On Parity Game Preorders and the Logic of Matching Plays

SOFSEM 2016

Maciej Gazda and Tim Willemse*

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

/ Department of Mathematics and Computer Science

Context

Parity games:

- machinery for deciding (bi)simulations and model checking
  - modal μ-calculus: $\mathcal{K} \models \nu X.\mu Y.(\langle a \rangle X \vee \langle b \rangle Y)$
  - same for first-order extensions of μ-calculus

Context

Parity games:

- machinery for deciding (bi)simulations and model checking
  - modal μ-calculus: $\mathcal{K} \models \nu X.\mu Y.(\langle a \rangle X \vee \langle b \rangle Y)$
  - same for first-order extensions of μ-calculus
- Provide semantics to fixpoint logics
  - BES: X in $(\mu X = (X \wedge Y) \vee Z)$ $(\mu Y = X)$ $(\nu Z = Z)$
  - LFP: $[\mathbf{lfp}\ Xst.s\ R\ t \vee \exists_u\ s\ R\ u \wedge Xut]s_0 t_0$
  - PBES: $X(s_0, t_0)$ in $(\mu X(s,t) = s\ R\ t \vee \exists_u\ s\ R\ u \wedge X(u,t))$

Context

Parity games:

- machinery for deciding (bi)simulations and model checking
  - modal μ-calculus: $\mathcal{K} \models \nu X.\mu Y.(\langle a \rangle X \vee \langle b \rangle Y)$
  - same for first-order extensions of μ-calculus
- Provide semantics to fixpoint logics
  - BES: X in $(\mu X = (X \wedge Y) \vee Z)\ (\mu Y = X)\ (\nu Z = Z)$
  - LFP: $[\mathbf{lfp}\ Xst.s\,R\,t \vee \exists_u\ s\,R\,u \wedge Xut]s_0t_0$
  - PBES: $X(s_0, t_0)$ in $(\mu X(s,t) = s\,R\,t \vee \exists_u\ s\,R\,u \wedge X(u,t))$

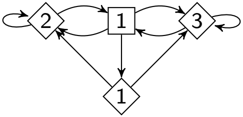
Underlying motivation for this work

- understand how to automatically/cheaply 'simplify' LFP/(P)BES formulae

Games

Parity games

- Two players: $\Diamond$ (Even) and $\Box$ (Odd)
- infinite duration
- played on a game graph



Games

Definition

A parity game is a tuple $(V, E, p, (V_\Diamond, V_\Box))$ where

- (V, E) is a directed graph
- V a set of vertices partitioned into $V_\Diamond$ and $V_\Box$
- E a total edge relation (i.e., at least one neighbour)
- $p : V \to \mathbb{N}$ a priority function (also called *colours*)

Games

Rules of the game:

Games

Rules of the game:

1. place a token on some vertex v
2. owner of the vertex v moves token to successor vertex v'
3. Repeat step 2

Games

Rules of the game:

1. place a token on some vertex v
2. owner of the vertex v moves token to successor vertex v'
3. Repeat step 2

Play: infinite sequence of vertices visited by token

Games

Rules of the game:

1. place a token on some vertex v
2. owner of the vertex v moves token to successor vertex v'
3. Repeat step 2

Play: infinite sequence of vertices visited by token

Winner of a play $\pi = v_1 v_2 v_3 \ldots$

- Let $\inf(\pi)$ be the set of priorities occurring infinitely often in π

Play π is winning for player $\Diamond$ iff $\min(\inf(\pi))$ is even.

Games

A strategy for player $\Diamond$ is a partial function $\varrho : V^* \times V_\Diamond \to V$

Games

A strategy for player $\Diamond$ is a partial function $\varrho : V^* \times V_\Diamond \to V$

$$\varrho(\underbrace{v_1 \ldots v_{n-1},}_{\text{history}} \underbrace{v_n}_{\text{currently}}) \in \underbrace{\{v \mid (v_n, v) \in E\}}_{\text{next position}}$$

Games

A strategy for player $\Diamond$ is a partial function $\varrho : V^* \times V_\Diamond \to V$

$$\varrho(\underbrace{v_1 \ldots v_{n-1},}_{\text{history}} \underbrace{v_n}_{\text{currently}}) \in \underbrace{\{v \mid (v_n, v) \in E\}}_{\text{next position}}$$

- Play $\pi = v_1 v_2 v_3 \ldots$ is consistent with strategy ϱ iff $\varrho(v_1 \ldots v_{i-1}, v_i) = v_{i+1}$ when $\varrho(v_1 \ldots v_{i-1}, v_i)$ is defined.
- Strategy ϱ is winning for $\Diamond$ from v iff $\Diamond$ wins all ϱ-consistent plays
- Player $\Diamond$ wins $W \subseteq V$ iff from all $v \in W$ she has a winning strategy

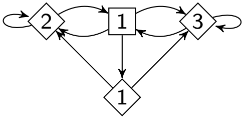
Games

Theorem (Positional determinacy)

- *Every vertex is won by either $\Diamond$ or $\Box$.*
- *Player $\Diamond$ wins a vertex w iff she has a memoryless strategy that is winning from w*

Theorem (Positional determinacy)

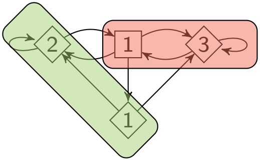
- *Every vertex is won by either ◊ or □.*
- *Player ◊ wins a vertex w iff she has a* ***memoryless strategy*** *that is winning from w*



Games

Theorem (Positional determinacy)

- *Every vertex is won by either $\Diamond$ or $\Box$.*
- *Player $\Diamond$ wins a vertex w iff she has a memoryless strategy that is winning from w*



Preorders for Games

Simplifying parity games (if game is explicit)

1. Use a behavioural equivalence relation (e.g. bisimulation)
2. Compute quotient (minimise)
3. Solve quotient game

Preorders for Games

Simplifying parity games (if game is explicit)

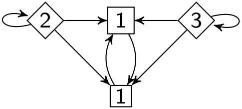
1. Use a behavioural equivalence relation (e.g. bisimulation)
2. Compute quotient (minimise)
3. Solve quotient game

Naively, a parity game $\approx$ Kripke structure

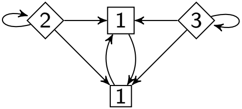
- Atomic propositions $AP = \mathbb{N} \times \{\Diamond, \Box\}$
- bisimulation ($\leftrightarroweq$) is 'sound':

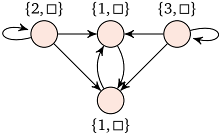
$$v \leftrightarroweq w \text{ implies } v \text{ and } w \text{ won by same player}$$

Preorders for Games
Example: bisimulation minimisation of a parity game



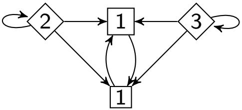
Example: bisimulation minimisation of a parity game

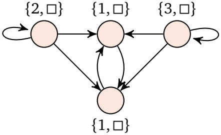


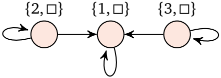


Preorders for Games

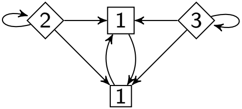
Example: bisimulation minimisation of a parity game

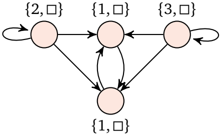


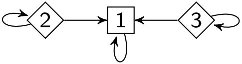


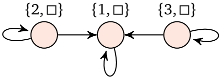


Example: bisimulation minimisation of a parity game



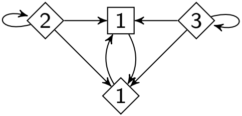




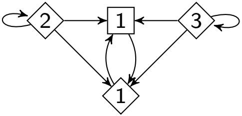


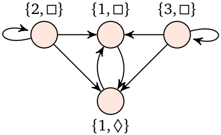
Preorders for Games
Example: bisimulation minimisation of a parity game

'Game-based' (bi)simulations required to minimise:

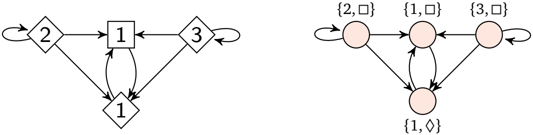


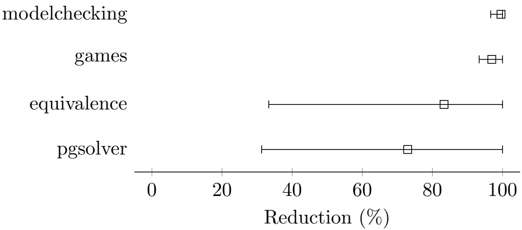
'Game-based' (bi)simulations required to minimise:





‘Game-based’ (bi)simulations required to minimise:





Preorders through Matching Plays

- Various ‘game-based’ (bi)simulations, eg:
  - direct simulation
  - delayed simulation
  - governed stuttering bisimulation
- Definitions seem ad-hoc
- Horrible proofs of soundness and transitivity

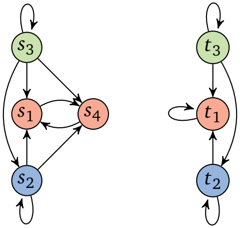
Preorders through Matching Plays

A more generic approach:

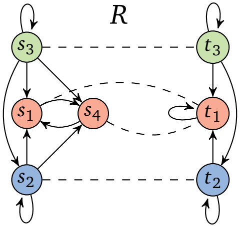
> *Any Kripke structure preorder defined through matching paths induces a parity game preorder by relating matching plays*

Preorders through Matching Plays

By example: matching paths for bisimulation

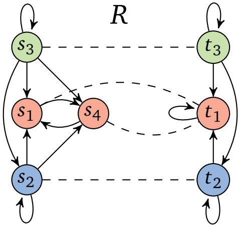


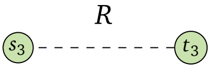
Preorders through Matching Plays
By example: matching paths for bisimulation



Preorders through Matching Plays

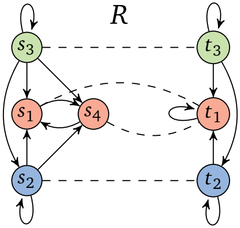
By example: matching paths for bisimulation

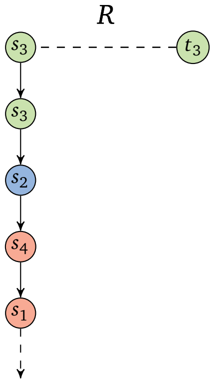




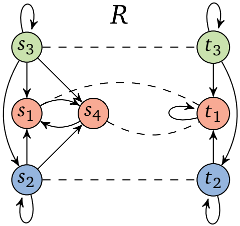
Preorders through Matching Plays

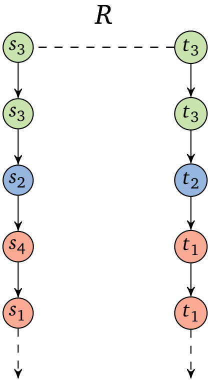
By example: matching paths for bisimulation



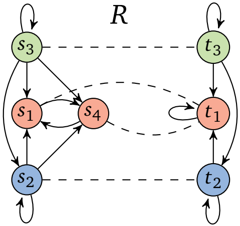


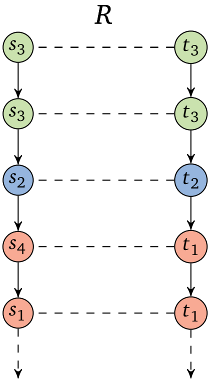
Preorders through Matching Plays
By example: matching paths for bisimulation





Preorders through Matching Plays
By example: matching paths for bisimulation





Preorders through Matching Plays

- Given a relation $R \subseteq S \times S$ on (states of) a Kripke structure
- Given a predicate $\mathrm{Rel}(R)$ which holds iff R is a Rel-relation
  - Think of $\mathrm{Rel}(R) \equiv$ 'R is a simulation relation'

Preorders through Matching Plays

- Given a relation $R \subseteq S \times S$ on (states of) a Kripke structure
- Given a predicate Rel(R) which holds iff R is a Rel-relation
  - Think of Rel(R) $\equiv$ 'R is a simulation relation'

Rel is characterised by a *path-matching* predicate Rel-match_R^L iff for all $R \subseteq S \times S$, Rel(R) holds iff for all $s\,R\,t$:

for all $\pi_s \in \text{Paths}(s)$ there is a $\pi_t \in \text{Paths}(t)$
such that $\text{Rel-match}_R^L(\pi_t, \pi_s)$

Preorders through Matching Plays

- Given a relation $R \subseteq S \times S$ on (states of) a Kripke structure
- Given a predicate $\mathsf{Rel}(R)$ which holds iff R is a Rel-relation
  - Think of $\mathsf{Rel}(R) \equiv$ 'R is a simulation relation'

Rel is characterised by a *path-matching* predicate Rel-match_R^L iff for all $R \subseteq S \times S$, $\mathsf{Rel}(R)$ holds iff for all $s\, R\, t$:

$$\text{for all } \pi_s \in \mathsf{Paths}(s) \text{ there is a } \pi_t \in \mathsf{Paths}(t) \text{ such that } \text{Rel-match}_R^L(\pi_t, \pi_s)$$

Example: *for Rel*$(R) \equiv$ *'*R *is a simulation relation', predicate Rel-match*$_R^L(\pi, \pi')$ *is given by* $\forall_i : L(\pi'_i) = L(\pi_i)$ *and* $\pi'_i\, R\, \pi_i$.

Preorders through Matching Plays

Rel	Rel-match$_R^L(\pi, \pi')$
trace inclusion	*for all* i, $L(\pi'_i) = L(\pi_i)$.
simulation	*for all* i, $L(\pi'_i) = L(\pi_i)$ *and* $\pi'_i \, R \, \pi_i$.
bisimulation	*for all* i, $L(\pi'_i) = L(\pi_i)$, $\pi'_i \, R \, \pi_i$ *and* $\pi_i \, R \, \pi'_i$.
stuttering simulation	*there is a non-decreasing, unbounded function* $f : \omega \to \omega$ *with* $f(1) = 1$ *such that for all* i *and all* $j \in [f(i), f(i+1))$, $L(\pi'_i) = L(\pi_j)$ *and* $\pi'_i \, R \, \pi_j$
stuttering bisimulation	*there is a non-decreasing, unbounded function* $f : \omega \to \omega$ *with* $f(1) = 1$ *such that for all* i *and all* $j \in [f(i), f(i+1))$, $L(\pi'_i) = L(\pi_j)$, $\pi'_i \, R \, \pi_j$ *and* $\pi_j \, R \, \pi'_i$.

Preorders through Matching Plays

Parity game relations through matching plays

Assume Rel (on KS) is characterised through matching paths

$R \subseteq V \times V$ is a parity game Rel-relation iff $v \, R \, w$ implies

for all $\Diamond$-strategies σ_v there is an $\Diamond$-strategy σ_w such that

for all σ_w-plays π_w there is a σ_v-play π_v satisfying:

$\text{Rel-match}_R^p(\pi_w, \pi_v)$

$v \sqsubseteq_{\mathsf{Rel}} w$ iff $v \, R \, w$ for some parity game Rel-relation R

Transitivity and Soundness

Theorem (Transitivity)

Relation $\sqsubseteq_{Rel}$ is transitive follows from:

1. *Monotonicity (in R) of Rel-match^p_R*
2. *For preorders R s.t. Rel(R), Rel-match^p_R is a preorder*

Transitivity and Soundness

Theorem (Transitivity)

Relation $\sqsubseteq_{Rel}$ is transitive follows from:

1. *Monotonicity (in R) of Rel-match$_R^p$*
2. *For preorders R s.t. Rel(R), Rel-match$_R^p$ is a preorder*

Consequence

- Every $\sqsubseteq_{\mathsf{Rel}}$, for Rel in the previous table, is a preorder

Transitivity and Soundness

Theorem (Soundness)

$v \sqsubseteq_{Rel} w$ implies if $\Diamond$ wins v then $\Diamond$ wins w follows if:

- *for all plays π_v won by $\Diamond$ and all plays π_w, if Rel-match$_R^p(\pi_w, \pi_v)$ then also π_w is won by $\Diamond$.*

Transitivity and Soundness

Theorem (Soundness)

$v \sqsubseteq_{Rel} w$ implies if $\Diamond$ wins v then $\Diamond$ wins w follows if:

- *for all plays π_v won by $\Diamond$ and all plays π_w, if Rel-match$^p_R(\pi_w, \pi_v)$ then also π_w is won by $\Diamond$.*

Consequence

- Every $\sqsubseteq_{\mathsf{Rel}}$, for Rel in the previous table, is sound

Existing and new parity game preorders

Theorem

- *direct simulation* $= \sqsubseteq_{simulation}$
- *governed bisimulation* $= \sqsubseteq_{bisimulation}$
- *governed stuttering bisimulation* $= \sqsubseteq_{stuttering\ bisimulation}$

Existing and new parity game preorders

Theorem

- *direct simulation* $= \sqsubseteq_{simulation}$
- *governed bisimulation* $= \sqsubseteq_{bisimulation}$
- *governed stuttering bisimulation* $= \sqsubseteq_{stuttering\ bisimulation}$

New parity game relations:

- governed trace inclusion
- governed stuttering simulation
- ...

Conclusions/Future work

- Uniform way of obtaining parity game relations
- ‘Easy’ proof of soundness and transitivity

Conclusions/Future work

- Uniform way of obtaining parity game relations
- 'Easy' proof of soundness and transitivity
- Logics that characterise parity game preorders
  - We have them...
  - Some are straightforward adaptations of those from KS
  - Some are subtly different
  - Apparently no generic way to transfer KS logic to PG logic

Conclusions/Future work

- Uniform way of obtaining parity game relations
- 'Easy' proof of soundness and transitivity
- Logics that characterise parity game preorders
    - We have them...
    - Some are straightforward adaptations of those from KS
    - Some are subtly different
    - Apparently no generic way to transfer KS logic to PG logic

- Decidability and complexity

Logic characterising parity game preorders

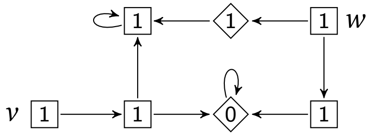
AHML $\phi, \psi ::= \mathbb{f} \mid \mathbb{t} \mid \neg\phi \mid \langle n \rangle_{\circledcirc} \phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle\langle n \rangle\rangle_{\circledcirc} \psi \mid \phi \langle\langle n \rangle\rangle_{\circledcirc}^{\infty} \psi$

Relation	Fragment	Grammar
$\sqsubseteq_{\text{simulation}}$	$\text{AHML}^{\leq}$	$\phi, \psi ::= \mathbb{t} \mid \langle n \rangle_{\Diamond} \phi \mid \phi \wedge \psi \mid \phi \vee \psi$
$\sqsubseteq_{\text{bisim}}$	$\text{AHML}^{\leftrightarrows}$	$\phi, \psi ::= \mathbb{t} \mid \neg\phi \mid \langle n \rangle_{\Diamond} \phi \mid \phi \wedge \psi \mid \phi \vee \psi$
$\sqsubseteq_{\text{stut. sim}}$	$\text{AHML}^{\leq_s}$	$\phi, \psi ::= \mathbb{t} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle\langle n \rangle\rangle_{\Diamond} \psi \mid \phi \langle\langle n \rangle\rangle_{\Diamond}^{\infty} \psi$
$\sqsubseteq_{\text{stut. bisim}}$	$\text{AHML}^{\leftrightarrows_s}$	$\phi, \psi ::= \mathbb{t} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \langle\langle n \rangle\rangle_{\Diamond} \psi \mid \phi \langle\langle n \rangle\rangle_{\Diamond}^{\infty} \psi$
$\sqsubseteq_{\text{trace inc}}$	$\text{AHML}^{\leq_t}$	$\phi ::= \mathbb{t} \mid \bigvee_{n \in N} \langle n \rangle_{\Diamond} \phi_n$ ($\emptyset \neq N \subset \mathbb{N}$ is a finite set of priorities)

Logic characterising parity game preorders

Arbitrary disjunctions are harmful in $\mathrm{AHML}^{\leq_t}$

$v \sqsubseteq_{\text{trace inc}} w$ and $w \sqsubseteq_{\text{trace inc}} v$.



- $w \models \langle 1 \rangle_{\Diamond} (\langle 1 \rangle_{\Diamond} \langle 1 \rangle_{\Diamond} \mathbb{t} \vee \langle 1 \rangle_{\Diamond} \langle 0 \rangle_{\Diamond} \mathbb{t})$
- $v \not\models \langle 1 \rangle_{\Diamond} (\langle 1 \rangle_{\Diamond} \langle 1 \rangle_{\Diamond} \mathbb{t} \vee \langle 1 \rangle_{\Diamond} \langle 0 \rangle_{\Diamond} \mathbb{t})$

Games

Algorithms (n nr. of vertices m nr. of edges, d nr. of priorities):

- 1993: *Recursive* alg. . . . . $\mathcal{O}(m\,n^{d})$
- 2000: *Small Progress Measures* alg. . . . . $\mathcal{O}(d\,m\,(\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$
- 2006: *Deterministic Subexponential* alg. . . . . $\mathcal{O}(n^{\sqrt{n}})$
- 2007: *Bigstep* alg. . . . . $\mathcal{O}(n^{d/3})$

Games

Algorithms (n nr. of vertices m nr. of edges, d nr. of priorities):

- 1993: *Recursive* alg. . . . . $\mathcal{O}(m\, n^{d})$
- 2000: *Small Progress Measures* alg. . . . . $\mathcal{O}(d\, m\, (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$
- 2006: *Deterministic Subexponential* alg. . . . . $\mathcal{O}(n^{\sqrt{n}})$
- 2007: *Bigstep* alg. . . . . $\mathcal{O}(n^{d/3})$

Exact complexity remains open