First-order theories

Gabriele Puppis

LaBRI / CNRS

Definition

Fix a class $\mathcal{C}$ of structures (e.g. graphs) and a logic $\mathcal{L}$ (e.g. FO).

The **$\mathcal{L}$-theory of $\mathcal{C}$** is the set of all formulas in $\mathcal{L}$ that can be *satisfied by some structure* in $\mathscr{C}$.

The theory is **decidable** if there is an algorithm that receives formulas as input and tells whether they are in the theory or not.

Definition

Fix a class $\mathcal{C}$ of structures (e.g. graphs) and a logic $\mathcal{L}$ (e.g. FO).

The **$\mathcal{L}$-theory of $\mathcal{C}$** is the set of all formulas in $\mathcal{L}$ that can be *satisfied by some structure* in $\mathscr{C}$.

The theory is **decidable** if there is an algorithm that receives formulas as input and tells whether they are in the theory or not.

Examples

- first-order theory of the class of all graphs
- monadic theory of the class of all linear orders
- monadic theory of $\mathbb{N}$
- monadic theory of the grid

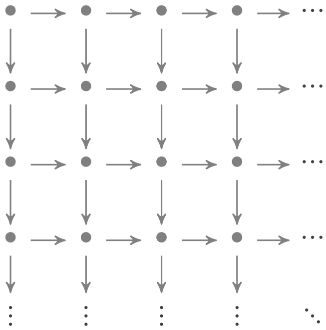
Undecidability of first-order theory

One cannot decide whether a given formula of $\mathrm{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

Undecidability of first-order theory

One cannot decide whether a given formula of $\mathsf{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

Given a Turing machine M, construct ψ_M defining its **halting runs**:

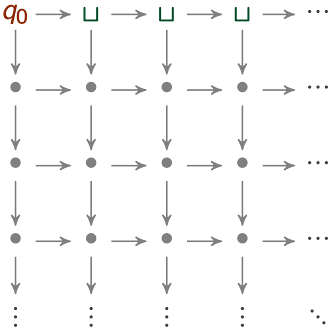


Undecidability of first-order theory

One cannot decide whether a given formula of $\mathrm{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

Given a Turing machine M, construct ψ_M defining its **halting runs**:

1. encode initial configuration by top row

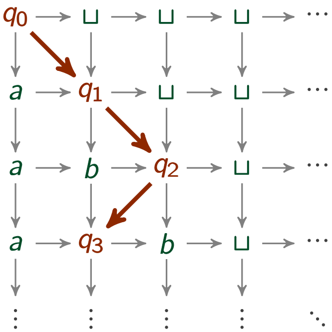


Undecidability of first-order theory

One cannot decide whether a given formula of $\mathsf{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

Given a Turing machine M, construct ψ_M defining its **halting runs**:

1. encode initial configuration by top row
2. encode next configurations by next rows

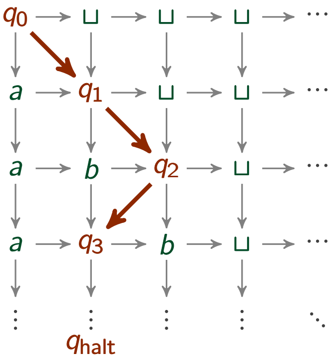


Undecidability of first-order theory

One cannot decide whether a given formula of $\mathrm{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

Given a Turing machine M, construct ψ_M defining its **halting runs**:

1. encode initial configuration by top row
2. encode next configurations by next rows
3. find a row with halting configuration



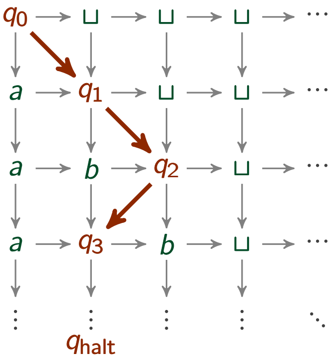
Undecidability of first-order theory

One cannot decide whether a given formula of $\mathsf{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

(and equally for $\mathsf{MSO}[E_1, E_2]$ over the grid $\mathbb{N} \times \mathbb{N}$)

Given a Turing machine M, construct ψ_M defining its **halting runs**:

1. encode initial configuration by top row
2. encode next configurations by next rows
3. find a row with halting configuration



MSO can even guess the labelling!

Undecidability of first-order theory

One cannot decide whether a given formula of $\mathrm{FO}[\Sigma, E_1, E_2]$ is satisfied over some **labelled grid**.

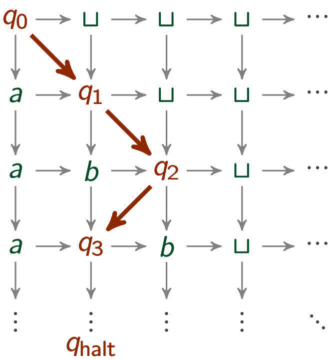
(and equally for $\mathrm{MSO}[E_1, E_2]$ over the grid $\mathbb{N} \times \mathbb{N}$)

uring machine M, construct ψ_M defining its **halting runs**:

configuration by top row

xt configurations by next rows

3. find a row with halting configuration



☞ MSO can even guess the labelling!

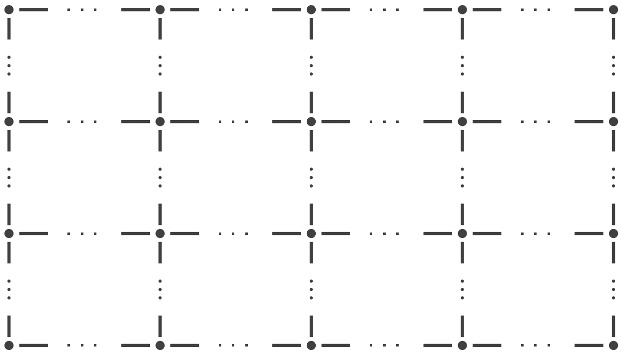
Consequences (Church '36, Turing '37, Trakhtenbrot '50, ...)

The FO theory of the class of all finite structures is undecidable (provided that signature contains a binary predicate besides =).

Consequences (Church '36, Turing '37, Trakhtenbrot '50, ...)

The FO theory of the class of all finite structures is undecidable (provided that signature contains a binary predicate besides =).

The MSO theory of any class of graphs with **unbounded grids as minors** is undecidable.

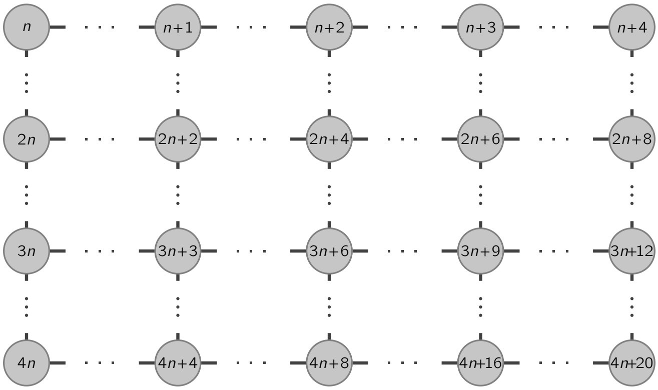


Consequences (Church '36, Turing '37, Trakhtenbrot '50, ...)

The FO theory of the class of all finite structures is undecidable (provided that signature contains a binary predicate besides =).

The MSO theory of any class of graphs with **unbounded grids as minors** is undecidable.

The MSO theory of $(\mathbb{N}, +)$ is undecidable.



Definition

Presburger arithmetic is the first-order theory of $(\mathbb{N}, +)$

Examples of Presburger formulas

$$\begin{aligned}
\psi_0 &= \exists x.\ \forall y.\ (x + y = y) \\
\varphi_{\leq}(x, y) &= \exists z.\ (y = x + z) \\
\varphi_{2\times}(x, y) &= (x + x = y) \\
\psi_\omega &= \forall x.\ \exists y.\ (x \leq y \ \wedge\ \neg x = y)
\end{aligned}$$

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

Originally proved by *quantifier elimination*. Here we use automata!

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

Originally proved by *quantifier elimination*. Here we use automata!

1. Encode numbers $x \in \mathbb{N}$ by **reverse binary expansions** $[x] \in \mathbb{B}^\star$

   e.g. $[4] = 001, \ [0] = \varepsilon, \ \ldots$

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

Originally proved by *quantifier elimination*. Here we use automata!

1. Encode numbers $x \in \mathbb{N}$ by **reverse binary expansions** $[x] \in \mathbb{B}^\star$

   e.g. $[4] = 001, \;\; [0] = \varepsilon, \;\; \ldots$

2. Encode sum relation $+ \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ by language $L_+ \subseteq (\mathbb{B} \times \mathbb{B} \times \mathbb{B})^\star$

   e.g. $\big[+(3, 1, 4)\big] \;=\; [3] \otimes [1] \otimes [4] \;=\; \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

Originally proved by *quantifier elimination*. Here we use automata!

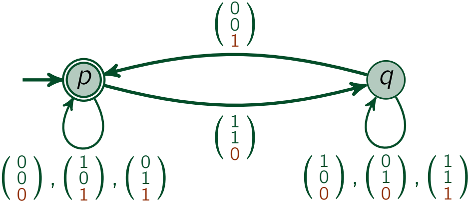
1. Encode numbers $x \in \mathbb{N}$ by **reverse binary expansions** $[x] \in \mathbb{B}^\star$

   e.g. $[4] = 001, \; [0] = \varepsilon, \; \ldots$

2. Encode sum relation $+ \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ by language $L_+ \subseteq (\mathbb{B} \times \mathbb{B} \times \mathbb{B})^\star$

   e.g. $\big[+(3,1,4)\big] = [3] \otimes [1] \otimes [4] = \begin{pmatrix}1\\1\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}$

$\mathcal{A}_+$:



Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

We inductively translate every Presburger formula $\varphi(x_1, \ldots, x_m)$ into a finite automaton $\mathcal{A}_\varphi$ over $\Sigma_m = \mathbb{B}^m$ such that

$$\mathscr{L}(\mathcal{A}_\varphi) \;=\; \left\{ [x_1] \otimes \cdots \otimes [x_m] \in \Sigma_m^\star \;\middle|\; (\mathbb{N}, +) \models \varphi(x_1, \ldots, x_m) \right\}$$

so as to reduce satisfiability of φ to emptiness of $\mathcal{L}(\mathcal{A}_\varphi)$.

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

We inductively translate every Presburger formula $\varphi(x_1, \ldots, x_m)$ into a finite automaton $\mathcal{A}_\varphi$ over $\Sigma_m = \mathbb{B}^m$ such that

$$\mathscr{L}(\mathcal{A}_\varphi) \; = \; \left\{ [x_1] \otimes \cdots \otimes [x_m] \in \Sigma_m^\star \;\middle|\; (\mathbb{N}, +) \vDash \varphi(x_1, \ldots, x_m) \right\}$$

so as to reduce satisfiability of φ to emptiness of $\mathcal{L}(\mathcal{A}_\varphi)$.

- For atomic formulas $x = y$ and $+(x, y, z)$, use automata $\mathcal{A}_=$ and $\mathcal{A}_+$

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

We inductively translate every Presburger formula $\varphi(x_1, \ldots, x_m)$ into a finite automaton $\mathcal{A}_\varphi$ over $\Sigma_m = \mathbb{B}^m$ such that

$$\mathscr{L}(\mathcal{A}_\varphi) = \left\{ [x_1] \otimes \cdots \otimes [x_m] \in \Sigma_m^\star \;\middle|\; (\mathbb{N}, +) \vDash \varphi(x_1, \ldots, x_m) \right\}$$

so as to reduce satisfiability of φ to emptiness of $\mathcal{L}(\mathcal{A}_\varphi)$.

- For atomic formulas $x = y$ and $+(x, y, z)$, use automata $\mathcal{A}_=$ and $\mathcal{A}_+$
- For disjunction $\varphi_1(\bar{x}) \vee \varphi_2(\bar{x})$, compute union of $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$

Decidability of Presburger arithmetic (Presburger '29)

One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

We inductively translate every Presburger formula $\varphi(x_1, \ldots, x_m)$ into a finite automaton $\mathcal{A}_\varphi$ over $\Sigma_m = \mathbb{B}^m$ such that

$$\mathscr{L}(\mathcal{A}_\varphi) \;=\; \left\{ [x_1] \otimes \cdots \otimes [x_m] \in \Sigma_m^\star \;\middle|\; (\mathbb{N}, +) \models \varphi(x_1, \ldots, x_m) \right\}$$

so as to reduce satisfiability of φ to emptiness of $\mathcal{L}(\mathcal{A}_\varphi)$.

- For atomic formulas $x = y$ and $+(x, y, z)$, use automata $\mathcal{A}_=$ and $\mathcal{A}_+$
- For disjunction $\varphi_1(\bar{x}) \vee \varphi_2(\bar{x})$, compute union of $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$
- For negation $\neg\varphi(\bar{x})$, compute the complement of $\mathcal{A}_\varphi$

Decidability of Presburger arithmetic (Presburger '29)

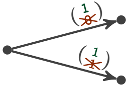
One can decide if a Presburger sentence ψ holds over $(\mathbb{N}, +)$.

We inductively translate every Presburger formula $\varphi(x_1, \ldots, x_m)$ into a finite automaton $\mathcal{A}_\varphi$ over $\Sigma_m = \mathbb{B}^m$ such that

$$\mathscr{L}(\mathcal{A}_\varphi) \;=\; \left\{ [x_1] \otimes \cdots \otimes [x_m] \in \Sigma_m^\star \;\middle|\; (\mathbb{N}, +) \vDash \varphi(x_1, \ldots, x_m) \right\}$$

so as to reduce satisfiability of φ to emptiness of $\mathcal{L}(\mathcal{A}_\varphi)$.

- For atomic formulas $x = y$ and $+(x, y, z)$, use automata $\mathcal{A}_=$ and $\mathcal{A}_+$
- For disjunction $\varphi_1(\bar{x}) \vee \varphi_2(\bar{x})$, compute union of $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$
- For negation $\neg\varphi(\bar{x})$, compute the complement of $\mathcal{A}_\varphi$
- For existential quantification $\exists y.\ \varphi(\bar{x}, y)$, project $\mathcal{A}_\varphi$ from Σ_{m+1} to Σ_m



Example of translation

Consider the (unsatisfiable) formula

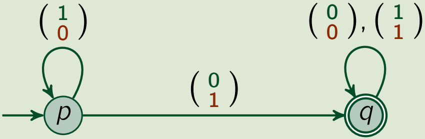
$$\psi = \exists x.\, \neg \exists y.\, (y = x + 1)$$

Example of translation

Consider the (unsatisfiable) formula

$$\psi = \exists x.\ \neg \exists y.\ (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$

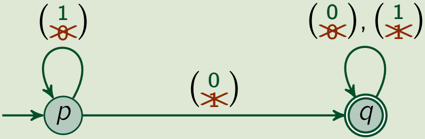


Example of translation

Consider the (unsatisfiable) formula

$$\psi \;=\; \exists x.\ \neg\, \exists y.\ (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$
2. Project away the encoding of y, thus capturing $\exists y.\ (y = x + 1)$

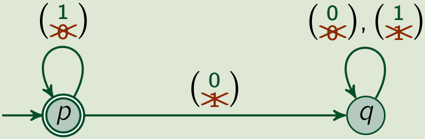


Example of translation

Consider the (unsatisfiable) formula

$$\psi \;=\; \exists x.\ \neg\, \exists y.\ (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$
2. Project away the encoding of y, thus capturing $\exists y.\ (y = x + 1)$
3. Complement the det. automaton, thus capturing $\neg\, \exists y.\ (y = x + 1)$

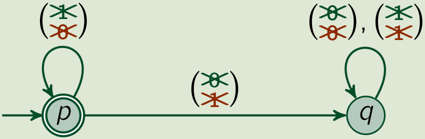


Example of translation

Consider the (unsatisfiable) formula

$$\psi = \exists x.\, \neg \exists y.\, (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$
2. Project away the encoding of y, thus capturing $\exists y.\, (y = x + 1)$
3. Complement the det. automaton, thus capturing $\neg \exists y.\, (y = x + 1)$
4. Project away the encoding of x, thus getting $\exists x.\, \neg \exists y.\, (y = x + 1)$

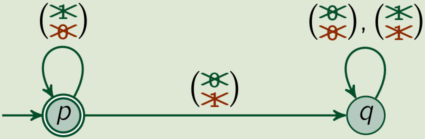


Example of translation

Consider the (unsatisfiable) formula

$$\psi \; = \; \exists x. \, \neg \, \exists y. \, (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$
2. Project away the encoding of y, thus capturing $\exists y. \, (y = x + 1)$
3. Complement the det. automaton, thus capturing $\neg \, \exists y. \, (y = x + 1)$
4. Project away the encoding of x, thus getting $\exists x. \, \neg \, \exists y. \, (y = x + 1)$



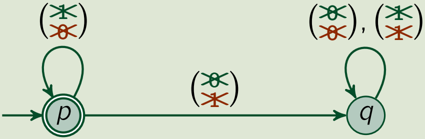
What's wrong??

Example of translation

Consider the (unsatisfiable) formula

$$\psi = \exists x.\ \neg \exists y.\ (y = x + 1)$$

1. Start from automaton $\mathcal{A}_{y=x+1}$
2. Project away the encoding of y, thus capturing $\exists y.\ (y = x + 1)$
3. Complement the det. automaton, thus capturing $\neg \exists y.\ (y = x + 1)$
4. Project away the encoding of x, thus getting $\exists x.\ \neg \exists y.\ (y = x + 1)$



What's wrong??

Languages of encodings should be closed under padding with 0's
After complement, keep only final states that are stable under 0.

The previous result can be generalized to many other structures:

Definition

An **automatic structure** is a structure that is isomorphic to

$$\big(L,\ R_1,\ \ldots,\ R_n \big)$$

where

- L is a regular language of words over Σ
  (each word identifies a precise element of the structure)
- each relation R_i has arity k_i and is represented
  by a regular language L_i over $\big(\Sigma \uplus \{\#\}\big)^{k_i}$
  (e.g. $(ab, abb) \in R_i$ iff $\binom{a}{a}\binom{b}{b}\binom{\#}{b} \in L_i$)

The previous result can be generalized to many other structures:

Definition

An **automatic structure** is a structure that is isomorphic to

$$\big(L,\ R_1,\ \ldots,\ R_n \big)$$

where

- L is a regular language of words over Σ (each word identifies a precise element of the structure)
- each relation R_i has arity k_i and is represented by a regular language L_i over $\big(\Sigma \uplus \{\#\}\big)^{k_i}$ (e.g. $(ab, abb) \in R_i$ iff $\binom{a}{a}\binom{b}{b}\binom{\#}{b} \in L_i$)

Examples of automatic structures

- $(\mathbb{N}, +, |_p)$, with $x \,|_p\, y$ iff $x = p^n$ divides y
- Binary tree with successor, ancestor, and equi-level predicates
- Unlabelled grid $(\mathbb{N} \times \mathbb{N}, \rightarrow, \downarrow)$

Theorem (Büchi '60, Hodgson '76, Khoussainov & Nerode '94)

Every automatic structure has a **decidable first-order theory**.

Theorem (Büchi '60, Hodgson '76, Khoussainov & Nerode '94)

Every automatic structure has a **decidable first-order theory**.

On the other hand...

Theorem

Some automatic structures have an **undecidable reachability problem**.

The **transition graph** of a Turing machine is automatic!

- configurations are encoded by words $a_1 \ldots a_{i-1}\, q\, a_i\, a_{i+1} \ldots a_n$
- transitions are of the following forms

$$a_1 \ldots a_{i-1}\, q\, a_i\, a_{i+1} \ldots a_n \quad a_1 \ldots a_{i-1}\, q\, a_i\, a_{i+1} \ldots a_n \quad a_1 \ldots a_{i-1}\, a_i\, q\, a_{i+1} \ldots a_n$$
$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$
$$a_1 \ldots a_{i-1}\, q'\, a'_i\, a_{i+1} \ldots a_n \quad a_1 \ldots a_{i-1}\, a'_i\, q'\, a_{i+1} \ldots a_n \quad a_1 \ldots a_{i-1}\, q'\, a'_i\, a_{i+1} \ldots a_n$$

Next