Lucas Waclawczyk

Decidability
of Logical Theories

Proseminar Theoretical Computer Science // Dresden, June 20, 2020

Can we mechanize mathematics?

Content

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 3 of 26

DRESDEN
concept

First-Order Logic

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \Big(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \Big) \right]$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \Big(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \Big) \right]$$

Formulas consist of:

- Quantifiers: $\forall$, $\exists$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q\, \exists p\, \forall x\, \forall y\, \left[R_1(p,q) \wedge \Big(\neg\big(R_1(x,1) \wedge R_1(y,1)\big) \vee R_2(x,y,p)\Big) \right]$$

Formulas consist of:

- Quantifiers: $\forall$, $\exists$
- Variables: $p, q, x, y, \ldots$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \; \left[R_1(p, q) \wedge \Big(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \Big) \right]$$

Formulas consist of:

- Quantifiers: $\forall$, $\exists$
- Variables: $p, q, x, y, \ldots$
- Boolean operators: $\wedge, \vee, \neg$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \left(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \right) \right]$$

Formulas consist of:

- Quantifiers: $\forall$, $\exists$
- Variables: $p, q, x, y, \ldots$
- Boolean operators: $\wedge, \vee, \neg$
- Relation symbols: $R_1, R_2, \ldots$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p,q) \wedge \left(\neg (R_1(x,1) \wedge R_1(y,1)) \vee R_2(x,y,p) \right) \right]$$

Formulas consist of:

- Quantifiers: $\forall$, $\exists$
- Variables: $p, q, x, y, \ldots$
- Boolean operators: $\wedge, \vee, \neg$
- Relation symbols: $R_1, R_2, \ldots$
- Special characters: $[,], (,)$

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \left(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \right) \right]$$

Formulas consist of:

- Quantifiers: $\forall, \exists$
- Variables: $p, q, x, y, \ldots$
- Boolean operators: $\wedge, \vee, \neg$
- Relation symbols: $R_1, R_2, \ldots$
- Special characters: $[,], (,)$

Simplified here:

- mostly "sentences"
  (no free variables)

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN
concept

Syntax

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \left(\neg(R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \right) \right]$$

Formulas consist of:

- Quantifiers: $\forall, \exists$
- Variables: $p, q, x, y, \ldots$
- Boolean operators: $\wedge, \vee, \neg$
- Relation symbols: $R_1, R_2, \ldots$
- Special characters: $[,], (,)$

Simplified here:

- mostly "sentences" (no free variables)
- only prenex normal form (all quantifiers on the left)

TECHNISCHE UNIVERSITÄT DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 5 of 26

DRESDEN concept

Semantics

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \land \Big(\neg (R_1(x, 1) \land R_1(y, 1)) \lor R_2(x, y, p) \Big) \right]$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN
concept

Semantics

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \Big(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \Big) \right]$$

$$\stackrel{!}{=} \text{"There are infinitely many prime numbers."}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN concept

Semantics

$$\forall q \,\exists p \,\forall x \,\forall y \,\left[R_1(p, q) \wedge \Big(\neg\big(R_1(x, 1) \wedge R_1(y, 1)\big) \vee R_2(x, y, p) \Big) \right]$$

- *universe* $\mathcal{U}$
  - possible values for variables
  - here $\mathbb{N}$

Semantics

$$\forall q \, \exists p \, \forall x \, \forall y \, \left[R_1(p, q) \wedge \left(\neg (R_1(x, 1) \wedge R_1(y, 1)) \vee R_2(x, y, p) \right) \right]$$

$$= \quad \forall q \, \exists p \, \forall x, y \, [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$$

- *universe $\mathcal{U}$*
  - possible values for variables
  - here $\mathbb{N}$

- *model $\mathcal{M}$*
  - universe + assignment of relations
  - here $(\mathbb{N}, >_2, (\times \neq)_3)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Wacławczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN
concept

Semantics

$$\forall q \, \exists p \, \forall x, y \; [p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$$

$$= \quad \text{"There are infinitely many prime numbers."}$$

- *universe $\mathcal{U}$*
  - possible values for variables
  - here $\mathbb{N}$

- *model $\mathcal{M}$*
  - universe + assignment of relations
  - here $(\mathbb{N}, >_2, (\times \neq)_3)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN
concept

Semantics

$$\forall q \, \exists p \, \forall x, y \; [p > q \land (x, y > 1 \to xy \neq p)]$$

$$\text{vs. } [(x + x = y) \lor (x \geq y)]$$

- *universe* $\mathcal{U}$
  - possible values for variables
  - here $\mathbb{N}$
- *model* $\mathcal{M}$
  - universe + assignment of relations
  - here $(\mathbb{N}, >_2, (\times \neq)_3)$

- *language* $\mathrm{L}(\mathcal{M})$
  - sentences that make sense in $\mathcal{M}$
  - here $\mathrm{L}(\mathbb{N}, >_2, (\times \neq)_3)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN
concept

Semantics

$$\forall q \, \exists p \, \forall x, y \; [p > q \land (x, y > 1 \rightarrow xy \neq p)]$$

$$\text{vs. } \forall y \, \exists x \; [\neg(xx \neq y)]$$

- *universe* $\mathcal{U}$
  - possible values for variables
  - here $\mathbb{N}$
- *model* $\mathcal{M}$
  - universe + assignment of relations
  - here $(\mathbb{N}, >_2, (\times \neq)_3)$

- *language* $\mathsf{L}(\mathcal{M})$
  - sentences that make sense in $\mathcal{M}$
  - here $\mathsf{L}(\mathbb{N}, >_2, (\times \neq)_3)$
- *theory* $\mathsf{Th}(\mathcal{M})$
  - true sentences formed with $\mathcal{M}$
  - here $\mathsf{Th}(\mathbb{N}, >_2, (\times \neq)_3)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 6 of 26

DRESDEN
concept

What Is Decidability?

- here: for logic (there is a more general definition)
- let $\mathcal{M}$ be a model, $\varphi \in L(\mathcal{M})$

$$\text{Th}(\mathcal{M}) \text{ decidable} :=$$

there is an algorithm that decides whether φ is true in $\mathcal{M}$

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 7 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Th $(\mathbb{N}, +)$ – A Decidable Theory

Theorem 1

Theorem

Th $(\mathbb{N}, +)$ *is decidable.*

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 9 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Theorem 1

Theorem

Th $(\mathbb{N}, +)$ *is decidable.*

i.e., there is an algorithm that can decide,
whether a sentence $\varphi \in \mathsf{L}(\mathbb{N}, +)$ is true or false.

Proof of Theorem 1

Idea: Construct an automaton that accepts an (almost) empty input iff the given sentence is true.

Let $i \in \mathbb{N} \setminus \{0\}$ and define $\Sigma_i := \{0, 1\}^i$ and $\Sigma_0 := \{()\}$.
An example for a word in Σ_3:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad \sim \quad \begin{pmatrix} 3 \\ 9 \\ 12 \end{pmatrix}$$

Review: Automata
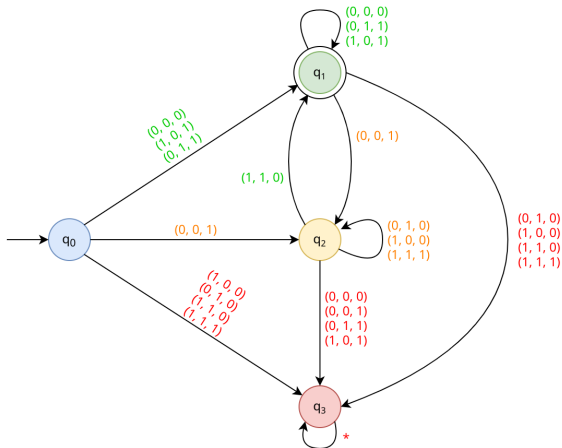
Example automaton that accepts $+_3$ encoded in Σ_3

Proof of Theorem 1

Now, let

- $k \in \{0, \ldots, l\}$

- $\varphi = Q_1 x_1 \ldots Q_l x_l \left[\psi(x_1, \ldots, x_l) \right] \in L(\mathbb{N}, +)$ where $Q_1, \ldots, Q_l \in \{\forall, \exists\}$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 12 of 26

DRESDEN
concept

Proof of Theorem 1

Now, let

- $k \in \{0, \dots, l\}$

- $\varphi = Q_1 x_1 \dots Q_l x_l \left[\psi(x_1, \dots, x_l) \right] \in L(\mathbb{N}, +)$ where $Q_1, \dots, Q_l \in \{\forall, \exists\}$

- $\varphi_k := Q_{k+1} x_{k+1} \dots Q_l x_l \left[\psi(x_1, \dots, x_l) \right]$
  $\Rightarrow \varphi_0 = \varphi$
  $\Rightarrow \varphi_l = \psi(x_1, \dots, x_l)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 12 of 26

DRESDEN
concept

Proof of Theorem 1

Now, let

- $k \in \{0, \dots, l\}$

- $\varphi = Q_1 x_1 \dots Q_l x_l \left[\psi(x_1, \dots, x_l)\right] \in L(\mathbb{N}, +)$ where $Q_1, \dots, Q_l \in \{\forall, \exists\}$

- $\varphi_k := Q_{k+1} x_{k+1} \dots Q_l x_l \left[\psi(x_1, \dots, x_l)\right]$
  $\Rightarrow \varphi_0 = \varphi$
  $\Rightarrow \varphi_l = \psi(x_1, \dots, x_l)$

- $\varphi_k(a_1, \dots, a_k) := Q_{k+1} x_{k+1} \dots Q_l x_l \left[\psi(a_1, \dots, a_k, x_{k+1}, \dots, x_l)\right]$
  where $a_1, \dots, a_k \in \mathbb{N}$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 12 of 26

DRESDEN
concept

Proof of Theorem 1

Now, let

- $k \in \{0, \ldots, l\}$

- $\varphi = Q_1 x_1 \ldots Q_l x_l [\psi(x_1, \ldots, x_l)] \in L(\mathbb{N}, +)$ where $Q_1, \ldots, Q_l \in \{\forall, \exists\}$

- $\varphi_k := Q_{k+1} x_{k+1} \ldots Q_l x_l [\psi(x_1, \ldots, x_l)]$
  $\Rightarrow \varphi_0 = \varphi$
  $\Rightarrow \varphi_l = \psi(x_1, \ldots, x_l)$

- $\varphi_k(a_1, \ldots, a_k) := Q_{k+1} x_{k+1} \ldots Q_l x_l [\psi(a_1, \ldots, a_k, x_{k+1}, \ldots, x_l)]$
  where $a_1, \ldots, a_k \in \mathbb{N}$

$\implies$ φ_l is only a Boolean expression.

TECHNISCHE UNIVERSITÄT DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 12 of 26

DRESDEN concept

Proof of Theorem 1

Construct an automaton A_l that behaves like $\varphi_l = \psi$, meaning A_l accepts exactly the tuples $(a_1, \ldots, a_l) \in \mathbb{N}^l$ for which $\varphi_l(a_1, \ldots, a_l)$ is true:
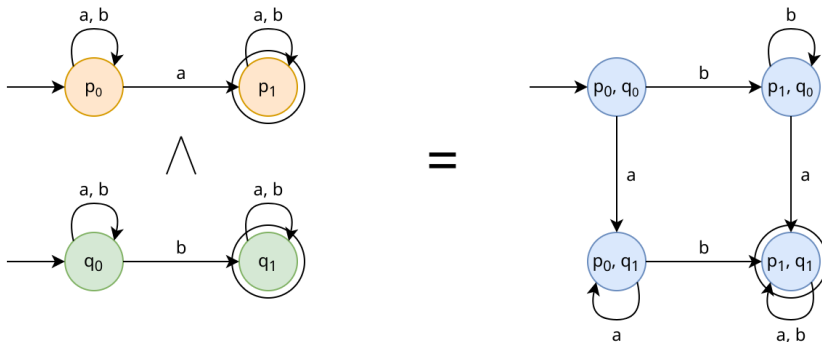
Proof of Theorem 1

Construct an automaton A_I that behaves like $\varphi_I = \psi$, meaning A_I accepts exactly the tuples $(a_1, \ldots, a_I) \in \mathbb{N}^I$ for which $\varphi_I(a_1, \ldots, a_I)$ is true:

- Take one addition automaton for each addition term in φ_I

Proof of Theorem 1
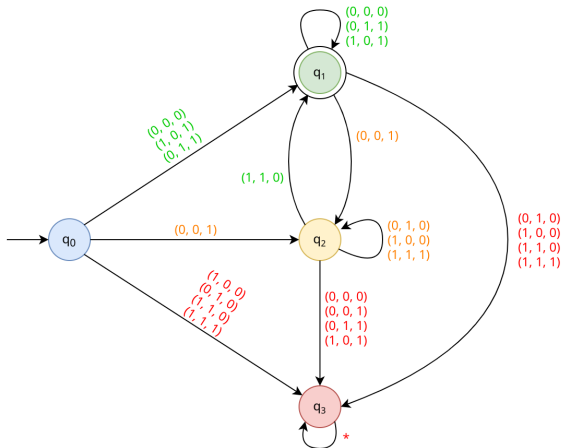
Addition automaton

Proof of Theorem 1

Construct an automaton A_I that behaves like $\varphi_I = \psi$, meaning A_I accepts exactly the tuples $(a_1, \ldots, a_I) \in \mathbb{N}^I$ for which $\varphi_I(a_1, \ldots, a_I)$ is true:

- Take one addition automaton for each addition term in φ_I
- Combine them:
  - automaton product for $\wedge$
  - automaton union for $\vee$
  - automaton complement for $\neg$

  in a way that they behave like φ_I.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Wacławczyk
Dresden, June 20, 2020

Slide 13 of 26

DRESDEN
concept

Proof of Theorem 1

Conjunction of two simple automata

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Proof of Theorem 1

Construct an automaton A_I that behaves like $\varphi_I = \psi$, meaning A_I accepts exactly the tuples $(a_1, \ldots, a_I) \in \mathbb{N}^I$ for which $\varphi_I(a_1, \ldots, a_I)$ is true:

- Take one addition automaton for each addition term in φ_I
- Combine them:
  - automaton product for $\land$
  - automaton union for $\lor$
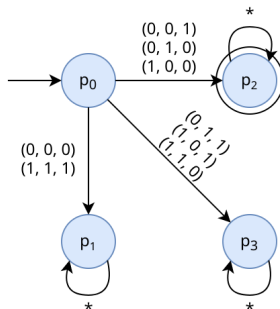 - automaton complement for $\neg$

  in a way that they behave like φ_I.

Important: There is an algorithm that constructs A_I from φ_I.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 13 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Proof of Theorem 1

Example construction of non-deterministic guessing

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

Example construction of non-deterministic guessing

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

If however $Q_l = \forall$, use complementation twice ($\forall x_l \, \varphi_l = \neg \exists x_l \neg \varphi_l$)

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

If however $Q_l = \forall$, use complementation twice ($\forall x_l \, \varphi_l = \neg \exists x_l \neg \varphi_l$)

$\implies$ Inductively construct $A_{l-2}, \ldots, A_0$

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

If however $Q_l = \forall$, use complementation twice ($\forall x_l \, \varphi_l = \neg \exists x_l \neg \varphi_l$)

$\implies$ Inductively construct $A_{l-2}, \ldots, A_0$

$\implies$ A_k accepts input $(a_1, \ldots, a_k) \in \mathbb{N}^k$ $\Leftrightarrow$ $\varphi_k(a_1, \ldots, a_k)$ is true

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

If however $Q_l = \forall$, use complementation twice ($\forall x_l\, \varphi_l = \neg\exists x_l \neg\varphi_l$)

$\implies$ Inductively construct $A_{l-2}, \ldots, A_0$

$\implies$ A_k accepts input $(a_1, \ldots, a_k) \in \mathbb{N}^k$ $\Leftrightarrow$ $\varphi_k(a_1, \ldots, a_k)$ is true

$\implies$ A_0 accepts input () $\Leftrightarrow$ $\varphi_0 = \varphi$ is true

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_l = \exists$, construct automaton A_{l-1} from A_l by

- copying states of A_l
- adding a new starting state
- making A_{l-1} guess the right a_l non-deterministically

If however $Q_l = \forall$, use complementation twice ($\forall x_l \, \varphi_l = \neg \exists x_l \neg \varphi_l$)

$\implies$ Inductively construct $A_{l-2}, \ldots, A_0$

$\implies$ A_k accepts input $(a_1, \ldots, a_k) \in \mathbb{N}^k$ $\Leftrightarrow$ $\varphi_k(a_1, \ldots, a_k)$ is true

$\implies$ A_0 accepts input () $\Leftrightarrow$ $\varphi_0 = \varphi$ is true

Let the algorithm return "$\varphi \in \text{Th}(\mathbb{N}, +)$" $\Leftrightarrow$ A_0 accepts input ()

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 14 of 26

DRESDEN
concept

Th $(\mathbb{N}, +, \times)$ – An Undecidable Theory

Theorem 2

Theorem

Th $(\mathbb{N}, +, \times)$ *is undecidable.*
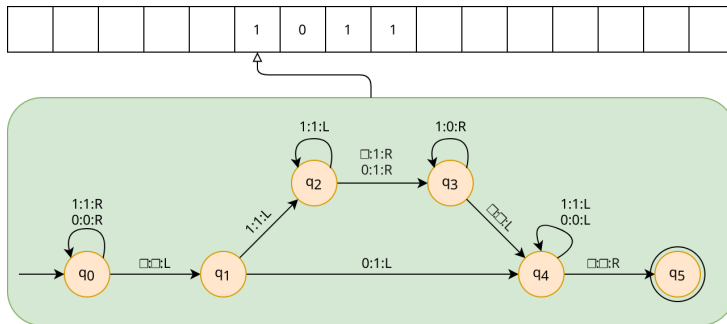
Theorem 2

Theorem

Th $(\mathbb{N}, +, \times)$ *is undecidable.*

i.e., there is <u>no</u> algorithm that can decide,
whether a sentence $\varphi \in \mathsf{L}(\mathbb{N}, +, \times)$ is true or false.

Turing Machines

Example turing machine
adapted from: [2]

Proof Idea for Theorem 2

- **The word problem for Turing machines is undecidable.** ([5], Satz 5.6)

Proof Idea for Theorem 2

- The word problem for Turing machines is undecidable. ([5], Satz 5.6)
- There is a mapping reduction that translates
  - a Turing machine M and a string w

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 18 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Proof Idea for Theorem 2

- The word problem for Turing machines is undecidable. ([5], Satz 5.6)
- There is a mapping reduction that translates
  - a Turing machine M and a string w
  - to a formula $\varphi_{M,w}(x) \in L(\mathbb{N}, +, \times)$ that contains only one free variable x, such that

Proof Idea for Theorem 2

- The word problem for Turing machines is undecidable. ([5], Satz 5.6)
- There is a mapping reduction that translates
  - a Turing machine M and a string w
  - to a formula $\varphi_{M,w}(x) \in L(\mathbb{N}, +, \times)$ that contains only one free variable x, such that
  - $\varphi_{M,w}(x)$ is true $\Leftrightarrow$ M accepts w ($x =$ computation history suitably encoded)

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Proof Idea for Theorem 2

- The word problem for Turing machines is undecidable. ([5], Satz 5.6)
- There is a mapping reduction that translates
  - a Turing machine M and a string w
  - to a formula $\varphi_{M,w}(x) \in L(\mathbb{N}, +, \times)$ that contains only one free variable x, such that
  - $\varphi_{M,w}(x)$ is true $\Leftrightarrow$ M accepts w (x = computation history suitably encoded)

Assume Th $(\mathbb{N}, +, \times)$ is decidable.
$\Longrightarrow$ The formulas $\exists x \, [\varphi_{M,w}(x)] \in L(\mathbb{N}, +, \times)$ are decidable.

Proof Idea for Theorem 2

- The word problem for Turing machines is undecidable. ([5], Satz 5.6)
- There is a mapping reduction that translates
  - a Turing machine M and a string w
  - to a formula $\varphi_{M,w}(x) \in L(\mathbb{N}, +, \times)$ that contains only one free variable x, such that
  - $\varphi_{M,w}(x)$ is true $\Leftrightarrow$ M accepts w ($x =$ computation history suitably encoded)

Assume $\text{Th}(\mathbb{N}, +, \times)$ is decidable.

$\implies$ The formulas $\exists x\, [\varphi_{M,w}(x)] \in L(\mathbb{N}, +, \times)$ are decidable.

$\implies$ The word problem for Turing machines is decidable.

Proof Idea for Theorem 2

- **The word problem for Turing machines is undecidable.** ([5], Satz 5.6) ⚡
- There is a mapping reduction that translates
  - a Turing machine M and a string w
  - to a formula $\varphi_{M,w}(x) \in L(\mathbb{N}, +, \times)$ that contains only one free variable x, such that
  - $\varphi_{M,w}(x)$ is true $\Leftrightarrow$ M accepts w ($x =$ computation history suitably encoded)

Assume Th $(\mathbb{N}, +, \times)$ is decidable.

$\implies$ The formulas $\exists x \, [\varphi_{M,w}(x)] \in L(\mathbb{N}, +, \times)$ are decidable.

$\implies$ The word problem for Turing machines is decidable.

$\implies$ ⚡

Thinking further...

Is there a true, unprovable sentence?

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 19 of 26

DRESDEN
concept

Thinking further...

Is there a true, unprovable sentence?

A proof is a series of implications and can be written as a string over some alphabet (here: $L(\mathbb{N}, +, \times)$). Assumptions:

A_1 Proofs can be checked by a machine.

A_2 Provable statements are true.

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 19 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Thinking further...

Is there a true, unprovable sentence?

A proof is a series of implications and can be written as a string over some alphabet (here: $L(\mathbb{N}, +, \times)$). Assumptions:

A_1 Proofs can be checked by a machine.

A_2 Provable statements are true.

Lemmas:

1. The provable statements of $\mathrm{Th}(\mathbb{N}, +, \times)$ are Turing recognizable.
   Proof idea: Just try all possible (suitably encoded) proofs.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 19 of 26

DRESDEN
concept

Thinking further...

Is there a true, unprovable sentence?

A proof is a series of implications and can be written as a string over some alphabet (here: $L(\mathbb{N}, +, \times)$). Assumptions:

A_1 Proofs can be checked by a machine.

A_2 Provable statements are true.

Lemmas:

1. The provable statements of $\text{Th}(\mathbb{N}, +, \times)$ are Turing recognizable.
   Proof idea: Just try all possible (suitably encoded) proofs.
2. There is a (true) statement in $\text{Th}(\mathbb{N}, +, \times)$ that is not provable.
   Proof idea: Contradiction to Theorem 2 by using 1.

Gödel's Incompleteness Theorem

A True, Unprovable Statement

Theorem
We can construct a true statement in Th $(\mathbb{N}, +, \times)$*, that is not provable.*

A True, Unprovable Statement

Theorem

We can construct a true statement in Th $(\mathbb{N}, +, \times)$, that is not provable.

Proof: Let *M* be a Turing machine that operates as follows.

- Delete the input
- Look for proof of $\varphi := \neg \exists x [\varphi_{M,0}(x)] \in \mathrm{Th}\,(\mathbb{N}, +, \times)$
- Go to final state $:\Leftrightarrow$ proof for φ has been found

A True, Unprovable Statement

Theorem

We can construct a true statement in Th $(\mathbb{N}, +, \times)$, that is not provable.

Proof: Let *M* be a Turing machine that operates as follows.

- Delete the input
- Look for proof of $\varphi := \neg \exists x[\varphi_{M,0}(x)] \in$ Th $(\mathbb{N}, +, \times)$
- Go to final state $:\Leftrightarrow$ proof for φ has been found

$\implies$ φ is the wanted statement.

TECHNISCHE UNIVERSITÄT DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 21 of 26

DRESDEN concept

Gödel's Method

- Construct the statement "This statement cannot be proved by the axioms."

Gödel's Method

- Construct the statement "This statement cannot be proved by the axioms."
- Argue against just adding this statement to the axiom.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 22 of 26

DRESDEN
concept

Gödel's Method

- Construct the statement "This statement cannot be proved by the axioms."
- Argue against just adding this statement to the axiom.

$\implies$ Incompleteness ☺

Conclusion

Conclusion

$\implies$ There are (very simple) indecidable logical theories.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 24 of 26

DRESDEN
concept

Conclusion

$\implies$ There are (very simple) indecidable logical theories.

$\implies$ No sound arithmetical system can be complete.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 24 of 26

DRESDEN
concept

Conclusion

$\implies$ There are (very simple) indecidable logical theories.

$\implies$ No sound arithmetical system can be complete.

$\implies$ Mathematics cannot be mechanized.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 24 of 26

DRESDEN
concept

References

References

[1] Martin Alessandro - Own work, CC BY-SA 4.0,
 `https://commons.wikimedia.org/w/index.php?curid=75082873`

[2] `https://www.inf-schule.de/grenzen/berechenbarkeit/turingmaschine/`
 `station_turingmaschine`

[3] Michael Sipser: Introduction to the Theory of Computation. Thomson
 Course Technology, 2006

[4] `https://www.youtube.com/watch?v=O4ndIDcDSGc`

[5] Prof. Dr. Franz Baader: Skript Theoretische Informatik und Logik
 (Sommersemester 2020), TU Dresden

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, June 20, 2020

Slide 26 of 26

DRESDEN
concept