Technische Universiteit
Eindhoven
University of Technology

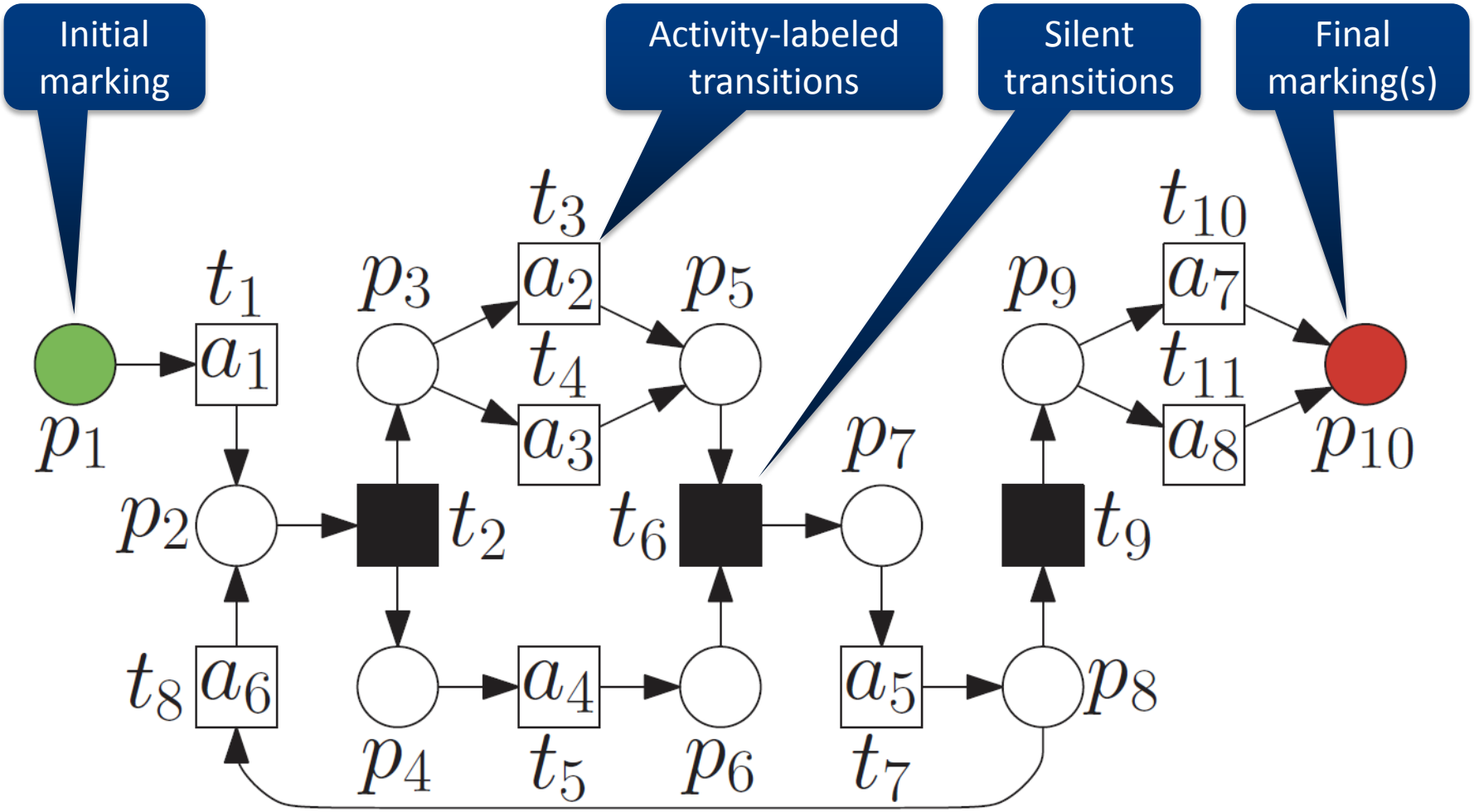
Decomposed Replay Using Hiding and Reduction

Eric Verbeek

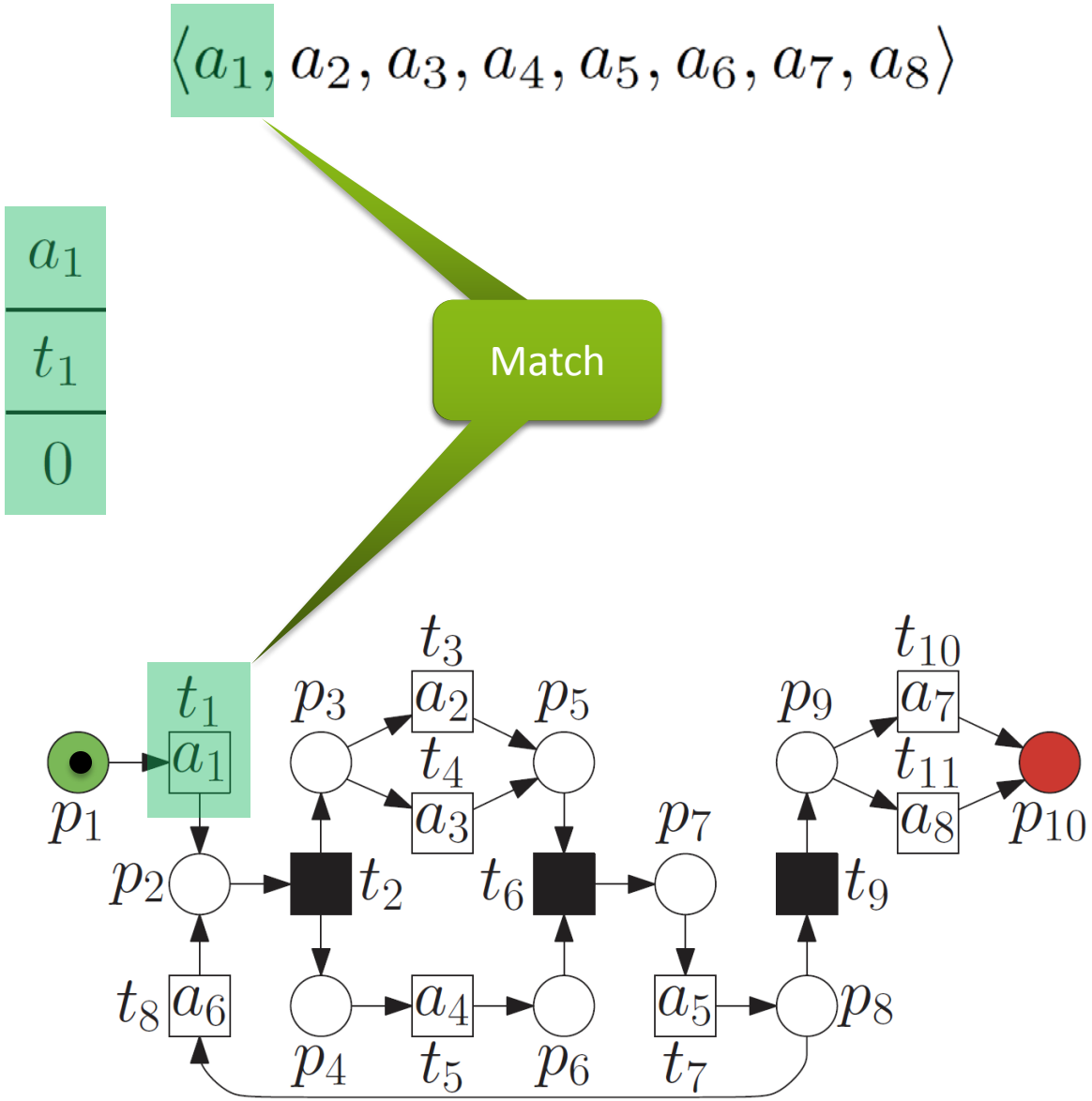
Where innovation starts

- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work

- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work


Initial marking
Activity-labeled transitions
Silent transitions
Final marking(s)
p_1
t_1
a_1
p_2
t_2
p_3
t_3
a_2
t_4
a_3
p_4
t_5
a_4
t_6
p_5
p_6
t_7
a_5
t_8
a_6
p_7
t_9
p_8
t_{10}
a_7
t_{11}
a_8
p_9
p_{10}

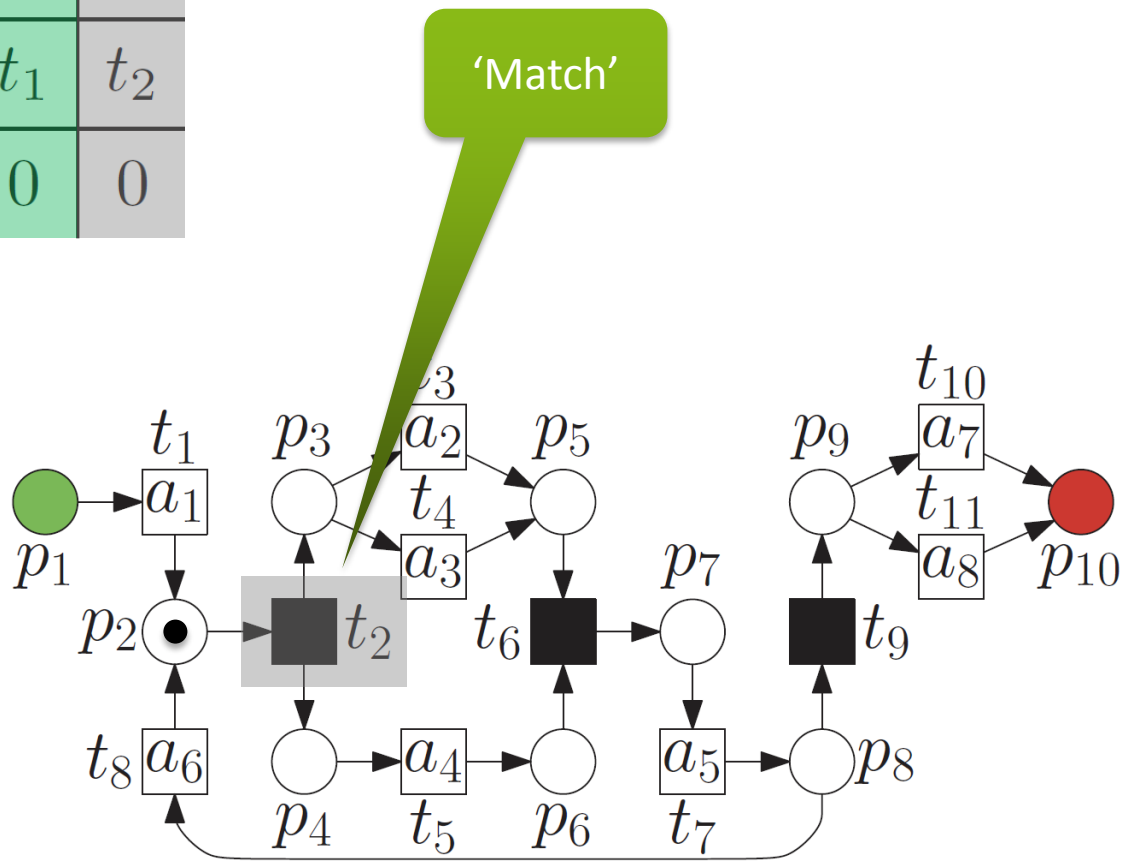
$$\langle a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \rangle$$



Synchronous move on a1 and t1

$$\langle a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \rangle$$

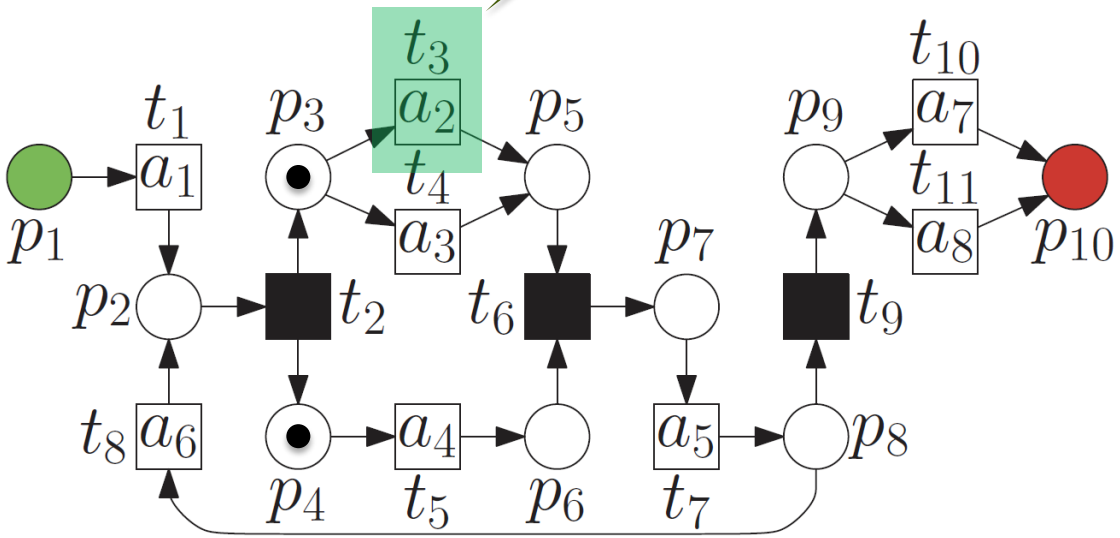
a_1	τ
t_1	t_2
0	0



$$\langle a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \rangle$$

a_1	τ	a_2
t_1	t_2	t_3
0	0	0

Match

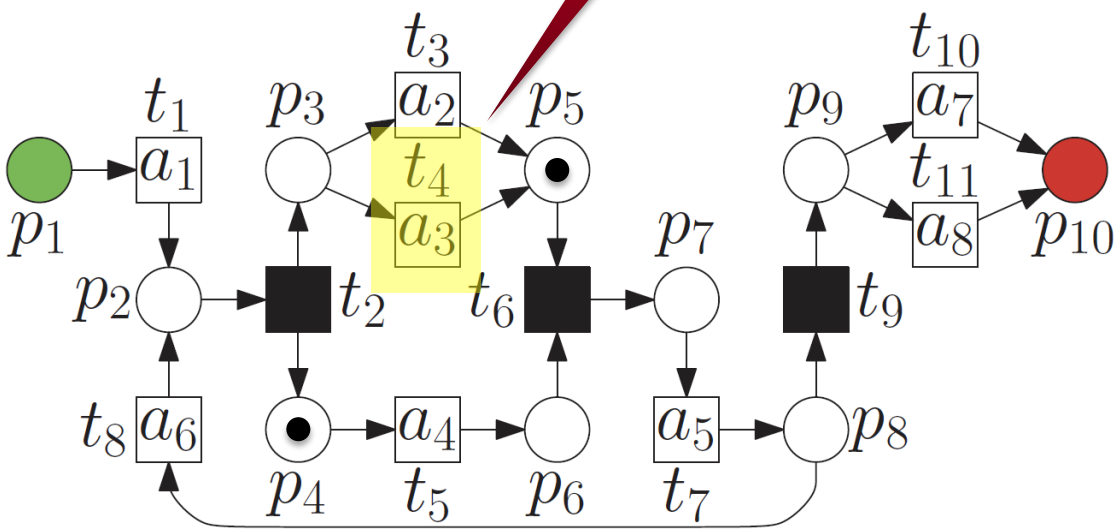


Synchronous move on a2 and t3

$$\langle a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \rangle$$

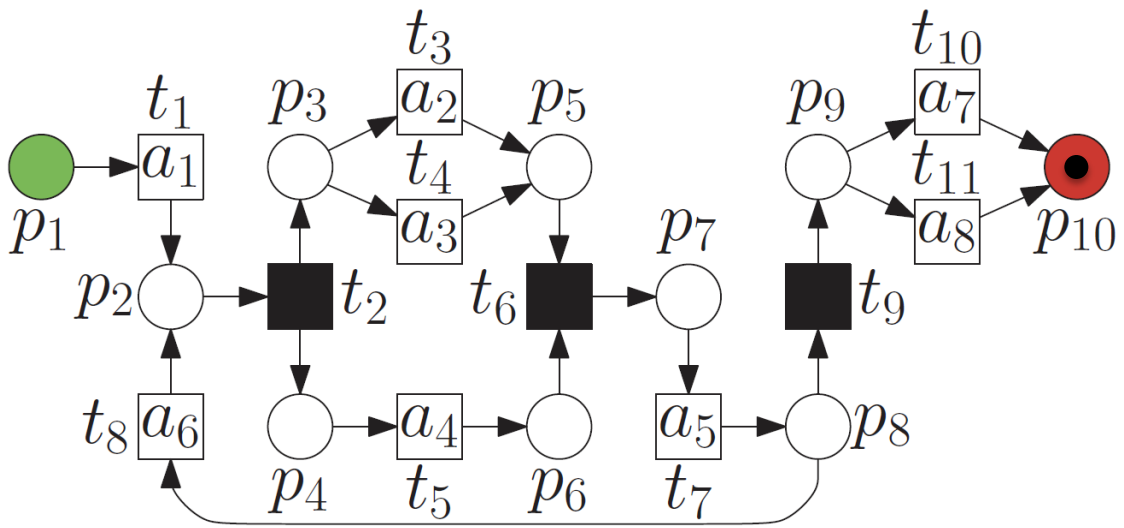
a_1	τ	a_2	a_3
t_1	t_2	t_3	$\gg$
0	0	0	10

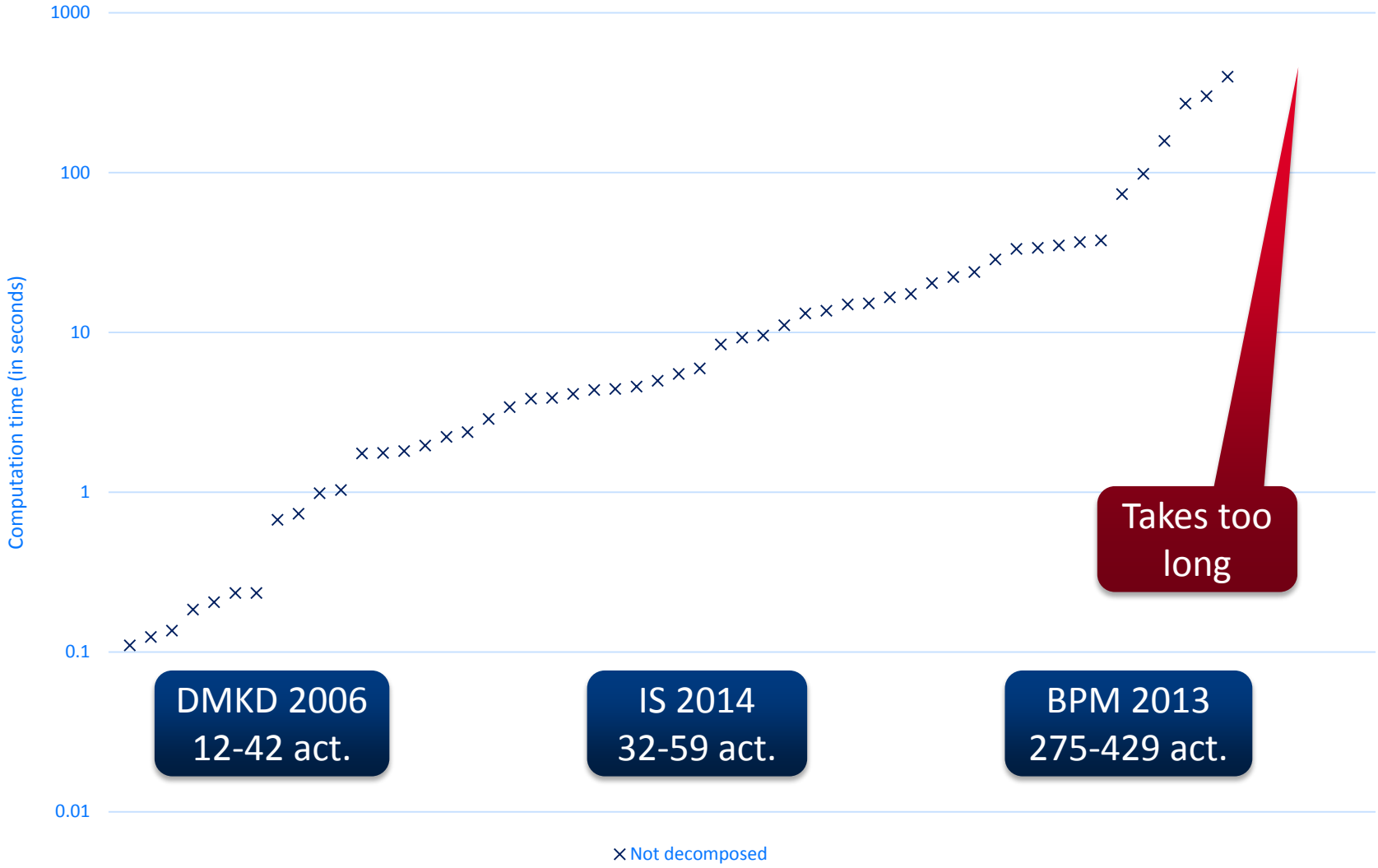
No match



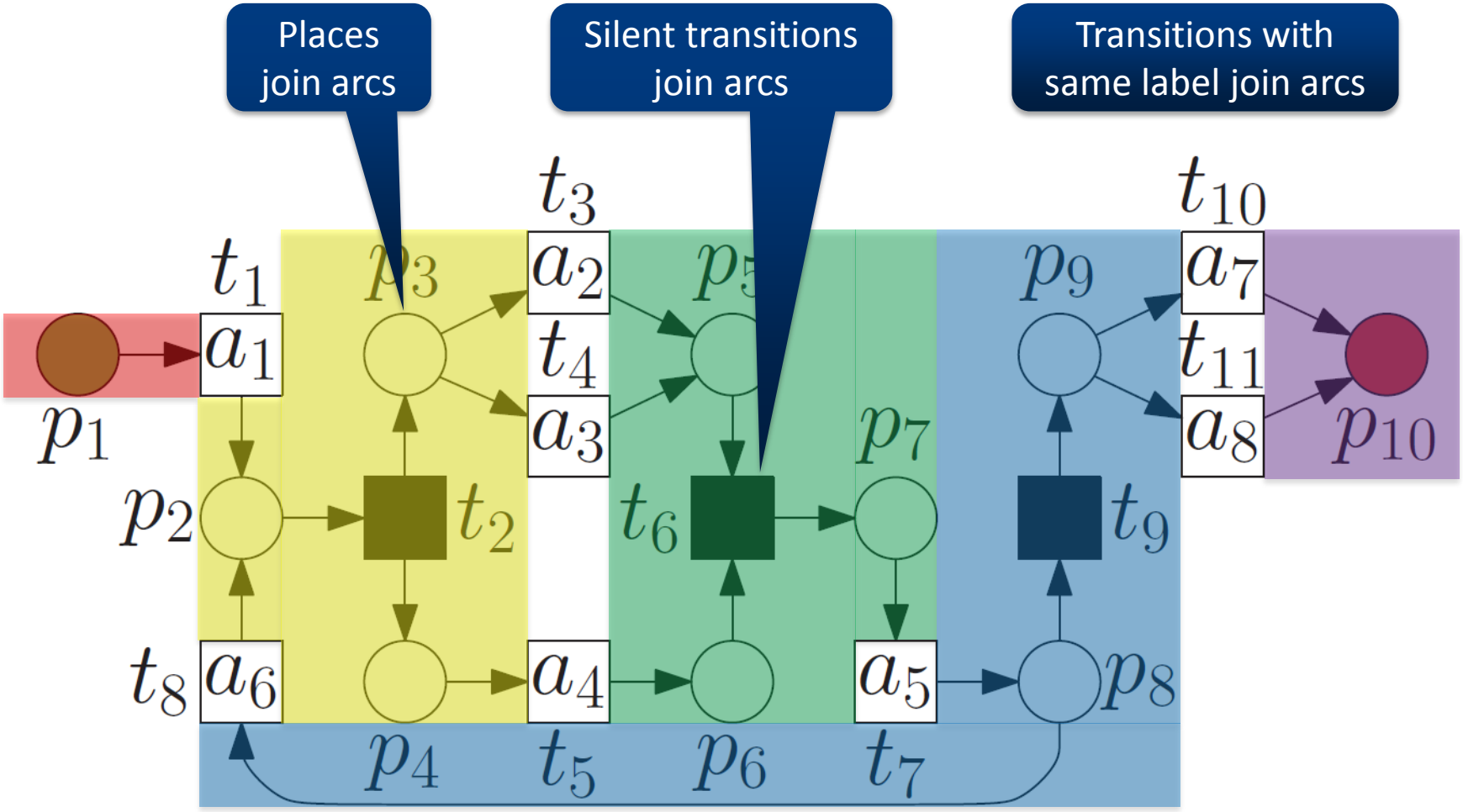
$$\langle a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 \rangle$$

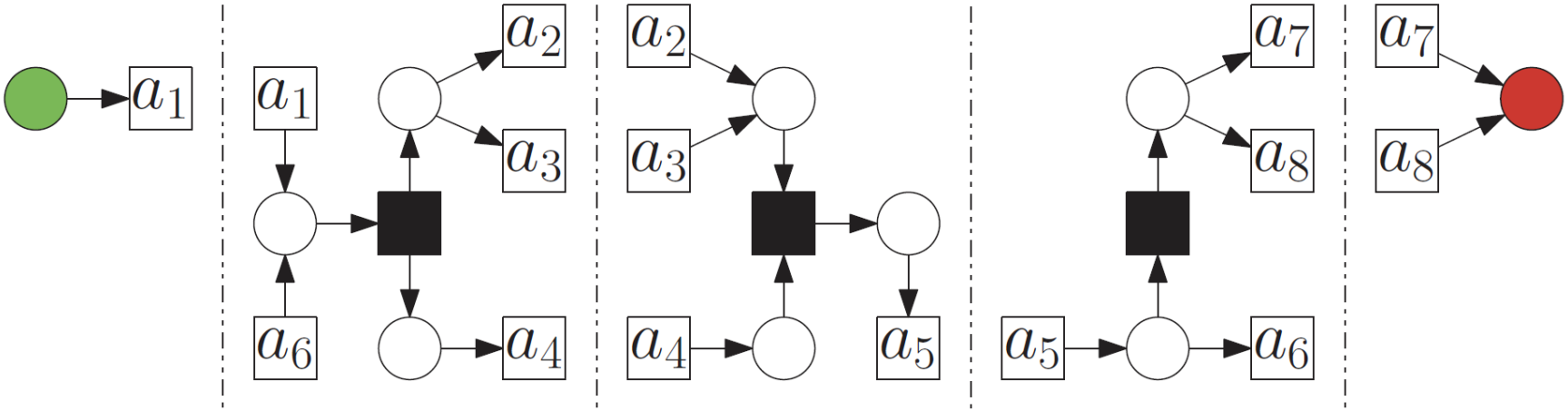
a_1	τ	a_2	a_3	a_4	τ	a_5	a_6	τ	a_7	a_8
t_1	t_2	t_3	$\gg$	t_5	t_6	t_7	$\gg$	t_9	t_{10}	$\gg$
0	0	0	10	0	0	0	10	0	0	10




1000
100
10
1
0.1
0.01
Computation time (in seconds)
Takes too long
DMKD 2006
12-42 act.
IS 2014
32-59 act.
BPM 2013
275-429 act.
Not decomposed

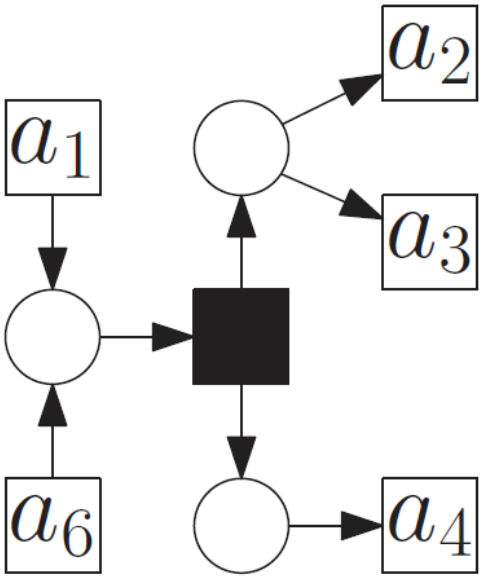
- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Merging alignments
  - Pseudo alignment
  - Alignment of alignments
  - Stitching rules
- Wrapping up
  - Conclusions
  - Future work


Places join arcs
Silent transitions join arcs
Transitions with same label join arcs
t_3
t_{10}
t_1
p_3
a_2
p_5
p_9
a_7
a_1
t_4
t_{11}
p_1
a_3
p_7
a_8
p_{10}
p_2
t_2
t_6
t_9
t_8
a_6
a_4
a_5
p_8
p_4
t_5
p_6
t_7


a_1
a_1
a_2
a_3
a_6
a_4
a_2
a_3
a_4
a_5
a_5
a_7
a_8
a_6
a_7
a_8

$$\langle a_1, a_2, a_3, a_4, a_6 \rangle$$

a_1	τ	a_2	a_3	a_4	a_6
t_1	t_2	t_3	$\gg$	t_5	$\gg$
0	0	0	5	0	5

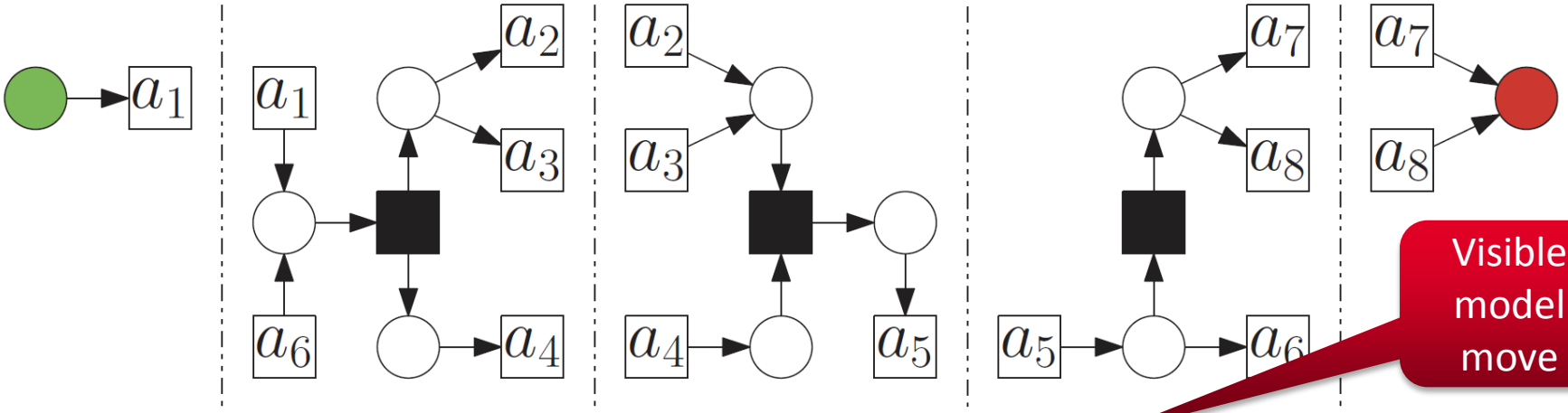


Adapted costs

a_1
t_1
0

a_2	a_3	a_4	τ	a_5
$\gg$	t_4	t_5	t_6	t_7
5	0	0	0	0

a_7	a_8
t_{10}	$\gg$
0	5

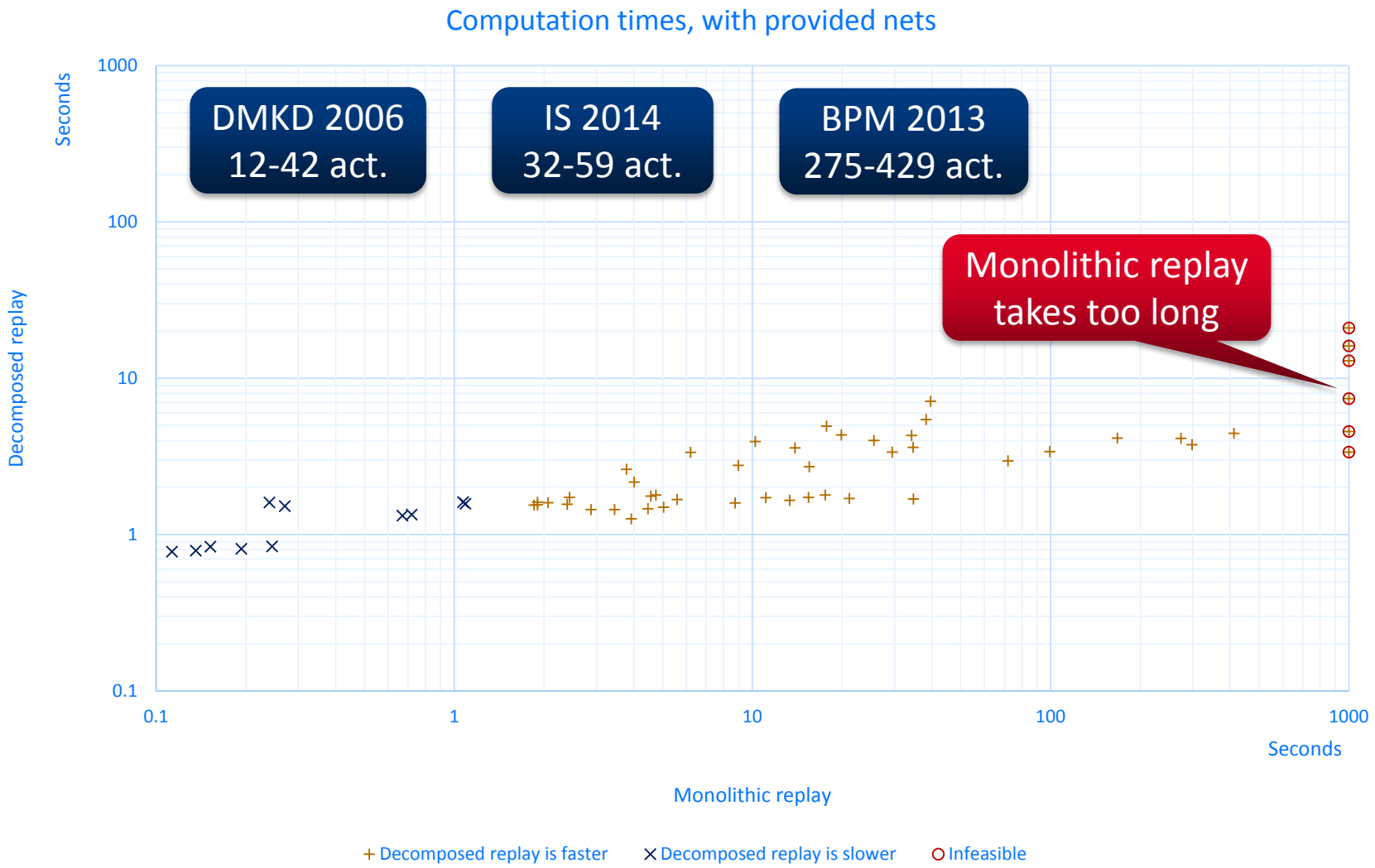


Visible model move

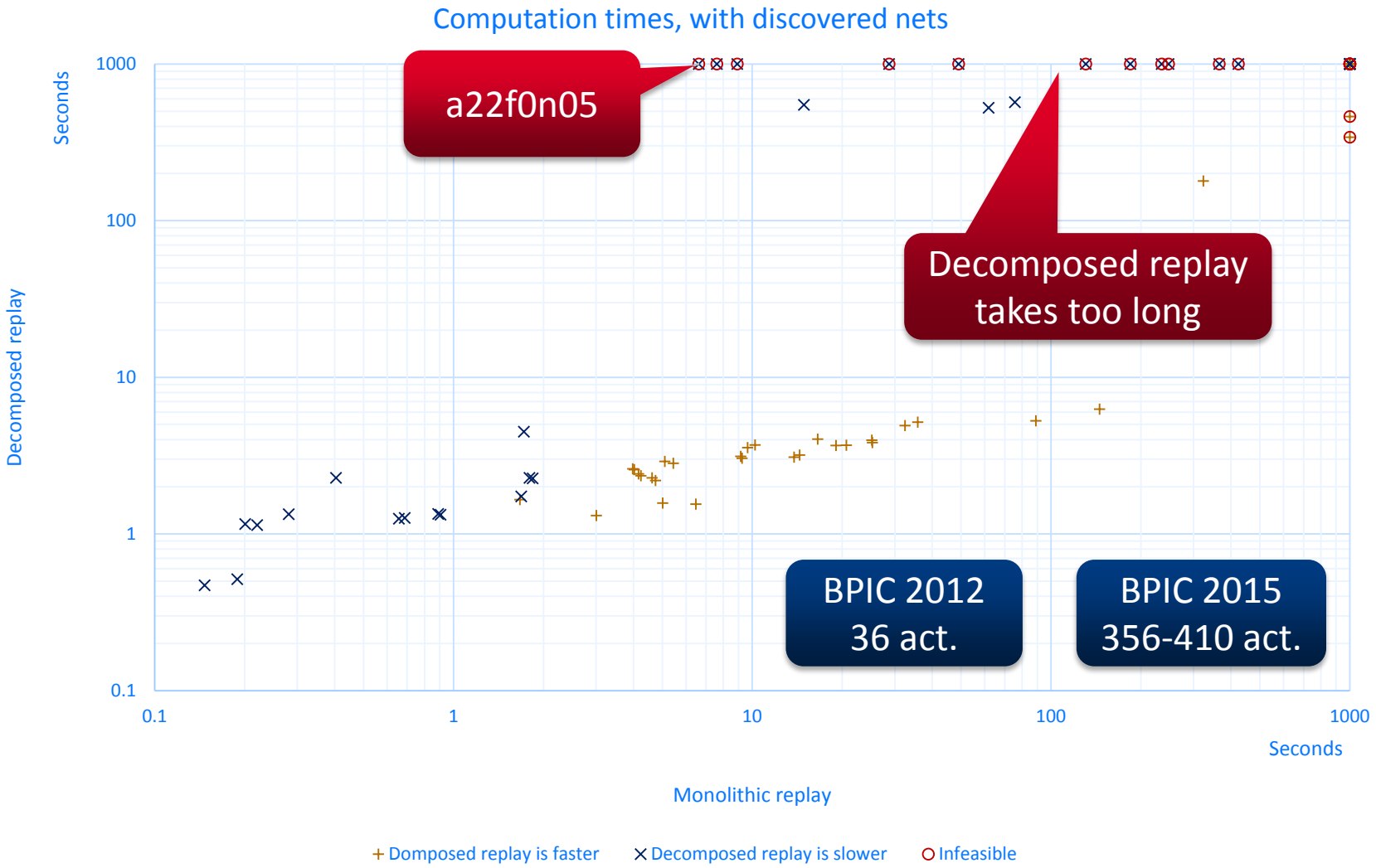
a_1	τ	a_2	a_3	a_4	a_6
t_1	t_2	t_3	$\gg$	t_5	$\gg$
0	0	0	5	0	5

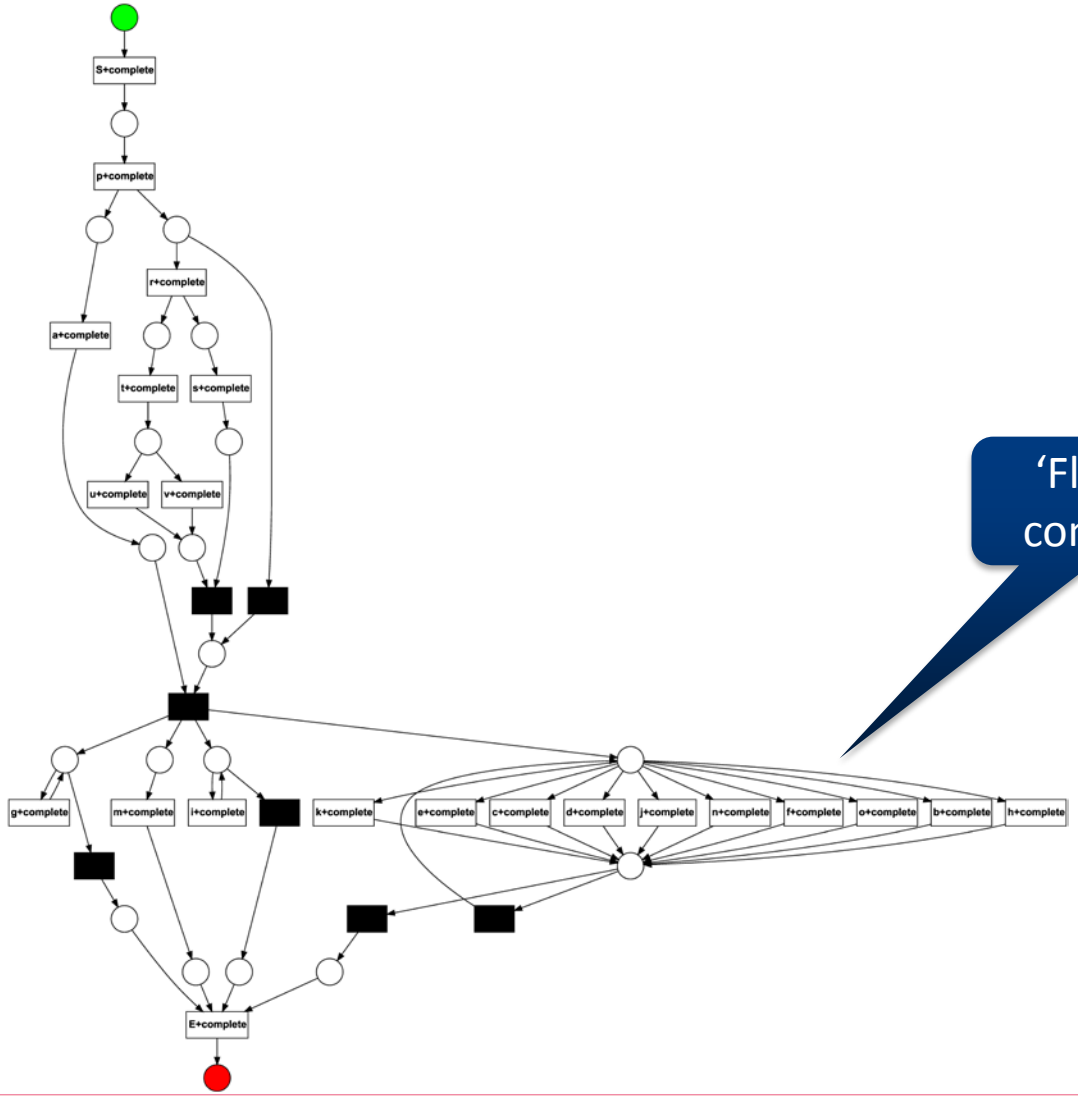
a_5	a_6	$\gg$	τ	a_7	a_8
t_7	t_8	t_7	t_9	t_{10}	$\gg$
0	0	2	0	0	5

- Non-decomposed costs 0 if and only if decomposed costs 0
- Decomposed costs less or equal to non-decomposed costs

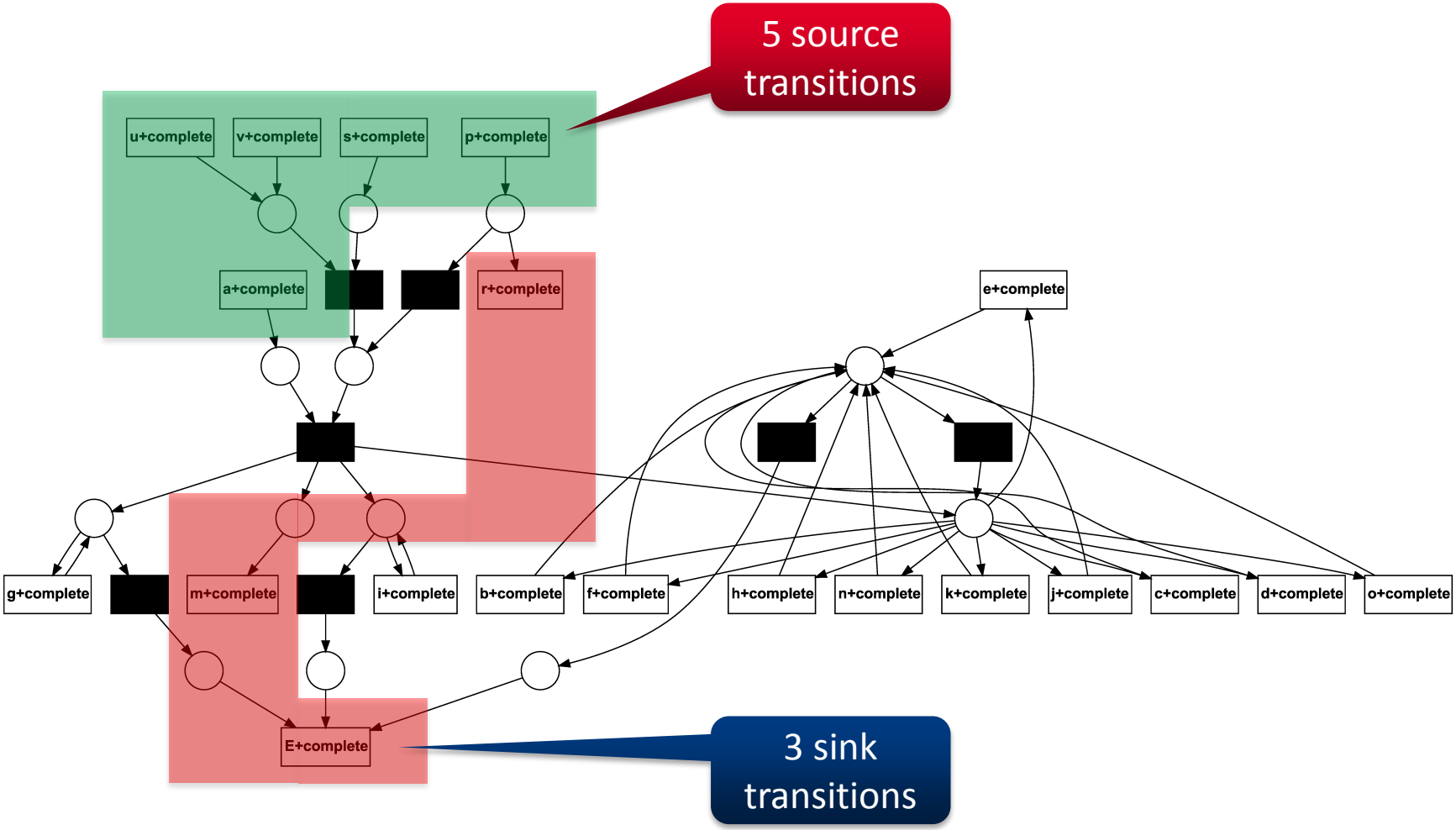

Computation times, with provided nets
DMKD 2006
12-42 act.
IS 2014
32-59 act.
BPM 2013
275-429 act.
Monolithic replay takes too long
Seconds
Decomposed replay
1000
100
10
1
0.1
0.1
1
10
100
1000
Seconds
Monolithic replay
+ Decomposed replay is faster
× Decomposed replay is slower
○ Infeasible

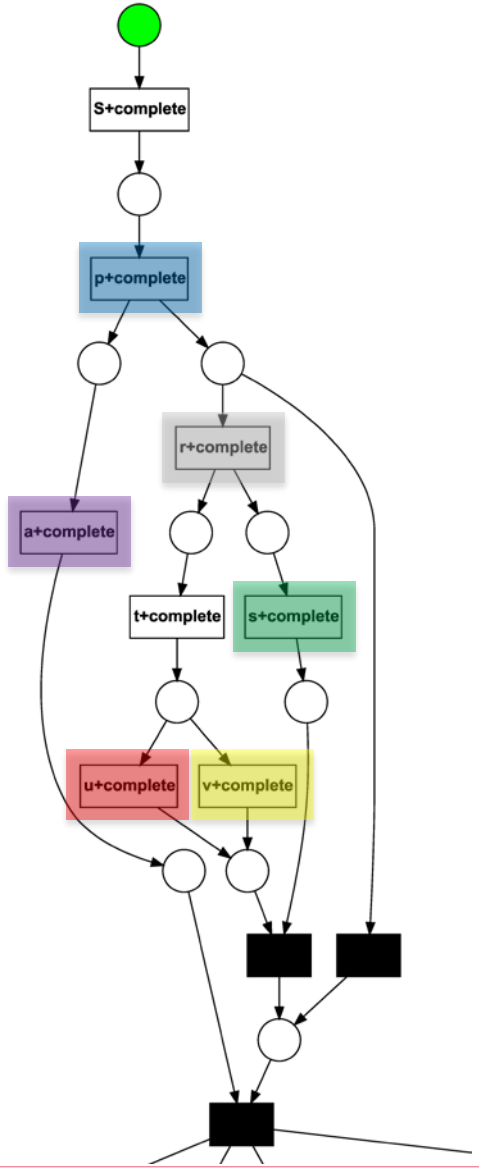
- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work

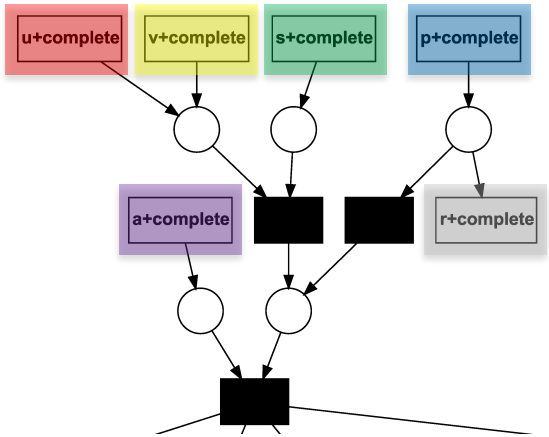

Computation times, with discovered nets
a22f0n05
Decomposed replay takes too long
BPIC 2012 36 act.
BPIC 2015 356-410 act.
Seconds
Decomposed replay
1000
100
10
1
0.1
0.1
1
10
100
1000
Seconds
Monolithic replay
+ Domposed replay is faster
× Decomposed replay is slower
○ Infeasible



Log a22f0n05 – Net discovered by Inductive Miner

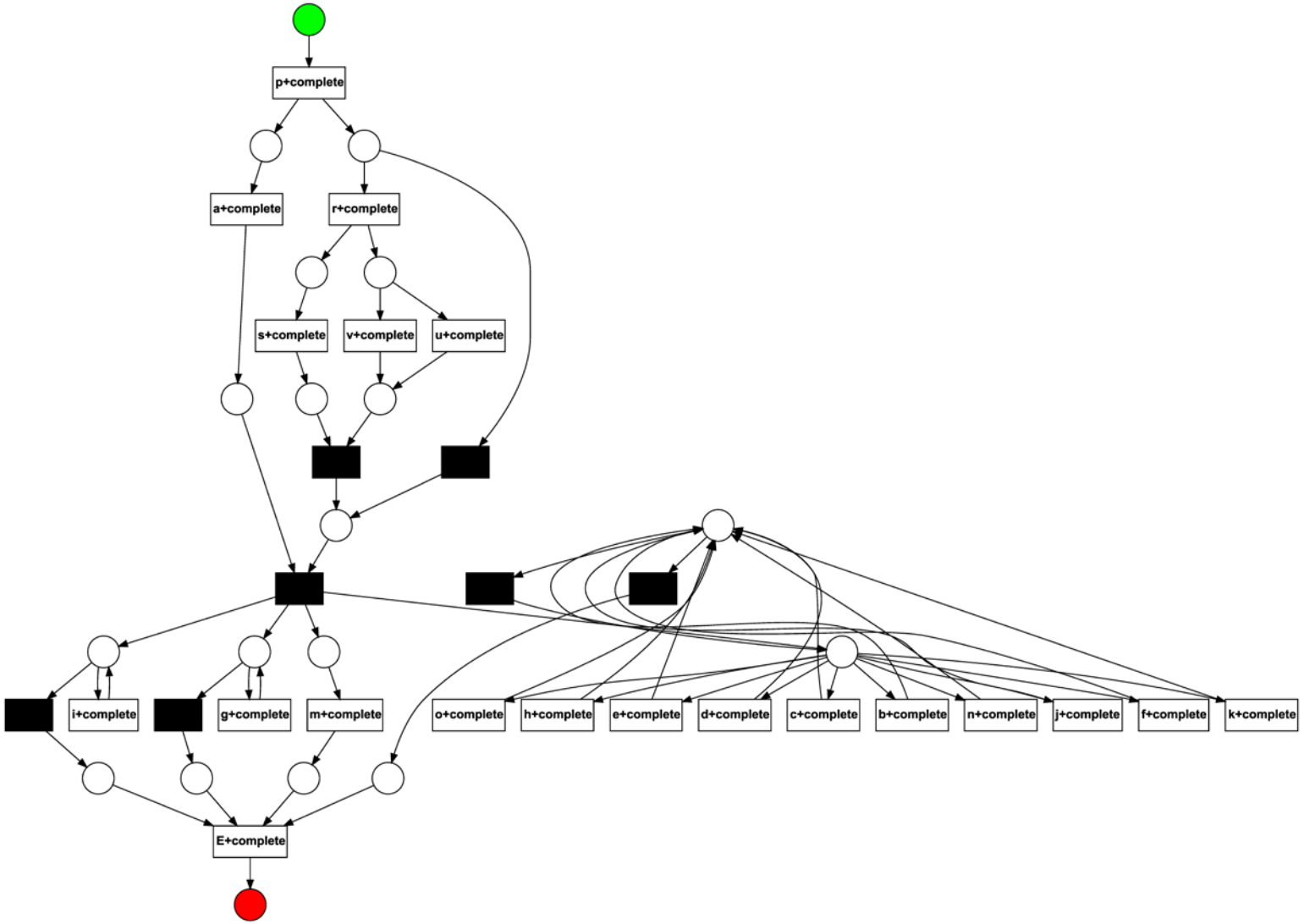

5 source transitions
u+complete
v+complete
s+complete
p+complete
a+complete
r+complete
e+complete
g+complete
m+complete
i+complete
b+complete
f+complete
h+complete
n+complete
k+complete
j+complete
c+complete
d+complete
o+complete
E+complete
3 sink transitions

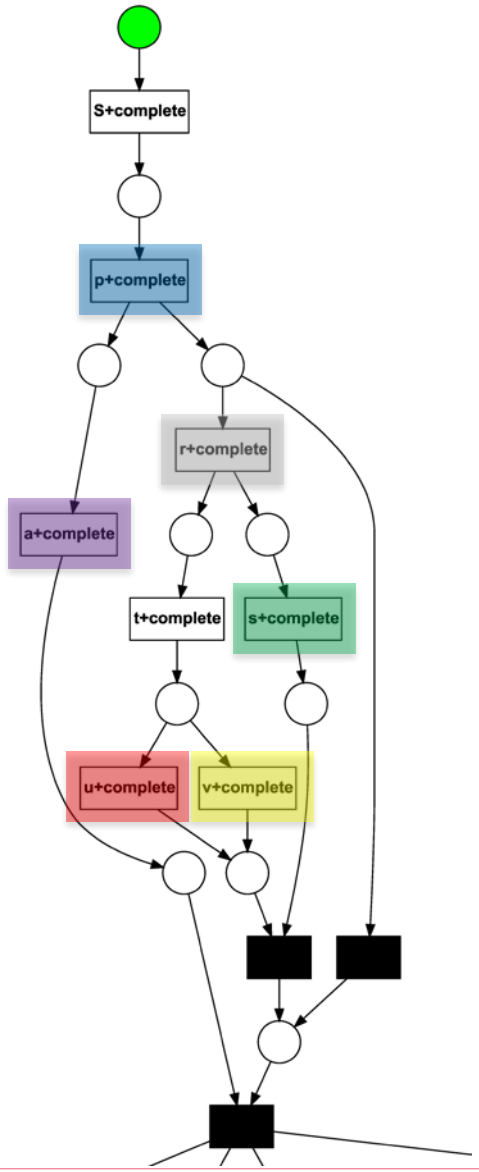

S+complete
p+complete
r+complete
a+complete
t+complete
s+complete
u+complete
v+complete


u+complete
v+complete
s+complete
p+complete
a+complete
r+complete

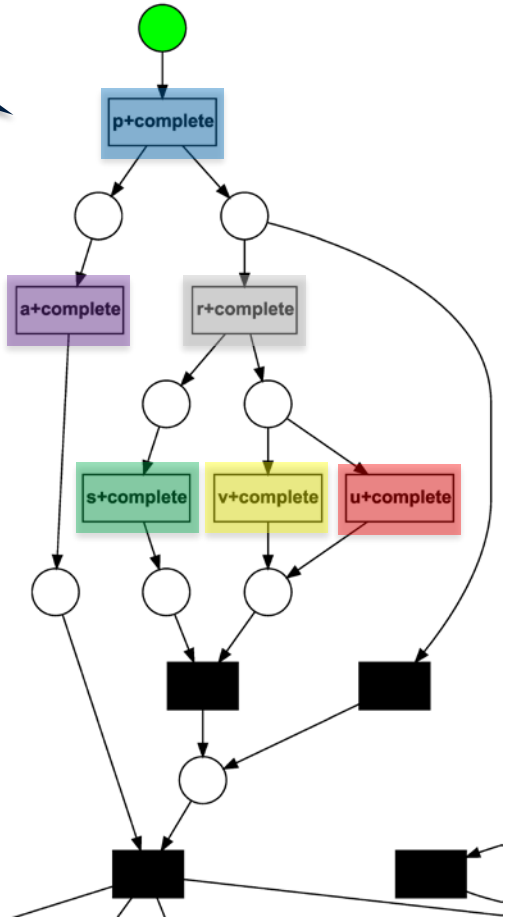
- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work

- Hide visible transition ‘not covered’ by subnet
- Apply behavior-preserving reduction rules afterwards

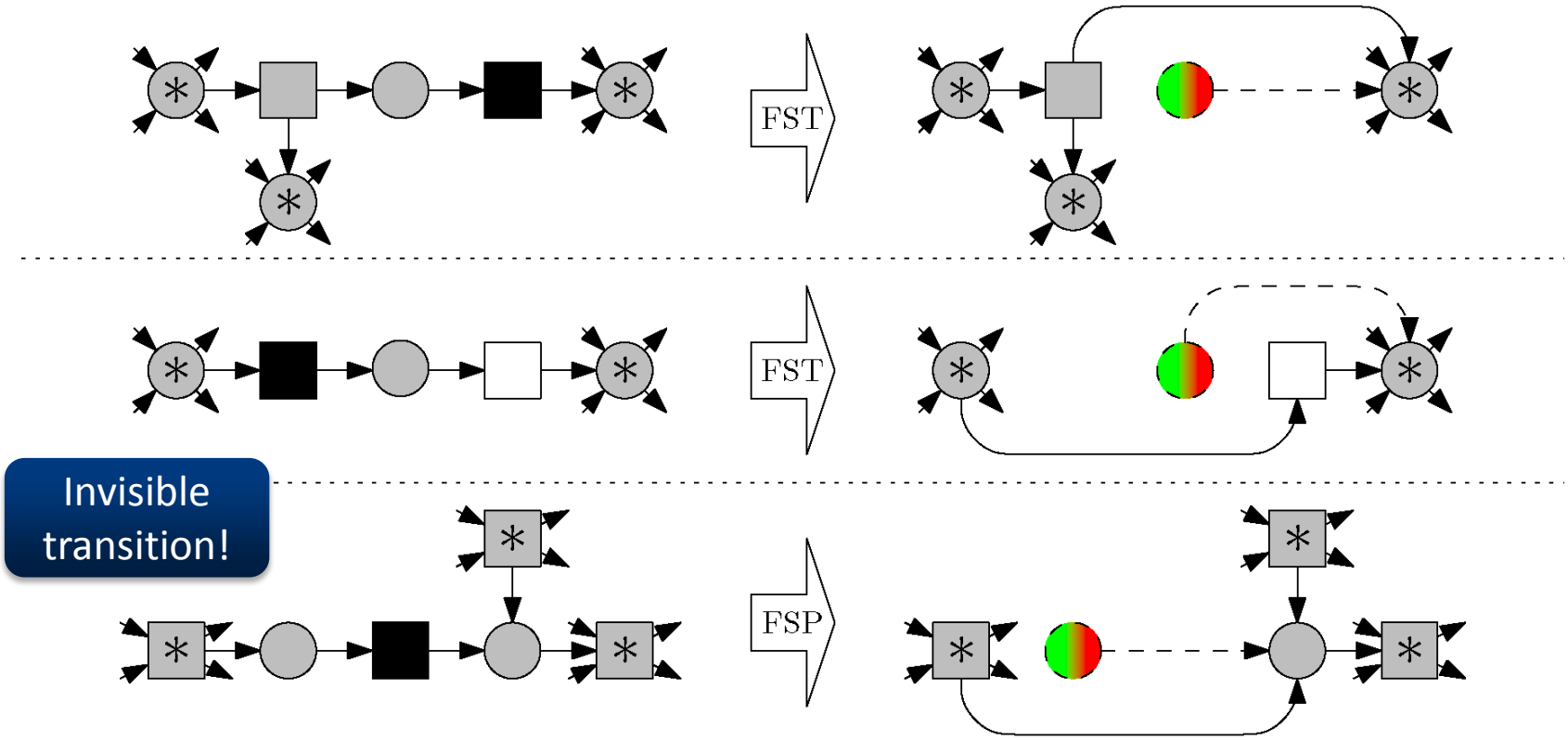

p+complete
a+complete
r+complete
s+complete
v+complete
u+complete
i+complete
g+complete
m+complete
o+complete
h+complete
e+complete
d+complete
c+complete
b+complete
n+complete
j+complete
f+complete
k+complete
E+complete



No source transitions

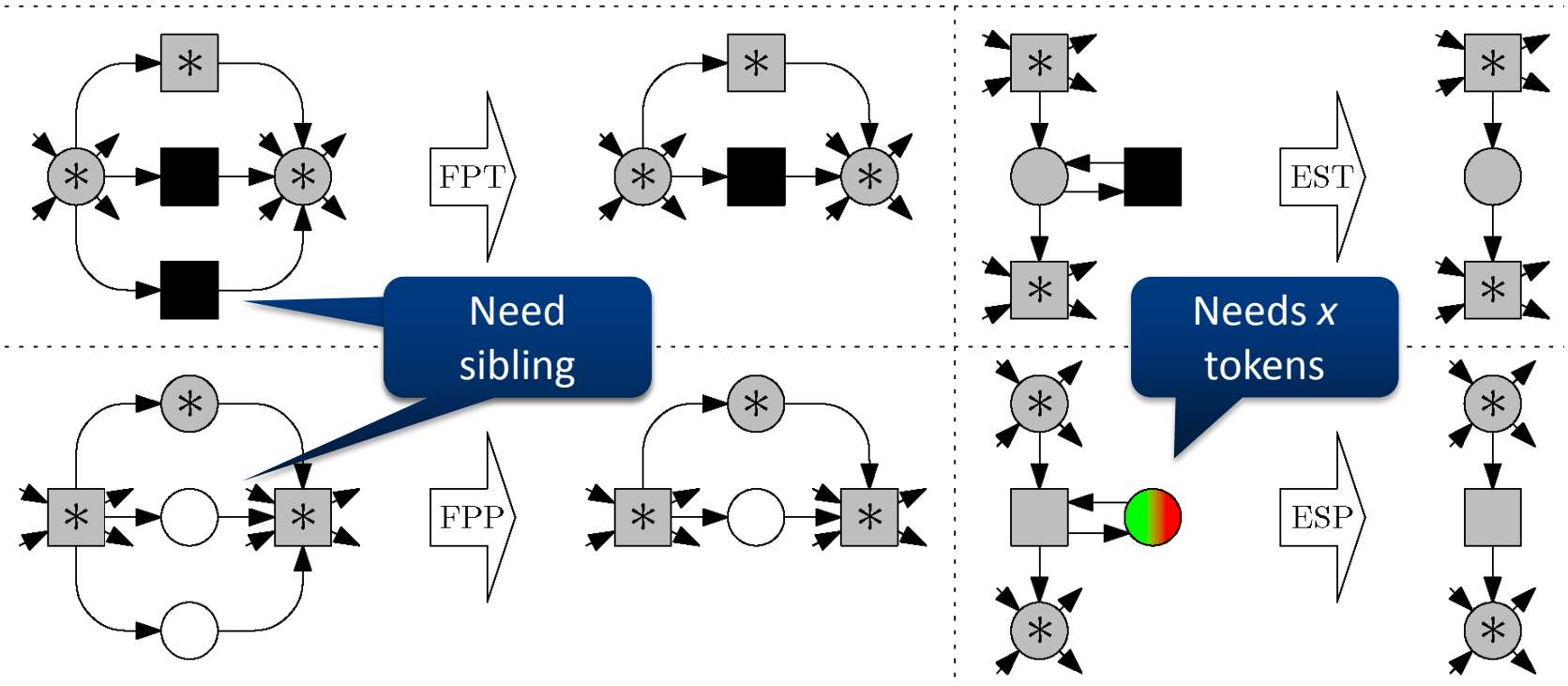


- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work



Legend

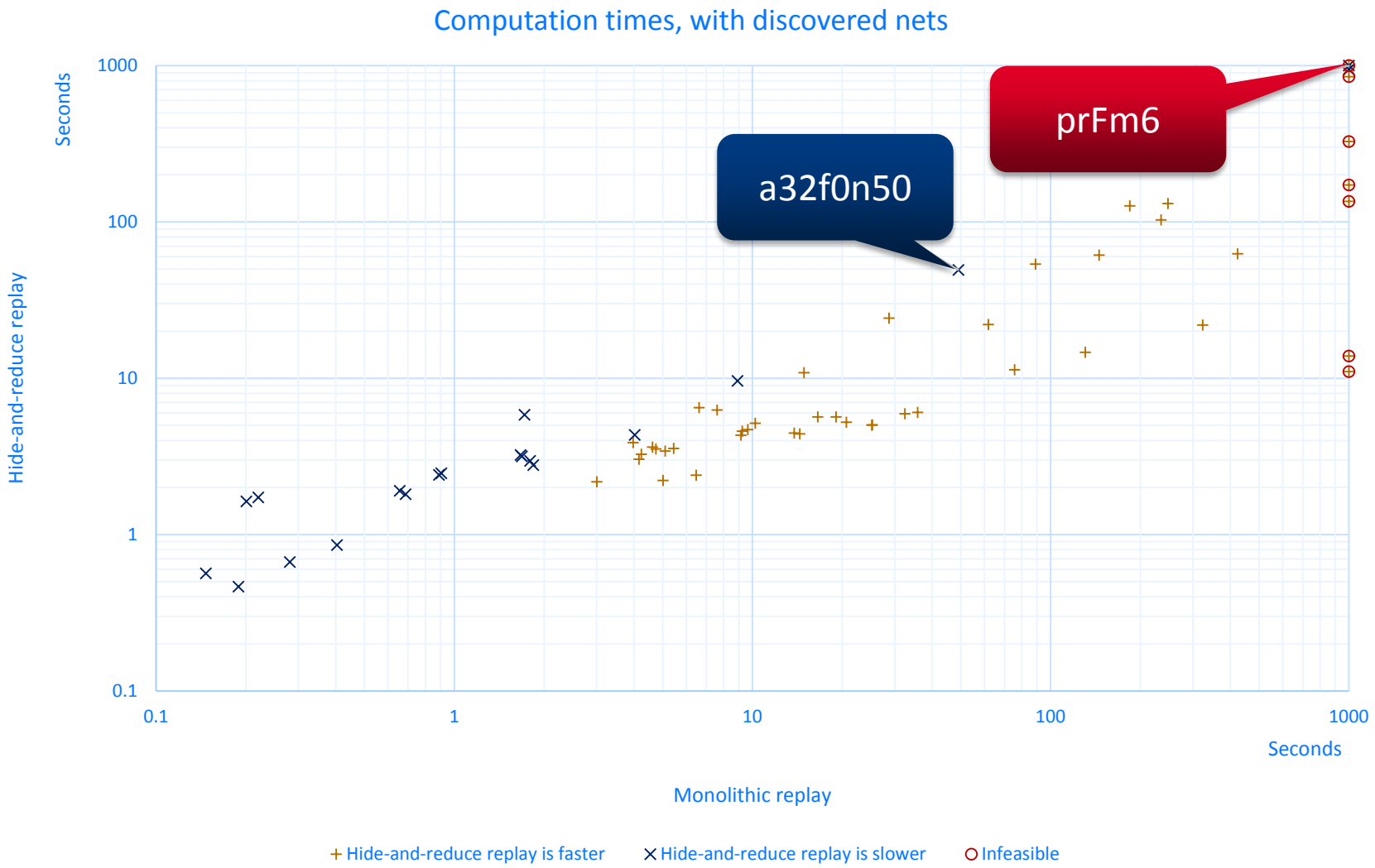
- an invisible transition
- a visible transition
- any transition
- a place containing x tokens in the initial and every final marking (where $x > 0$)
- an unmarked place
- any place
- $*$ any number of these objects (includes connected arcs)
- updating initial and final markings if needed



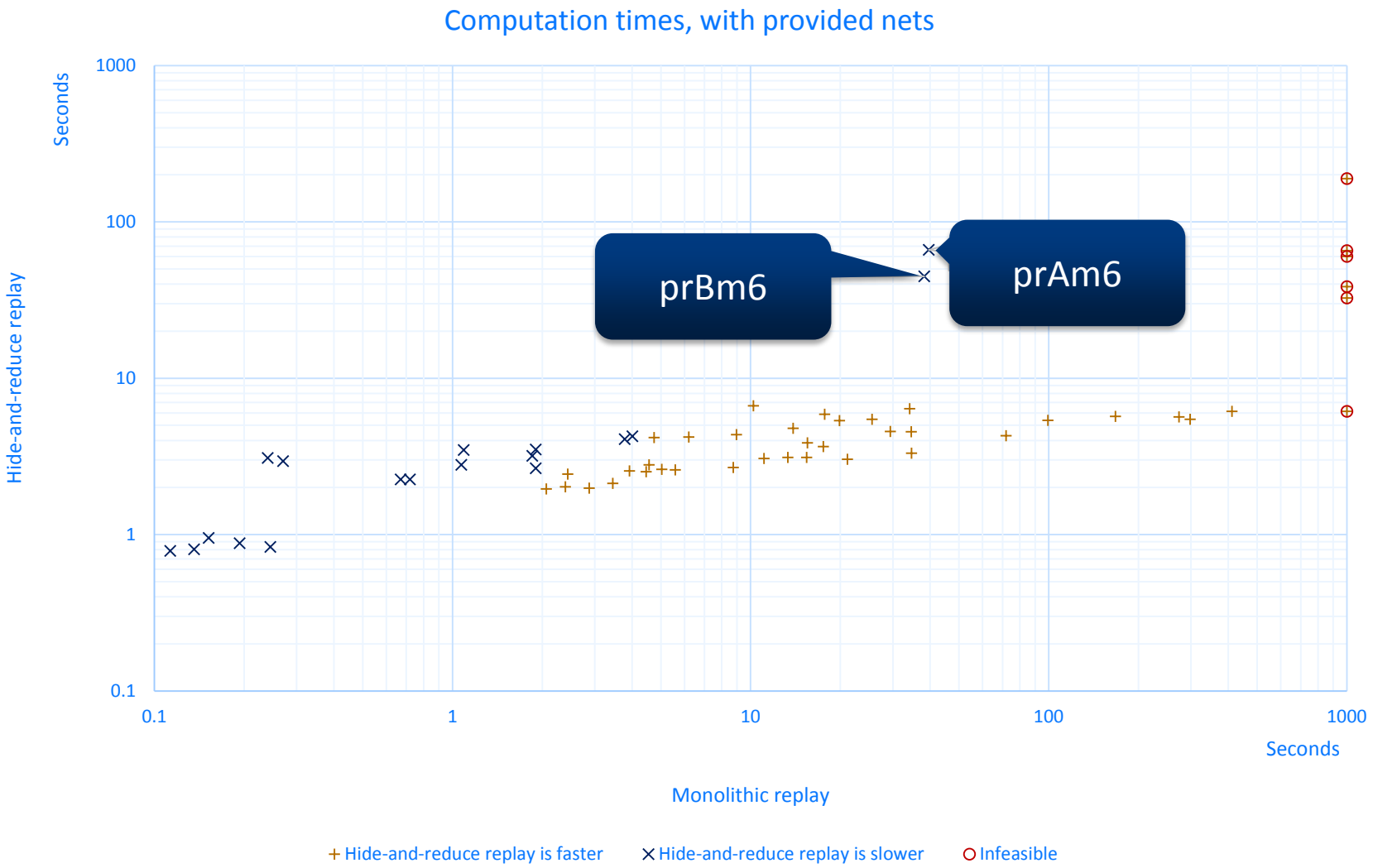
Legend

- an invisible transition
- a visible transition
- any transition
- a place containing x tokens in the initial and every final marking (where $x > 0$)
- an unmarked place
- any place
- $*$ any number of these objects (includes connected arcs)
- updating initial and final markings if needed

- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work



Computation times for H&R replay – discovered nets



Computation times for H&R replay – provided nets

- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work

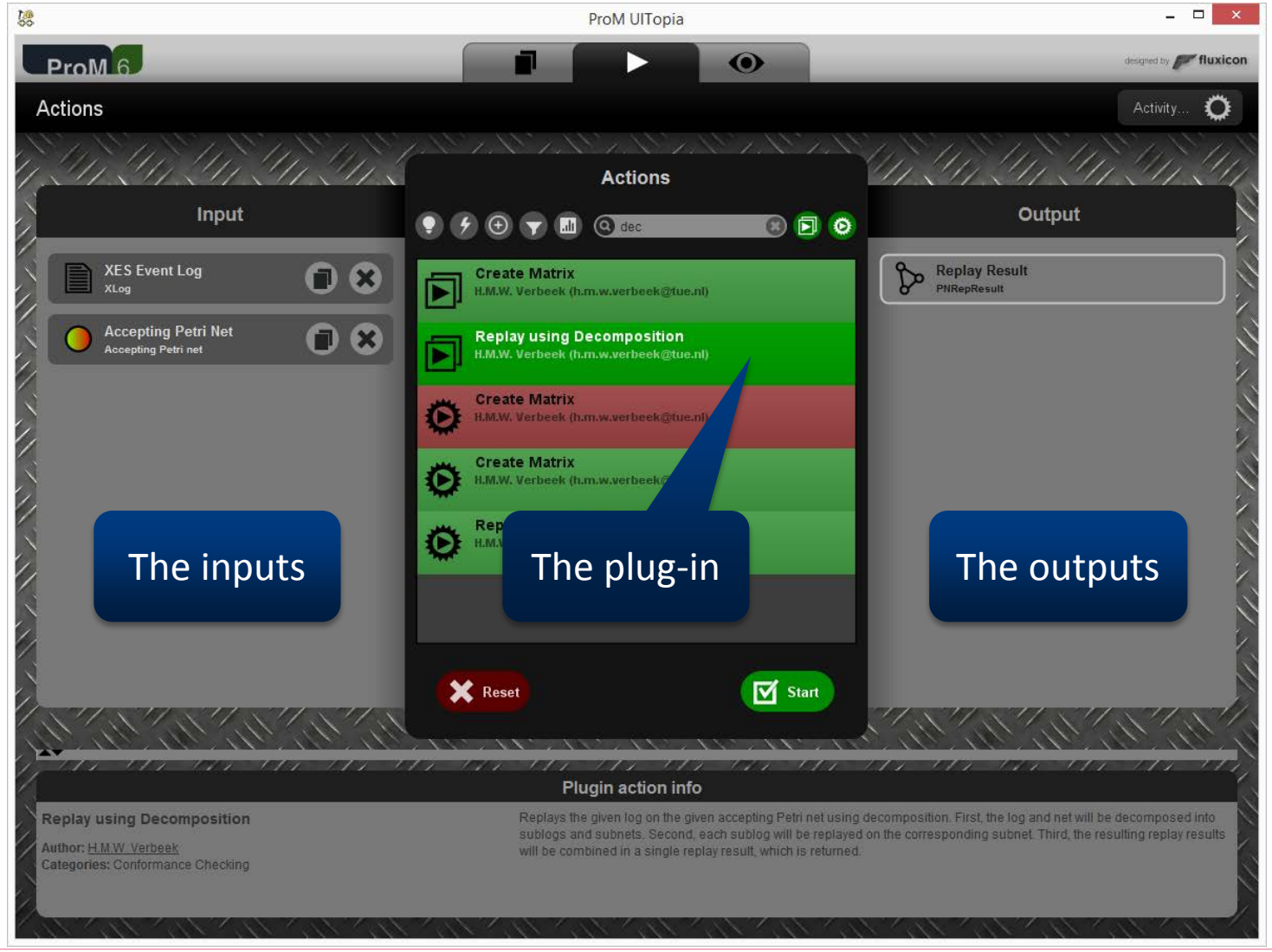
- Decomposed replay sometimes much worse
  - On discovered nets
  - From less than 10 seconds to more than 1000 seconds
- Hide&Reduce replay never much worse
  - On discovered or provided nets
  - Typically faster if it takes more than 10 seconds
  - From more than 1000 seconds to just above 10 seconds

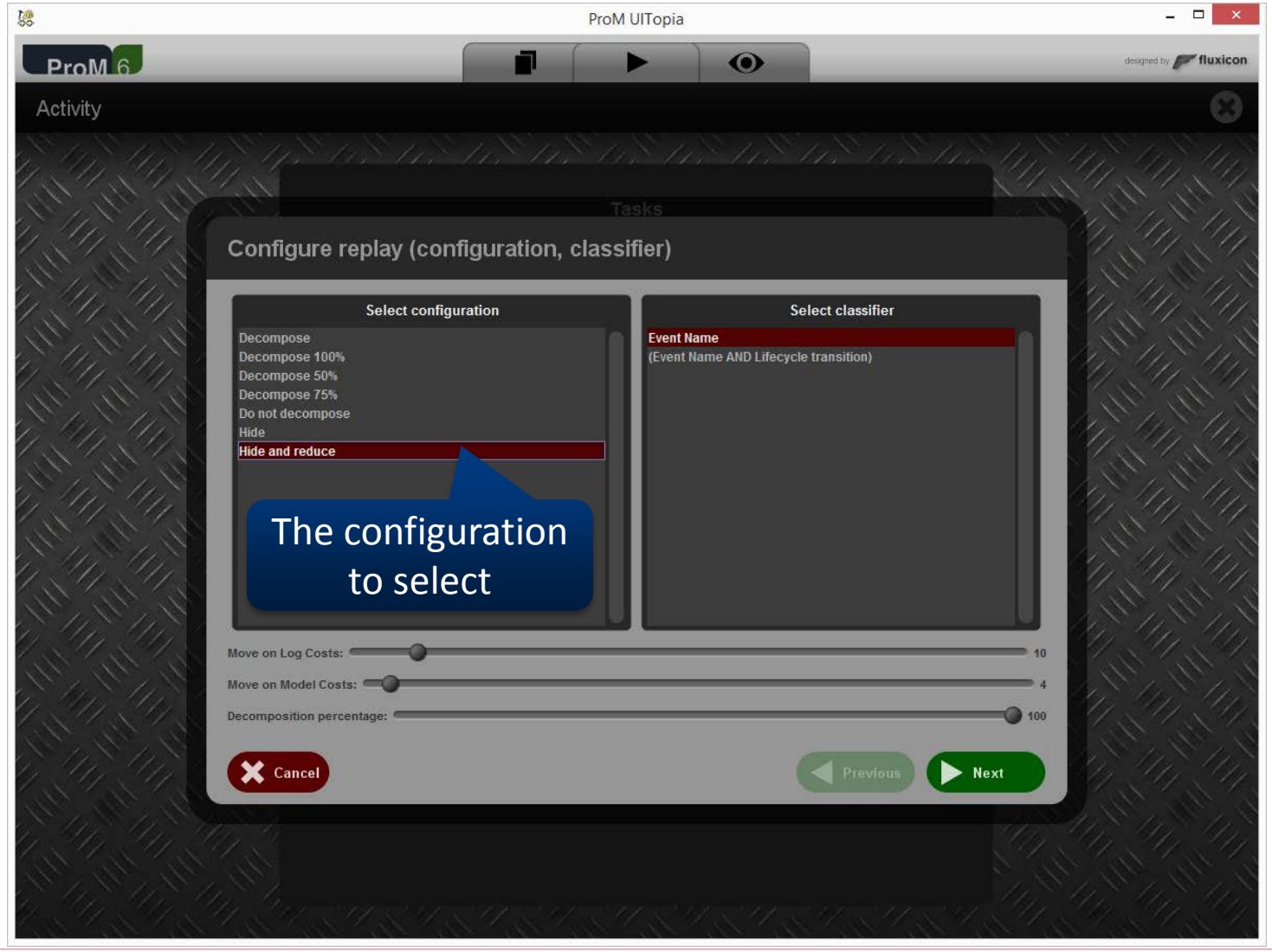
- On provided nets
  - Decomposed replay
    - as it is fastest
- On discovered nets
  - Hide&Reduce replay
    - as it provides the most answers in 1000 seconds

The Process Mining Toolkit

/ Wil van der Aalst / Peter van den Brand / Massimiliano de Leoni / Boudewijn van Dongen / Dirk Fahland / Christian Günther / Bart Hompes / Maikel Leemans / Sander Leemans / Xixi Lu / Felix Mannhardt / Eric Verbeek / Michael Westergaard

TU/e Technische Universiteit Eindhoven University of Technology
Where innovation starts


ProM UITopia
ProM 6
designed by fluxicon
Actions
Activity...
Input
XES Event Log
XLog
Accepting Petri Net
Accepting Petri net
Actions
dec
Create Matrix
H.M.W. Verbeek (h.m.w.verbeek@tue.nl)
Replay using Decomposition
H.M.W. Verbeek (h.m.w.verbeek@tue.nl)
Create Matrix
H.M.W. Verbeek (h.m.w.verbeek@tue.nl)
Create Matrix
Reset
Start
Output
Replay Result
PNRepResult
The inputs
The plug-in
The outputs
Plugin action info
Replay using Decomposition
Author: H.M.W. Verbeek
Categories: Conformance Checking
Replays the given log on the given accepting Petri net using decomposition. First, the log and net will be decomposed into sublogs and subnets. Second, each sublog will be replayed on the corresponding subnet. Third, the resulting replay results will be combined in a single replay result, which is returned.


ProM UITopia
ProM 6
designed by fluxicon
Activity
Tasks
Configure replay (configuration, classifier)
Select configuration
Decompose
Decompose 100%
Decompose 50%
Decompose 75%
Do not decompose
Hide
Hide and reduce
Select classifier
Event Name
(Event Name AND Lifecycle transition)
The configuration to select
Move on Log Costs:
10
Move on Model Costs:
4
Decomposition percentage:
100
Cancel
Previous
Next

- Preliminaries
  - Monolithical replay
  - Decomposed replay
- Hiding and Reduction
  - Problem with decomposed replay
  - Approach
  - Reduction rules
  - Results
- Wrapping up
  - Conclusions
  - Future work

- Cost-preserving reduction rules
  - Any reduction rule is fine, as long as it preserves the costs for any possible trace
  - Behavior-preserving implies cost-preserving, but might not equal
  - More rules?
    - Leading to better reduction?

Questions?