A Multilevel Algorithm based on Binary Decision Diagrams

Johann Schuster

Department of Computer Science
University of the Federal Armed Forces Munich

14.11.2007

Outline of the talk

- Introduction
- Multilevel algorithm
- Symbolic encoding
- Adapted data structures
- Symbolic algorithm
- Results and conlcusion

Task

Calculate a steady state distribution of a time homogeneous continuous time Markov chain (CTMC) $\mathcal{M}$ with generator matrix Q, i.e. solve

$$\begin{aligned} \vec{\pi} \cdot Q &= 0 \\ \textstyle\sum_i \pi_i &= 1 \end{aligned} \quad (1)$$

Problem

Direct solution infeasible, numerical solution often slow.

Universität der Bundeswehr München

Task

Calculate a steady state distribution of a time homogeneous continuous time Markov chain (CTMC) $\mathcal{M}$ with generator matrix Q, i.e. solve

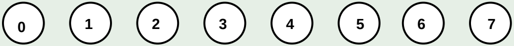
$$\begin{aligned} \vec{\pi} \cdot Q &= 0 \\ \sum_i \pi_i &= 1 \end{aligned} \tag{1}$$

Problem

Direct solution infeasible, numerical solution often slow.

Grouping states

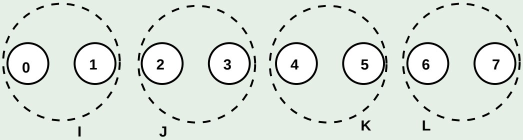
Example



System l_0

Grouping states

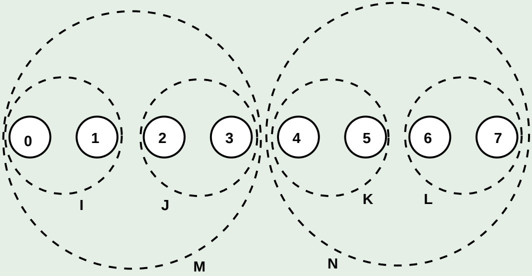
Example



System $l_0 - 1$

Grouping states

Example

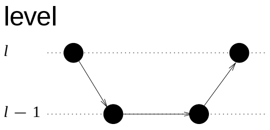


System $l_0 - 2$

The algorithm (I)

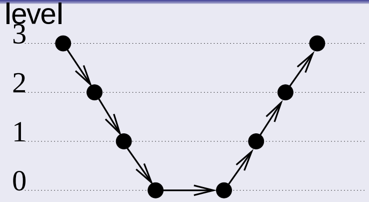
A two-level step consists of the following parts

- Aggregation: From the rate matrix and current approximation of the solution vector on a certain level (l) derive a smaller system $(l-1)$
- Solution: Solve the smaller system $(l-1)$ (i.e. directly or by another multilevel step)
- Disaggregation: Correct the solution of system (l) by the solution of system $(l-1)$



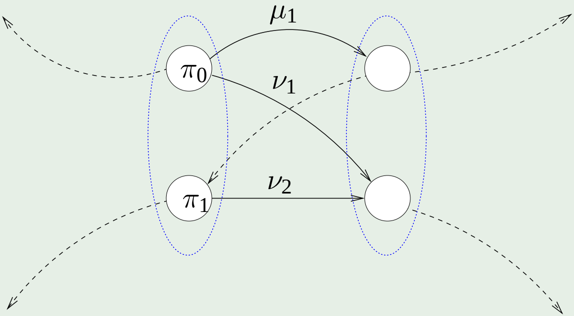
Multilevel algorithm - V-cycle

Basic V-cycle



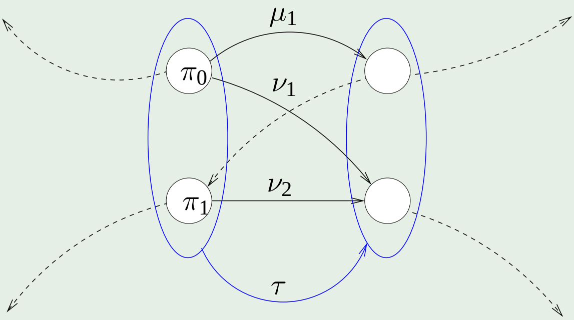
Aggregation - idea

Example



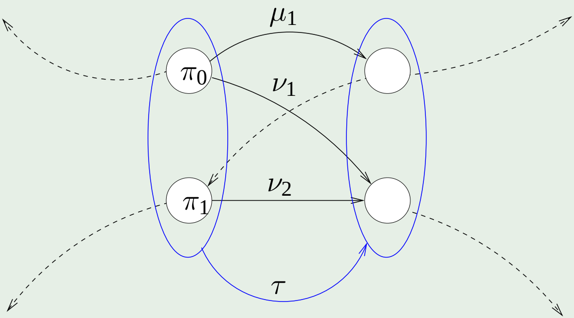
Aggregation - idea

Example



Aggregation - idea

Example



$$\tau = \frac{\pi_0 \cdot (\mu_1 + \nu_1) + \pi_1 \cdot \nu_2}{\pi_0 + \pi_1}$$

Aggregation - continued

Aggregation equations

Partitioning a statespace of n states $\{i, j, \ldots\}$ into N macro states $I, J, \ldots$ leads to

$$q_{IJ} = \frac{\sum_{i \in I} \pi_i \sum_{j \in J} q_{ij}}{\sum_{i \in I} \pi_i} \tag{2}$$

Disaggregation

Disaggregation equations

The Disaggregation step scales all the fine states $i \in I$ with the same factor:

$$\pi_i^{new} = \frac{\vec{\pi}_I^{agg,new}}{\vec{\pi}_I^{agg}} \cdot \vec{\pi}_i$$

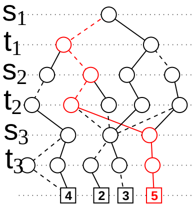
Universität der Bundeswehr München

Symbolic encoding of a Matrix

Store nonzero entries as bitstrings with weights

$s_1s_2s_3$ \ $t_1t_2t_3$	000	001	010	011	100	101	110	111
000		2					4	
001	3			5				4
010		2						
011	3							
100								
101								
110		2						2
111	3			5			3	

row 0 0 1
column 0 1 1
bitstring 000111

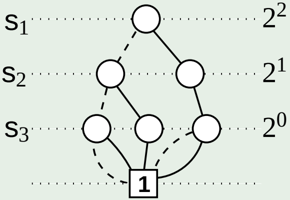


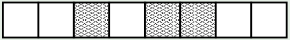
Example path: (001,011)=5, resulting bitstring: 000111

Reachability

Symbolic reachability analysis leads to Binary Decision Diagram (BDD) *reach* encoding reachable states

Example



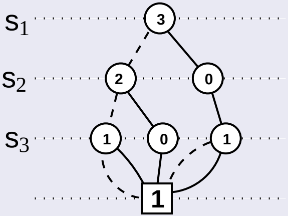


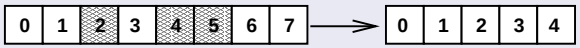
Reachablility BDD mapping a potential statespace of $2^3 = 8$ states to 5 reachable states

Offset-Labelling

- The mapping from potential state space to reachable statespace is stored in *reach* as offset information [Parker 02]

offset labelled BDD



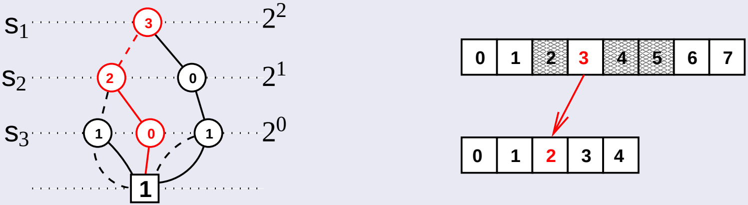


mapping potential to reachable states

offset labelled BDD *reach*

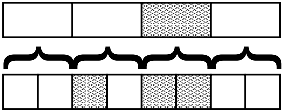
Offset labelling example path

- potential state index: $2^1 + 2^0 = 3$
- reachable state index (=offset): $2 + 0 = 2$

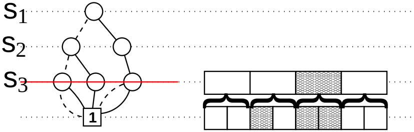


Aggregation of neighbouring states

Neighbouring states are grouped to macro states



The same can be achieved by aggregation of the lowest BDD variable in *reach*.



- Introduction
- Multilevel algorithm
- Symbolic encoding
- *Adapted data structures*
- Symbolic algorithm
- Results and conlcusion

Symbolic multilevel ingredients

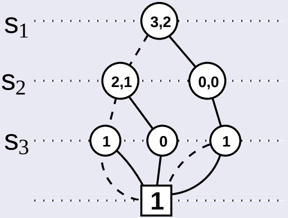
Two concepts are needed to perform the symbolic multilevel algorithm

- Multi-offset-labelling to compactly store the iteration vectors
- Compact storage of aggregated transition matrices
- Speed-up by intermediate sparse matrices

Symbolic aggregation basics (I)

Multi-offset-labelling is introduced that describes the reachable state space for each aggregated system

Multi-offsets for aggregation of variable s_3



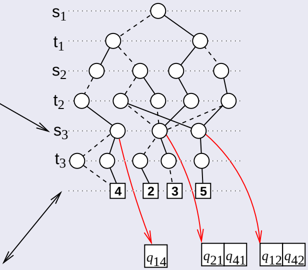
Symbolic aggregation basics (II)

Aggregated transition matrix entries are stored at the row nodes of the corresponding aggregation variable.

Matrix storage (aggregation at s_3)

	q_{12}		q_{14}
q_{21}			
q_{41}	q_{42}		

	2					4	
3			5				4
	2						
3							
	2						2
3			5			3	

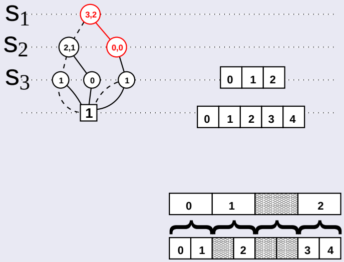


Diagonal elements are stored in a separate vector.

- Introduction
- Multilevel algorithm
- Symbolic encoding
- Adapted data structures
- *Symbolic algorithm*
- Results and conlcusion

Aggregation, example path

section of Depth first traversal (DFS), aggregation at s_3



$$\pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$$

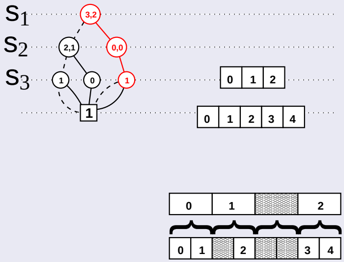
s-Offsets:

$$\begin{aligned} \mathit{coarse} &= 2 \\ \mathit{fine} &= 3 \end{aligned}$$

$$\bar{\pi} = (\pi_0 + \pi_1, \pi_2, 0)$$

Aggregation, example path

section of Depth first traversal (DFS), aggregation at s_3



$$\pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$$

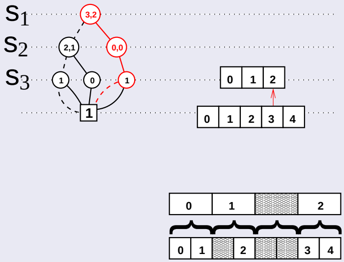
s-Offsets:

$$\begin{aligned} coarse &= 2 \\ fine &= 3 \end{aligned}$$

$$\bar{\pi} = (\pi_0 + \pi_1, \pi_2, 0)$$

Aggregation, example path

section of Depth first traversal (DFS), aggregation at s_3



$$\pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$$

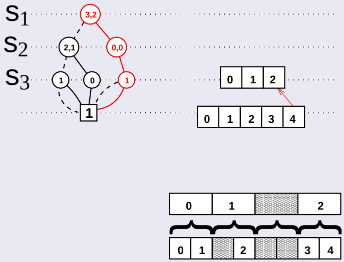
s-Offsets:

$$\begin{aligned} coarse &= 2 \\ fine &= 3 \end{aligned}$$

$$\bar{\pi} = (\pi_0 + \pi_1, \pi_2, \pi_3)$$

Universität der Bundeswehr München

section of Depth first traversal (DFS), aggregation at s_3



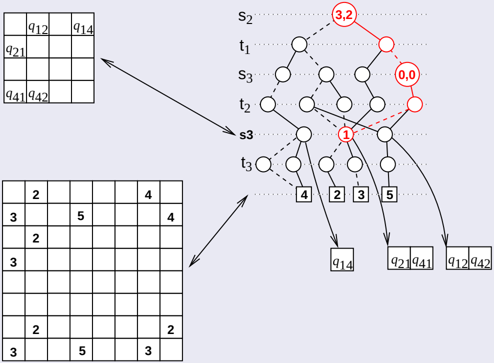
$$\pi = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$$

s-Offsets:

$$\begin{aligned} \textit{coarse} &= 2 \\ \textit{fine} &= 4 \end{aligned}$$

$$\bar{\pi} = (\pi_0 + \pi_1, \pi_2, \pi_3 + \pi_4)$$

Symbolic aggregation



Let
$\vec{\pi} = (\pi_0, \pi_1, \pi_2, \pi_3, \pi_4)$
Processing the last red node:

$$q_{41} = 2\pi_3 + 3\pi_4$$

Dividing by the coarse vector element with offset 2 leads to:

$$q_{41} = \frac{2\pi_3 + 3\pi_4}{\pi_3 + \pi_4}$$

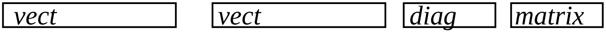
By an iterative solver or another multilevel step get a solution $\vec{\pi}^{agg,new}$ of the aggregated system.

The Disaggregation step scales all the fine states $i \in I$ with the factor:

$$\pi_i^{new} = \frac{\vec{\pi}_I^{agg,new}}{\vec{\pi}_I^{agg}} \cdot \vec{\pi}_i$$

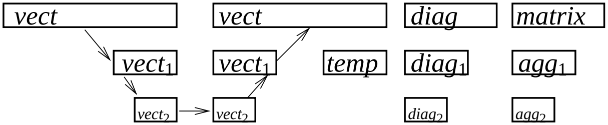
This can be done in a depth first traversal of the reachability graph similar to the aggregation

Memory considerations



Memory for a basic Jacobi iteration

Memory considerations

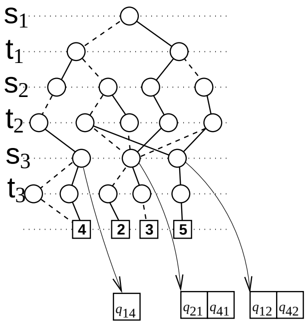


Multilevel overhead

- $matrix + diag + vect + max(vect, agg) + sm$
- $agg_i = diag_i + 2 * vect_i + rates$
- $agg = \sum_{i \in agg\_level} agg_i + temp$

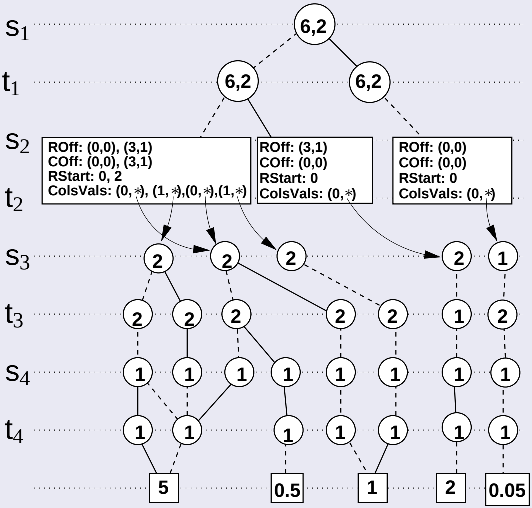
Improving speed - adding sparse matrices

Up to lowest aggregation level sparse matrices can be used [Parker 02]



otherwise intermediate sparse matrices have to be used

intermediate sparse matrices



- Introduction
- Multilevel algorithm
- Symbolic encoding
- Adapted data structures
- Symbolic algorithm
- *Results and conlcusion*

Conclusion

- Framework for symbolic multilevel solvers is given
- Solution times depend on the model
- Current speedup up to 2
- Memory requirements depend on position and number of aggregation levels

Future work

- Try different orderings of the aggregated systems
- Parallelise parts of the algorithm

P. Buchholz and T. Dayar. *Comparison of multilevel methods for kronecker-based markovian representations*. Computing, 73(4):349-371, 2004

G. Horton and S. Leutenegger. *A Multi-Level Solution Algorithm for Steady-State Markov Chains*. ACM Performance Evaluation Review, 22(1):191-200, May 1994. Proceedings of the ACM Sigmetrics and Performance 1994, International Conference on Measurement and Modeling of Computer Systems

D. Parker. *Implementation of symbolic model checking for probabilistic systems*. PhD thesis, School of Computer Science, Faculty of Science, University of Birmingham, 2002.