Back to Basics: Homogeneous Representations of Multi-Rate Synchronous Dataflow Graphs

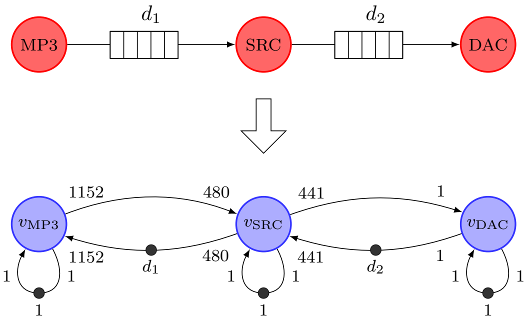
Robert de Groote, Philip Hölzenspies, Jan Kuper, and Hajo Broersma

UNIVERSITY OF TWENTE.

Computer Architectures for Embedded Systems Group
Dept. of Electrical Engineering, Mathematics and Computer Science
University of Twente, Enschede, The Netherlands
http://caes.ewi.utwente.nl

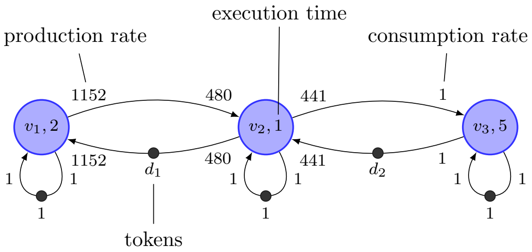
MEMOCODE 2013

Multi-Rate Synchronous Dataflow Graphs (1/3)



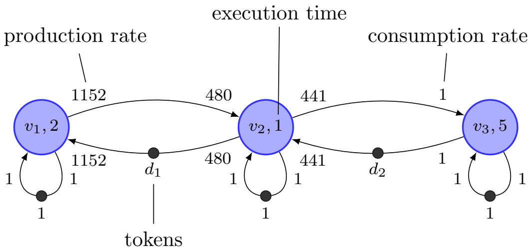
- Capture task graphs
- Potential parallelism and interactions explicit
- Well suited for modelling DSP applications
- Annotations for analysis

Multi-Rate Synchronous Dataflow Graphs (2/3)



- Rates, auto-concurrency
- Consistency, Iteration, Periodicity
- Homogeneous, Cyclo-Static, Scenario-Aware, ...

Multi-Rate Synchronous Dataflow Graphs (3/3)



Throughput Analysis

- (Average) number of graph iterations per time unit
- Find critical cycle

Buffer Analysis

- Determine buffer capacities required for minimal throughput
- Make all cycles equally critical

MRSDF Graphs - Exact Analysis (1/3)

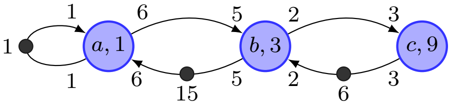
Throughput Analysis

- Algorithms available for Homogeneous SDF Graphs (marked graphs)
- Transform MRSDF graph into HSDF graph
- Transformation described in [1], [2], ...

[1] Lee, Edward A., and David G. Messerschmitt. "Synchronous data flow." Proceedings of the IEEE 75.9 (1987): 1235-1245.

[2] Sriram, Sundararajan, and Shuvra S. Bhattacharyya. Embedded multiprocessors: Scheduling and synchronization. CRC press, 2009.

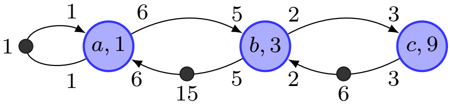
MRSDF Graphs - Exact Analysis (2/3)



MRSDF to HSDF Transformation

- Represent individual firings in an iteration

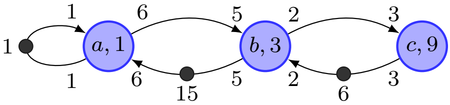
MRSDF Graphs - Exact Analysis (2/3)



MRSDF to HSDF Transformation

- Represent individual firings in an iteration
- Represent each token by a single edge

MRSDF Graphs - Exact Analysis (2/3)

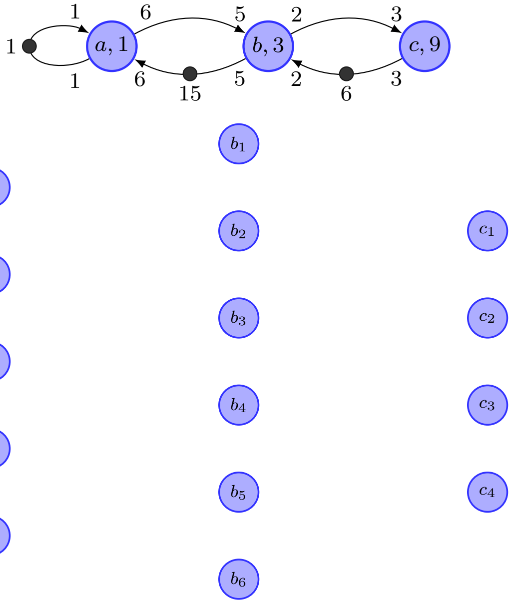


MRSDF to HSDF Transformation

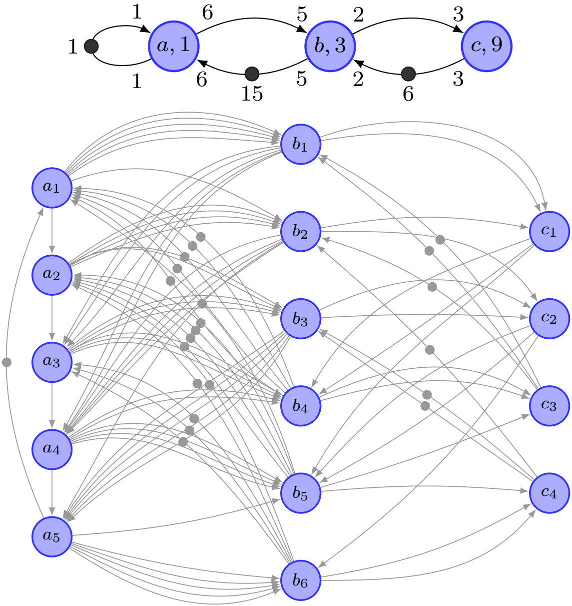
- Represent individual firings in an iteration
- Represent each token by a single edge

Analysis: compute critical cycle (MCR)

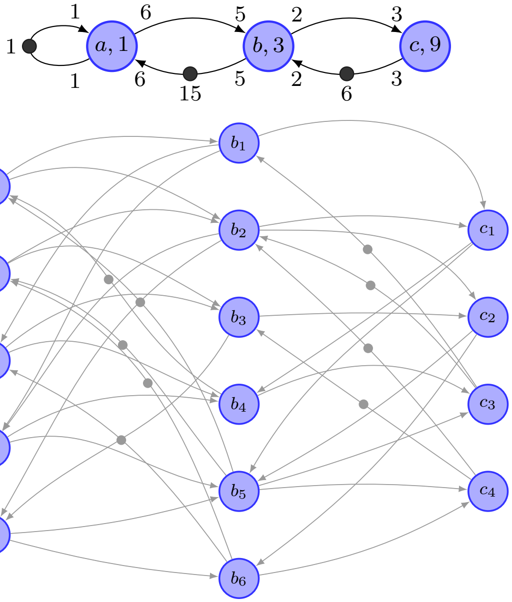
MRSDF Graphs - Exact Analysis (2/3)



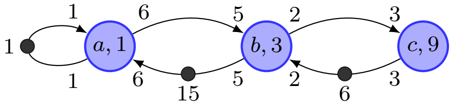
MRSDF Graphs - Exact Analysis (2/3)

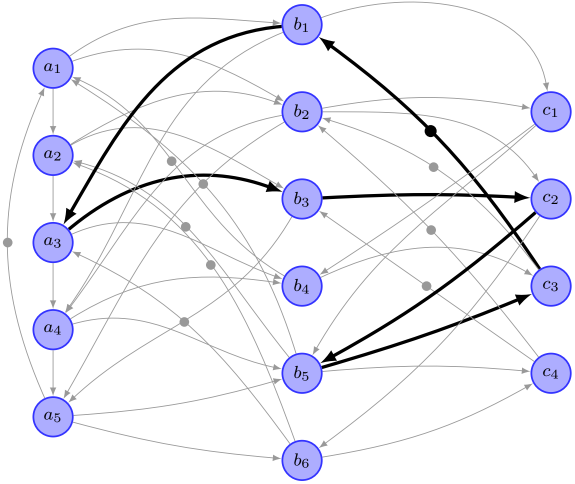


MRSDF Graphs - Exact Analysis (2/3)



MRSDF Graphs - Exact Analysis (2/3)





MRSDF Graphs - Exact Analysis (3/3)

HSDF-based approach abandoned due to high complexity

- State-Space Exploration used instead [1]

[1] A. H. Ghamarian, M. C. W. Geilen, S. Stuijk, T. Basten, B. D. Theelen, M. R. Mousavi, A. J. M. Moonen, and M. J. G. Bekooij, “Throughput Analysis of Synchronous Data Flow Graphs,” *ACSD*, 2006.

MRSDF Graphs - Exact Analysis (3/3)

HSDF-based approach abandoned due to high complexity

- State-Space Exploration used instead [1]

Exact Analysis: costly, but useful?

[1] A. H. Ghamarian, M. C. W. Geilen, S. Stuijk, T. Basten, B. D. Theelen, M. R. Mousavi, A. J. M. Moonen, and M. J. G. Bekooij, "Throughput Analysis of Synchronous Data Flow Graphs," *ACSD*, 2006.

MRSDF Graphs - Exact Analysis (3/3)

HSDF-based approach abandoned due to high complexity

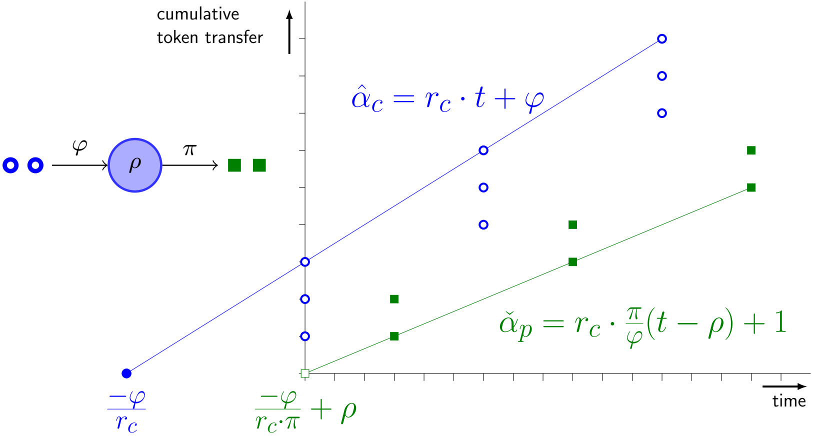
- State-Space Exploration used instead [1]

Exact Analysis: costly, but useful?

- Only need guarantees

[1] A. H. Ghamarian, M. C. W. Geilen, S. Stuijk, T. Basten, B. D. Theelen, M. R. Mousavi, A. J. M. Moonen, and M. J. G. Bekooij, "Throughput Analysis of Synchronous Data Flow Graphs," *ACSD*, 2006.

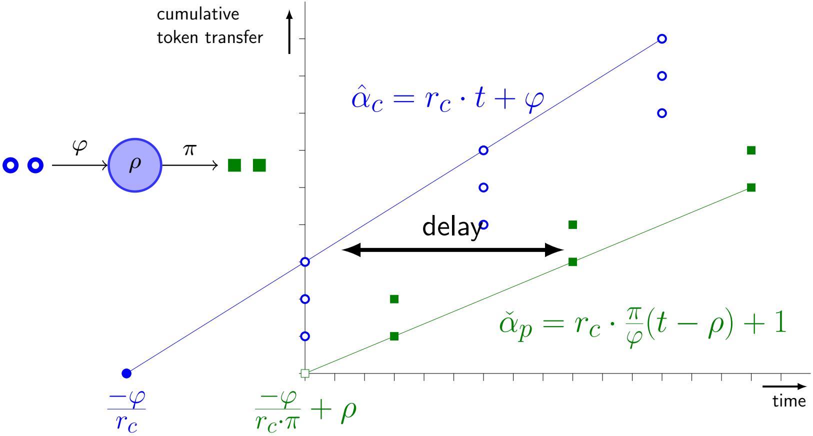
MRSDF Graphs - Conservative Analysis (1/2)



Construct linear bounds:

- Upper bound on token consumption times: $\hat{\alpha}_c$
- Lower bound on token production times: $\check{\alpha}_p$

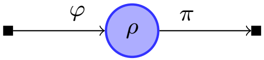
MRSDF Graphs - Conservative Analysis (1/2)

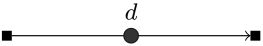


Construct linear bounds:

- Upper bound on token consumption times: $\hat{\alpha}_c$
- Lower bound on token production times: $\check{\alpha}_p$

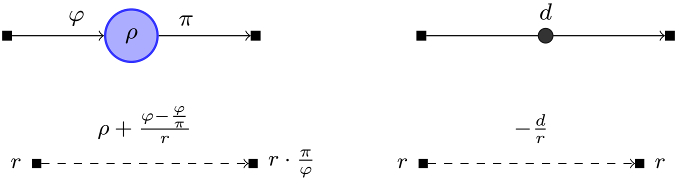
SDF Graphs - Conservative Analysis (2/2)





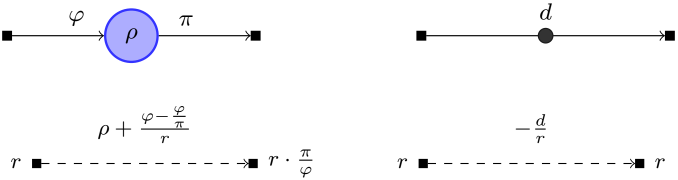
Hausmans, J.P.H.M., et al. "Compositional temporal analysis model for incremental hard real-time system design." Proceedings of the tenth ACM international conference on Embedded software (EMSOFT). ACM, 2012.

SDF Graphs - Conservative Analysis (2/2)



Hausmans, J.P.H.M., et al. "Compositional temporal analysis model for incremental hard real-time system design." Proceedings of the tenth ACM international conference on Embedded software (EMSOFT). ACM, 2012.

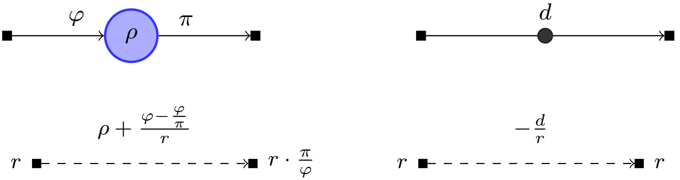
SDF Graphs - Conservative Analysis (2/2)



Translate each actor and channel into an edge (i, j):

- γ: Transfer rate ratio
- ϵ: Rate-independent delay
- δ: Rate-dependent delay

Hausmans, J.P.H.M., et al. "Compositional temporal analysis model for incremental hard real-time system design." Proceedings of the tenth ACM international conference on Embedded software (EMSOFT). ACM, 2012.

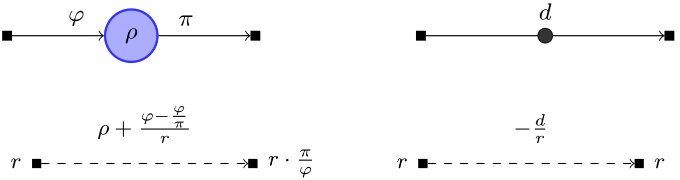


Translate each actor and channel into an edge (i, j):

- γ: Transfer rate ratio
- ϵ: Rate-independent delay
- δ: Rate-dependent delay
- s: Firing start time
- Compute maximum rate, r

Hausmans, J.P.H.M., et al. "Compositional temporal analysis model for incremental hard real-time system design." Proceedings of the tenth ACM international conference on Embedded software (EMSOFT). ACM, 2012.

SDF Graphs - Conservative Analysis (2/2)



Translate each actor and channel into an edge (i, j):

- γ: Transfer rate ratio
- ϵ: Rate-independent delay
- δ: Rate-dependent delay
- s: Firing start time
- Compute maximum rate, r

maximize r

s.t. $s(j) \geq s(i) + \epsilon(i, j) + \frac{\delta(i,j)}{r(i)}$

$r(j) = \gamma(i, j) \cdot r(i)$

Hausmans, J.P.H.M., et al. "Compositional temporal analysis model for incremental hard real-time system design." Proceedings of the tenth ACM international conference on Embedded software (EMSOFT). ACM, 2012.

Back to Basics

Existing exact analysis of MRSDF graphs

- Data-driven transformation into HSDF
- Redundancy in resulting HSDF

Existing approximate analysis

- No upper bound on rate - no sense of error
- Opaque solution from an LP

Back to Basics

Existing exact analysis of MRSDF graphs

- Data-driven transformation into HSDF
- Redundancy in resulting HSDF

Existing approximate analysis

- No upper bound on rate - no sense of error
- Opaque solution from an LP

No common ground!

Status Quo on analysis:

Wiggers, M. H., Bekooij, M.J. and Smit, G.J.M. "Efficient computation of buffer capacities for cyclo-static dataflow graphs." Design Automation Conference, 2007. DAC'07. 44th ACM/IEEE. IEEE, 2007. (67 citations)

Stuijk, S., Geilen, M., and Basten, T. (2006, July). Exploring trade-offs in buffer requirements and throughput constraints for synchronous dataflow graphs. In Proceedings of the 43rd annual Design Automation Conference (pp. 899-904). ACM. (114 citations)

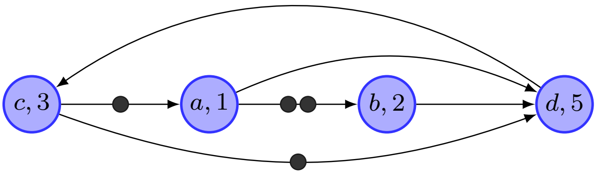
A. H. Ghamarian, M. C. W. Geilen, S. Stuijk, T. Basten, B. D. Theelen, M. R. Mousavi, A. J. M. Moonen, and M. J. G. Bekooij, "Throughput Analysis of Synchronous Data Flow Graphs," *ACSD*, 2006. (127 citations)

Back to Basics

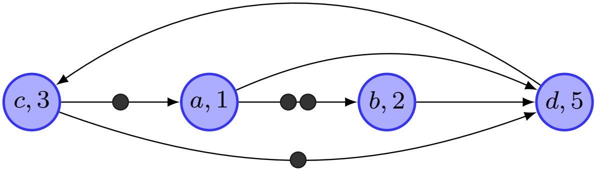
Periodic timed synchronous systems

- Mathematics: Max-Plus algebra (constraints)
- HSDF Graph: Linear Shift-Invariant system
- MRSDF Graph: Linear Shift-varying system

Back to Basics - HSDF, Max-Plus

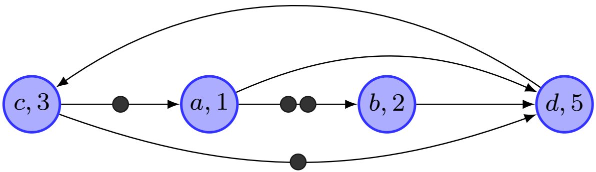


Back to Basics - HSDF, Max-Plus



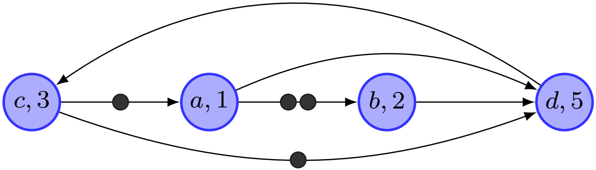
$$t_a(k) = t_c(k-1) + 1$$
$$t_b(k) = t_a(k-2) + 2$$
$$t_c(k) = t_d(k) + 3$$
$$t_d(k) = \max\{t_b(k), t_a(k), t_c(k)\} + 5$$

Back to Basics - HSDF, Max-Plus



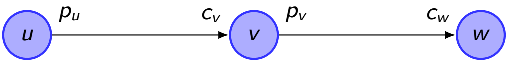
$$\begin{bmatrix} t_a \\ t_b \\ t_c \\ t_d \end{bmatrix}(k) = \bigoplus_i A_i \begin{bmatrix} t_a \\ t_b \\ t_c \\ t_d \end{bmatrix} \otimes (k-i)$$

Back to Basics - HSDF, Max-Plus



$$\begin{bmatrix} t_a \\ t_b \\ t_c \\ t_d \end{bmatrix} (k_0 + k) = \begin{bmatrix} t_a \\ t_b \\ t_c \\ t_d \end{bmatrix} (k_0) + 9k$$

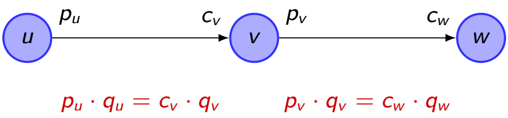
Back to Basics - MRSDF



Structural invariants:

- Repetition vector, q

Back to Basics - MRSDF

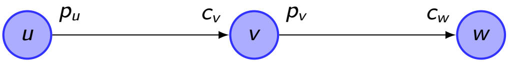


Structural invariants:

- Repetition vector, q

Back to Basics - MRSDF

$$c_v \cdot s_{uv} = p_v \cdot s_{vw}$$



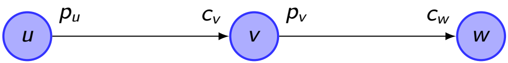
$$p_u \cdot q_u = c_v \cdot q_v \qquad p_v \cdot q_v = c_w \cdot q_w$$

Structural invariants:

- Repetition vector, q

Back to Basics - MRSDF

$$c_v \cdot s_{uv} = p_v \cdot s_{vw}$$



$$p_u \cdot q_u = c_v \cdot q_v \qquad p_v \cdot q_v = c_w \cdot q_w$$

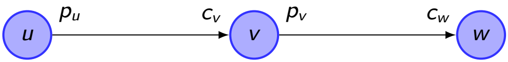
$$\mathcal{N} = p_u \cdot q_u \cdot s_{uv} = c_v \cdot q_v \cdot s_{uv} = p_v \cdot q_v \cdot s_{vw} = \ldots$$

Structural invariants:

- Repetition vector, q
- Normalisation vector, s

Back to Basics - MRSDF

$$c_v \cdot s_{uv} = p_v \cdot s_{vw}$$



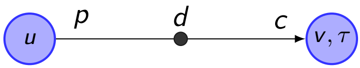
$$p_u \cdot q_u = c_v \cdot q_v \qquad p_v \cdot q_v = c_w \cdot q_w$$

$$\mathcal{N} = p_u \cdot q_u \cdot s_{uv} = c_v \cdot q_v \cdot s_{uv} = p_v \cdot q_v \cdot s_{vw} = \ldots$$

Structural invariants:

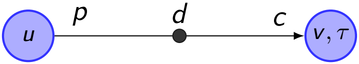
- Repetition vector, q
- Normalisation vector, s
- Normalised token count, $\mathcal{N}$

MRSDF Analysis - Exact Homogeneous Representations



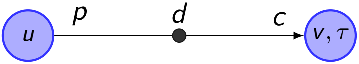
$t_v(k) = t_u(...) + \tau$

MRSDF Analysis - Exact Homogeneous Representations



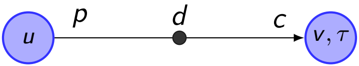
$$t_v(k) = t_u(\ldots k \cdot c - d \ldots) + \tau$$

MRSDF Analysis - Exact Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

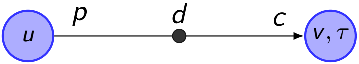
MRSDF Analysis - Exact Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

$$t_v(k + mq_v) = \ldots$$

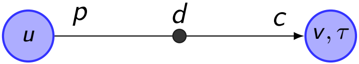
MRSDF Analysis - Exact Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

$$t_v(k + mq_v) = t_u\left(\left\lceil \frac{(k + mq_v) \cdot c - d}{p} \right\rceil\right) + \tau$$

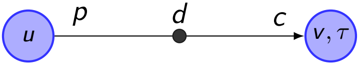
MRSDF Analysis - Exact Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

$$t_v(k + mq_v) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil + mq_u\right) + \tau$$

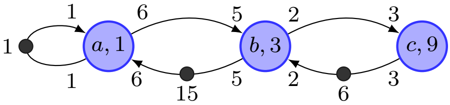
MRSDF Analysis - Exact Homogeneous Representations



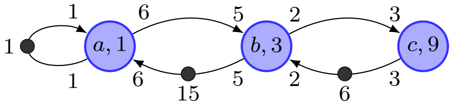
$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

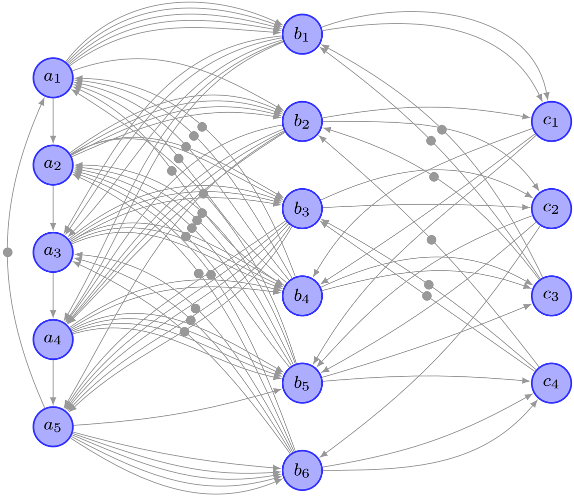
$$t_v(k + mq_v) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil + mq_u\right) + \tau = t_u(k + mq_u - C) + \tau$$

MRSDF Analysis - Exact Homogeneous Representations

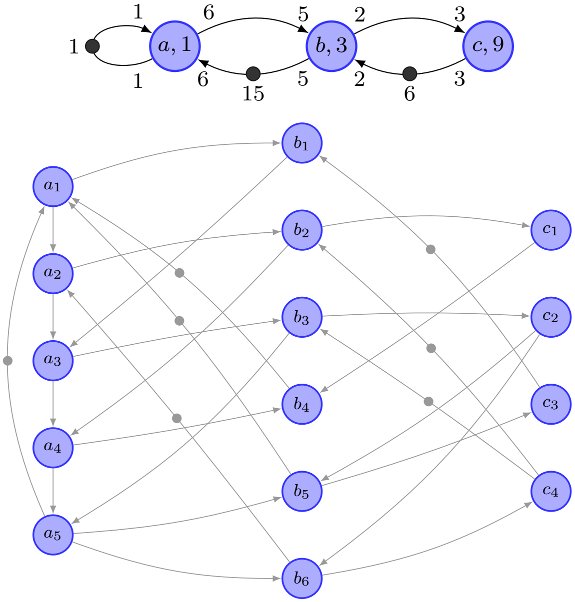


MRSDF Analysis - Exact Homogeneous Representations

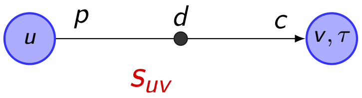




MRSDF Analysis - Exact Homogeneous Representations

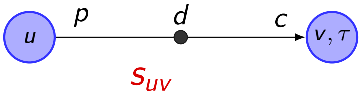


MRSDF Analysis - Approx. Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

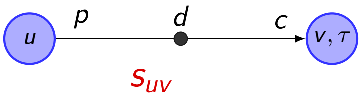
MRSDF Analysis - Approx. Homogeneous Representations



$$t_v(k) = t_u\left(\left\lceil \frac{k \cdot c - d}{p} \right\rceil\right) + \tau$$

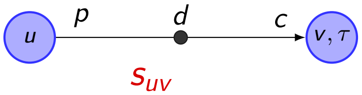
Obtain shift-invariance by changing counting units

MRSDF Analysis - Approx. Homogeneous Representations



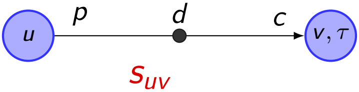
$$t_v\left(\frac{k}{q_v}\right) =$$

MRSDF Analysis - Approx. Homogeneous Representations



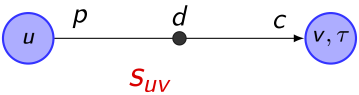
$$t_v(\kappa) =$$

MRSDF Analysis - Approx. Homogeneous Representations



$$t_v(\kappa) = t_u\left(\frac{1}{q_u}\left\lceil\frac{\kappa \cdot q_v \cdot c - d}{p}\right\rceil\right) + \tau$$

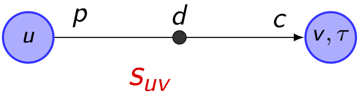
MRSDF Analysis - Approx. Homogeneous Representations



$$t_v(\kappa) = t_u\left(\frac{1}{q_u}\left\lceil\frac{\kappa \cdot q_v \cdot c - d}{p}\right\rceil\right) + \tau$$

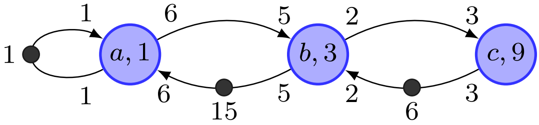
$$= t_u\left(\frac{1}{q_u}\left\lfloor\frac{\kappa \cdot q_v \cdot c - d + p - 1}{p}\right\rfloor\right) + \tau$$

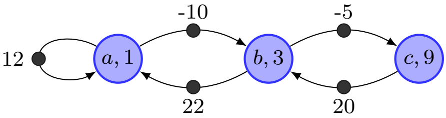
MRSDF Analysis - Approx. Homogeneous Representations



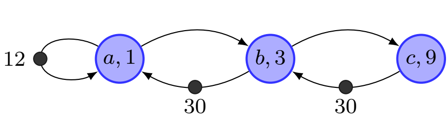
$$\hat{t}_v(k) = \hat{t}_u\left(k - s_{uv} \cdot d\right) + \tau$$
$$= \hat{t}_u\left(k - s_{uv} \cdot (d - p + 1)\right) + \tau$$

MRSDF Analysis - Approx. Homogeneous Representations



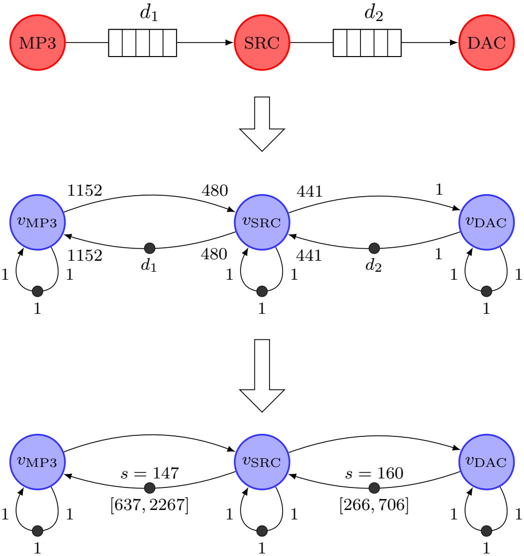


Pessimistic



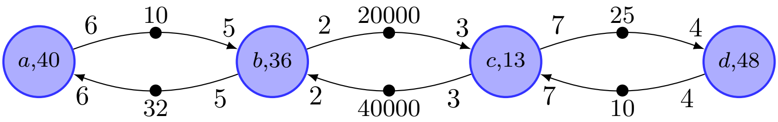
Optimistic

MRSDF Analysis - Example use case



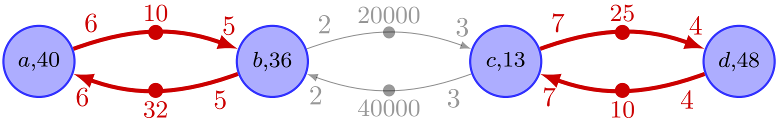
MP3 decoder: $\tau = 1603621$, SRC: $\tau = 1320974$, DAC: $\tau = 5000$

Future Work - Towards Incremental Analysis



Goal: Close the gap between exact and approximate analysis

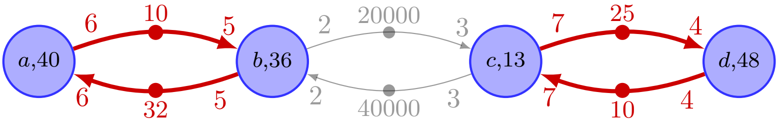
Future Work - Towards Incremental Analysis



Goal: Close the gap between exact and approximate analysis

- Critical Subgraph

Future Work - Towards Incremental Analysis



$54.3 \leq \lambda^* \leq 69.1$

$48.8 \leq \lambda^* \leq 65.7$

$9.8 \cdot 10^{-3} \leq \lambda^* \leq 9.8 \cdot 10^{-3}$

Goal: Close the gap between exact and approximate analysis

- Critical Subgraph
- Use bounds to zoom in on critical subgraph

Conclusions

Back to Basics!

- Gives us a *natural* transformation from MRSDF into HSDF...

Conclusions

Back to Basics!

- Gives us a *natural* transformation from MRSDF into HSDF...
- ...from which we can derive bounding HSDF graphs

Conclusions

Back to Basics!

- Gives us a *natural* transformation from MRSDF into HSDF...
- ...from which we can derive bounding HSDF graphs

Properties:

- Buffer weights direct further optimization

Conclusions

Back to Basics!

- Gives us a *natural* transformation from MRSDF into HSDF...
- ...from which we can derive bounding HSDF graphs

Properties:

- Buffer weights direct further optimization
- Approximation gets better for large repetition vectors (= large HSDF graphs)

Conclusions

Back to Basics!

- Gives us a *natural* transformation from MRSDF into HSDF...
- ...from which we can derive bounding HSDF graphs

Properties:

- Buffer weights direct further optimization
- Approximation gets better for large repetition vectors (= large HSDF graphs)
- Perfectly suited to balance analysis accuracy and runtime

Questions ?

robert.degroote@utwente.nl