Games with Hierarchical Information and Model Checking

Łukasz Kaiser
kaiser@informatik.rwth-aachen.de

Mathematische Grundlagen der Informatik
RWTH Aachen

Lausanne 2007

Overview

Games

Automatic Structures

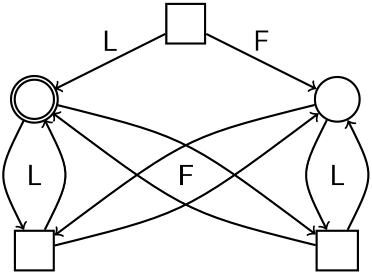
Game Quantifier

Model Checking

Conclusions and Future Work

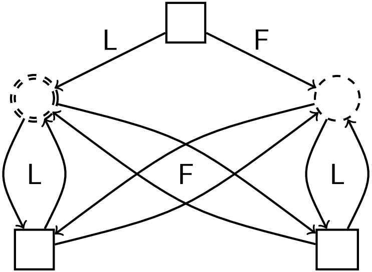
Games with Perfect Information

Imagine a coin, which players can **flip** or **leave** intact.



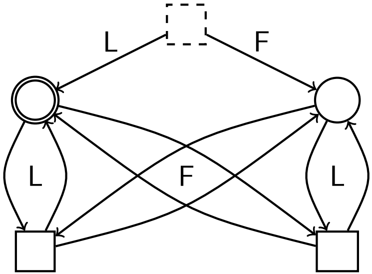
Games with Semiperfect Information

Player II can not see the actions of player I.



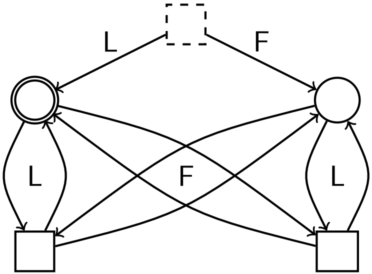
Games with Hierchical Information

- Two different players in **coalition** II
- The player on top makes a move **not noticed** by others



Games with Hierchical Information

- Two different players in **coalition** II
- The player on top makes a move **not noticed** by others



Players in coalition II can win together:

- the top player flips the coin
- the bottom one repeats the last move of player I

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level,

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$

- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1
  - **there exists a strategy for** I **on level** 2

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**,
  two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$
  but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy**
  good **for all counter-strategies**
  but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1
  - **there exists a strategy for** I **on level** 2
  - **so that for all strategies of** II **on level** 2

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1
  - **there exists a strategy for** I **on level** 2
  - **so that for all strategies of** II **on level** 2
  - ...

Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1
  - **there exists a strategy for** I **on level** 2
  - **so that for all strategies of** II **on level** 2
  - ...
  - **there exists a strategy for** I **on level** N

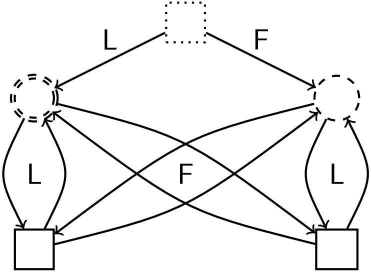
Winning with Hierarchical Information

- Two coalitions I and II on **N levels of information**, two players on each level ($2N$ players).

  On level i players **see moves on levels** $j \leq i$ but **can not see moves on levels** $j > i$
- usually winning defined when there **exists a winning strategy** good **for all counter-strategies** but this contradicts information advantage, here strategies must be given level-by-level, i.e. **coalition** I wins when
  - **there exists a strategy for** I **on level** 1
  - **so that for all strategies of** II **on level** 1
  - **there exists a strategy for** I **on level** 2
  - **so that for all strategies of** II **on level** 2
  - ...
  - **there exists a strategy for** I **on level** N
  - **so that for all strategies of** II **on level** N I wins the resulting play

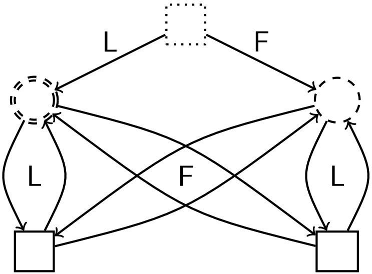
Games with Hierchical Information

- Moves of the player on top **not noticed** by any other player
- Moves of player I **not seen** by the bottom player



Games with Hierchical Information

- Moves of the player on top **not noticed** by any other player
- Moves of player I **not seen** by the bottom player



Player I can win:

- strategy of the bottom player given first
- player I makes a move to assure that the coin will be flipped

Making Moves in Turn and Deciding the Winners

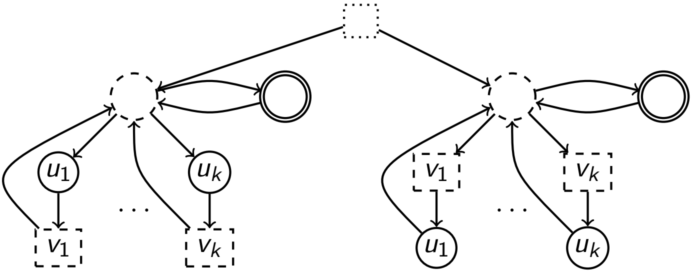
- A game is **alternating** when players make moves in turn, always in the same order, no matter what actions are taken.

Making Moves in Turn and Deciding the Winners

- A game is **alternating** when players make moves in turn, always in the same order, no matter what actions are taken.
- Given any hierarchical game, determining which coalition wins is **undecidable**, but it can be decided for alternating games.

Making Moves in Turn and Deciding the Winners

- A game is **alternating** when players make moves in turn, always in the same order, no matter what actions are taken.
- Given any hierarchical game, determining which coalition wins is **undecidable**, but it can be decided for alternating games.



Overview

Inductively Defined Structures

Structures in logic

$$\mathfrak{A} = (A, R_1, \ldots, R_k, f_1, \ldots, f_l)$$

- universe A is any set
- relations and functions are abstract subsets of A^n

Inductively Defined Structures

Structures in logic

$$\mathfrak{A} = (A, R_1, \ldots, R_k, f_1, \ldots, f_l)$$

- universe A is any set
- relations and functions are abstract subsets of A^n

Finitely presented structures

- universe defined inductively, elements can be trees or words
- relations and functions represented algorithmically
  - computed by a Turing machine
  - given by rewrite rules
  - recognised by an automaton

Inductively Defined Structures

Structures in logic

$$\mathfrak{A} = (A, R_1, \ldots, R_k, f_1, \ldots, f_l)$$

- universe A is any set
- relations and functions are abstract subsets of A^n

Finitely presented structures

- universe defined inductively, elements can be trees or words
- relations and functions represented algorithmically
  - computed by a Turing machine
  - given by rewrite rules
  - recognised by an automaton

Presentation: for an abstract structure find a finitely presented one.

- $\mathbb{N} \rightsquigarrow \{0,1\}^*$
- $\mathcal{P}(\mathbb{N}) \rightsquigarrow \{0,1\}^\omega$

Automatic Relations (Synchronous Transducers)

Relation $R(w_1, \dots, w_k)$ over (ω)-words (or trees) is **automatic** if it is recognised by a **synchronous** (Büchi, tree) automaton.

Such automaton works in fact over k-**tuples** of letters, i.e. over **convolution** of the words $w_1, \dots, w_k$.

$$w_1 \otimes \dots \otimes w_k = \begin{bmatrix} w_1[0] \\ \vdots \\ w_k[0] \end{bmatrix} \begin{bmatrix} w_1[1] \\ \vdots \\ w_k[1] \end{bmatrix} \begin{bmatrix} w_1[2] \\ \vdots \\ w_k[2] \end{bmatrix} \dots$$

Automatic Relations (Synchronous Transducers)

Relation $R(w_1, \ldots, w_k)$ over (ω)-words (or trees) is **automatic** if it is recognised by a **synchronous** (Büchi, tree) automaton.

Such automaton works in fact over k-**tuples** of letters, i.e. over **convolution** of the words $w_1, \ldots, w_k$.

$$w_1 \otimes \ldots \otimes w_k = \begin{bmatrix} w_1[0] \\ \vdots \\ w_k[0] \end{bmatrix} \begin{bmatrix} w_1[1] \\ \vdots \\ w_k[1] \end{bmatrix} \begin{bmatrix} w_1[2] \\ \vdots \\ w_k[2] \end{bmatrix} \ldots$$

Automatic Structures:

- have decidable first-order theory (automata manipulation)
- nonelementary complexity in the number of quantifier alternations
- other interesting model-theoretic properties
- software model checking, shape analysis

Interesting Automatic Structures

Presburger Arithmetic $(\mathbb{N}, +)$

- $n \rightsquigarrow$ **binary encoding**
- $+ \rightsquigarrow$ **bitwise** $+$ **with carry-over**

Interesting Automatic Structures

Presburger Arithmetic $(\mathbb{N}, +)$

- $n \rightsquigarrow$ **binary encoding**
- $+ \rightsquigarrow$ **bitwise** $+$ **with carry-over**

Skolem Arithmetic $(\mathbb{N}, \cdot)$

- $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n} \rightsquigarrow$ **binary encodings of** $\alpha_1, \alpha_2, \ldots, \alpha_n$
- $\cdot \rightsquigarrow +$ **on corresponding** α_i **components**

Interesting Automatic Structures

Presburger Arithmetic $(\mathbb{N}, +)$

- $n \rightsquigarrow$ **binary encoding**
- $+ \rightsquigarrow$ **bitwise** $+$ **with carry-over**

Skolem Arithmetic $(\mathbb{N}, \cdot)$

- $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_n^{\alpha_n} \rightsquigarrow$ **binary encodings of** $\alpha_1, \alpha_2, \dots, \alpha_n$
- $\cdot \rightsquigarrow +$ **on corresponding** α_i **components**

Boolean Algebra $(\mathcal{P}(\mathbb{N}), \cup, \cap, {}^{\mathrm{C}}, \emptyset, \mathbb{N})$

- $A \subseteq \mathbb{N} \rightsquigarrow w_A : w_A[i] = 1 \iff i \in A$
- $\cup \rightsquigarrow \max$
- $\cap \rightsquigarrow \min$
- ${}^{\mathrm{C}} \rightsquigarrow 1 - x$

Overview

Game Quantifier

Our structure: $(\Sigma^{\leq \omega}, R_1, \ldots, R_K)$

$\Game xy \; \varphi(x, y)$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = a$

Player II $\quad y =$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = a$

Player II $\quad y = b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = \ a\ b$

Player II $\quad y = \ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = a\ b$

Player II $\quad y = b\ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = \ a\ b\ b$

Player II $\quad y = \ b\ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\reflectbox{G}xy\ \varphi(x, y)$

Player I $\quad x = \ a\ b\ b$

Player II $\quad y = \ b\ b\ a$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = \ a\ b\ b\ a$

Player II $\quad y = \ b\ b\ a$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = \ a\ b\ b\ a$

Player II $\quad y = \ b\ b\ a\ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $x = a\ b\ b\ a\ a$

Player II $y = b\ b\ a\ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \dots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $x =\ a\ b\ b\ a\ a$

Player II $y =\ b\ b\ a\ b\ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy \ \varphi(x, y)$

Player I $\quad x = \ a \ b \ b \ a \ a \ldots$

Player II $\quad y = \ b \ b \ a \ b \ b$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \dots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = a\ b\ b\ a\ a \dots$

Player II $\quad y = b\ b\ a\ b\ b \dots$

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = \; a\ b\ b\ a\ a \ldots$

Player II $\quad y = \; b\ b\ a\ b\ b \ldots$

Can **Player** I play so that however **Player** II plays $\varphi(x, y)$ holds?

Game Quantifier

Our structure: $(\Sigma^{\leq\omega}, R_1, \ldots, R_K)$

$\Game xy\ \varphi(x, y)$

Player I $\quad x = a\ b\ b\ a\ a \ldots$

Player II $\quad y = b\ b\ a\ b\ b \ldots$

Can **Player** I play so that however **Player** II plays $\varphi(x, y)$ holds?

Considered before in infinitary logic:

$$\Game xy\ \varphi(x, y) \iff (\exists a_1 \forall b_1 \exists a_2 \forall b_2 \ldots)\ \varphi(a_1 a_2 \ldots, b_1 b_2 \ldots).$$

Decidability of FO+⅁

Game quantifier makes automata alternating

Decidability of FO+⅁

Game quantifier makes automata alternating

Lemma

If $R(\overline{x}, \overline{y}, \overline{z})$ *is* ω*-regular then* $⅁\overline{xy}\ R(\overline{x}, \overline{y}, \overline{z})$ *is* ω*-regular as well.*

Decidability of FO+$\Game$

Game quantifier makes automata alternating

Lemma

If $R(\overline{x}, \overline{y}, \overline{z})$ is ω-regular then $\Game\overline{xy}\ R(\overline{x}, \overline{y}, \overline{z})$ is ω-regular as well.

Proof method

$$\delta_{\Game}(q, \overline{z}) = \bigvee_{\overline{x} \in \Gamma^k} \bigwedge_{\overline{y} \in \Gamma^l} \delta_R(q, \overline{x} \otimes \overline{y} \otimes \overline{z})$$

where k is the size of $\overline{x}$, l the size of $\overline{y}$, δ are transition functions.

Decidability of FO+$\Game$

Game quantifier makes automata alternating

Lemma

If $R(\overline{x}, \overline{y}, \overline{z})$ is ω-regular then $\Game \overline{x} \overline{y}\ R(\overline{x}, \overline{y}, \overline{z})$ is ω-regular as well.

Proof method

$$\delta_{\Game}(q, \overline{z}) = \bigvee_{\overline{x} \in \Gamma^k} \bigwedge_{\overline{y} \in \Gamma^l} \delta_R(q, \overline{x} \otimes \overline{y} \otimes \overline{z})$$

where k is the size of $\overline{x}$, l the size of $\overline{y}$, δ are transition functions.

Alternating automata can be determinized.

Overview

Games

Automatic Structures

Game Quantifier

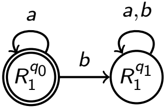
Model Checking

Conclusions and Future Work

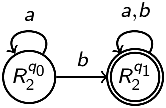
Model Checking Game Example

Take the formula $\exists x\ (R_1(x) \wedge R_2(x))$

$R_1 = \{a^\omega\}$



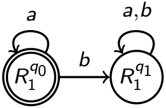
$R_2 = \{a, b\}^\omega \setminus \{a^\omega\}$



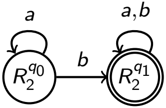
Model Checking Game Example

Take the formula $\exists x\ (R_1(x) \wedge R_2(x))$

$R_1 = \{a^\omega\}$



$R_2 = \{a, b\}^\omega \setminus \{a^\omega\}$



x must be given before $\wedge$ branch is chosen

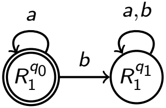
$\wedge$ on a **higher level of information** than $\exists x$

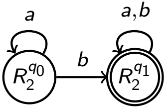
Model Checking Game Example

Take the formula $\exists x \; (R_1(x) \land R_2(x))$

$\exists x \; (R_1(x) \land R_2(x))$

$R_1 = \{a^\omega\}$ $\quad$ $R_2 = \{a, b\}^\omega \setminus \{a^\omega\}$





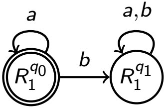
x must be given before $\land$ branch is chosen

$\land$ on a **higher level of information** than $\exists x$

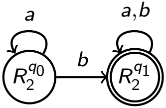
Model Checking Game Example

Take the formula $\exists x \ (R_1(x) \land R_2(x))$

$R_1 = \{a^\omega\}$

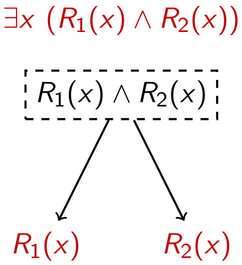


$R_2 = \{a, b\}^\omega \setminus \{a^\omega\}$



x must be given before $\land$ branch is chosen

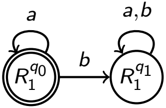
$\land$ on a **higher level of information** than $\exists x$



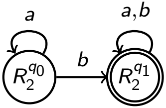
Model Checking Game Example

Take the formula $\exists x\ (R_1(x) \wedge R_2(x))$

$R_1 = \{a^\omega\}$



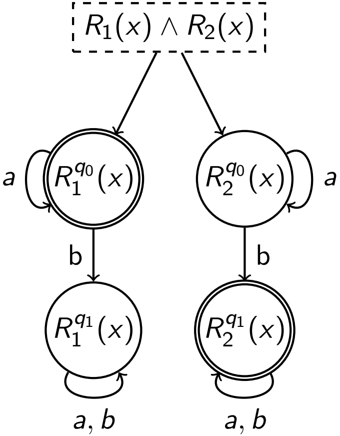
$R_2 = \{a, b\}^\omega \backslash \{a^\omega\}$



x must be given before $\wedge$ branch is chosen

$\wedge$ on a **higher level of information** than $\exists x$

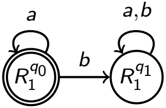
$\exists x\ (R_1(x) \wedge R_2(x))$



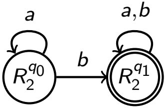
Model Checking Game Example

Take the formula $\exists x\ (R_1(x) \wedge R_2(x))$

$R_1 = \{a^\omega\}$



$R_2 = \{a, b\}^\omega \setminus \{a^\omega\}$

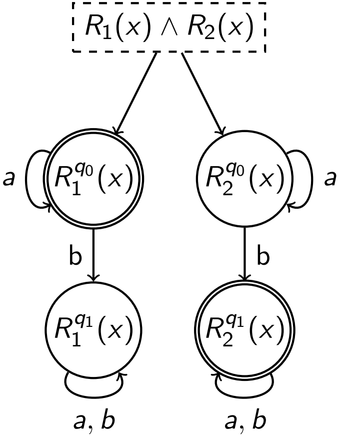


x must be given before $\wedge$ branch is chosen

$\wedge$ on a **higher level of information** than $\exists x$

- quantifier alternation $\rightsquigarrow$ different levels of information
- game quantifier $\rightsquigarrow$ opposing players on the same level of information

$\exists x\ (R_1(x) \wedge R_2(x))$



Hierarchical Games and FO+$\Game$

Alternating hierarchical (Muller) games are exactly model checking games for FO+$\Game$ on ω-automatic structures.

- for each FO+$\Game$ formula an alternating game can be built
- for every alternating game an automatic structure and a FO+$\Game$ formula can be given

Hierarchical Games and FO+∂

Alternating hierarchical (Muller) games are exactly model checking games for FO+∂ on ω-automatic structures.

- for each FO+∂ formula an alternating game can be built
- for every alternating game an automatic structure and a FO+∂ formula can be given

For an alternating game G with N levels let $W_{\mathrm{I}}^{G,v_0}(\overline{x})$ hold when the play respecting $\overline{x}$ starting from v_0 is winning for I.

Hierarchical Games and FO+$\rotatebox[origin=c]{180}{∂}$

Alternating hierarchical (Muller) games are exactly model checking games for FO+$\Game$ on ω-automatic structures.

- for each FO+$\Game$ formula an alternating game can be built
- for every alternating game an automatic structure and a FO+$\Game$ formula can be given

For an alternating game G with N levels let $W_{\mathrm{I}}^{G,v_0}(\overline{x})$ hold when the play respecting $\overline{x}$ starting from v_0 is winning for I.

Theorem

For any **alternating** *hierarchical Muller game G coalition I wins G starting from v_0 exactly if in $(\Sigma^{\leq\omega}, W_{\mathrm{I}}^{G,v_0})$ holds*

$$\Game x_1 y_1 \ldots \Game x_N y_N \; W_{\mathrm{I}}^{G,v_0}(x_1, y_1, \ldots, x_N, y_N).$$

Overview

What Good are these Games?

Can these games be efficiently solved algorithmically?

- can the lattice of antichains algorithm (Chatterjee, Doyen, Henzinger, Raskin) be extended to these games?
- efficient algorithm for model checking on automatic structures?

What Good are these Games?

Can these games be efficiently solved algorithmically?

- can the lattice of antichains algorithm (Chatterjee, Doyen, Henzinger, Raskin) be extended to these games?
- efficient algorithm for model checking on automatic structures?

In the asynchronous case, can these games lead to good heuristics?

- there are many heuristic algorithms for games
- better insight for program abstraction?

What Good are these Games?

Can these games be efficiently solved algorithmically?

- can the lattice of antichains algorithm (Chatterjee, Doyen, Henzinger, Raskin) be extended to these games?
- efficient algorithm for model checking on automatic structures?

In the asynchronous case, can these games lead to good heuristics?

- there are many heuristic algorithms for games
- better insight for program abstraction?

Other ways to extend the games and results.

- With stochastic moves, discounts and quantitative predicates can we check probabilistic or hybrid systems?
- Are these still model checking games when the arena is infinite?

Thank You