

Game Quantification on Automatic Structures

Hierarchical Model Checking Games

Łukasz Kaiser

kaiser@informatik.rwth-aachen.de

Mathematische Grundlagen der Informatik
RWTH Aachen

Computer Science Logic 2006

Introduction

Logic with Game Quantifier

Hierarchical Model Checking Games

Conclusions and Future Work

Build formulas using infinite string of (alternating) quantifiers

$$(\exists x_1 \forall y_1 \exists x_2 \forall y_2 \dots) R(x_1, y_1, x_2, y_2, \dots)$$

- ▶ R is a set of infinite sequences, classically open or closed
 $R = \bigvee_i R_i(x_1, y_1, \dots, x_i, y_i)$
- ▶ semantics given using Gale-Stewart games, first player wins $G(\exists\forall, R)$
- ▶ duality under negation follows from determinacy for Borel R (Martin)
- ▶ traditionally compared to infinitary or second-order logic

Automatic Structures

We are working on structures on finite and infinite words which are **presentations of ω -automatic structures**

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

Each R_i is recognised by a Muller automaton over $(\Sigma \cup \{\square\})^{\text{arity}(R_i)}$.
Finite words are encoded by adding \square^ω suffix.

$$w^1 \otimes \dots \otimes w^k = \begin{bmatrix} x_1^1 \\ \vdots \\ x_1^k \end{bmatrix} \begin{bmatrix} x_2^1 \\ \vdots \\ x_2^k \end{bmatrix} \dots \begin{bmatrix} \square \\ \vdots \\ x_l^k \end{bmatrix} \begin{bmatrix} \square \\ \vdots \\ x_{l+1}^k \end{bmatrix} \dots$$

Introduction

Logic with Game Quantifier

Hierarchical Model Checking Games

Conclusions and Future Work

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

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

Player I $x = a$

Player II $y =$

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

Player I $x = a$

Player II $y = b$

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

Player I $x = a \ b$

Player II $y = b$

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

Player I $x = a \ b$

Player II $y = b \ b$

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

Player I $x = a b b$

Player II $y = b b$

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

Player I $x = a b b$

Player II $y = b b a$

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

Player I $x = a b b a$

Player II $y = b b a$

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

Player I $x = a b b a$

Player II $y = b b a b$

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

Player I $x = a b b a a$

Player II $y = b b a b$

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

Player I $x = a b b a a$

Player II $y = b b a b b$

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

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

Player II $y = b b a b b$

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

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

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

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

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

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

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

Game Quantifier Formally

$$\exists xy \varphi(x, y) \iff$$

$$(\exists \text{ well-formed } f : \Gamma^* \times \Gamma^* \rightarrow \Gamma)$$

$$(\forall \text{ well-formed } g : \Gamma^* \times \Gamma^* \rightarrow \Gamma) \varphi(x_{fg}, y_{fg}),$$

$\Gamma = \Sigma \cup \{\square\}$, x_{fg} and y_{fg} are constructed inductively using f and g

$$x_{fg}[n] = f(x_{fg}|_{n-1}, y_{fg}|_{n-1})$$

$$y_{fg}[n] = g(x_{fg}|_n, y_{fg}|_{n-1})$$

well-formed f : if f outputs \square then it continues to output \square infinitely

Game Quantifier Formally

$$\exists xy \varphi(x, y) \iff$$

$$(\exists \text{ well-formed } f : \Gamma^* \times \Gamma^* \rightarrow \Gamma)$$

$$(\forall \text{ well-formed } g : \Gamma^* \times \Gamma^* \rightarrow \Gamma) \varphi(x_{fg}, y_{fg}),$$

$\Gamma = \Sigma \cup \{\square\}$, x_{fg} and y_{fg} are constructed inductively using f and g

$$x_{fg}[n] = f(x_{fg}|_{n-1}, y_{fg}|_{n-1})$$

$$y_{fg}[n] = g(x_{fg}|_n, y_{fg}|_{n-1})$$

well-formed f : if f outputs \square then it continues to output \square infinitely
Coincides with the classical definition

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

Game Formula Example

$$R(u, w, s, t) := \exists xy (y = u \rightarrow x = s) \wedge (y = w \rightarrow x = t)$$

Game Formula Example

$$R(u, w, s, t) := \exists xy (y = u \rightarrow x = s) \wedge (y = w \rightarrow x = t)$$

common prefix of s and t is **longer** than the **common prefix** of u and w

$$R(u, w, s, t) \equiv |s \sqcap t| > |u \sqcap w| \quad (s \neq t, u \neq w)$$

Game Formula Example

$$R(u, w, s, t) := \exists xy (y = u \rightarrow x = s) \wedge (y = w \rightarrow x = t)$$

common prefix of s and t is **longer** than the **common prefix** of u and w

$$R(u, w, s, t) \equiv |s \sqcap t| > |u \sqcap w| \quad (s \neq t, u \neq w)$$

(\Leftarrow) assume $|s \sqcap t| > |u \sqcap w|$

Player II will have to choose $y = u$ or $y = w$ before **Player I** chooses if $x = s$ or if $x = t$.

Game Formula Example

$$R(u, w, s, t) := \exists xy (y = u \rightarrow x = s) \wedge (y = w \rightarrow x = t)$$

common prefix of s and t is **longer** than the **common prefix** of u and w

$$R(u, w, s, t) \equiv |s \sqcap t| > |u \sqcap w| \quad (s \neq t, u \neq w)$$

(\Leftarrow) assume $|s \sqcap t| > |u \sqcap w|$

Player II will have to choose $y = u$ or $y = w$ before **Player I** chooses if $x = s$ or if $x = t$.

(\Rightarrow) in the other case

Player II knows if $x = s$ or if $x = t$ before choosing whether $y = u$ or $y = w$ and can therefore win.

Negating game quantifier reverses move order

$$\neg \exists \overline{xy} \varphi(\overline{x}, \overline{y}, \overline{z}) \iff \exists \overline{yx} \neg \varphi(\overline{x}, \overline{y}, \overline{z})$$

Negating game quantifier reverses move order

$$\neg \exists \overline{xy} \varphi(\overline{x}, \overline{y}, \overline{z}) \iff \exists \overline{yx} \neg \varphi(\overline{x}, \overline{y}, \overline{z})$$

- ▶ On automatic structures it follows from determinacy of finitely coloured Muller games.
- ▶ Can be proved using the general theorem of Martin for all Borel relations.

Negating game quantifier reverses move order

$$\neg \exists \overline{xy} \varphi(\overline{x}, \overline{y}, \overline{z}) \iff \exists \overline{yx} \neg \varphi(\overline{x}, \overline{y}, \overline{z})$$

- ▶ On automatic structures it follows from determinacy of finitely coloured Muller games.
- ▶ Can be proved using the general theorem of Martin for all Borel relations.

Negation normal form for $\text{FO}+\exists$.

Game quantifier makes automata alternating

Game quantifier makes automata alternating

Lemma

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

Game quantifier makes automata alternating

Lemma

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

Proof method

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

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

Game quantifier makes automata alternating

Lemma

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

Proof method

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

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

Alternating automata can be determinized with

double exponential blowup.

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$
use boolean combinations for $<, \leq, \geq, =$

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$
use boolean combinations for $<, \leq, \geq, =$
 $x \sqsubseteq y \equiv (\forall z \neq x) |x \sqcap z| \leq |x \sqcap y|$

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$

use boolean combinations for $<, \leq, \geq, =$

$$x \sqsubseteq y \equiv (\forall z \neq x) |x \sqcap z| \leq |x \sqcap y|$$

$$|x| \leq |y| \equiv (\forall x' \neq x) (\exists y' \neq y) |x \sqcap x'| \leq |y \sqcap y'|$$

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$

use boolean combinations for $<, \leq, \geq, =$

$x \sqsubseteq y \equiv (\forall z \neq x) |x \sqcap z| \leq |x \sqcap y|$

$|x| \leq |y| \equiv (\forall x' \neq x) (\exists y' \neq y) |x \sqcap x'| \leq |y \sqcap y'|$

Game quantifier gives us **prefix** and **equal length**.

Expressive power of $\text{FO}+\exists$

We have already defined $|s \sqcap t| > |u \sqcap w|$

use boolean combinations for $<, \leq, \geq, =$

$x \sqsubseteq y \equiv (\forall z \neq x) |x \sqcap z| \leq |x \sqcap y|$

$|x| \leq |y| \equiv (\forall x' \neq x) (\exists y' \neq y) |x \sqcap x'| \leq |y \sqcap y'|$

Game quantifier gives us **prefix** and **equal length**.

$\text{FO}+\exists$ can express **all regular relations**

- ▶ on the binary tree with successor relations
- ▶ on binary coded numbers with addition

Invariance under Inductive Automorphisms

Definition

The bijection $\pi : \Sigma^{\leq\omega} \rightarrow \Sigma^{\leq\omega}$ is **inductive** when there is a family of permutations of Σ $\{\pi_w\}_{w \in \Sigma^*}$ so that for each word u

$$\pi(u)[n] = \pi_{u|_{n-1}}(u[n]).$$

Invariance under Inductive Automorphisms

Definition

The bijection $\pi : \Sigma^{\leq\omega} \rightarrow \Sigma^{\leq\omega}$ is **inductive** when there is a family of permutations of Σ $\{\pi_w\}_{w \in \Sigma^*}$ so that for each word u

$$\pi(u)[n] = \pi_{u|_{n-1}}(u[n]).$$

Theorem

If ϕ is an inductive automorphism of $\mathfrak{A} = (\Sigma^{\leq\omega}, R_1, \dots, R_k)$ and R is definable in $FO+\exists$ on \mathfrak{A} , then

$$R(\bar{x}) \iff R(\overline{\phi(x)}).$$

Introduction

Logic with Game Quantifier

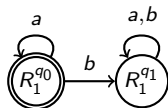
Hierarchical Model Checking Games

Conclusions and Future Work

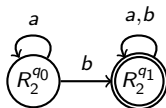
Model Checking Game Example

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

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



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

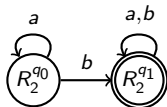
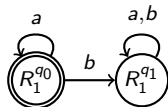


Model Checking Game Example

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 $\exists x$

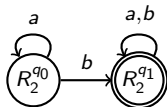
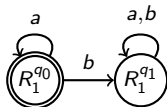
Model Checking Game Example

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

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

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

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



x must be given before \wedge branch is chosen
 \wedge on a **higher level of information** than $\exists x$

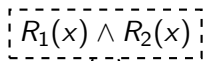
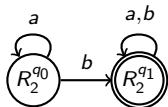
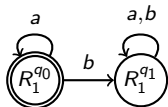
Model Checking Game Example

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

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

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

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



x must be given before \wedge branch is chosen
 \wedge on a **higher level of information** than $\exists x$

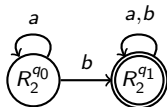
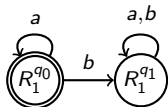
Model Checking Game Example

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

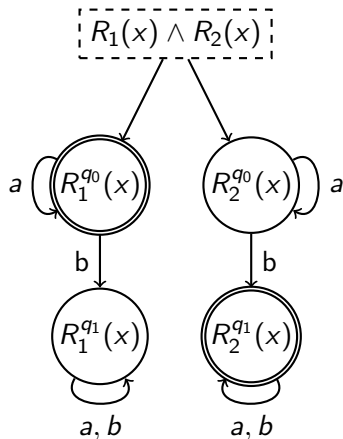
$\exists x (R_1(x) \wedge R_2(x))$

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

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



x must be given before \wedge branch is chosen
 \wedge on a **higher level of information** than $\exists x$



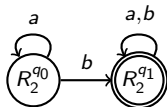
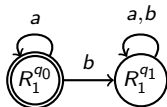
Model Checking Game Example

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

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

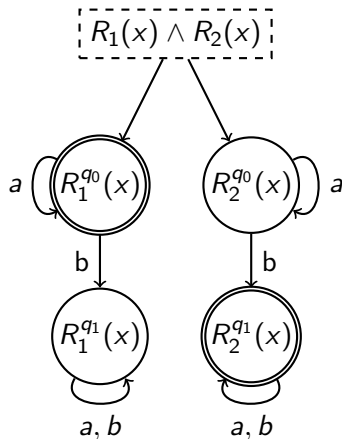
$$R_1 = \{a^\omega\}$$

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



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



Hierarchical Muller Games

- ▶ 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 $j > i$**

Hierarchical Muller Games

- ▶ 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,

Hierarchical Muller Games

- ▶ 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. **I** wins when
 - ▶ **there exists a strategy for I on level 1**

Hierarchical Muller Games

- ▶ 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. **I** wins when
 - ▶ **there exists a strategy for I on level 1**
 - ▶ **so that for all strategies of II on level 1**

Hierarchical Muller Games

- ▶ 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. **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**

Hierarchical Muller Games

- ▶ 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. **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**

Hierarchical Muller Games

- ▶ 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. **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**
 - ▶ ...

Hierarchical Muller Games

- ▶ 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. **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**

Hierarchical Muller Games

- ▶ 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. **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

Making Moves in Turn is Important

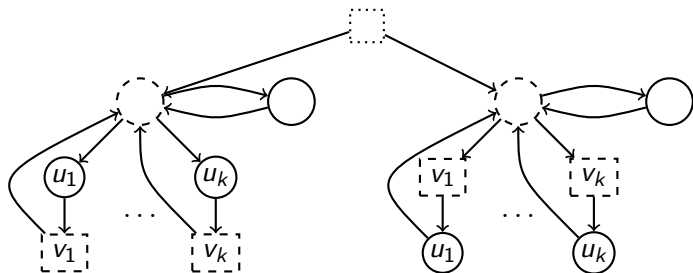
- ▶ A game is **alternating** when players make moves in turn, always in the same order, no matter what actions are taken.

Making Moves in Turn is Important

- ▶ 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 is Important

- ▶ 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.



Alternating hierarchical Muller games are exactly model checking games for $\text{FO}+\exists$ on ω -automatic structures.

- ▶ for each $\text{FO}+\exists$ formula an alternating game can be built
- ▶ for every alternating game an automatic structure and a $\text{FO}+\exists$ formula can be given

Alternating hierarchical Muller games are exactly model checking games for $\text{FO}+\exists$ on ω -automatic structures.

- ▶ for each $\text{FO}+\exists$ formula an alternating game can be built
- ▶ for every alternating game an automatic structure and a $\text{FO}+\exists$ formula can be given

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

Alternating hierarchical Muller games are exactly model checking games for $\text{FO}+\exists$ on ω -automatic structures.

- ▶ for each $\text{FO}+\exists$ formula an alternating game can be built
- ▶ for every alternating game an automatic structure and a $\text{FO}+\exists$ formula can be given

For an alternating game G with N levels let $W_I^{G, v_0}(\bar{x})$ hold when the play respecting \bar{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 $(\Sigma^\omega, W_I^{G, v_0})$ holds

$$\exists x_1 y_1 \dots \exists x_N y_N W_I^{G, v_0}(x_1, y_1, \dots, x_N, y_N)$$

Introduction

Logic with Game Quantifier

Hierarchical Model Checking Games

Conclusions and Future Work

Future Work (1): What Good are these Games?

Can these games be used or extended **beyond automatic structures?**

- ▶ these games capture automata determinisation in an abstract way
- ▶ tree-automatic relations can (most probably) be captured

Future Work (1): What Good are these Games?

Can these games be used or extended **beyond automatic structures?**

- ▶ these games capture automata determinisation in an abstract way
- ▶ tree-automatic relations can (most probably) be captured
- ▶ can xml-automatic relations can be captured?
- ▶ can simple set-based relations be captured? Rado graph?
- ▶ can we use these games for classes of formulas on complex structures?

Future Work (1): What Good are these Games?

Can these games be used or extended **beyond automatic structures?**

- ▶ these games capture automata determinisation in an abstract way
- ▶ tree-automatic relations can (most probably) be captured
- ▶ can xml-automatic relations can be captured?
- ▶ can simple set-based relations be captured? Rado graph?
- ▶ can we use these games for classes of formulas on complex structures?

Other interesting questions.

- ▶ Are these still model checking games when the arena is infinite?
- ▶ Are these model checking games for Henkin quantifiers when information flow is not hierarchical?
- ▶ We used long definition with alternation, can we do better?
- ▶ Can games with two levels be solved efficiently?
 - ▶ Are semiperfect-information games related to $\exists\forall$ -fragment?

Conclusions and Future Work (2)

$\text{FO}+\exists$ is a natural extension of first-order logic on structures on words

- ▶ **preserves regularity**
- ▶ **more expressive** than FO on weaker automatic structures
- ▶ invariant under **inductive automorphisms**
- ▶ what can be expressed using one, two, three \exists quantifiers?
- ▶ when can formulas that use \exists be written in FO?
- ▶ how does a relation defined in $\text{FO}+\exists$ depend on the presentation?

Thank You