Lucas Waclawczyk

Decidability
of Logical Theories

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

Can we mechanize mathematics?

Inhalt

First-Order Logic

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

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

Gödel's Incompleteness Theorem

Conclusion

References

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

Slide 3 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

First-Order Logic

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, May 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: $\forall$, $\exists$

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: $p, q, x, y, \dots$

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

Slide 5 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

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: $\wedge, \vee, \neg$

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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, May 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: $[,], (,)$

TECHNISCHE UNIVERSITÄT DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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)

TECHNISCHE UNIVERSITÄT DRESDEN

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

Slide 5 of 26

DRESDEN concept

Semantics

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide   6 of 26

DRESDEN
concept

Semantics

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

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

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

Slide 6 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Semantics

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 6 of 26

DRESDEN
concept

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 Waclawczyk
Dresden, May 20, 2020

Slide 6 of 26

DRESDEN
concept

Semantics

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

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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* $\mathrm{L}(\mathcal{M})$
  - sentences that make sense in $\mathcal{M}$
  - here $\mathrm{L}(\mathbb{N}, >_2, (\times \neq)_3)$
- *theory* $\mathrm{Th}(\mathcal{M})$
  - true sentences formed with $\mathcal{M}$
  - here $\mathrm{Th}(\mathbb{N}, >_2, (\times \neq)_3)$

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 6 of 26

DRESDEN
concept

Semantics

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

$=$ "There are infinitely many prime numbers."

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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, $\varphi \in \mathsf{L}(\mathcal{M})$

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

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

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

Slide 7 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Theorem 1

Theorem

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

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 $\varphi \in L(\mathbb{N}, +)$ is true or false.

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 9 of 26

DRESDEN
concept

Review: Automata

Example automaton that accepts all correct binary additions

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 10 of 26

DRESDEN
concept

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:

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad \sim \quad \begin{pmatrix} 1 \\ 5 \\ 6 \end{pmatrix}$$

Proof of Theorem 1

Now, let

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

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

Proof of Theorem 1

Now, let

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

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

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

Proof of Theorem 1

Now, let

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

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

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

- $\varphi_i(a_1, \ldots, a_i) := Q_{i+1} x_{i+1} \ldots Q_l x_l \, [\psi(a_1, \ldots, a_i, x_{i+1}, \ldots, x_l)]$

Proof of Theorem 1

Now, let

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

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

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

- $\varphi_i(a_1, \ldots, a_i) := Q_{i+1} x_{i+1} \ldots Q_l x_l \, [\psi(a_1, \ldots, a_i, x_{i+1}, \ldots, x_l)]$

$$\implies \quad \varphi_l \text{ is only a Boolean expression.}$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 12 of 26
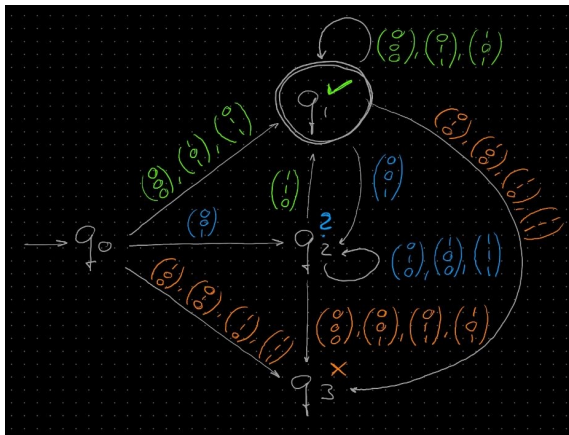
DRESDEN
concept

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:

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

Slide 13 of 26

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 13 of 26

DRESDEN
concept

Proof of Theorem 1

Addition automaton

TECHNISCHE
UNIVERSITÄT
DRESDEN

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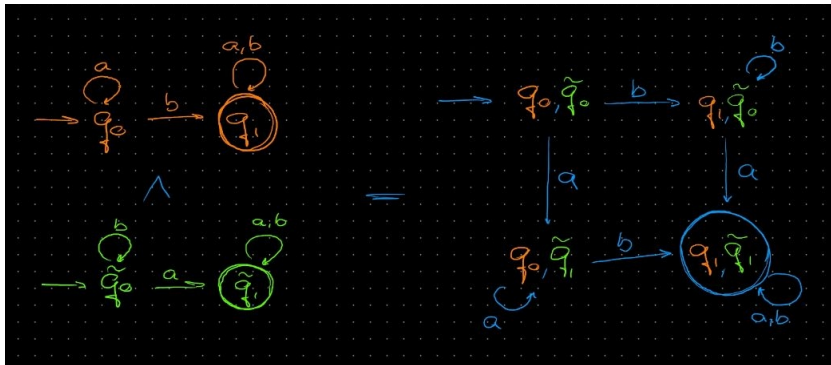Lucas Waclawczyk
Dresden, May 20, 2020

Slide 13 of 26

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 $\wedge$
  – automaton union for $\vee$
  – automaton complement for $\neg$

  in a way that they behave like φ_I.

Proof of Theorem 1

Conjunction of two simple automata

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 13 of 26

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 $\wedge$
  - automaton union for $\vee$
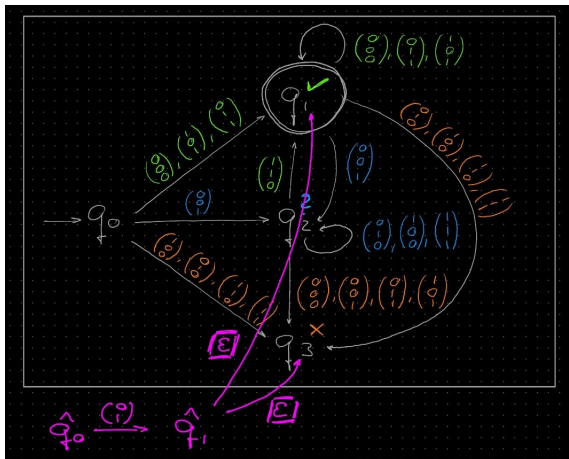 - automaton complement for $\neg$

  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, May 20, 2020

Slide 13 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_i = \exists$, construct automaton A_i from A_{i+1} by

- copying A_{i+1}
- adding a new start state and one state for each character in Σ_i
- making A_i guess the right a_{i+1} non-deterministically

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

Example construction of non-deterministic guessing

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_i = \exists$, construct automaton A_i from A_{i+1} by

- copying A_{i+1}
- adding a new start state and one state for each character in Σ_i
- making A_i guess the right a_{i+1} non-deterministically

If else $Q_i = \forall$, use complementation twice ($\forall x_i \, \varphi_{i+1} = \neg \exists x_i \neg \varphi_{i+1}$)

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_i = \exists$, construct automaton A_i from A_{i+1} by

- copying A_{i+1}
- adding a new start state and one state for each character in Σ_i
- making A_i guess the right a_{i+1} non-deterministically

If else $Q_i = \forall$, use complementation twice ($\forall x_i \, \varphi_{i+1} = \neg \exists x_i \neg \varphi_{i+1}$)

$$\implies \quad A_i \text{ accepts input } (a_1, \ldots, a_i) \in \mathbb{N}^i \quad \Leftrightarrow \quad \varphi_i \text{ is true}$$

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_i = \exists$, construct automaton A_i from A_{i+1} by

- copying A_{i+1}
- adding a new start state and one state for each character in Σ_i
- making A_i guess the right a_{i+1} non-deterministically

If else $Q_i = \forall$, use complementation twice ($\forall x_i\, \varphi_{i+1} = \neg\exists x_i \neg\varphi_{i+1}$)

$\implies$ A_i accepts input $(a_1, \ldots, a_i) \in \mathbb{N}^i$ $\Leftrightarrow$ φ_i is true

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

Proof of Theorem 1

If $Q_i = \exists$, construct automaton A_i from A_{i+1} by

- copying A_{i+1}
- adding a new start state and one state for each character in Σ_i
- making A_i guess the right a_{i+1} non-deterministically

If else $Q_i = \forall$, use complementation twice ($\forall x_i\, \varphi_{i+1} = \neg\exists x_i \neg\varphi_{i+1}$)

$\implies$ A_i accepts input $(a_1, \ldots, a_i) \in \mathbb{N}^i$ $\Leftrightarrow$ φ_i is true

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 14 of 26

DRESDEN
concept

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

Theorem 2

Theorem

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

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

Slide 16 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Theorem 2

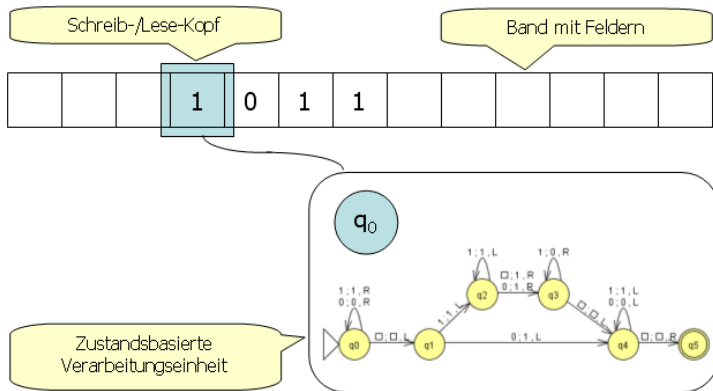## Theorem

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

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

Turing Machines

Example turing machine
source: [2]

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

Slide 17 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

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

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 (with the computation history suitably encoded in x)

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 (with the computation history suitably encoded in x)

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

TECHNISCHE UNIVERSITÄT DRESDEN

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

Slide 18 of 26

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 (with the computation history suitably encoded in x)

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

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

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 18 of 26

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 (with the computation history suitably encoded in x)

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

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

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

$\implies$ ⚡

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

Slide 18 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

Thinking further...

Is there a true, unprovable sentence?

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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.

TECHNISCHE
UNIVERSITÄT
DRESDEN

Decidability of Logical Theories
Lucas Waclawczyk
Dresden, May 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 $\mathrm{Th}(\mathbb{N}, +, \times)$ are Turing recognizable.
   Proof idea: Just try all possible (suitably encoded) proofs.
2. There is a (true) statement in $\mathrm{Th}(\mathbb{N}, +, \times)$ that is not provable.
   Proof idea: Contradiction to Theorem 2 by using 1.

Gödel's Incompleteness Theorem

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

A True, Unprovable Statement

Theorem

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

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

Slide 21 of 26

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

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 $\varphi := \neg\exists x[\varphi_{M,0}(x)] \in \text{Th}(\mathbb{N}, +, \times)$
- Accept input $\Leftrightarrow$ proof for φ has been found

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 21 of 26

DRESDEN
concept

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 $\varphi := \neg \exists x[\varphi_{M,0}(x)] \in \text{Th}\,(\mathbb{N}, +, \times)$
- Accept input $\Leftrightarrow$ proof for φ has been found

$\implies$ φ is the wanted statement.

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, May 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 ☺

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 22 of 26

DRESDEN
concept

Conclusion

Conclusion

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 24 of 26

DRESDEN
concept

Conclusion

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

$\implies$ Mathematics cannot be mechanized.

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 24 of 26

DRESDEN
concept

Conclusion

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

$\implies$ Mathematics cannot be mechanized.

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

TECHNISCHE
UNIVERSITÄT
DRESDEN

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

Slide 26 of 26

DRESDEN
concept