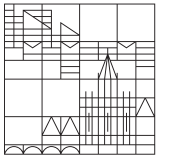
Universität Konstanz



Thesis Proposal Summary

Fabian Klopfer

Modelling of Complex Self-Organizing Systems Group, October 14, 2020

Introduction

In the last presentation . . .

- introduced CRNs & ODEs
- looked at Tribastone et al. [1]
- compared the algorithms informally
- tried to find representations between CRNs & WAs

What we discussed then

- Look at classical lumping rather than CRN species lumping
- Construct “middle ground” between WAs & CTMCs using fully probabilistic Segala
- Elaborate on connection between Kiefer [2] and partition refinement-based lumping e.g. Valmari et al. [3]

Schützenberger's construction & Bisimulation I

Schützenberger [4] vs. Buchholz [5]:

$$\overrightarrow{\mu}(\sigma) = \overrightarrow{F}\mu(\sigma)\overrightarrow{F}_R^{-1} \qquad \overrightarrow{F}\mu(\sigma) = \overrightarrow{\mu}(\sigma)\overrightarrow{F} \tag{1}$$

$$\hat{P} = WPV \qquad WP = \hat{P}V_R^{-1} \tag{2}$$

With $W \cdot V = I$ and P the transition matrix.

$$W \cdot V = I$$
$$W \cdot V \cdot V_R^{-1} = I \cdot V_R^{-1}$$
$$W = V_R^{-1}$$

Schützenberger's construction & Bisimulation II

Thus we get

$$\overrightarrow{\mu}(\sigma) = \overrightarrow{F}\,\mu(\sigma)\,\overrightarrow{F}_R^{-1} \qquad \overrightarrow{F}\,\mu(\sigma) = \overrightarrow{\mu}(\sigma)\,\overrightarrow{F} \tag{3}$$

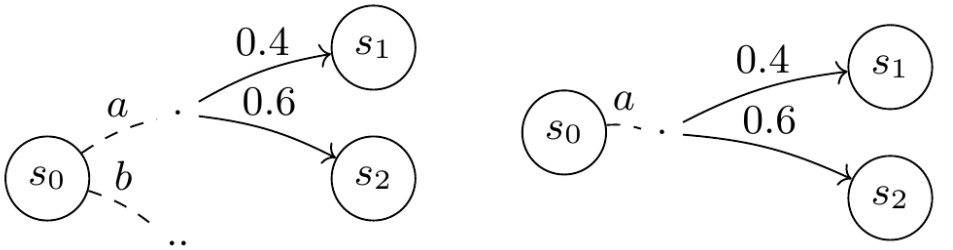
$$\hat{P} = WPW_R^{-1} \qquad WP = \hat{P}W \tag{4}$$

WA $\leftrightarrow$ labelled CTMC I

WA $\leftrightarrow$ CTMC: A labelled CTMC is already a WA.
scale WA by row sum per row and letter to get (sort of) a labeled DTMC which can be interpreted as uniformized labelled CTMC (sort of).

WA $\leftrightarrow$ labelled CTMC II

Problem: Only one labelled transition possible per state.
What we get is actually not a MC but rather a deterministic Rabin PA.

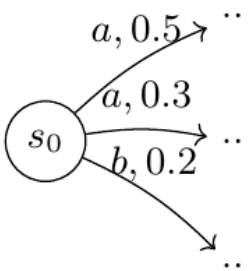


WA $\leftrightarrow$ labelled CTMC III

$\Rightarrow$ A **WA** may be normalized by left. mult. vector of sum of row sums, yielding a **f.p. Segala**, i.e. the row sum of all matrices together equals 1.
Drop the labels of the **f.p. Segala** and add the matrices together to get a **DTMC**, which can be viewed as **uniformized CTMC**.

$\Leftarrow$ A **CTMC** can be uniformized to a **DTMC**. Scaling each probability by the number of letters yields a **f.p. Segala**, which is already a **WA**

WA $\leftrightarrow$ labelled CTMC IV



Problem: Paths & Probabilities are preserved but what are the labels?

WA $\leftrightarrow$ labelled CTMC V

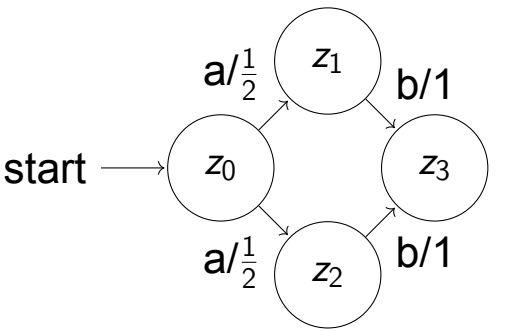
Labelled CTMC is already a WA.
WA $\rightarrow$ CTMC: Resort to the following:

1. Generate the language, i.e. all weights with all words
2. for every word build a LCTMC with the adequate labels and all transition probabilities set to 1.
3. sum the weights for all words.
4. connect all generated LCTMCs to an extra state, add an extra empty label transition matrix with transitions from extra state to respective word with probability $\frac{\text{word weight}}{\text{weight sum}}$

Connection between the algorithms I

Weighted automaton $\mathcal{A} = (n, \Sigma, \mu, \alpha, \eta)$ with

- $n = 3$,
- $\Sigma = \{a, b\}$,
- $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$,
- $\eta = (\eta_0, \eta_1, \eta_2, \eta_3)$,
- $\mu(a) = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$,
- $\mu(b) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$.



Connection between the algorithms II

$$\mu(a)^2 = (0); \quad \mu(b)^2 = (0); \quad \mu(b)\mu(a) = 0$$

$\Rightarrow$ Words to consider: a, b, ab

$$r^{(i)} = \begin{pmatrix} r_{a,0}^{(i)} & r_{a,1}^{(i)} & r_{a,2}^{(i)} & r_{a,3}^{(i)} \\ r_{b,0}^{(i)} & r_{b,1}^{(i)} & r_{b,2}^{(i)} & r_{b,3}^{(i)} \end{pmatrix}$$

$$v_i = \alpha\mu(a)r_a^{(i)} + \alpha\mu(b)r_b^{(i)} + \alpha\mu(a)\mu(b)r_{ab}^{(i)}$$

$$= \left(0, \frac{1}{2}\alpha_0, \frac{1}{2}\alpha_0, 0\right) r_a^{(i)} + (0, 0, 0, \alpha_1 + \alpha_2)\, r_b^{(i)} + (0, 0, 0, \alpha_0)\, r_{ab}^{(i)}$$

$$= \left(0, \frac{1}{2}r_{a,0}^{(i)}\alpha_0, \frac{1}{2}r_{a,0}^{(i)}\alpha_0, r_{b,0}^{(i)}[\alpha_1 + \alpha_2] + r_{a,0}^{(i)}r_{b,1}^{(i)}[\alpha_0]\right)$$

Connection between the algorithms III

$$\overrightarrow{F} = \begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ 0 & \frac{1}{2} r_{a,0}^{(1)} \alpha_0 & \frac{1}{2} r_{a,0}^{(1)} \alpha_0 & r_{b,0}^{(1)} [\alpha_1 + \alpha_2] + r_{a,0}^{(1)} r_{b,1}^{(1)} [\alpha_0] \\ 0 & \frac{1}{2} r_{a,0}^{(2)} \alpha_0 & \frac{1}{2} r_{a,0}^{(2)} \alpha_0 & r_{b,0}^{(2)} [\alpha_1 + \alpha_2] + r_{a,0}^{(2)} r_{b,1}^{(2)} [\alpha_0] \end{pmatrix}$$

Using partition refinement we get $\{s_1, s_2\}$, thus

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Connection between the algorithms IV

Chose $\alpha = (1, 0, 0, 0)$.
Now Kiefer restricts the $r^{(i)}_{\sigma,k}$ to $\mathbb{N}_+$.
Resulting in that it is not possible to find parameters such that $\overrightarrow{F} = W$.
Even when we loosen that restriction, chose $r^{(1)}_{a,0} = 1, r^{(1)}_{b,0} = r^{(1)}_{b,1} = 0, r^{(2)}_{a,0} = 0, r^{(2)}_{b,0} = 1$.
Still impossible as $\overrightarrow{F}\, i = j = 41$.

Summary I

- Kiefers et al. [2] basis construction not able to produce the same basis as partition refinement
- yields non-sparse transition matrices
- induces computational overhead, possibly more overhead than gaining by dimensionality reduction
- Alternative: Use Householder reflectors for finding the basis [6], yield potentially sparser transition matrices while numerically stable wo. hacks
- Partition-refinement does **not** explicitly construct a basis.
- rather constructs new transition matrix than transforming old one

Summary II

LUMPCTMC(P, Q)

1 $L :=$ blocks of P
2 while $L \neq \emptyset$
3 $\quad S :=$ POP(L)
4 $\quad$ SPLIT(S, P, L)
5 $n' :=$ # of blocks in P
6 allocate $n' \times n'$ matrix Q'
7 initialize Q' to zero
8 for every block B_k of P
9 $\quad x_i :=$ arbitrary state in B_k
10 $\quad$ for every x_j such that $x_i \to x_j$
11 $\quad\quad$ Let B_l be $[x_j]_P$
12 $\quad\quad Q'(B_k, B_l) := Q'(B_k, B_l) + Q(x_i, x_j)$
13 return Q'

Algorithm 1. Pseudocode of the lumping algorithm.

$Q'(B_k, B_l)$, the rate from B_k to B_l in the quotient chain, is $q(x_i, B_l) = \sum_{x_j \in B_l} Q(x_i, x_j)$, where x_i is an

SPLIT(S, P, L)

1 $L', L'' := \emptyset$
2 for every $x_j \in S$
3 $\quad$ for every $x_i \to x_j$
4 $\quad\quad x_i.\text{sum} := 0$
5 for every $x_j \in S$
6 $\quad$ for every $x_i \to x_j$
7 $\quad\quad x_i.\text{sum} := x_i.\text{sum} + Q(x_i, x_j)$
8 $\quad\quad L' := L' \cup \{x_i\}$
9 for each $x_i \in L'$
10 $\quad B :=$ block of x_i
11 $\quad$ delete x_i from B
12 $\quad$ INSERT(B_T, x_i)
13 $\quad$ if $B \notin L''$ add B to L''
14 for every $B \in L''$
15 $\quad B_l :=$ largest block of $\{B, V_{k_1}, \ldots, V_{k_{|B_T|}}\}$
16 $\quad L := L \cup \{B, V_{k_1}, \ldots, V_{k_{|B_T|}}\} - \{B_l\}$

Algorithm 2. Pseudocode of SPLIT procedure.

Bibliography

L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, “Maximal aggregation of polynomial dynamical systems,” Proceedings of the National Academy of Sciences of the United States of America, vol. 114 38, pp. 10 029–10 034, 2017.

S. Kiefer, A. S. Murawski, J. Ouaknine, B. Wachter, and J. Worrell, “On the complexity of equivalence and minimisation for q-weighted automata,” Logical Methods in Computer Science, vol. 9, 2013.

A. Valmari and G. Franceschinis, “Simple o (m logn) time markov chain lumping,” in International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2010, pp. 38–52.

M. P. Schützenberger, “On the definition of a family of automata,” Information and control, vol. 4, no. 2-3, pp. 245–270, 1961.

P. Buchholz, “Bisimulation relations for weighted automata,” Theoretical Computer Science, vol. 393, no. 1-3, pp. 109–123, 2008.

S. Kiefer and B. Wachter, “Stability and complexity of minimising probabilistic automata,” in International Colloquium on Automata, Languages, and Programming, Springer, 2014, pp. 268–279.