Partial Order Reduction for Reachability Games

September 12, 2019

Frederik M. Bønneland, Peter G. Jensen, Kim G. Larsen, Marco Muñiz, and Jiří Srba

Department of Computer Science
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

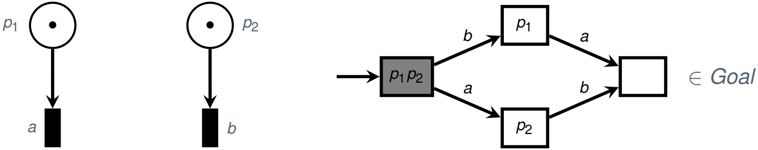
Overview

- (Generalized) Partial Order Reductions for games.
- Stable (Strategy Preserving) Reductions.
  - Stable reductions for Petri net games.
- Experiments
- Conclusion

Setting

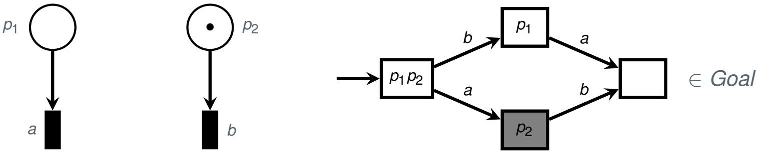
2-player Reachability games

- Pruning of redundant action interleavings.

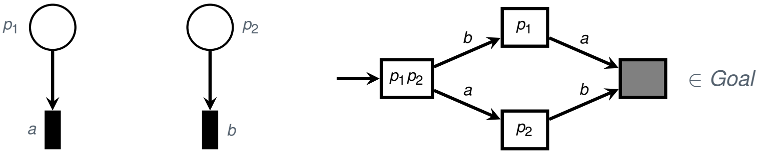


Partial Order Reductions

- Pruning of redundant action interleavings.

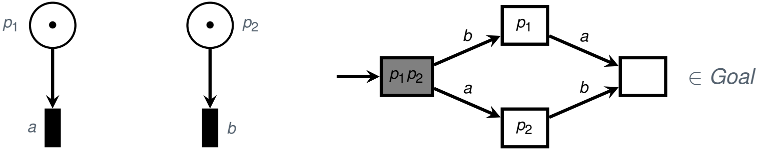


- Pruning of redundant action interleavings.

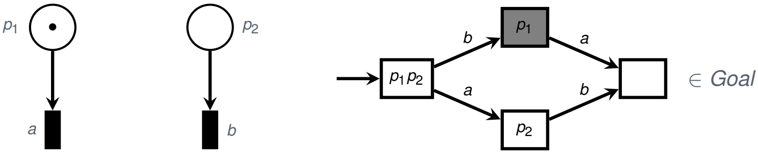


Partial Order Reductions

- Pruning of redundant action interleavings.

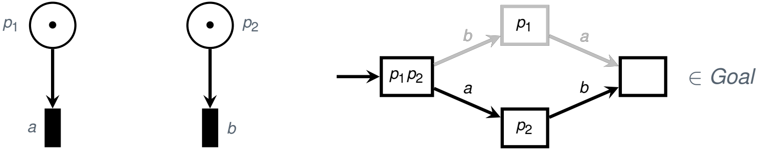


- Pruning of redundant action interleavings.



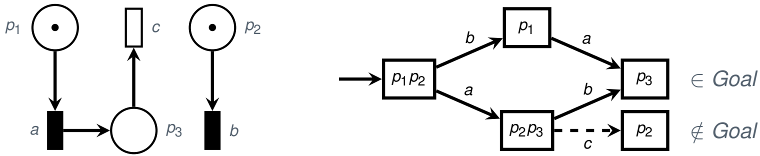
Partial Order Reductions

- Pruning of redundant action interleavings.



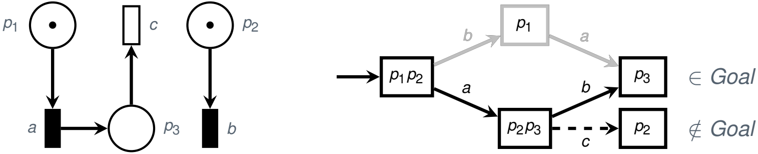
Partial Order Reductions

- Pruning of redundant action interleavings.



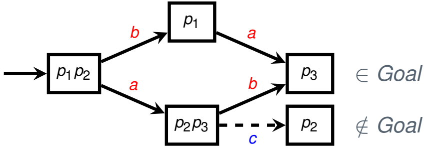
Partial Order Reductions

- Pruning of redundant action interleavings.



Preliminaries - Game Labelled Transition System

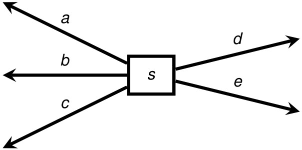
- Game Labelled Transition System (GLTS): $G = (\mathcal{S}, A_1, A_2, \rightarrow, Goal)$



- $en_1(s) = \{a \in A_1 \mid \exists s' \in \mathcal{S}.\ s \xrightarrow{a} s'\}$ for $s \in \mathcal{S}$.
- Similar definition for $en_2(s)$.
- **Example**: $en_1(p_2p_3) = \{b\}$ and $en_2(p_2p_3) = \{c\}$.
- $en(s) = en_1(s) \cup en_2(s)$.

Preliminaries - Reduction

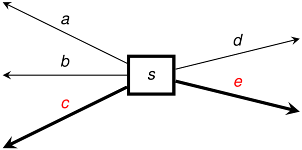
- Game Labelled Transition System (GLTS): $G = (\mathcal{S}, A_1, A_2, \rightarrow, Goal)$
- Reduction: $St : \mathcal{S} \rightarrow 2^A$



- **Example**: $St(s) = \{c, e\}$
- Reduced GLTS: $G_{St} = (\mathcal{S}, A_1, A_2, \underset{St}{\longrightarrow}, Goal)$
  - $s \overset{a}{\underset{St}{\longrightarrow}} s'$ iff $s \overset{a}{\rightarrow} s'$ and $a \in St(s)$

Preliminaries - Reduction

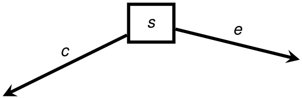
- Game Labelled Transition System (GLTS): $G = (\mathcal{S}, A_1, A_2, \rightarrow, Goal)$
- Reduction: $St : \mathcal{S} \rightarrow 2^A$



- **Example**: $St(s) = \{c, e\}$
- Reduced GLTS: $G_{St} = (\mathcal{S}, A_1, A_2, \underset{St}{\longrightarrow}, Goal)$
  - $s \xrightarrow[St]{a} s'$ iff $s \xrightarrow{a} s'$ and $a \in St(s)$

Preliminaries - Reduction

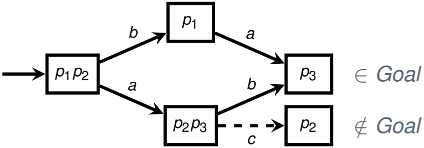
- Game Labelled Transition System (GLTS): $G = (\mathcal{S}, A_1, A_2, \rightarrow, Goal)$
- Reduction: $St : \mathcal{S} \rightarrow 2^A$



- **Example**: $St(s) = \{c, e\}$
- Reduced GLTS: $G_{St} = (\mathcal{S}, A_1, A_2, \xrightarrow[St]{}, Goal)$
  - $s \xrightarrow[St]{a} s'$ iff $s \xrightarrow{a} s'$ and $a \in St(s)$

Strategy: $\sigma : \mathcal{S} \to A_1 \cup \{\perp\}$

- $\sigma(s) = \perp$ implies $en_1(s) = \emptyset$



Example

$\sigma(p_1 p_2) = b$, $\sigma(p_1) = a$ and $\sigma(p_3) = \perp$.

Preliminaries - Winning Strategies

- Paths: $\pi = s_0 s_1 \cdots$ and $i \in \mathbb{N}^0$
- Maximal runs from a state s subject to a strategy: $\Pi^{max}_{G,\sigma}(s)$

Winning Strategies

σ is winning from s, iff

- for all $\pi \in \Pi^{max}_{G,\sigma}(s)$ there exists a position i s.t. $\pi_i \in \mathit{Goal}$

Reachability Conditions

Purpose

Iff $s \in S$ is winning in G then s is winning in G_{st}

Conditions

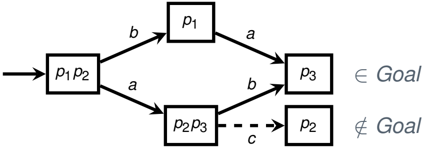
- **I** : No reduction for mixed states
- **W** : Non-stubborn transitions commute
- **R** : (Sufficient) Preservation of reachability
- **G1** and **G2** : Preservation of paths to mixed states
- **S** : Safety from mixed states
- **C** : Preservation of $P2$-cycles
- **D** : Preservation of Deadlocks

Theorem

If St *satisfies* ***I*****,** ***W*****,** ***R*****,** ***G1*****,** ***G2*****,** ***S*****,** ***C*****,** ***D*** *then for all* $s \in \mathcal{S}$ *state s is winning for the controller in G iff state s is winning for the controller in* G_{St}*.*

Reachability Conditions - Mixed States

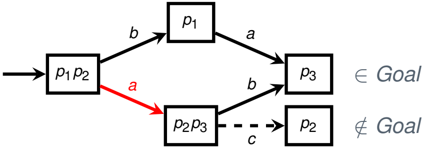
- **I**: If s is a mixed state then $en(s) \subseteq St(s)$.



- **Example**: State $p_2 p_3$ is a mixed state.

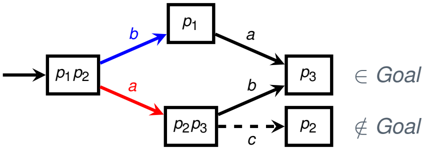
Reachability Conditions - Commutativity

- **W**: For all $w \in \overline{St(s)}^*$ and all $a \in St(s)$ if $s \xrightarrow{wa} s'$ then $s \xrightarrow{aw} s'$.



- **Example**: $St(p_1 p_2) = \{a\}$ is sufficient to satisfy **W**.

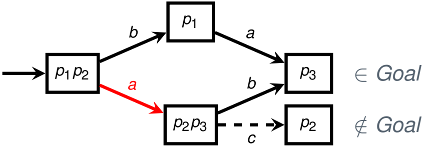
- **R**: $A_s(Goal) \subseteq St(s)$ for some interesting set of actions $A_s(Goal)$.



- If $s \notin Goal$, $w = a_1 \cdots a_n$, $s \xrightarrow{w} s'$, and $s' \in Goal$ then there exists $1 \leq i \leq n$ s.t. $a_i \in A_s(Goal)$.
- **Example**: Either $A_{p_1 p_2}(\{p_3\}) = \{a\}$ or $A_{p_1 p_2}(\{p_3\}) = \{b\}$ are both viable interesting sets.

Reachability Conditions - Game 1

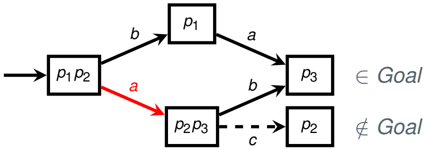
- **G1**: If $en_2(s) = \emptyset$, then for all $w \in \overline{St(s)}^*$ where $s \xrightarrow{w} s'$ then $en_2(s') = \emptyset$.



- Paths to mixed or environment states are preserved.
- **G2** is symmetric.

Reachability Conditions - Safe

- **S**: $en_1(s) \cap St(s) \subseteq safe(s)$ or $en_1(s) \subseteq St(s)$.



- $a \in A_1$ is safe in s if whenever $w \in (A_1 \setminus \{a\})^*$ and $s \xrightarrow{w} s'$ and $en_2(s') = \emptyset$ and $s \xrightarrow{aw} s''$ then $en_2(s'') = \emptyset$.
- Actions shifted to the front due to **W** may never lead to mixed or environment states.

Reachability Conditions - Deadlock & Cycle

- **C**: For all $a \in A_2$ if there exists $w \in A_2^{\omega}$ s.t. $s \xrightarrow{w}$ and a occurs infinitely often in w then $a \in St(s)$.
  - In order to preserve infinite paths of environment actions in the reduced GLTS.
- **D**: If $en_2(s) \neq \emptyset$ then there exists $a \in en_2(s) \cap St(s)$ s.t. for all $w \in \overline{St(s)}^{*}$ where $s \xrightarrow{w} s'$ we have $a \in en_2(s')$.
  - In order to preserve deadlocks in the reduced GLTS.

Instationation to Petri Net Games

Mixed states

- If $en_1(M) \neq \emptyset$ and $en_2(M) \neq \emptyset$ then $en(M) \subseteq St(M)$.

Transition Commutativity

- **W**: For all $w \in \overline{St(M)}^*$ and all $t \in St(M)$ if $M \xrightarrow{wt} M'$ then $M \xrightarrow{tw} M'$.

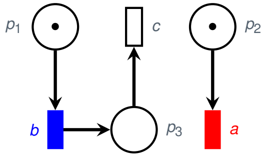
Petri Net Games - Preserving Environment Cycles

Conditions G1 and G2

If $en_2(M) = \emptyset$ then $T_2 \subseteq St(M)$.
Symmetric for Player 2.

Condition R

- If goals are all markings where there is 0 tokens in p_2 then all the transitions that decrease the number of tokens in p_2 are interesting.
- $\{a\}$ is a sufficient interesting set for this goal.



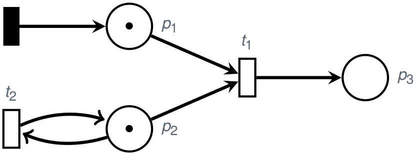
Condition S

Lemma

If $t^+ \cap {}^\bullet T_2 = \emptyset$ and ${}^-t \cap {}^\circ T_2 = \emptyset$ then t is safe in any marking.

- b is *not* safe since it contributes tokens to p_3.

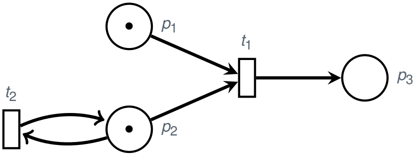
Condition C



- $\mathit{Finite} = \{\}$
- $\mathit{Marked} = \{\}$

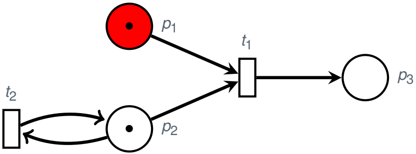
Petri Net Games - Preserving Environment Cycles

Condition C



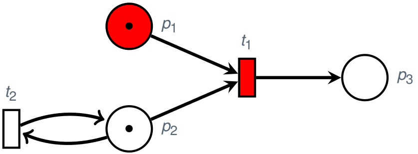
- *Finite* $= \{\}$
- *Marked* $= \{\}$

Condition C



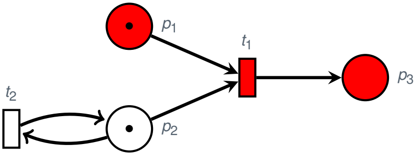
- $\mathit{Finite} = \{p_1\}$
- $\mathit{Marked} = \{\}$

Condition C



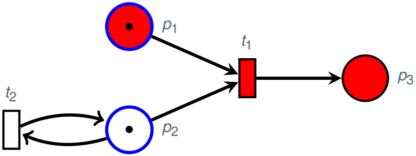
- *Finite* $= \{p_1, t_1\}$
- *Marked* $= \{\}$

Condition C



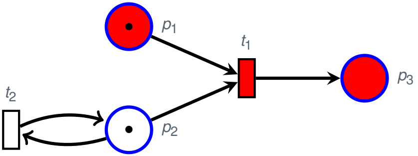
- $Finite = \{p_1, t_1, p_3\}$
- $Marked = \{\}$

Condition C



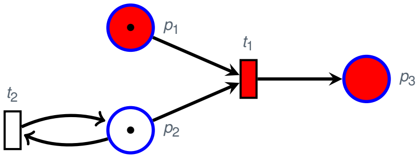
- *Finite* $= \{p_1, t_1, p_3\}$
- *Marked* $= \{p_1, p_2\}$

Condition C



- $Finite = \{p_1, t_1, p_3\}$
- $Marked = \{p_1, p_2, p_3\}$

Condition C

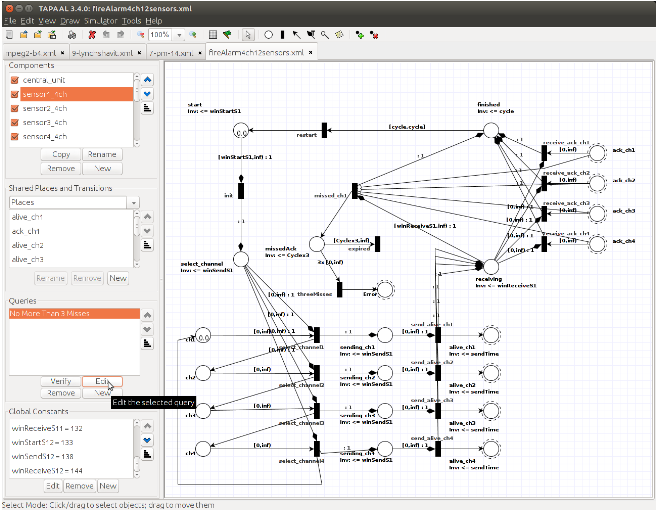


- $Finite = \{p_1, t_1, p_3\}$
- $Marked = \{p_1, p_2, p_3\}$

- A transition t may occur infinitely often if $t \notin Finite$ and all its pre-places can be marked ${}^\bullet t \subseteq Marked$.
- If t can occur infinitly often, $t \in St(s)$.

Implementation & TAPAAL

- Implemented in the TAPAAL verification tool suite.



Models

- (MW) Manufacturing Workflow.
  - Scaled on the number of requested features.
- (OW) Order Workflow.
  - Scaled on re-initialising.
- (NIM) Nim.
  - Scaled on the number of allowed pebbles and the allowable amount to add each round.
- (PCP) Producer Consumer System.
  - Scaled on the number of producers and consumers.
- (AIM) Autonomous Intersection Management.
  - Scaled on the number of cars, intersections, lane length, and speeds.
- (Lyngby) Railway Scheduling Problem.
  - Scaled on the number of moving trains.

Experiments

	Time (seconds)		Markings ×1000		Reduction	
Model	NORMAL	POR	NORMAL	POR	%T	%M
MW-40	735.2	0.2	69439	9	100	100
MW-50	1952.0	0.2	135697	11	100	100
MW-60	4417.0	0.3	234570	13	100	100
OW-10000	0.9	0.7	320	240	22	25
OW-100000	11.1	7.8	3200	2400	30	25
OW-1000000	137.7	109.8	32000	24000	20	25
NIM-5-49500	9.2	3.4	5054	892	63	82
NIM-7-49500	32.7	3.9	24039	1159	88	95
NIM-9-49500	165.1	4.7	114235	1522	97	99
NIM-11-49500	710.7	8.2	533516	1877	99	100

	Time (seconds)		Markings $\times 1000$		Reduction	
Model	NORMAL	POR	NORMAL	POR	%T	%M
PCS-2-2	24.0	19.9	9629	6554	17	32
PCS-2-3	116.1	90.9	61990	39114	22	37
PCS-2-4	399.1	283.3	240510	145109	29	40
AIM-13-100-6-11	117.9	46.6	1702	514	60	70
AIM-13-100-6-16	173.9	63.2	2464	746	64	70
AIM-13-150-9-16	337.0	219.9	3696	2454	35	34
AIM-13-150-9-21	408.0	294.1	4853	3331	28	31
AIM-14-150-9-16	449.9	278.4	4259	2865	38	33
AIM-15-150-9-16	534.5	384.9	4861	3204	28	34
LyngbySmall-2	3.4	1.0	1803	444	71	75
LyngbySmall-3	23.1	26.1	11473	10701	-13	7
LyngbySmall-4	144.3	193.3	65371	70008	-34	-7
Lyngby-2	3292.0	215.5	1511749	87214	93	94

Conclusion

- Partial Order Reduction for Games
  - Stable (Strategy Preserving)
- Reductions for both players
- Implemented in TAPAAL
- Encouraging experimental results

Future Work

- Timed Games?

Related Work

Start Pruning When Time Gets Urgent:
Partial Order Reduction for Timed Systems (CAV'18)

F. M. Bønneland, P. G. Jensen, K. G. Larsen, M. Muñiz and J. Srba