+}b_{0,0} & b_{0,1} & b_{0,2} & a_{0,1} & & \\ b_{1,0} & a_{0,0}{+}b_{1,1} & b_{1,2} & & a_{0,1} & \\ b_{2,0} & b_{2,1} & a_{0,0}{+}b_{2,2} & & & a_{0,1} \\ \hline a_{1,0} & & & a_{1,1}{+}b_{0,0} & b_{0,1} & b_{0,2} \\ & a_{1,0} & & b_{1,0} & a_{1,1}{+}b_{1,1} & b_{1,2} \\ & & a_{1,0} & b_{2,0} & b_{2,1} & a_{1,1}{+}b_{2,2} \end{array}\right]$$

Kronecker sum can express asynchronous behavior

Kronecker encoding of transition matrices

Kronecker description of the next-state function

$\mathcal{N} : \mathcal{X}_{pot} \rightarrow 2^{\mathcal{X}_{pot}}$ can be thought of as a boolean matrix $\mathbf{N} \in \mathbb{B}^{\mathcal{X}_{pot} \times \mathcal{X}_{pot}}$

The model is Kronecker-consistent if we can write $\mathbf{N} = \sum_{\alpha \in \mathcal{E}} \mathbf{N}_\alpha = \sum_{\alpha \in \mathcal{E}} \left(\bigotimes_{L \geq k \geq 1} \mathbf{N}_{k,\alpha} \right)$

where $\bigotimes$ is the Kronecker product operator and all operations are performed in boolean algebra

In other words, $\mathcal{N} = \bigvee_{\alpha \in \mathcal{E}} \mathcal{N}_e$ and each $\mathcal{N}_\alpha = \bigwedge_{L \geq k \geq 1} \mathcal{N}_{k,\alpha}$ where

$\mathcal{N}_{k,\alpha} : \mathcal{X}_k \rightarrow 2^{\mathcal{X}_k}$ is encoded by the boolean matrix $\mathbf{N}_{k,\alpha} \in \mathbb{B}^{|\mathcal{X}_k| \times |\mathcal{X}_k|}$

Locality: Often, the k^{th} local state does not affect and is not affected by event α, $\mathbf{N}_{k,\alpha} = \mathbf{I}$

encode a huge $\mathbf{N}$ using just $L \cdot |\mathcal{E}|$ small matrices $\mathbf{N}_{k,\alpha}$

each $\mathbf{N}_{k,\alpha}$ is a $|\mathcal{X}_k| \times |\mathcal{X}_k|$ boolean matrix, usually very sparse

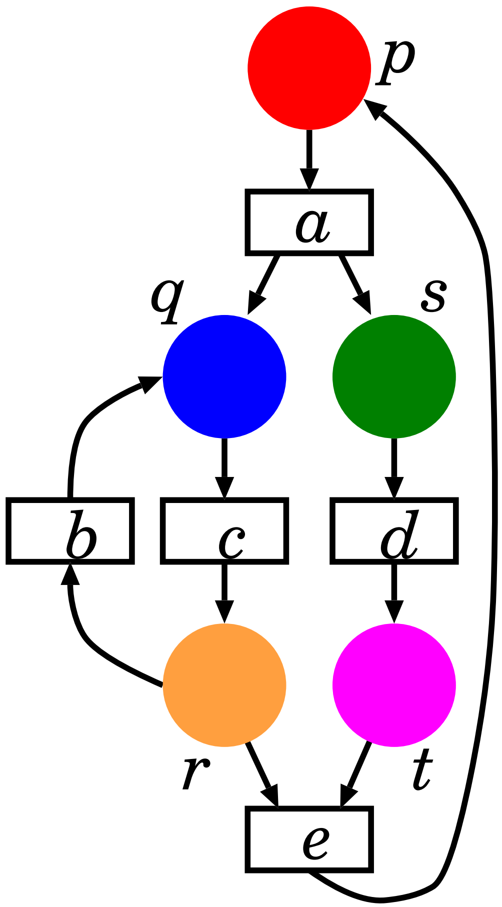
Using structural information to encode $\mathcal{N}$ $(L = 5)$

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

EVENTS $\rightarrow$

LEVELS $\downarrow$

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



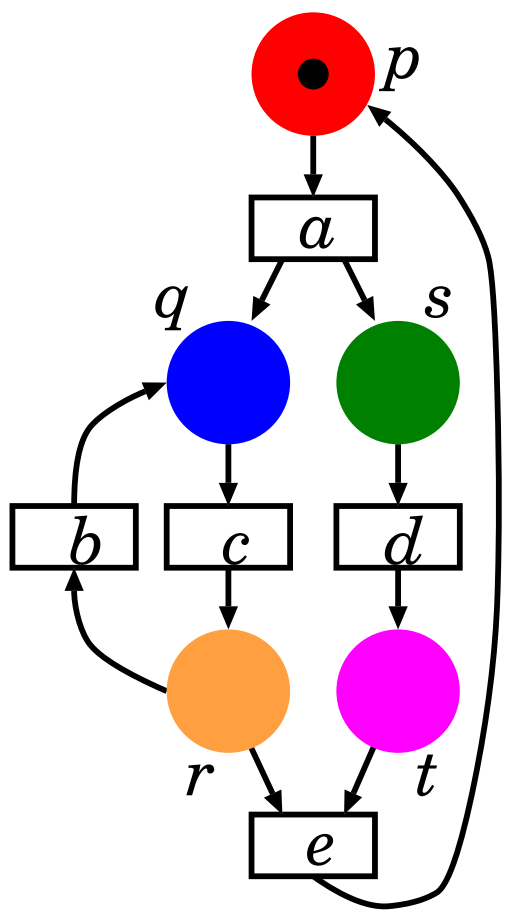
we can determine a priori from the model whether $\mathbf{N}_{k,\alpha} = \mathbf{I}$

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

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

EVENTS $\rightarrow$; LEVELS $\downarrow$

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



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

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

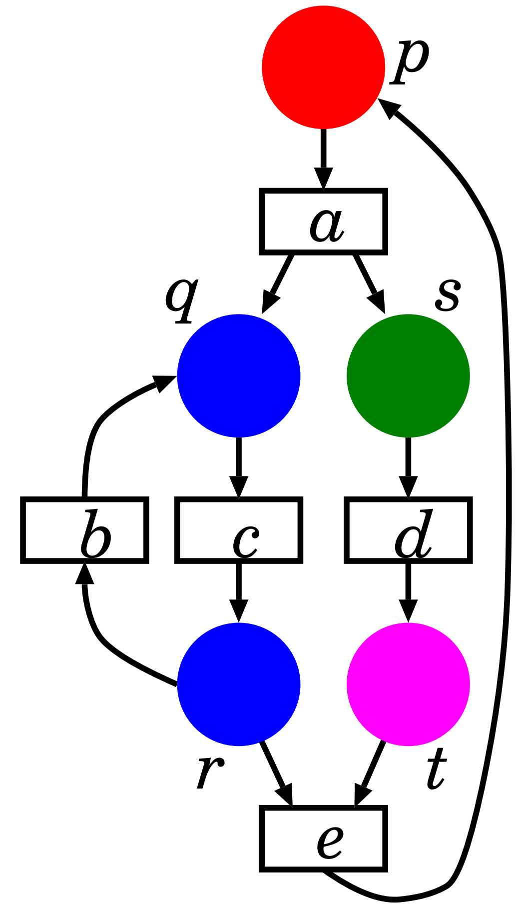
EVENTS $\rightarrow$

LEVELS $\downarrow$

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

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

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



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

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

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

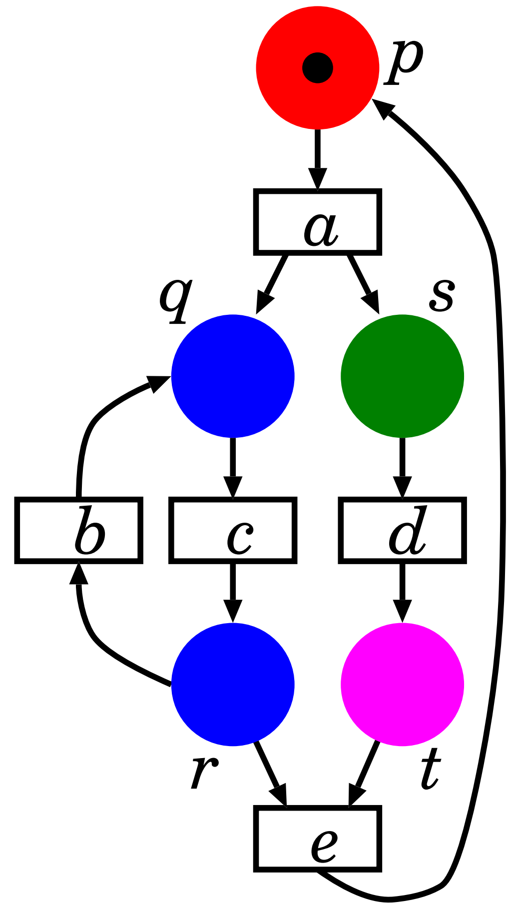
EVENTS →

LEVELS ↓

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

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

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



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

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

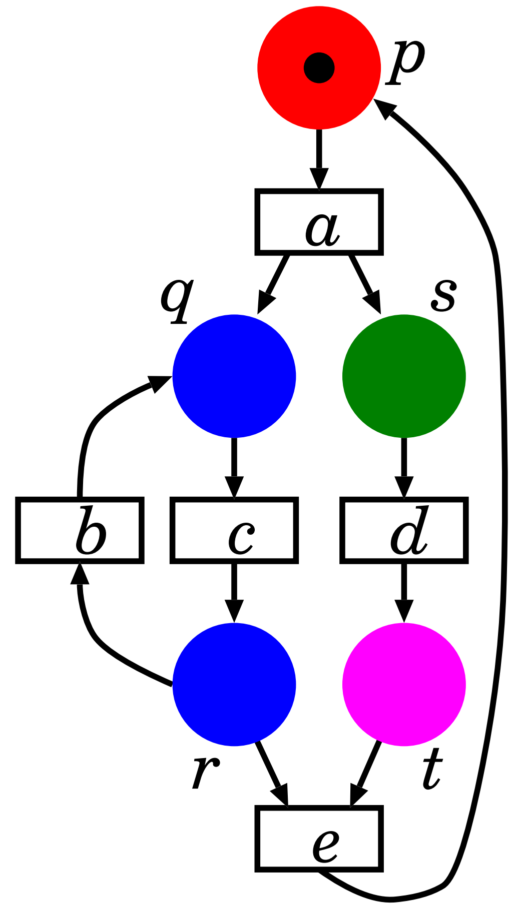
EVENTS →

LEVELS ↓

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

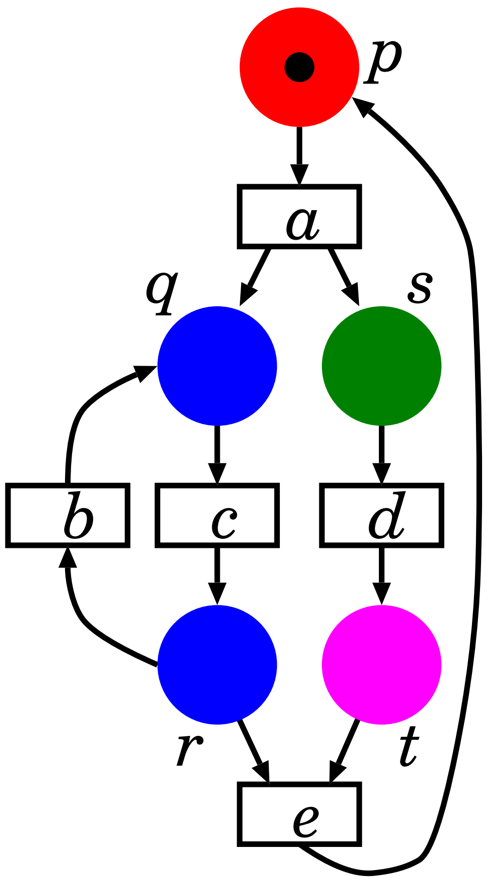
$Top(a) : 4$ $Top(bc) : 3$ $Top(d) : 2$ $Top(e) : 4$

$Bot(a) : 2$ $Bot(bc) : 3$ $Bot(d) : 1$ $Bot(e) : 1$



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

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

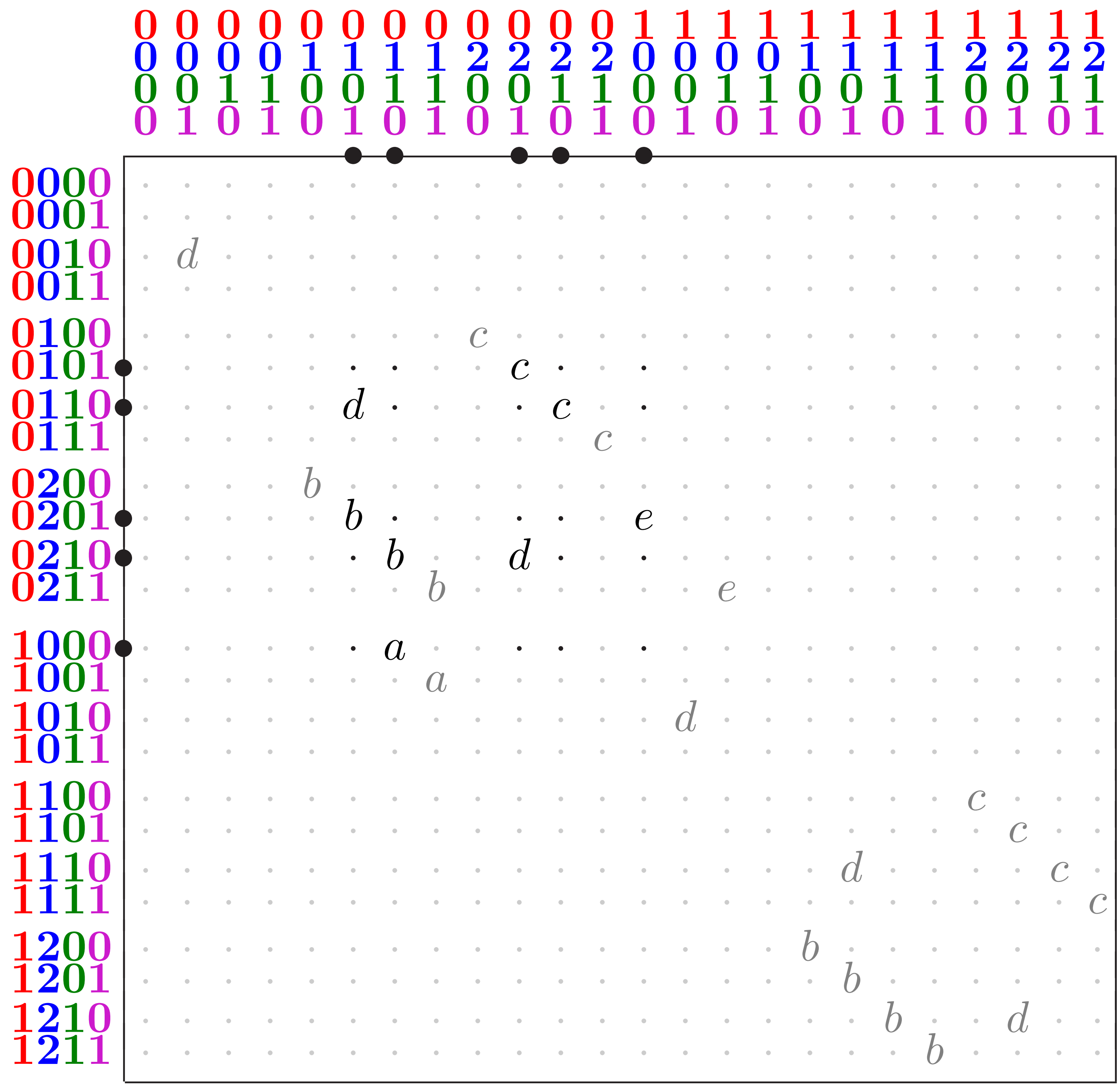


$\{p^0, p^1\} \equiv \{0,1\}$

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

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

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



a, b, c, d, and e indicate the Petri net transition causing $\mathbf{N}[\mathbf{i}, \mathbf{j}] = 1$

Kronecker description of the transition rate matrix of a CTMC

- Parallel composition of L submodels with overall event set $\mathcal{E}$ (synchronizing vs. local)
- Global state $\mathbf{i}$ is a L-tuple $(\mathbf{i}_L, ..., \mathbf{i}_1)$ of local states $\qquad \mathcal{X}_{rch} \subseteq \mathcal{X}_{pot} = \mathcal{X}_L \times \cdots \times \mathcal{X}_1$
- Transition rate matrix $\mathbf{R} = \mathbf{R}_{pot}[\mathcal{X}_{rch}, \mathcal{X}_{rch}]$ where $\mathbf{R}_{pot} = \sum_{\alpha \in \mathcal{E}} \bigotimes_{L \geq k \geq 1} \mathbf{R}_{k,\alpha}$
- $\mathbf{R}_{k,\alpha}[\mathbf{i}_k, \mathbf{j}_k] = \begin{cases} \lambda_{k,\alpha}(\mathbf{i}_k) \cdot \Delta_{k,\alpha}(\mathbf{i}_k, \mathbf{j}_k) & \text{if } \alpha \text{ and submodel } k \text{ are dependent} \\ 1 & \text{if } \alpha \text{ and submodel } k \text{ are independent and } \mathbf{i}_k = \mathbf{j}_k \\ 0 & \text{if } \alpha \text{ and submodel } k \text{ are independent and } \mathbf{i}_k \neq \mathbf{j}_k \end{cases}$

encode a huge $\mathbf{R}$ with $L \cdot |\mathcal{E}|$ "small" matrices

On the stochastic structure of parallelism and synchronisation models for distributed algorithms
Plateau [SIGMETRICS 1985]

factor L slowdown, still needs a probability vector of size $|\mathcal{X}_{rch}|$

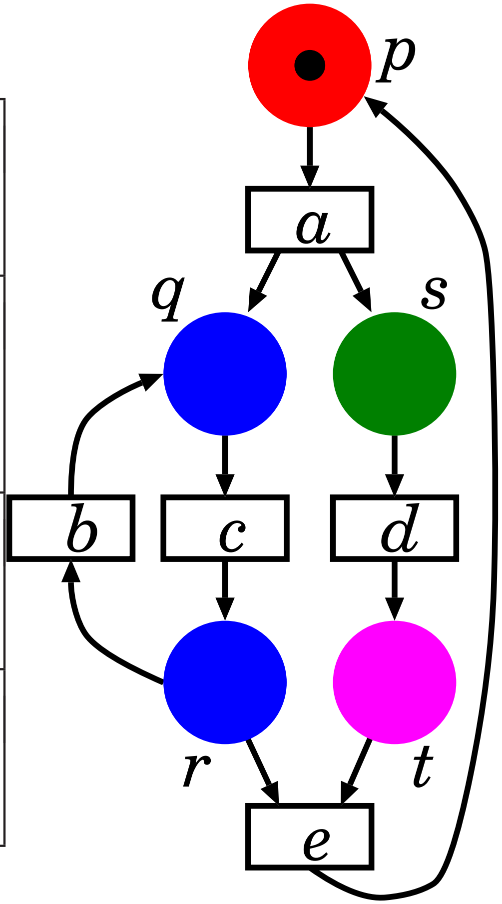
Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models Buchholz, Ciardo, Donatelli, Kemper [INFORMS J. Comp. 2000]

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

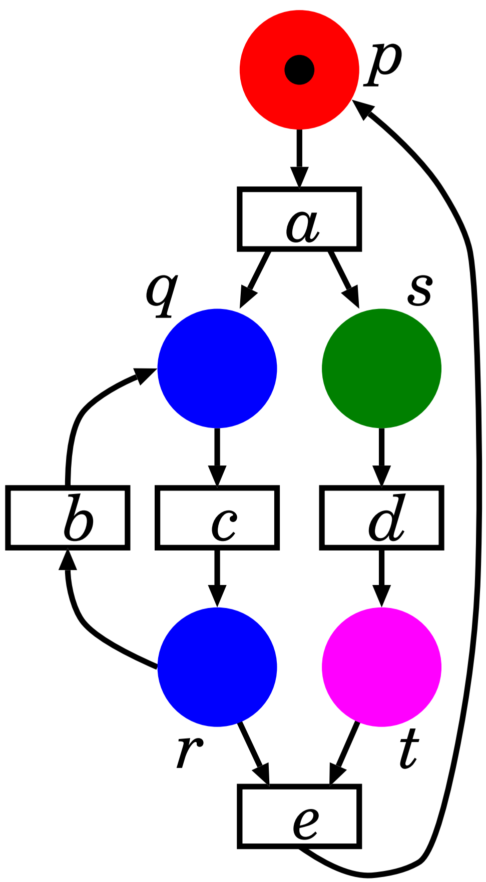
$\mathcal{X}_4 : \{p^0, p^1\} \equiv \{0,1\}$ $\mathcal{X}_3 : \{q^0r^0, q^1r^0, q^0r^1\} \equiv \{0,1,2\}$ $\mathcal{X}_2 : \{s^0, s^1\} \equiv \{0,1\}$ $\mathcal{X}_1 : \{t^0, t^1\} \equiv \{0,1\}$

EVENTS $\rightarrow$ / LEVELS $\downarrow$

$\mathbf{R}_{4,a}$: $\begin{bmatrix}0 & 0\\ \alpha & 0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{R}_{4,e}$: $\begin{bmatrix}0 & \epsilon\\ 0 & 0\end{bmatrix}$
$\mathbf{R}_{3,a}$: $\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$	$\mathbf{R}_{3,b}$: $\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & \beta & 0\end{bmatrix}$	$\mathbf{R}_{3,c}$: $\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & \gamma\\ 0 & 0 & 0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{R}_{3,e}$: $\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 1 & 0 & 0\end{bmatrix}$
$\mathbf{R}_{2,a}$: $\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{R}_{2,d}$: $\begin{bmatrix}0 & 0\\ \delta & 0\end{bmatrix}$	$\mathbf{I}$
$\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\mathbf{R}_{1,d}$: $\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}$	$\mathbf{R}_{1,e}$: $\begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}$



we can (conservatively) determine when $\mathbf{R}_{k,\alpha} = \mathbf{I}$ from the model

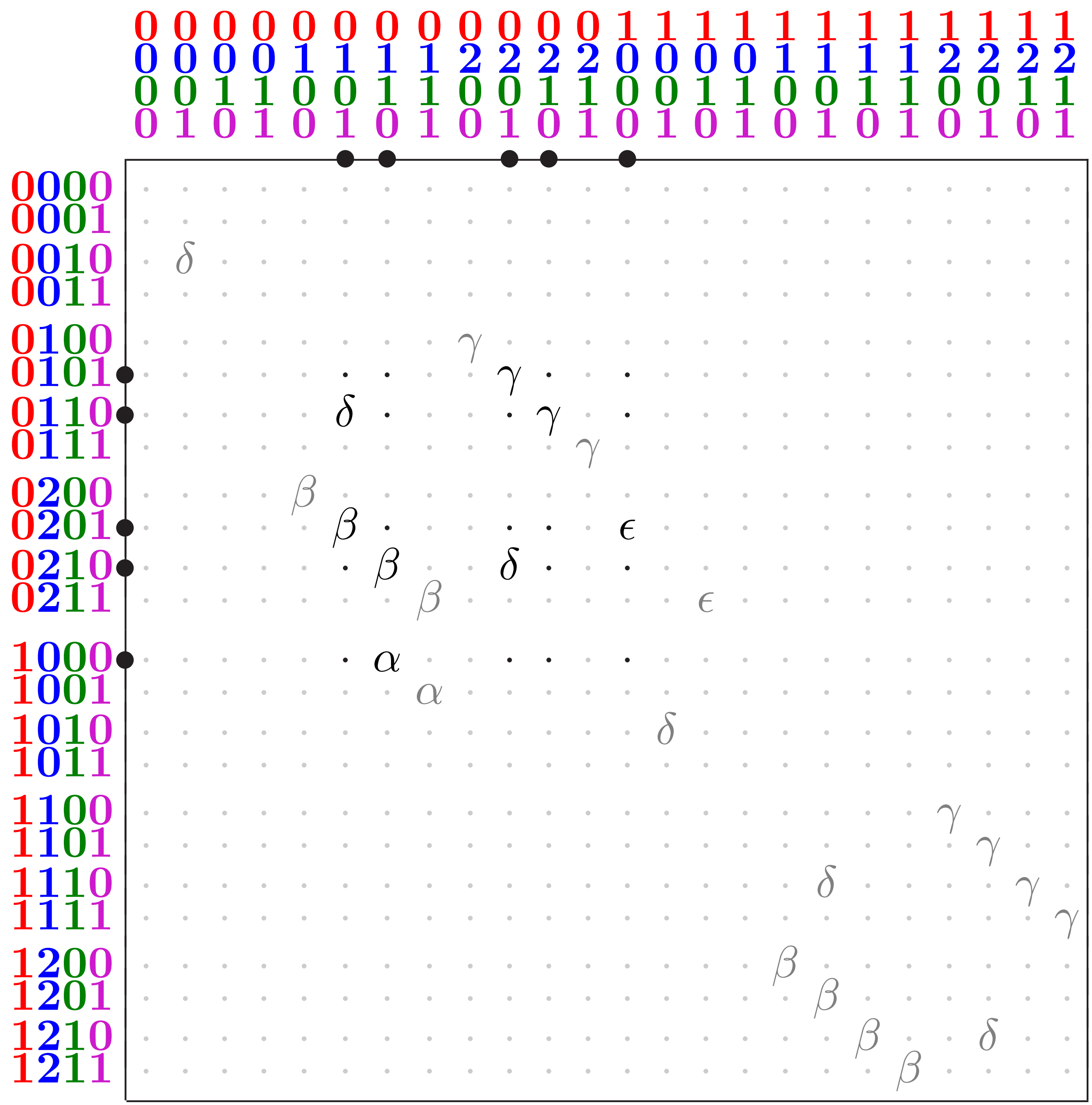


$\{p^0, p^1\} \equiv \{0, 1\}$

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

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

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



State indexing options: potential ψ_{pot} vs. actual ψ_{rch}

For Markov analysis, we can generate $\mathcal{X}_{rch}$ first, starting from $\mathcal{X}_{init}$ and $\mathcal{N}_{pot} : \mathcal{X}_{pot} \to 2^{\mathcal{X}_{pot}}$

Once we know $\mathcal{X}_{rch}$:

- We can restrict $\mathcal{N}_{pot}$ to $\mathcal{N}_{rch} : \mathcal{X}_{rch} \to 2^{\mathcal{X}_{rch}}$ (if needed for further logical analysis)
- We can store $\mathbf{R}_{pot} : \mathcal{X}_{pot} \times \mathcal{X}_{pot} \to \mathbb{R}$ or $\mathbf{R}_{rch} : \mathcal{X}_{rch} \times \mathcal{X}_{rch} \to \mathbb{R}$
- We can choose algorithms that use $\pi_{pot} : \mathcal{X}_{pot} \to \mathbb{R}$ or $\boldsymbol{\pi}_{rch} : \mathcal{X}_{rch} \to \mathbb{R}$

Strictly **explicit** methods: using actual, or reachable, $\mathbf{R}_{rch}$ and $\boldsymbol{\pi}_{rch}$ is the obvious choice

Strictly **implicit** methods: decision diagrams usually don't work well to store π_{pot} or $\boldsymbol{\pi}_{rch}$

We often resort to **hybrid** methods, but they, too, have tradeoffs:

- Storing $\boldsymbol{\pi}_{rch}$ instead of π_{pot} (as a full vector) is practically unavoidable when $|\mathcal{X}_{pot}| \gg |\mathcal{X}_{rch}|$
- Symbolic storage for $\mathbf{R}_{pot}$ often requires less memory than for $\mathbf{R}_{rch}$
- However, using $\mathbf{R}_{pot}$ in conjunction with $\boldsymbol{\pi}_{rch}$ complicates indexing...
- ...forcing us to store $\psi_{rch} : \mathcal{X}_{pot} \to \{0, 1, ..., |\mathcal{X}_{rch}| - 1\} \cup \{\text{null}\}$, using an EV$^+$MDD...
- ...instead of the easier $\psi_{pot} : \mathcal{X}_{pot} \to \{0, 1, ..., |\mathcal{X}_{pot}| - 1\}$, using mixed-base indexing

EV$^+$MDDs to store the lexicographic indexing function ψ_{rch}

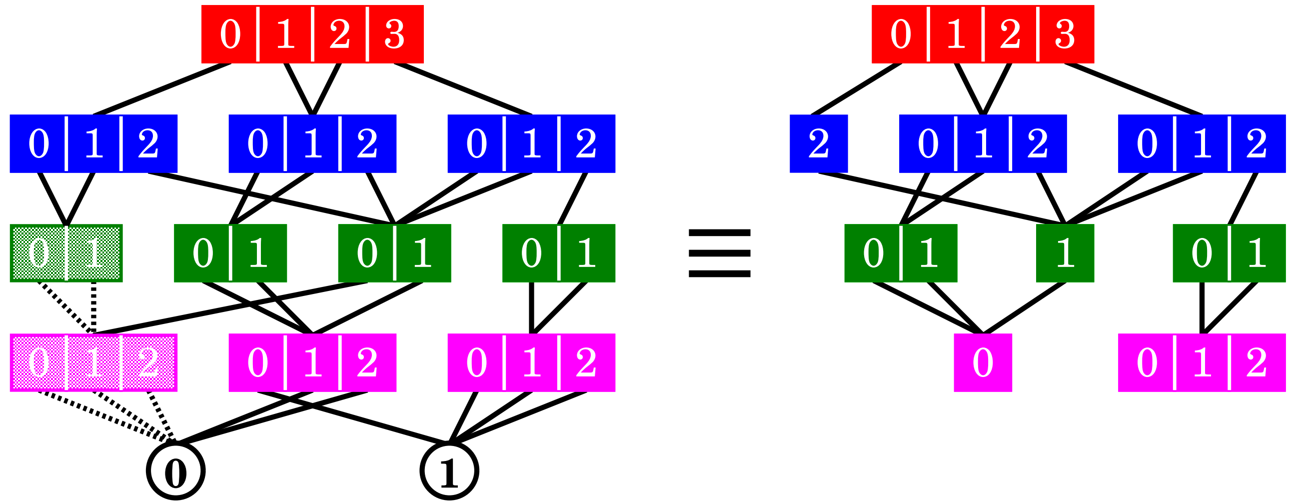
We use MDDs to store the reachable set of states $\mathcal{X}_{rch}$:

$$\mathcal{X}_4 = \{0, 1, 2, 3\}$$

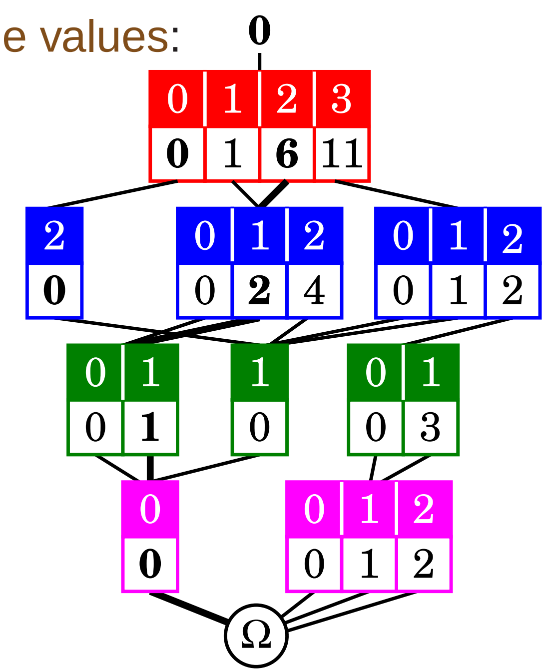
$$\mathcal{X}_3 = \{0, 1, 2\}$$

$$\mathcal{X}_2 = \{0, 1\}$$

$$\mathcal{X}_1 = \{0, 1, 2\}$$



To compute the lexicographic index $\psi_{rch}(\mathbf{i})$ of state $\mathbf{i} \in \mathcal{X}_{rch}$ we use edge values:



- Sum the values found on the corresponding path:

$$\psi_{rch}(2, 1, 1, 0) = \mathbf{0} + \mathbf{6} + \mathbf{2} + \mathbf{1} + \mathbf{0} = \mathbf{9}$$

- State $\mathbf{i}$ is unreachable if its path is not complete:

$$\psi_{rch}(0, 2, 0, 0) = \mathbf{0} + \mathbf{0} + \mathbf{0} + \infty^+ = \infty^+$$

(a missing edge has the default value of ∞^+)

THEOREM: the EV$^+$MDD encoding ψ_{rch} is isomorphic to the MDD encoding $\mathcal{X}_{rch}$

Potential Kronecker: the shuffle algorithm PSh

First algorithm proposed for the solution of Kronecker-encoded CTMCs [Plateau SIGMETRICS 1985]

PSh computes $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{x}} \cdot \bigotimes_{L \geq k \geq 1} \mathbf{A}_k$

PSh^+ computes $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{x}} \cdot \mathbf{I}_{n_L \cdots n_{k+1}} \otimes \mathbf{A}_k \otimes \mathbf{I}_{n_{k-1} \cdots n_1}$

Based on the equality [Davio IEEE-TC 1981]

$$\bigotimes_{L \geq k \geq 1} \mathbf{A}_k = \prod_{L \geq k \geq 1} \mathbf{S}^T_{(n_L \cdots n_{k+1}, n_k \cdots n_1)} \cdot (\mathbf{I}_{|\mathcal{X}_{pot}|/n_k} \otimes \mathbf{A}_k) \cdot \mathbf{S}_{(n_L \cdots n_{k+1}, n_k \cdots n_1)}$$

where $\mathbf{S}_{(a,b)} \in \mathbb{B}^{a \cdot b \times a \cdot b}$ is the matrix describing an (a, b)-perfect shuffle permutation:

$$\mathbf{S}_{(a,b)}[i, j] = \begin{cases} 1 & \text{if } j = (i \text{ mod } a) \cdot b + (i \text{ div } a) \\ 0 & \text{otherwise} \end{cases}$$

Requires

- L vector permutations and
- L multiplications $\mathbf{x} \cdot (\mathbf{I}_{|\mathcal{X}_{pot}|/n_k} \otimes \mathbf{A}_k)$

Complexity of the k-th multiplication: $O(|\mathcal{X}_{pot}|/n_k \cdot \eta[\mathbf{A}_k])$

Potential Kronecker: the shuffle algorithm PSh

PSh(in: $n_L, \ldots, n_1$, $\mathbf{A}_L, \ldots, \mathbf{A}_1$; inout: $\widehat{\mathbf{x}}$, $\widehat{\mathbf{y}}$);

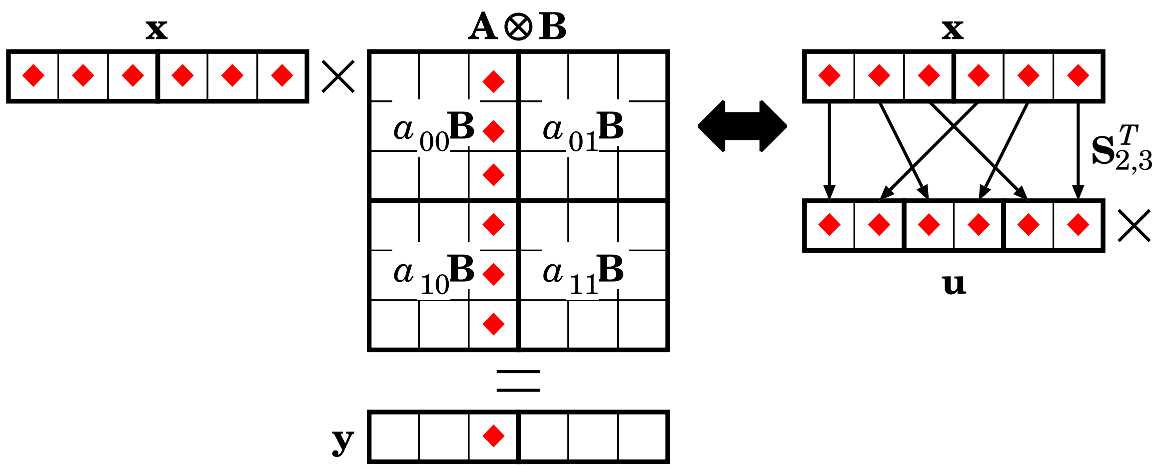
```
n_left ← 1;
n_right ← n_{L-1} ··· n_1;
for k = L down to 1
    base ← 0;
    jump ← n_k · n_right;
    if A_k ≠ I then
        for block = 0 to n_left − 1
            for offset = 0 to n_right − 1
                index ← base + offset;
                for h = 0 to n_k − 1
                    z_h ← x̂_index;
                    index ← index + n_right;
                z' ← z · A_k;
                index ← base + offset;
                for h = 0 to n_k − 1
                    ŷ_index ← z'_h;
                    index ← index + n_right;
            base ← base + jump;
        x̂ ← ŷ;
    n_left ← n_left · n_k;
    n_right ← n_right / n_{k-1};
```

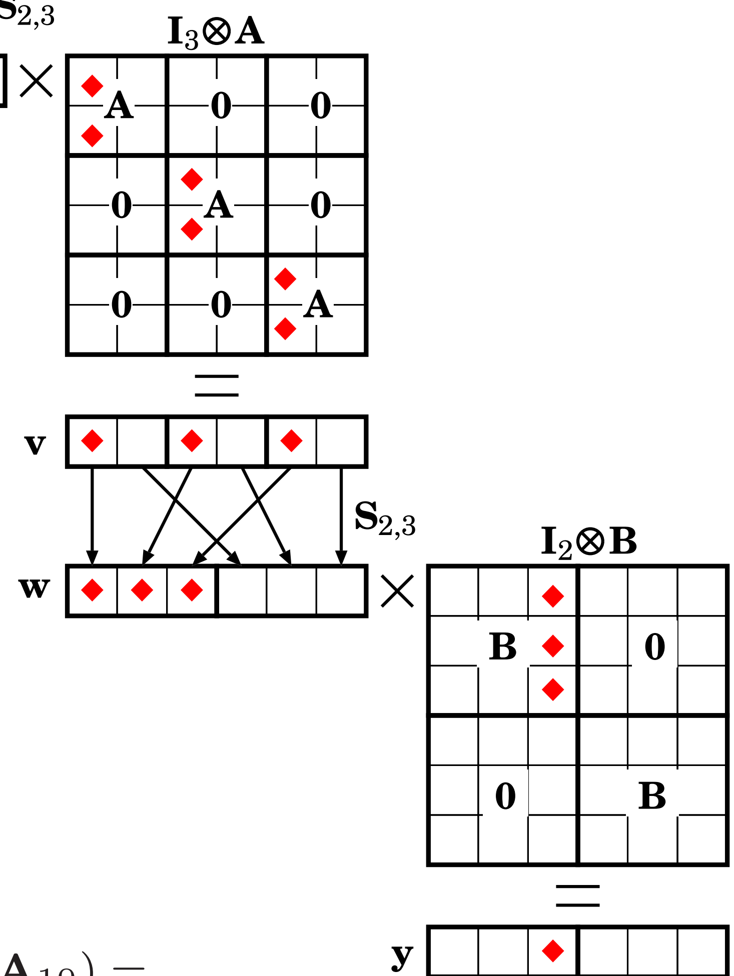
Let n_0 be 1

Example of shuffle computation

$\mathbf{y} \leftarrow \mathbf{x} \cdot (\mathbf{A} \otimes \mathbf{B})$

Follow the entries marked with a diamond to obtain $\mathbf{y}_2$





$$\mathbf{y} \leftarrow \underbrace{\underbrace{\underbrace{\underbrace{\mathbf{x} \cdot \mathbf{S}_{2,3}^T}_{\mathbf{u}} \cdot (\mathbf{I}_3 \otimes \mathbf{A})}_{\mathbf{v}} \cdot \mathbf{S}_{2,3}}_{\mathbf{w}} \cdot \mathbf{S}_{6,1}^T \cdot (\mathbf{I}_2 \otimes \mathbf{B}) \cdot \mathbf{S}_{6,1}}_{\mathbf{y}}$$

$\mathbf{y}_2 \leftarrow$

$\mathbf{B}_{02}\mathbf{w}_0 + \mathbf{B}_{12}\mathbf{w}_1 + \mathbf{B}_{22}\mathbf{w}_2 =$

$\mathbf{B}_{02}\mathbf{v}_0 + \mathbf{B}_{12}\mathbf{v}_2 + \mathbf{B}_{22}\mathbf{v}_4 =$

$\mathbf{B}_{02}(\mathbf{u}_0\mathbf{A}_{00} + \mathbf{u}_1\mathbf{A}_{10}) + \mathbf{B}_{12}(\mathbf{u}_2\mathbf{A}_{00} + \mathbf{u}_3\mathbf{A}_{10}) + \mathbf{B}_{22}(\mathbf{u}_4\mathbf{A}_{00} + \mathbf{u}_5\mathbf{A}_{10}) =$

$\mathbf{B}_{02}(\mathbf{x}_0\mathbf{A}_{00} + \mathbf{x}_3\mathbf{A}_{10}) + \mathbf{B}_{12}(\mathbf{x}_1\mathbf{A}_{00} + \mathbf{x}_4\mathbf{A}_{10}) + \mathbf{B}_{22}(\mathbf{x}_2\mathbf{A}_{00} + \mathbf{x}_5\mathbf{A}_{10}) =$

$\mathbf{A}_{00}\mathbf{B}_{02}\mathbf{x}_0 + \mathbf{A}_{00}\mathbf{B}_{12}\mathbf{x}_1 + \mathbf{A}_{00}\mathbf{B}_{22}\mathbf{x}_2 + \mathbf{A}_{10}\mathbf{B}_{02}\mathbf{x}_3 + \mathbf{A}_{10}\mathbf{B}_{12}\mathbf{x}_4 + \mathbf{A}_{10}\mathbf{B}_{22}\mathbf{x}_5$

Complexity of PSh and PSh^+

Let α be the average number of nonzero entries per row in $\mathbf{A}_k$

PSh has complexity $O\left(\sum_{L \geq k \geq 1} |\mathcal{X}_{pot}|/n_k \cdot \eta[\mathbf{A}_k]\right) = O\left(|\mathcal{X}_{pot}| \cdot L \cdot \alpha\right)$

Even when $\mathcal{X}_{pot} = \mathcal{X}_{rch}$, PSh is faster than $Ordinary$ explicit multiplication only if

$$|\mathcal{X}_{pot}| \cdot L \cdot \alpha < |\mathcal{X}_{pot}| \cdot \alpha^L \quad \Leftrightarrow \quad \alpha > L^{\frac{1}{L-1}}$$

PSh^+ has complexity $O\left(|\mathcal{X}_{pot}|/n_k \cdot \eta[\mathbf{A}_k]\right) = O\left(|\mathcal{X}_{pot}| \cdot \alpha\right)$

Complexity of computing $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + \widehat{\mathbf{x}} \cdot \bigoplus_{L \geq k \geq 1} \mathbf{A}_k$:

$$O\left(\sum_{L \geq k \geq 1} |\mathcal{X}_{pot}|/n_k \cdot \eta[\mathbf{A}_k]\right) = O\left(|\mathcal{X}_{pot}| \sum_{L \geq k \geq 1} \frac{\eta[\mathbf{A}_k]}{n_k}\right) = O\left(|\mathcal{X}_{pot}| \cdot L \cdot \alpha\right)$$

$Ordinary$ is faster than PSh when $\alpha \leq 1$

PSh^+ saves space, but not time, w.r.t. $Ordinary$

Potential Kronecker: PRw and PRw^+

$PRwEl$(in: $\mathbf{i}, x, n_L, \ldots, n_1, \mathbf{A}_L, \ldots, \mathbf{A}_1$; inout: $\widehat{\mathbf{y}}$)

for each $\mathbf{j}_L$ s.t. $\mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L] > 0$
    $j'_L \leftarrow \mathbf{j}_L$; $a_L \leftarrow \mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L]$;
    for each $\mathbf{j}_{L-1}$ s.t. $\mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}] > 0$
        $j'_{L-1} \leftarrow j'_L \cdot n_{L-1} + \mathbf{j}_{L-1}$; $a_{L-1} \leftarrow a_L \cdot \mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}]$;

$\ldots$

        for each $\mathbf{j}_1$ s.t. $\mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1] > 0$
            $j'_1 \leftarrow j'_2 \cdot n_1 + \mathbf{j}_1$; $a_1 \leftarrow a_2 \cdot \mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1]$;
            $\widehat{\mathbf{y}}_{j'_1} \leftarrow \widehat{\mathbf{y}}_{j'_1} + x \cdot a_1$;

PRw(in: $\widehat{\mathbf{x}}, n_L, \ldots, n_1, \mathbf{A}_L, \ldots, \mathbf{A}_1$; inout: $\widehat{\mathbf{y}}$)

for $\mathbf{i} = 0$ to $|\mathcal{X}_{pot}| - 1$
    $PRwEl(i, \widehat{\mathbf{x}}_i, n_L, \ldots, n_1, \mathbf{A}_L, .., \mathbf{A}_1, \widehat{\mathbf{y}})$;

$PRwEl^+$(in: $n_k, n_{k-1} \cdots n_1, i_k^-, \mathbf{i}_k, i_k^+, x, \mathbf{A}_k$; inout: $\widehat{\mathbf{y}}$)

for each $\mathbf{j}_k$ s.t. $\mathbf{A}_k[\mathbf{i}_k, \mathbf{j}_k] > 0$
    $j' \leftarrow (i_k^- \cdot n_k + \mathbf{j}_k) \cdot n_{k-1} \cdots n_1 + i_k^+$;
    $\widehat{\mathbf{y}}_{j'} \leftarrow \widehat{\mathbf{y}}_{j'} + x \cdot \mathbf{A}_k[\mathbf{i}_k, \mathbf{j}_k]$;

PRw^+(in: $\widehat{\mathbf{x}}, n_L \cdots n_{k+1}, n_k, n_{k-1} \cdots n_1, \mathbf{A}_k$; inout: $\widehat{\mathbf{y}}$)

for $i \equiv (i_k^-, \mathbf{i}_k, i_k^+) = 0$ to $n_L \cdots n_{k+1} \cdot n_k \cdot n_{k-1} \cdots n_1 - 1$
    $PRwEl^+(n_k, n_{k-1} \cdots n_1, i_k^-, \mathbf{i}_k, i_k^+, \widehat{\mathbf{x}}_i, \mathbf{A}_k, \widehat{\mathbf{y}})$;

Complexity of PRw and PRw^+

PRw computes $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + \widehat{\mathbf{x}} \cdot \mathbf{A}$, according to the definition of Kronecker product
Requires sparse row-wise format for each $\mathbf{A}_k$

$PRwEl$ computes the contribution of $\widehat{\mathbf{x}}_i$ to each entry of $\widehat{\mathbf{y}}$ as

$$\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + \widehat{\mathbf{x}}_i \cdot \mathbf{A}_{i,\mathcal{X}_{pot}}$$

$PRwEl$ reaches statement $\quad a_k \leftarrow a_{k-1} \cdot \mathbf{A}_k[\mathbf{i}_k, \mathbf{j}_k] \quad O(\alpha^k)$ times

PRw makes $|\mathcal{X}_{pot}|$ calls to $PRwEl$, hence has complexity

$$O\left(|\mathcal{X}_{pot}| \cdot \sum_{L \geq k \geq 1} \alpha^k\right) = \begin{cases} O\left(|\mathcal{X}_{pot}| \cdot L\right) = O(L \cdot \eta[\mathbf{A}]) & \text{if } \alpha \leq 1 \\ O(|\mathcal{X}_{pot}| \cdot \alpha^L) = O(\eta[\mathbf{A}]) & \text{if } \alpha > 1 \end{cases}$$

PRw^+ has complexity $O\left(|\mathcal{X}_{pot}| \cdot \dfrac{\eta[\mathbf{A}_k]}{n_k}\right) = O\left(|\mathcal{X}_{pot}| \cdot \alpha\right)$

Complexity of computing $\widehat{\mathbf{y}} \leftarrow \widehat{\mathbf{y}} + \widehat{\mathbf{x}} \cdot \bigoplus_{L \geq k \geq 1} \mathbf{A}_k$ using PRw^+: $O\left(|\mathcal{X}_{pot}| \cdot L \cdot \alpha\right)$

PRw amortizes the multiplications for a_{L-1},...,a_2 only if $\alpha \gg 1$
PRw^+ saves space, but not time, w.r.t. $Ordinary$

Potential Kronecker: $PRwCl$ and $PRwCl^+$

$PRwCl$(in: $\widehat{\mathbf{x}}, n_L, \ldots, n_1, \mathbf{A}_L, \ldots, \mathbf{A}_1$; inout: $\widehat{\mathbf{y}}$)

for $\mathbf{i}_L = 0$ to $n_L - 1$
    for each $\mathbf{j}_L$ s.t. $\mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L] > 0$
        $j'_L \leftarrow \mathbf{j}_L;\ a_L \leftarrow \mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L];$
        for $\mathbf{i}_{L-1} = 0$ to $n_{L-1} - 1$
            for each $\mathbf{j}_{L-1}$ s.t. $\mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}] > 0$
                $j'_{L-1} \leftarrow j'_L \cdot n_{L-1} + \mathbf{j}_{L-1};\ a_{L-1} \leftarrow a_L \cdot \mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}];$
$\ldots$
                for $\mathbf{i}_1 = 0$ to $n_1 - 1$
                    for each $\mathbf{j}_1$ s.t. $\mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1] > 0$
                        $j'_1 \leftarrow j'_2 \cdot n_1 + \mathbf{j}_1;\ a_1 \leftarrow a_2 \cdot \mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1];$
                        $\widehat{\mathbf{y}}_{j'_1} \leftarrow \widehat{\mathbf{y}}_{j'_1} + \widehat{\mathbf{x}}_{\mathbf{i}} \cdot a_1;$

The overall complexity is $O\left(|\mathcal{X}_{pot}| \cdot \alpha^L\right)$

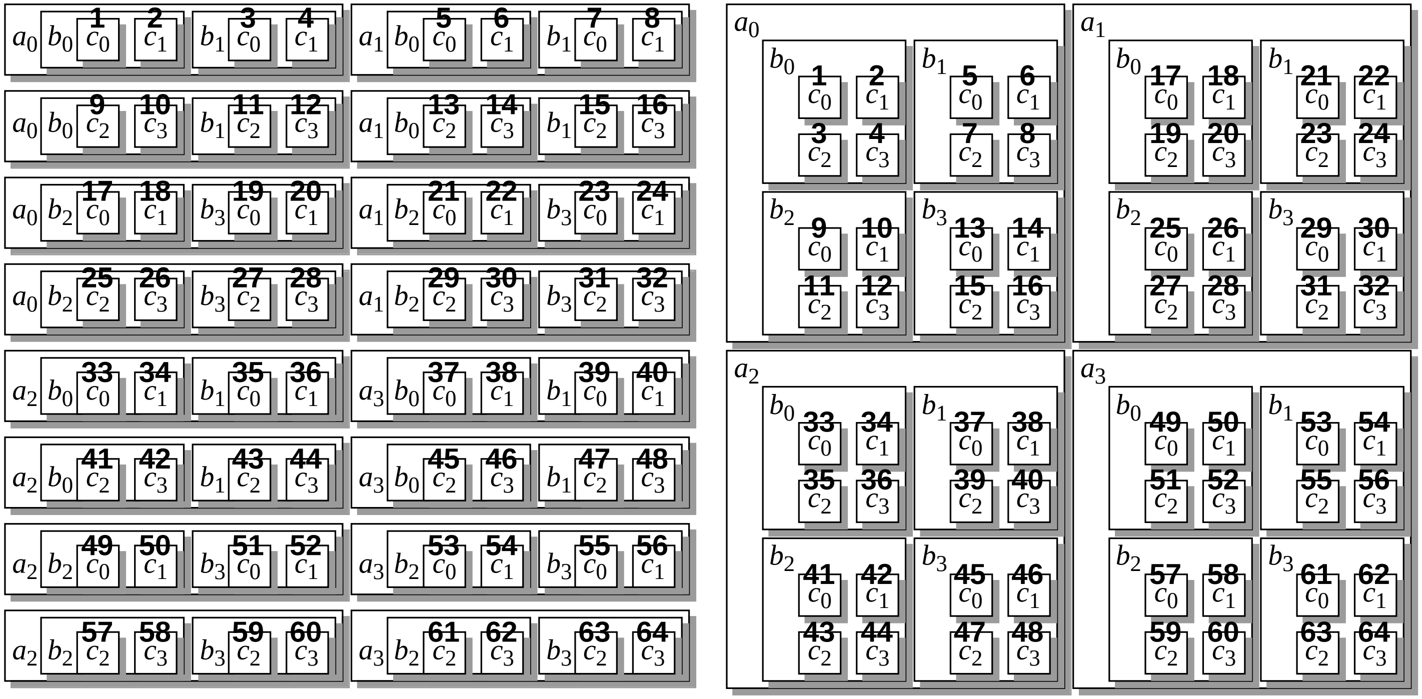
$PRwCl^+$(in: $\widehat{\mathbf{x}}, n_L \cdots n_{k+1}, n_k, n_{k-1} \cdots n_1, \mathbf{A}_k$; inout: $\widehat{\mathbf{y}}$)

for $i_k^- = 0$ to $n_L \cdots n_{k+1} - 1$
    for $\mathbf{i}_k = 0$ to $n_k - 1$
        for each $\mathbf{j}_k$ s.t. $\mathbf{A}_k[\mathbf{i}_k, \mathbf{j}_k] > 0$
            $j'_k \leftarrow i_k^- \cdot n_k + \mathbf{j}_k;$
            for $i_k^+ = 0$ to $n_{k-1} \cdots n_1 - 1$
                $j'_L \leftarrow j'_k \cdot n_{k-1} \cdots n_1 + i_k^+;$
                $\widehat{\mathbf{y}}_{j'_L} \leftarrow \widehat{\mathbf{y}}_{j'_L} + \widehat{\mathbf{x}}_{(i_k^-, \mathbf{i}_k, i_k^+)} \cdot \mathbf{A}_k[\mathbf{i}_k, \mathbf{j}_k];$

Potential Kronecker: $PRwCl$

$$\widehat{\mathbf{x}} \cdot \mathbf{A} = \widehat{\mathbf{x}} \cdot \left(\begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 & b_1 \\ b_2 & b_3 \end{bmatrix} \otimes \begin{bmatrix} c_0 & c_1 \\ c_2 & c_3 \end{bmatrix} \right)$$

PRw



$PRwCl$

Each "b" and "c" box corresponds to one multiplication:

- $\mathbf{A}$ contains $8 \times 8 = 64$ entries of the form $a_i b_j c_l$
- Computing each entry from scratch: $64 \times 2 = 128$ multiplications
- Using PRw: $64 + 32 = 96$ multiplications
- Using $PRwCl$: $64 + 16 = 80$ multiplications: interleaving helps!

the entries of $\mathbf{A}$ are not generated in row or column order

Actual Kronecker: ARw

ARw(in: $\mathbf{x}$, $\mathbf{A}_L, \ldots, \mathbf{A}_1, \mathcal{X}_{rch}$; inout: $\mathbf{y}$)

1 for each $\mathbf{i} \in \mathcal{X}_{rch}$
2 $\quad I \leftarrow \psi(\mathbf{i})$;
3 $\quad$ for each $\mathbf{j}_L$ s.t. $\mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L] > 0$
4 $\quad\quad a_L \leftarrow \mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L]$;
5 $\quad\quad$ for each $\mathbf{j}_{L-1}$ s.t. $\mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}] > 0$
6 $\quad\quad\quad a_{L-1} \leftarrow a_L \cdot \mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}]$;

...

7 $\quad\quad\quad$ for each $\mathbf{j}_1$ s.t. $\mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1] > 0$
8 $\quad\quad\quad\quad a_1 \leftarrow a_2 \cdot \mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1]$;
9 $\quad\quad\quad\quad J \leftarrow \psi(\mathbf{j})$;
10 $\quad\quad\quad\quad \mathbf{y}_J \leftarrow \mathbf{y}_J + \mathbf{x}_I \cdot a_1$;

Statement 9 computes the index $J = \psi(\mathbf{j})$ of state $\mathbf{j}$ in the array $\mathbf{y}$

$$O\left(|\mathcal{X}_{rch}| \cdot \left(\sum_{L \geq k \geq 1} \alpha^k + \alpha^L \cdot \log|\mathcal{X}_{rch}|\right)\right) = \begin{cases} O\left(|\mathcal{X}_{rch}| \cdot (L + \log|\mathcal{X}_{rch}|)\right) & \text{if } \alpha \leq 1 \\ O\left(|\mathcal{X}_{rch}| \cdot \alpha^L \cdot \log|\mathcal{X}_{rch}|\right) & \text{if } \alpha > 1 \end{cases}$$

if $L < \log|\mathcal{X}_{rch}|$: ARw has a $\log|\mathcal{X}_{rch}|$ overhead w.r.t. $Ordinary$

Actual Kronecker: $ARwCl$ and $ARwCl^+$

$ARwCl$(in: $\mathbf{x}$, $\mathbf{A}_L, ..., \mathbf{A}_1, \mathcal{X}_{rch}$; inout: $\mathbf{y}$)

for each $\mathbf{i}_L \in \mathcal{X}_L$ — *all local states $\mathbf{i}_L$*
  $I_L \leftarrow \psi_L(\mathbf{i}_L)$;
  for each $\mathbf{j}_L$ s.t. $\mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L] > 0$
    $J_L \leftarrow \psi_L(\mathbf{j}_L)$;
    if $J_L \neq$ null then
      $a_L \leftarrow \mathbf{A}_L[\mathbf{i}_L, \mathbf{j}_L]$;
      for each $\mathbf{i}_{L-1} \in \mathcal{X}_{L-1}(\mathbf{i}_L)$ — *all $\mathbf{i}_{L-1}$ compatible with $\mathbf{i}_L$*
        $I_{L-1} \leftarrow \psi_{L-1}(I_L, \mathbf{i}_{L-1})$;
        for each $\mathbf{j}_{L-1}$ s.t. $\mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}] > 0$
          $J_{L-1} \leftarrow \psi_{L-1}(J_L, \mathbf{j}_{L-1})$;
          if $J_{L-1} \neq$ null then
            $a_{L-1} \leftarrow a_L \cdot \mathbf{A}_{L-1}[\mathbf{i}_{L-1}, \mathbf{j}_{L-1}]$;

...

            for each $\mathbf{i}_1 \in \mathcal{X}_1(\mathbf{i}_L, ..., \mathbf{i}_2)$ — *all $\mathbf{i}_1$ compatible with $\mathbf{i}_L, ..., \mathbf{i}_2$*
              $I_1 \leftarrow \psi_1(I_2, \mathbf{i}_1)$;
              for each $\mathbf{j}_1$ s.t. $\mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1] > 0$
                $J_1 \leftarrow \psi_1(J_2, \mathbf{j}_1)$;
                if $J_1 \neq$ null then
                  $a_1 \leftarrow a_2 \cdot \mathbf{A}_1[\mathbf{i}_1, \mathbf{j}_1]$;
                  $\mathbf{y}_{J_1} \leftarrow \mathbf{y}_{J_1} + \mathbf{x}_{I_1} \cdot a_1$;

we use EV$^+$MDDs to index the state space

Actual Kronecker: $ARwCl$

Complexity of $ARwCl$: $O\left(\sum_{L\geq k\geq 1} |\mathcal{X}_1|\cdots|\mathcal{X}_k| \cdot \alpha^k \cdot \log n_k\right) = O(|\mathcal{X}_{rch}| \cdot \alpha^L \cdot \log n_L)$

assuming that $|\mathcal{X}_1|\cdots|\mathcal{X}_{L-1}| \ll |\mathcal{X}_{rch}|$

Complexity of $ARwCl^+$: $O(|\mathcal{X}_{rch}| \cdot \alpha \cdot \log n_L)$ regardless of k

The resulting complexity of computing

$$\mathbf{y} \leftarrow \mathbf{y} + \mathbf{x} \cdot \left(\bigoplus_{L\geq k\geq 1} \mathbf{A}_k\right)_{\mathcal{X}_{rch},\mathcal{X}_{rch}}$$

using $ARwCl^+$ is $O\left(L \cdot |\mathcal{X}_{rch}| \cdot \alpha \cdot \log n_L\right)$

only $\log n_L$ overhead w.r.t. $Ordinary$ for any sparsity level

but it cannot be used in a Gauss-Seidel iteration

Beyond Kronecker

Kronecker-consistent decomposition of a CTMC model

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

- the potential transition rate matrix $\widehat{\mathbf{R}}$ is additively partitioned $\widehat{\mathbf{R}} = \sum_{\alpha \in \mathcal{E}} \widehat{\mathbf{R}}_\alpha$
- $\widehat{\mathcal{S}} = \mathcal{X}_L \times \cdots \times \mathcal{X}_1$, a global state $\mathbf{i}$ consists of L local states $\mathbf{i} = (\mathbf{i}_L, \ldots, \mathbf{i}_1)$
- and, most importantly, we can multiplicatively partition each $\widehat{\mathbf{R}}_\alpha$, that is, we can write

$\lambda_\alpha(\mathbf{i}) = \lambda_{L,\alpha}(\mathbf{i}_L) \cdots \lambda_{1,\alpha}(\mathbf{i}_1)$ and $\Delta_\alpha(\mathbf{i},\mathbf{j}) = \Delta_{L,\alpha}(\mathbf{i}_L,\mathbf{j}_L) \cdots \Delta_{1,\alpha}(\mathbf{i}_1,\mathbf{j}_1)$

$$\widehat{\mathbf{R}}_\alpha = \mathbf{R}_{L,\alpha} \otimes \cdots \otimes \mathbf{R}_{\mathbf{1},\alpha}$$

for stochastic Petri nets with transition rates depending on at most one place, any partition of the places into L subsets is consistent (even with inhibitor, reset, or probabilistic arcs)

in general, however, a CTMC model with L submodels and $|\mathcal{E}|$ events does not have a Kronecker representation (unless we reduce L by merging submodels or increase $|\mathcal{E}|$ by splitting events)

Multiterminal multiway decision diagrams

From BDDs to MDDs: allow multiway choices at each nonterminal node [Kam PhD 1995]

From BDDs to MTBDDs: allow multiple terminal nodes, not just 0 and 1 [Clarke IWLS 1993]

From BDDs to MTMDDs combine both generalizations

We can use a quasi-reduced MTMDD to encode a real matrix $\mathbf{A} : \mathcal{X}_{pot} \times \mathcal{X}_{pot} \to \mathbb{R}$

- Nodes are organized into $2L+1$ levels
  - Map variables $(\mathbf{i}_L, \mathbf{j}_L, ..., \mathbf{i}_1, \mathbf{j}_1)$ onto levels $(2L, ..., 1)$ interleaved order is usually best
  - Level $2L$ contains the unique root node
  - Levels $2L-1$ through 1 contain one or more nodes, no duplicate nodes allowed
  - Level 0 contains as many nodes as the different entries in $\mathbf{A}$
- A node at a level corresponding to $\mathbf{i}_k$ or $\mathbf{j}_k$ has $|\mathcal{X}_k|$ arcs pointing to nodes at the level below

$\mathbf{A}[\mathbf{i}, \mathbf{j}] = x \;\Leftrightarrow\;$ path labeled $(\mathbf{i}_L, \mathbf{j}_L, ..., \mathbf{i}_1, \mathbf{j}_1)$ leads to node x at level 0

MTMDDs encoding of the transition rate matrix

When using MTMDDs to store the transition rate matrix, we have a choice:

- Store $\mathbf{R}_{pot} : \mathcal{X}_{pot} \times \mathcal{X}_{pot} \to \mathbb{R}$
  - $\mathbf{R}_{pot}[\mathbf{i},\mathbf{j}] = 0$ if $\mathbf{i} \in \mathcal{X}_{rch}$ and $\mathbf{j} \notin \mathcal{X}_{rch}$
  - but it is possible to have $\mathbf{R}_{pot}[\mathbf{i},\mathbf{j}] > 0$ for $\mathbf{i} \notin \mathcal{X}_{rch}$ and $\mathbf{j} \in \mathcal{X}_{rch}$
  - a natural choice if we use a compositional approach

- Store $\mathbf{R}_{rch} : \mathcal{X}_{rch} \times \mathcal{X}_{rch} \to \mathbb{R}$
  - strictly speaking, still $\mathcal{X}_{pot} \times \mathcal{X}_{pot} \to \mathbb{R}$, but $\mathbf{R}[\mathbf{i},\mathbf{j}] = 0$ if $\mathbf{i} \notin \mathcal{X}_{rch}$ or $\mathbf{j} \notin \mathcal{X}_{rch}$
  - usually requires more MTMDD nodes
  - can be built by enumerating the entries explicitly and storing them implicitly in an MTMDD or...
  - ...by setting to zero the rows corresponding to $\mathcal{X}_{pot} \setminus \mathcal{X}_{rch}$ in the MTMDD encoding of $\mathbf{R}_{pot}$
    $\Rightarrow$ filter the enties of $\mathbf{R}_{pot}$ using $\mathcal{X}_{rch}$

Canonical matrix diagrams (MxDs) [Miner PhD 2000]

- There is a single root node r, with an associated value $\rho \in \mathbb{R}^{\geq 0}$; $\langle \rho, r \rangle$ is the root edge
- Each non-terminal node p is at a level $p.lvl \in \{L, .., 1\}$
- There is a single terminal node Ω, at level 0
- A node p at level $k > 0$ has $n_k \times n_k$ edges of the form $p[i_k, j_k] = \langle \sigma, q \rangle$, where
  - the value associated with the edge satisfies $\sigma \in [0, 1]$
  - the destination node q satisfies $q.lvl < k$
  - at least one edge has $\sigma = 1$ and, if $\sigma = 0$, then $q = \Omega$
- There is no identity node, i.e., p at level $k > 0$ such that $p[i_k, j_k] = \begin{cases} \langle 1, q \rangle & \text{if } i_k = j_k \\ \langle 0, \Omega \rangle & \text{if } i_k \neq j_k \end{cases}$

$\langle \sigma, p \rangle$ w.r.t. $l \geq k = p.lvl$ encodes matrix $\mathbf{A}_{(l, \langle \sigma, p \rangle)} : \mathcal{X}_l \times \cdots \times \mathcal{X}_1 \times \mathcal{X}_l \times \cdots \times \mathcal{X}_1 \to \mathbb{R}^{\geq 0}$

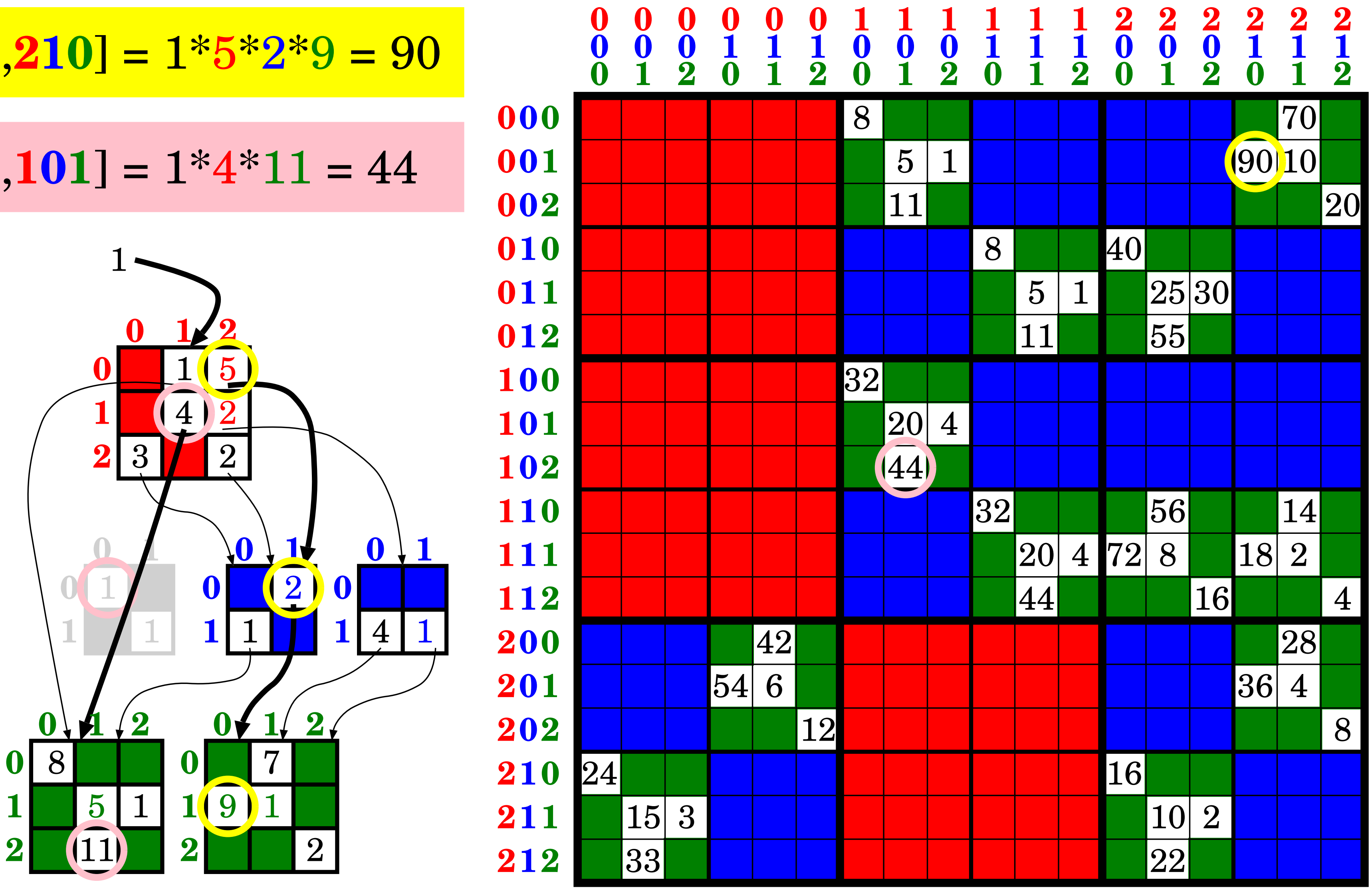
$$\begin{cases} \mathbf{I}_{n_l \times n_l} \otimes \mathbf{A}_{(l-1, \langle \sigma, p \rangle)} & \text{if } l > k \\ \sigma \cdot \left[\begin{array}{c|c|c} \mathbf{A}_{(l-1, p[0,0])} & \cdots & \mathbf{A}_{(l-1, p[0, n_l - 1])} \\ \hline \cdots & \cdots & \cdots \\ \hline \mathbf{A}_{(l-1, p[n_l - 1, 0])} & \cdots & \mathbf{A}_{(l-1, p[n_l - 1, n_l - 1])} \end{array} \right] & \text{if } l = k \\ \sigma & \text{if } l = k = 0 \text{, thus } p = \Omega \end{cases}$$

MxDs can **canonically** encode matrices $\mathcal{X}_{pot} \times \mathcal{X}_{pot} \to \mathbb{R}^{\geq 0}$

An example of MxD

R[001,210] = 1*5*2*9 = 90

R[102,101] = 1*4*11 = 44



- A generalization of the Kronecker encoding to non-Kronecker-consistent systems
- Can filter $\mathbf{R}_{pot}$ to enforce knowledge of the reachable states $\mathcal{X}_{rch}$ and store $\mathbf{R}_{rch}$
- Analogous to a $2L$-level edge-valued decision diagram with a special identity reduction rule

Empirical comparison

Memory consumption in bytes for: $\mathcal{X}_{rch}$ (MDD), $\mathbf{R}_{rch}$ (Sparse), $\mathbf{R}_{pot}$ (Kronecker), $\mathbf{R}_{pot}$ and $\mathbf{R}_{rch}$ (Pot/Act MxD), $\mathbf{R}_{pot}$ and $\mathbf{R}_{rch}$ (Pot/Act MTMDD)

Model	N	$\lvert\mathcal{X}_{pot}\rvert$	$\lvert\mathcal{X}_{rch}\rvert$	**MDD**	**Sparse**	**Kron**	**Pot MxD**	**Act MxD**	**Pot MTMDD**	**Act MTMDD**
qn4	2	324	324	333	14,256	772	586	722	22,784	22,784
	6	38,416	38,416	499	2,524,480	3,092	2,494	2,870	36,864	36,864
	10	527,076	527,076	905	38,524,464	7,076	5,778	6,522	62,720	62,720
qn8	2	6,561	324	681	14,256	1,204	738	1,688	43,776	49,152
	6	5,764,801	38,416	1,119	2,524,480	2,404	1,674	5,872	55,040	70,912
	10	214,358,881	527,076	1,953	38,524,464	3,604	2,610	12,040	66,304	98,560
mserv2	3	1,485	495	705	23,352	4,124	3,246	3,952	34,560	40,704
	6	6,345	2,115	3,176	111,408	17,468	13,998	16,432	111,104	135,168
	10	18,495	6,165	8,846	342,720	52,228	42,278	49,032	306,560	378,460
mserv4	3	14,256	495	1,174	23,352	5,568	4,098	4,916	68,864	79,616
	6	106,596	2,115	8,453	111,408	22,920	17,502	20,054	254,360	298,856
	10	488,268	6,165	33,739	342,720	67,560	52,342	58,934	873,896	998,552
mserv6	3	32,076	495	1,333	23,352	5,724	4,066	5,316	86,784	101,376
	6	239,841	2,115	8,614	111,408	23,076	17,470	20,238	298,596	347,956
	10	1,098,603	6,165	33,900	342,720	67,716	52,310	59,118	982,396	1,112,684

Model	N	$\lvert\mathcal{X}_{pot}\rvert$	$\lvert\mathcal{X}_{rch}\rvert$	**MDD**	**Sparse**	**Kron**	**Pot MxD**	**Act MxD**	**Pot MTMDD**	**Act MTMDD**
molloy4	5	4,536	91	660	4,204	1,316	1,148	2,534	23,552	28,160
	8	32,805	285	1,215	14,676	2,528	2,300	5,216	27,648	38,656
	10	87,846	506	1,766	27,104	3,556	3,288	7,504	31,232	47,360
molloy5	5	7,776	91	846	4,204	1,100	792	4,298	28,416	37,120
	8	59,049	285	1,545	14,676	1,592	1,188	9,356	31,232	50,944
	10	161,051	506	2,223	27,104	1,920	1,452	13,778	33,280	61,952
kan3	1	160	160	264	8,032	500	412	544	18,432	18,432
	3	58,400	58,400	937	5,590,400	7,572	6,786	8,134	66,816	67,072
	5	2,546,432	2,546,432	5,646	303,705,920	45,660	41,816	48,780	303,776	303,776
kan4	1	256	160	332	8,032	420	354	602	23,552	24,576
	3	160,000	58,400	628	5,590,400	2,500	2,216	3,284	44,032	50,176
	5	9,834,496	2,546,432	1,532	303,705,920	7,940	7,118	9,950	92,928	110,592
kan16	1	65,536	160	1,275	8,032	2,148	866	3,000	95,232	107,520
	3	—	58,400	1,902	5,590,400	3,236	1,746	10,566	115,456	151,808
	5	—	2,546,432	3,149	303,705,920	4,324	2,626	24,106	135,168	216,320
fms5	1	2,100	84	535	3,228	1,456	604	1,808	36,096	40,960
	3	9,432,500	20,600	3,294	1,554,080	8,304	5,224	24,320	151,296	247,040
	5	2,016,379,008	852,012	30,490	82,727,748	34,484	24,664	138,244	654,892	1,255,108
fms21	1	4,194,304	84	2,050	3,228	3,132	1,132	7,396	126,976	148,224
	3	—	20,600	6,777	1,554,080	5,028	2,328	68,762	176,896	437,760
	5	—	852,012	22,038	82,727,748	6,924	3,524	255,988	235,008	1,393,932

A (quasi-reduced) EV*MDD on $\mathbf{x} = (x_L, ..., x_1)$ is a directed acyclic edge-labeled multi-graph:

- Ω is the only terminal node — $\Omega.var = x_0$
- A nonterminal node p is associated with a variable $p.var = x_k$, $k \in \{L, ..., 1\}$ — $\mathcal{X}_p = \mathcal{X}_k$
- and has an edge for each $i \in \mathcal{X}_p$, associated with a value in $[0,1]$ — $p[i] = \langle \rho, q \rangle = \langle p[i].v, p[i].d \rangle$
- If $p.var = x_k$, then $q.var = x_{k-1}$ or $q = \Omega$ and $\rho = 0$ — $\max_{i \in \mathcal{X}_p} p[i].v = 1$
- There are no duplicates: if $p.var = q.var$ and $p[i] = q[i]$ for all $i \in \mathcal{X}_p$, then $p = q$

The node reached from p through $\alpha = (i_k, i_{k-1}, ..., i_h) \in \mathcal{X}_k \times \cdots \times \mathcal{X}_h$, for $L \geq k \geq h \geq 1$, is

$$p[\alpha].d = \begin{cases} (p[i_k].d)[i_{k-1}, ..., i_h].d & \text{if } p[i_k].d \neq \Omega \\ \Omega & \text{otherwise} \end{cases}$$

and the value associated with this path is

$$p[\alpha].v = \begin{cases} p[i_k].v \cdot (p[i_k].d)[i_{k-1}, ..., i_h].v & \text{if } p[i_k].d \neq \Omega \\ p[i_k].v & \text{otherwise} \end{cases}$$

Edge $\langle \rho, p \rangle$ with $p.var = x_k$ encodes function $f(\alpha) = \rho \cdot p[\alpha].v$, for $\alpha \in \mathcal{X}_k \times \cdots \times \mathcal{X}_1$

In particular, edge $\langle \rho, \Omega \rangle$ encodes the constant ρ

Since $\max_{i \in \mathcal{X}_p} p[i].v = 1$ for each node p, we have that $\rho = \max(f)$ — $\rho \in \mathbb{R}^{\geq 0}$

EV*MDDs can canonically encode functions $f : \mathcal{X}_L \times \cdots \times \mathcal{X}_1 \to \mathbb{R}^{\geq 0}$

EV*MDDs by example

One way to think about EV*MDDs is "EV$^+$MDD $= -\log($EV*MDD$)$":

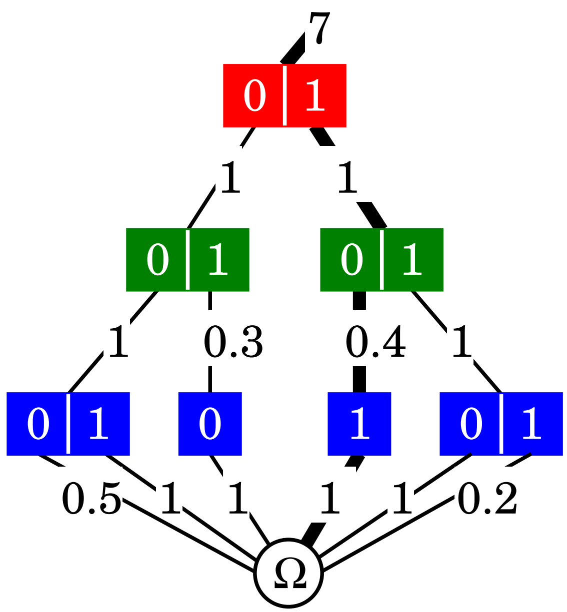
$0 \Leftrightarrow 1$

edge values $\in [0, \infty^+] \Leftrightarrow$ edge values $\in [0, 1]$

root incoming edge $\in (\infty^-, \infty^+] \Leftrightarrow$ root incoming edge $\in [0, \infty^+)$

values add along the path $\Leftrightarrow$ values multiply along the path

$\mathbf{i}_3$	0	0	0	0	1	1	1	1
$\mathbf{i}_2$	0	0	1	1	0	0	1	1
$\mathbf{i}_1$	0	1	0	1	0	1	0	1
f	3.5	7	2.1	0	0	2.8	7	1.4



Canonicity: all edge values leaving a node are in $[0, 1]$ and at least one is 1

In canonical form, the root incoming edge has value $\max_{\mathbf{i} \in \mathcal{X}_{pot}} f(\mathbf{i})$

Conclusion

EV$^+$MDDs are ideal to store the indexing function $\psi_{rch} : \mathcal{X}_{pot} \to \{0, 1, ..., |\mathcal{X}_{rch}| - 1\} \cup \{\text{null}\}$

- The EV$^+$MDD storing ψ_{rch} is isomorphic to the MDD storing the state space $\mathcal{X}_{rch}$

EV*MDDs are likely the best choice to store the transition rate matrix of a structured CTMC model

- They are completely general (like MTMDDs, unlike Kronecker)
- They can exploit locality in the high-level model (unlike ordinary MTMDDs, like Kronecker)
- They can be exponentially more compact than Kronecker and MTMDDs (for different reasons)
- They have less than $\times 2$ memory overhead w.r.t. Kronecker or MTMDDs in the worst case
- They are very similar to Matrix Diagrams when encoding matrices but, unlike Matrix Diagrams, identical rows in a node (or even in different nodes at the same level) are not stored multiple times
- They suggest an interesting approximation algorithm
  Approximate steady-state analysis of large Markov models based on the structure of their decision diagram encoding
  Performance Evauation, v. 68, p. 463-486, 2011 (another talk...)
- They can approach the time efficiency of explicit sparse matrices for vector-matrix multiplication
  A two-phase Gauss-Seidel algorithm for the stationary solution of EVMDD-encoded CTMCs
  accepted at *QEST 2012* (another talk...)

End