

LOGIC AND GAMES ON AUTOMATIC STRUCTURES

Łukasz Kaiser

Mathematische Grundlagen der Informatik
RWTH Aachen

OVERVIEW

Logic and Games

Automatic Structures

Imperfect Information Games

Generalized Quantifiers

MODEL CHECKING PROBLEM

$$\mathcal{A} \models \varphi ?$$

MODEL CHECKING PROBLEM

$$\mathcal{A} \models \varphi ?$$

Deductive approach

$$\text{Th}(\mathbb{N}, +) \models \varphi ?$$

- **Objects:** formulas

MODEL CHECKING PROBLEM

$$\mathcal{A} \models \varphi ?$$

Deductive approach

$$\text{Th}(\mathbb{N}, +) \models \varphi ?$$

- **Objects:** formulas

Direct approach

$$(\mathbb{N}, +) \models \varphi ?$$

- **Objects:** numbers

MODEL CHECKING PROBLEM

$$\mathcal{A} \models \varphi ?$$

Deductive approach

$$\text{Th}(\mathbb{N}, +) \models \varphi ?$$

- **Objects:** formulas

Direct approach

$$(\mathbb{N}, +) \models \varphi ?$$

- **Objects:** numbers

Presentation approach

$$(\mathbb{N}_2, \mathcal{A}_+) \models \varphi ?$$

- **Objects:** sequences of digits

MODEL CHECKING GAME

Idea: Construct a game \mathcal{G} so that

$\mathcal{A} \models \varphi ? \quad \rightsquigarrow \quad \text{who wins } \mathcal{G} ?$

MODEL CHECKING GAME

Idea: Construct a game \mathcal{G} so that

$\mathcal{A} \models \varphi ? \quad \rightsquigarrow \quad \text{who wins } \mathcal{G} ?$

Players:

MODEL CHECKING GAME

Idea: Construct a game \mathcal{G} so that

$\mathcal{A} \models \varphi ? \quad \rightsquigarrow \quad \text{who wins } \mathcal{G} ?$

Players:

- **Eloïse:** φ is true on \mathcal{A} !



MODEL CHECKING GAME

Idea: Construct a game \mathcal{G} so that

$\mathcal{A} \models \varphi ? \rightsquigarrow$ who wins \mathcal{G} ?

Players:

- **Eloïse:** φ is true on \mathcal{A} !



- **Abélard:** No, φ is false on \mathcal{A} !



MODEL CHECKING GAME: THEORY

$$\text{Th}(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

MODEL CHECKING GAME: THEORY

$$\text{Th}(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$



I will prove it by **induction**.

MODEL CHECKING GAME: THEORY

$$\text{Th}(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$



I will prove it by **induction**.



Show me for $m = 0$.

MODEL CHECKING GAME: THEORY

$$\text{Th}(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$



I will prove it by **induction**.



Show me for $m = 0$.



Let $n = 0$. $0 = 0 + 0$ is an **axiom**.

MODEL CHECKING GAME: THEORY

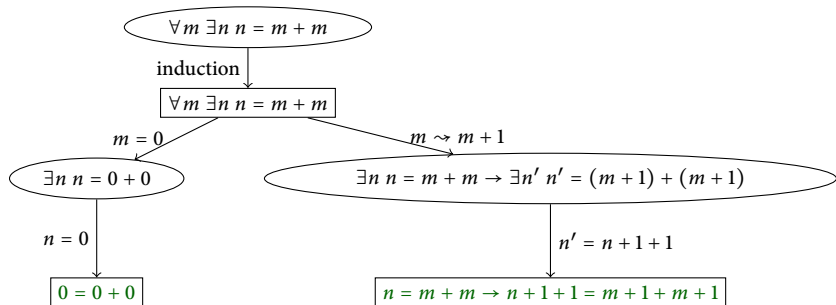
$$\text{Th}(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$



I will prove it by **induction**.

Show me for $m = 0$.

Let $n = 0$. $0 = 0 + 0$ is an **axiom**.



MODEL CHECKING GAME: STRUCTURE

$$(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

MODEL CHECKING GAME: STRUCTURE

$$(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

Position $\forall m$, Abélard chooses m .



Let $m = 17$.

MODEL CHECKING GAME: STRUCTURE

$$(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

Position $\forall m$, Abélard chooses m .



Let $m = 17$.

Position $\exists n$, Eloïse chooses n .



Let $n = 34$.

MODEL CHECKING GAME: STRUCTURE

$$(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

Position $\forall m$, Abélard chooses m .



Let $m = 17$.

Position $\exists n$, Eloïse chooses n .



Let $n = 34$.

Eloïse wins as $34 = 17 + 17$.

MODEL CHECKING GAME: STRUCTURE

$$(\mathbb{N}, +) \models \forall m \exists n n = m + m?$$

Position $\forall m$, Abélard chooses m .



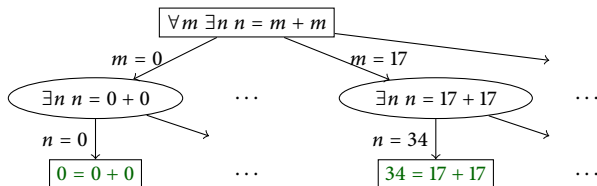
Let $m = 17$.

Position $\exists n$, Eloïse chooses n .



Let $n = 34$.

Eloïse wins as $34 = 17 + 17$.



MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$

MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$



Chooses the **first digit** of m .

MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$



Chooses the **first digit** of m .



Chooses the **first digit** of n .

MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$



Chooses the **first digit** of m .



Chooses the **first digit** of n but **do not show Abélard!**

MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$

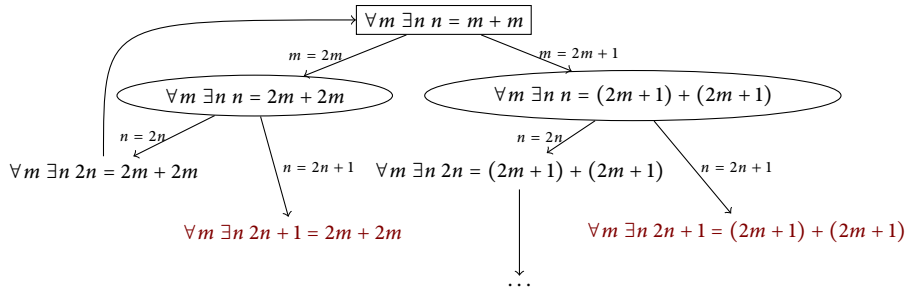


Chooses the **first digit** of m .



Chooses the **first digit** of n but **do not show Abélard!**

Make a step of $+$ (simplify) and **continue ...**



MODEL CHECKING GAME: PRESENTATION

$$(\mathbb{N}_2, \mathcal{A}_+) \models \forall m \exists n n = m + m?$$

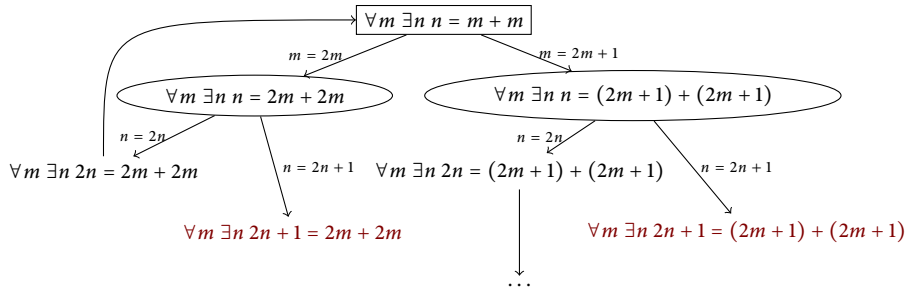


Chooses the **first digit** of m .



Chooses the **first digit** of n but **do not show Abélard!**

Make a step of $+$ (simplify) and **continue ...**



Games with imperfect information, unbounded duration, many players.

OVERVIEW

Logic and Games

Automatic Structures

Imperfect Information Games

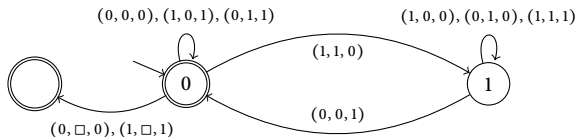
Generalized Quantifiers

AUTOMATIC STRUCTURES

Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation

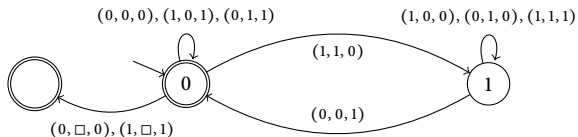
AUTOMATIC STRUCTURES

Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



AUTOMATIC STRUCTURES

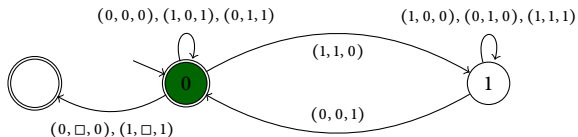
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



1	0	0	1	=	9	
+	1	0	1	□	=	5
	0	1	1	1	=	14

AUTOMATIC STRUCTURES

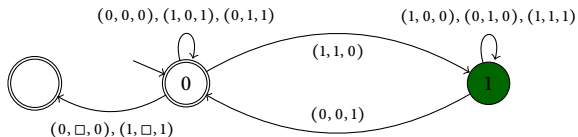
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



		1	0	0	1	=	9
+		1	0	1	□	=	5
		0	1	1	1	=	14

AUTOMATIC STRUCTURES

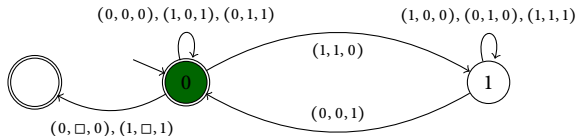
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



		1	0	0	1	=	9
+		1	0	1	□	=	5
<hr/>							
		0	1	1	1	=	14

AUTOMATIC STRUCTURES

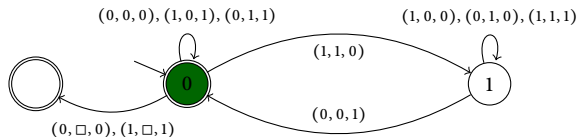
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



	1	0	0	1	=	9
+	1	0	1	□	=	5
	0	1	1	1	=	14

AUTOMATIC STRUCTURES

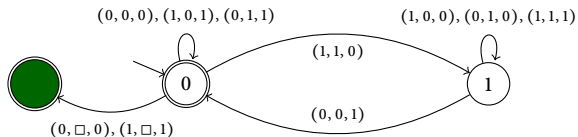
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



	1	0	0	1	= 9
+	1	0	1	□	= 5
	0	1	1	1	= 14

AUTOMATIC STRUCTURES

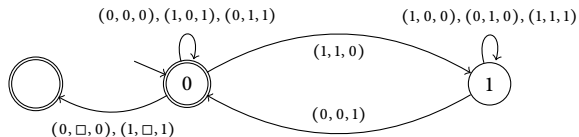
Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



	1	0	0	1	=	9
+	1	0	1	\square	=	5
<hr/>						
	0	1	1	1	=	14

AUTOMATIC STRUCTURES

Example: $+ = \{(x_2, y_2, z_2) : x + y = z\}$ is a regular relation



$$\begin{array}{r|cccccc}
 & 1 & 0 & 0 & 1 & = & 9 \\
 + & 1 & 0 & 1 & \square & = & 5 \\
 \hline
 & 0 & 1 & 1 & 1 & = & 14
 \end{array}$$

Relational structure \mathfrak{A} is automatic if

$$\mathfrak{A} \cong (L, R_1, \dots, R_n)$$

with L, R_i regular

VARIOUS AUTOMATIC STRUCTURES

Presburger Arithmetic $(\mathbb{N}, +)$ [word-automatic]

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

VARIOUS AUTOMATIC STRUCTURES

Presburger Arithmetic $(\mathbb{N}, +)$ [word-automatic]

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

Skolem Arithmetic (\mathbb{N}, \cdot) [tree-automatic]

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

VARIOUS AUTOMATIC STRUCTURES

Presburger Arithmetic $(\mathbb{N}, +)$ [word-automatic]

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

Skolem Arithmetic (\mathbb{N}, \cdot) [tree-automatic]

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

Boolean Algebra $(\mathcal{P}(\mathbb{N}), \cup, \cap, \overset{C}{}, \emptyset, \mathbb{N})$ [ω -word-automatic]

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

OVERVIEW

Logic and Games

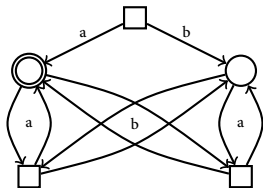
Automatic Structures

Imperfect Information Games

Generalized Quantifiers

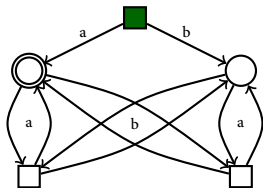
GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.



GAMES WITH PERFECT INFORMATION

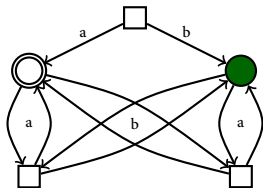
Games are represented by labelled directed graphs.



Plays are sequences of positions:

GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.



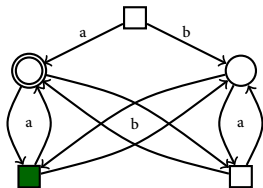
Plays are sequences of positions:



Chooses b .

GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.

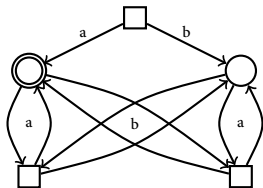


Plays are sequences of positions:



 Chooses b .  Chooses b again.

GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.

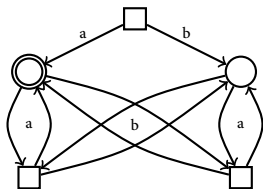


Plays are sequences of positions:



 Chooses b .  Chooses b again. **and so on ...**


GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.



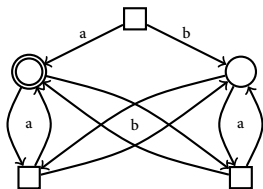
Plays are sequences of positions:

 Chooses b .  Chooses b again. **and so on ...**



Winning condition: Eloïse wants to visit  infinitely often.

GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.



Plays are sequences of positions:

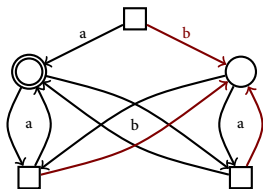
 Chooses b .  Chooses b again. **and so on ...**

Winning condition: Eloïse wants to visit \bigcirc infinitely often.



Winning strategy **guarantees** winning all consistent plays.

GAMES WITH PERFECT INFORMATION

Games are represented by labelled directed graphs.




Plays are sequences of positions:

 Chooses b .  Chooses b again. **and so on ...**

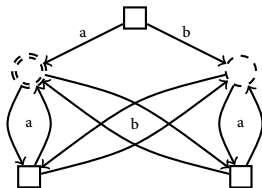
Winning condition: Eloïse wants to visit \bigcirc infinitely often.

Winning strategy guarantees winning all consistent plays.

 I will choose a on the right and b elsewhere.

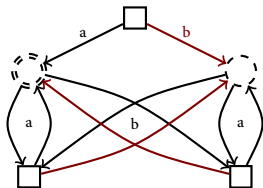
GAMES WITH IMPERFECT INFORMATION

Abélard cannot see the moves of Eloïse



GAMES WITH IMPERFECT INFORMATION

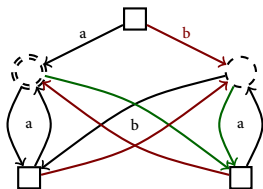
Abélard cannot see the moves of Eloïse



Let me try: I will always choose *b*.

GAMES WITH IMPERFECT INFORMATION

Abélard cannot see the moves of Eloïse



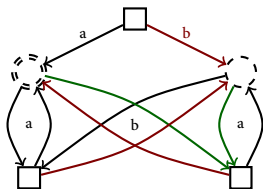
Let me try: I will always choose b .



Then I will choose b on the left and a on the right.

GAMES WITH IMPERFECT INFORMATION

Abélard cannot see the moves of Eloïse



Let me try: I will always choose b .



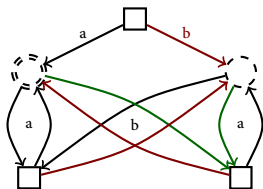
Then I will choose b on the left and a on the right.



You have no winning strategy!

GAMES WITH IMPERFECT INFORMATION

Abélard cannot see the moves of Eloïse



Let me try: I will always choose b .



Then I will choose b on the left and a on the right.



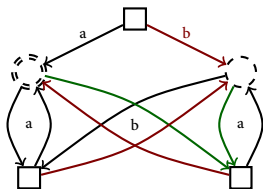
You have no winning strategy!



Neither do you!

GAMES WITH IMPERFECT INFORMATION

Abélard cannot see the moves of Eloïse



Let me try: I will always choose b .



Then I will choose b on the left and a on the right.



You have no winning strategy!



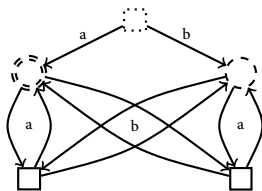
Neither do you!

In such imperfect information game it is **Abélard** who loses.

GAMES WITH HIERARCHICAL IMPERFECT INFORMATION



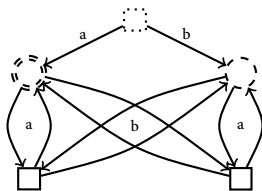
Eloïse, meet my friend **Adam**:



GAMES WITH HIERARCHICAL IMPERFECT INFORMATION



Eloïse, meet my friend **Adam**:



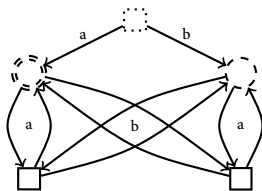
Hierarchy of information:

- **Adam** can see what everyone does
- **Eloïse** can only see what **Abélard** does
- **Abélard** can not see the moves of other players

GAMES WITH HIERARCHICAL IMPERFECT INFORMATION



Eloïse, meet my friend **Adam**:



Hierarchy of information:

- **Adam** can see what everyone does
- **Eloïse** can only see what **Abélard** does
- **Abélard** can not see the moves of other players

We say that Eloïse wins this game, because:

- for every strategy of **Abélard**
- there is a counter-strategy of **Eloïse**
- that leads to a winning play whatever **Adam** does

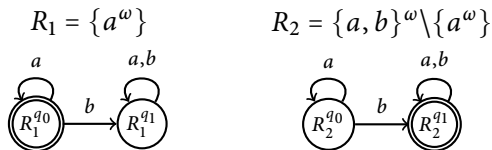
BUILDING THE MODEL CHECKING GAME

Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

BUILDING THE MODEL CHECKING GAME

Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

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



Note: x given before \wedge branch is chosen

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

BUILDING THE MODEL CHECKING GAME

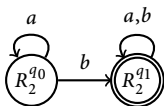
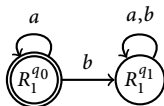
Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

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

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

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

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



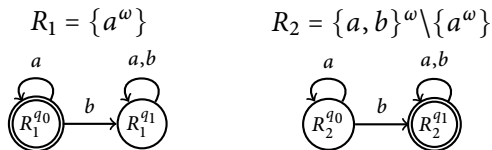
Note: x given before \wedge branch is chosen

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

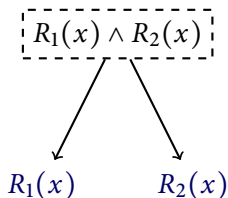
BUILDING THE MODEL CHECKING GAME

Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

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



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



Note: x given before \wedge branch is chosen

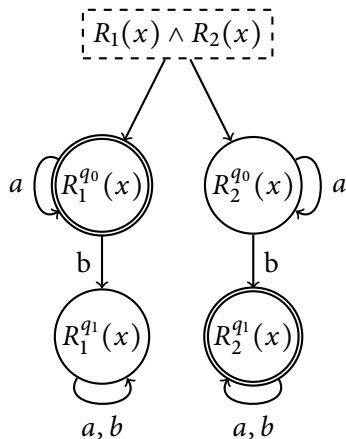
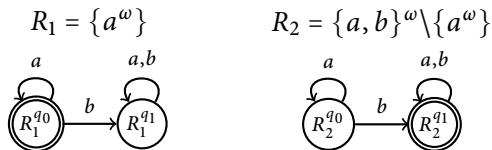
\wedge on a **higher level of information** than $\exists x$

BUILDING THE MODEL CHECKING GAME

Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

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

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



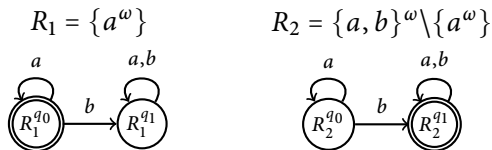
Note: x given before \wedge branch is chosen

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

BUILDING THE MODEL CHECKING GAME

Goal: $\mathcal{A} \models \varphi \rightsquigarrow$ who wins \mathcal{G} .

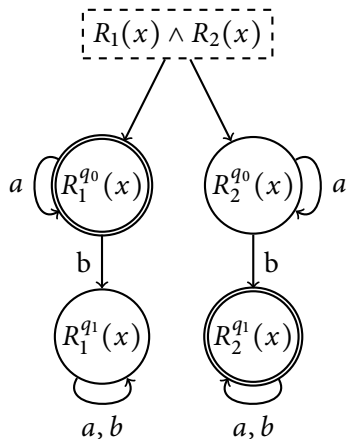
Take the formula $\exists x (R_1(x) \wedge R_2(x))$



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

- **quantifier alternation** \rightsquigarrow
different levels of information

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



HIERARCHICAL GAMES AND LOGIC

Theorem

Coalition \bigcirc wins the game $\text{MC}(\mathfrak{A}, \varphi)$ iff $\mathfrak{A} \models \varphi$.

HIERARCHICAL GAMES AND LOGIC

Theorem

Coalition \bigcirc wins the game $MC(\mathfrak{A}, \varphi)$ iff $\mathfrak{A} \models \varphi$.

Is there a strict correspondence between games and logic?

HIERARCHICAL GAMES AND LOGIC

Theorem

Coalition \bigcirc wins the game $\text{MC}(\mathfrak{A}, \varphi)$ iff $\mathfrak{A} \models \varphi$.

Is there a strict correspondence between games and logic?

Define $W_{\bigcirc}^G(\bar{x}) \iff$ the play according to \bar{x} is winning for \bigcirc .

Theorem

For any strictly hierarchical game \mathcal{G} , coalition \bigcirc wins \mathcal{G} iff

$$(\Sigma^{\leq \omega}, W_{\bigcirc}^G) \models \exists x_1 \forall x_2 \dots Qx_N W_{\bigcirc}^G(x_1, x_2, \dots, x_N).$$

OVERVIEW

Logic and Games

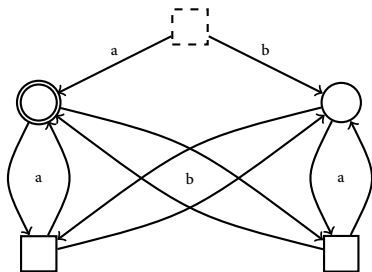
Automatic Structures

Imperfect Information Games

Generalized Quantifiers

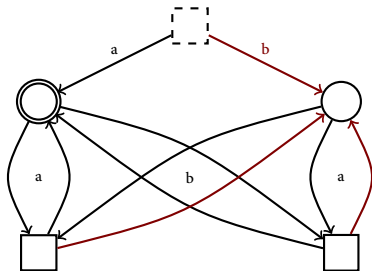
HIERCHICAL GAMES WITH SAME-LEVEL PLAYERS

- **Eloïse** and **Abélar** can see each others moves
- but **not** the moves of **Adam** (he can see everything)



HIERCHICAL GAMES WITH SAME-LEVEL PLAYERS

- **Eloïse** and **Abélar**d can see each others moves
- but **not** the moves of **Adam** (he can see everything)

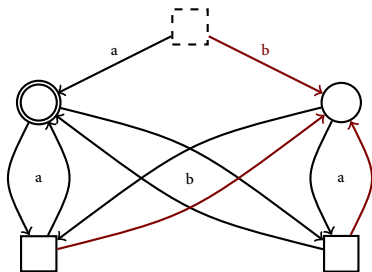


Now **Abélar** and **Adam** can win together:

- **Adam** chooses *b*
- **Abélar** repeats the last move of **Eloïse**

HIERCHICAL GAMES WITH SAME-LEVEL PLAYERS

- **Eloïse** and **Abélar**d can see each others moves
- but **not** the moves of **Adam** (he can see everything)



Now **Abélar**d and **Adam** can win together:

- **Adam** chooses *b*
- **Abélar**d repeats the last move of **Eloïse**

What does it correspond to in logic?

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a$

Abélard

$y =$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse $x = a$

Abélard $y = b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b$

Abélard

$y = b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b$

Abélard

$y = b b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b$

Abélard

$y = b b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b$

Abélard

$y = b b a$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a$

Abélard

$y = b b a$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a$

Abélard

$y = b b a b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a a$

Abélard

$y = b b a b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a a$

Abélard

$y = b b a b b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a a \dots$

Abélard

$y = b b a b b$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse

$x = a b b a a \dots$

Abélard

$y = b b a b b \dots$

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse $x = a b b a a \dots$

Abélard $y = b b a b b \dots$

Can **Eloïse** play so that however **Abélard** plays $\varphi(x, y)$ holds?

GAME QUANTIFIER

Our structure: $(\{a, b\}^\omega, R_1, \dots, R_K)$

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

Eloïse $x = a b b a a \dots$

Abélard $y = b b a b b \dots$

Can **Eloïse** play so that however **Abélard** plays $\varphi(x, y)$ holds?

Considered before in infinitary logic:

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

DECIDABILITY OF $\text{FO}[\exists]$

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.

Corollary

The logic $\text{FO}[\exists]$ is decidable on automatic structures.

DECIDABILITY OF FO[\exists]

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.

Corollary

The logic FO[\exists] is decidable on automatic structures.

There is a strict correspondence between FO[\exists] and hierarchical games

Corollary

It is decidable whether \bigcirc win in a hierarchical game \mathcal{G} .

DECIDABILITY OF FO[\exists]

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.

Corollary

The logic FO[\exists] is decidable on automatic structures.

There is a strict correspondence between FO[\exists] and hierarchical games

Corollary

It is decidable whether \bigcirc win in a hierarchical game \mathcal{G} .

Only for hierarchical games where players alternate!

GENERALIZED UNARY QUANTIFIERS

We consider the following **counting** quantifiers:

- **infinity quantifier**

$$\exists^\infty x \varphi(x) \equiv \{x : \varphi(x)\} \text{ is infinite}$$

- **modulo counting quantifiers**

$$\exists^{k \bmod m} x \varphi(x) \equiv |\{x : \varphi(x)\}| = k \bmod m$$

- **uncountability quantifier** (on ω -automatic structures)

$$\exists^{>\omega} x \varphi(x) \equiv \{x : \varphi(x)\} \text{ is uncountable}$$

Theorem

The logic **FO[C]** collapses to **FO** and thus is **decidable** on automatic structures.

MORE ON GENERALIZED UNARY QUANTIFIERS

These are **no other** generalized unary quantifiers that preserve regularity

- we obtained **full characterization** up to definability

MORE ON GENERALIZED UNARY QUANTIFIERS

These are **no other** generalized unary quantifiers that preserve regularity

- we obtained **full characterization** up to definability

Extends to structures with interpreted equality

- Example: **atomless boolean algebra**
- Technical difficulty with the **uncountability quantifier**
- Method: investigate **semigroups recognizing transitive relations**
- **Corollary:** every **countable ω -automatic structure** is automatic

MORE ON GENERALIZED UNARY QUANTIFIERS

These are **no other** generalized unary quantifiers that preserve regularity

- we obtained **full characterization** up to definability

Extends to structures with interpreted equality

- Example: **atomless boolean algebra**
- Technical difficulty with the **uncountability quantifier**
- Method: investigate **semigroups recognizing transitive relations**
- **Corollary:** every **countable ω -automatic structure** is automatic

Extends to **generalized-automatic structures**

SUMMARY

Problem studied:

$\mathcal{A} \models \varphi$? where \mathcal{A} **automatic (presentation)** and $\varphi \in \text{FO}[C, \exists]$

SUMMARY

Problem studied:

$\mathfrak{A} \models \varphi?$ where \mathfrak{A} **automatic (presentation)** and $\varphi \in \text{FO}[C, \exists]$

Main results:

- model-checking games for automatic structures
- decidability of hierarchical games
- analysis of memory structures for infinite games
- study of game quantification on automatic structures
- elimination of cardinality and counting quantifiers

SUMMARY

Problem studied:

$\mathfrak{A} \models \varphi?$ where \mathfrak{A} **automatic (presentation)** and $\varphi \in \text{FO}[C, \exists]$

Main results:

- model-checking games for automatic structures
- decidability of hierarchical games
- analysis of memory structures for infinite games
- study of game quantification on automatic structures
- elimination of cardinality and counting quantifiers

Thank You