# Theory of Computing
# Selected Topics

Ming-Hsien Tsai

Department of Information Management
National Taiwan University

Spring 2019

(original created by Bow-Yaw Wang)

# Decidability of Logical Theories

- Consider the following mathematical statements over integers:
  1. $\forall q \exists p \forall x, y[p > q \wedge (x, y > 1 \rightarrow xy \neq p)]$
  2. $\forall a, b, c, n[(a, b, c > 0 \wedge n > 2) \rightarrow a^n + b^n \neq c^n]$; and
  3. $\forall q \exists p \forall x, y[p > q \wedge x, y > 1 \rightarrow (xy \neq p \wedge xy \neq p + 2)]$.
- In words, they are
  1. "there are infinitely many prime numbers."
  2. "the equation $a^n + b^n = c^n$ does not have non-trivial solution when $n > 2$." (Fermat's last theorem)
  3. "there are infinitely many twin primes."
- Would it be wonderful if we could check whether a given mathematical statement is true ?

# A Language of True Mathematical Statements

- As usual, we define a language for mathematical statements.
- Consider the following alphabet

$$\{\wedge, \vee, \neg, (, ), [, ], \forall, x, \exists, R_1, \ldots, R_k\}$$

  - $\wedge, \vee, \neg$ are <u>Boolean opearations</u>;
  - ( and ) are <u>parentheses</u>;
  - $\forall$ and $\exists$ are <u>quantifiers</u>;
  - $x$ denotes <u>variables</u>;
    - $x_i$ is denoted by $\underbrace{x \cdots x}_{i}$.
  - $R_1, \ldots, R_k$ are <u>relations</u>.

# A Language of True Mathematical Statements

- A string of the form $R_i(x_1, \ldots, x_j)$ is an atomic formula with arity $j$.
- A well-formed formula is defined as follows.
  - An atomic formula a well-formed;
  - If $\phi_1$ and $\phi_2$ are well-formed, $\phi_1 \wedge \phi_2$, $\phi_1 \vee \phi_2$, and $\neg\phi_1$ are well-formed; and
  - $\exists x_i[\phi_1]$ and $\forall x_i[\phi_1]$ are wellformed if $\phi_1$ is well-formed.
- A formula is in prenex normal form if its quantifiers appear first.
  - Any formula can be rewritten in prenex normal form.
- We only consider formula in prenex normal form.
- A variable not bound by any quantifier is a free variable.
- A formula without free variables is a sentence or statement.
- Examples.
  - $R_1(x_1) \wedge R_2(x_1, x_2, x_3)$ (or $R_1(x) \wedge R_2(x, xx, xxx)$)
  - $\forall x_1[R_1(x_1) \wedge R_2(x_1, x_2, x_3)]$
  - $\forall x_1 \exists x_2 \exists x_3[R_1(x_1) \wedge R_2(x_1, x_2, x_3)]$

# A Language of True Mathematical Statements

- A universe is where the variables take values.
- A model (or interpretation, structure) consists of a universe and an assignment of relations to relation symbols.
- Formally, a model $\mathcal{M} = (U, P_1, \ldots, P_k)$ consists of a universe $U$ and relations $P_i$ assigned to symbols $R_i$ ($i = 1, \ldots, k$).
- If $\phi$ is true in a model $\mathcal{M}$, $\mathcal{M}$ is a model of $\phi$.
- The theory of a model $\mathcal{M}$ (written $\mathrm{Th}(\mathcal{M})$) is the collection of true sentences in $\mathcal{M}$.

# Examples

- Consider $\mathcal{M}_1 = (\mathbb{N}, \leq)$.
- Let $\phi$ be the sentence $\forall x_1 \forall x_2 [R_1(x_1, x_2) \vee R_1(x_2, x_1)]$.
- $\phi$ is true in $\mathcal{M}_1$.
    - We assign the relation $\leq$ to the symbol $R_1$.
- $\mathcal{M}_1$ is a model of $\phi$.
- $\phi \in \mathrm{Th}(\mathcal{M}_1)$.
- For simplicity, we will also write $\phi$ as $\forall x_1 \forall x_2 [x_1 \leq x_2 \vee x_2 \leq x_1]$.
- Now consider $\mathcal{M}_1' = (\mathbb{N}, <)$.
- Then $\phi$ is not true in $\mathcal{M}_1'$.

# Examples

- Define a 3-ary relation $PLUS = \{(a, b, c) : a + b = c\}$.
- Consider $\mathcal{M}_2 = (\mathbb{R}, PLUS)$.
- Let $\psi$ be the sentence $\forall x_1 \exists x_2 [R_1(x_2, x_2, x_1)]$ (or $\forall x_1 \exists x_2 [x_2 + x_2 = x_1]$).
- $\mathcal{M}_2$ is a model of $\psi$.
- $\psi \in \mathrm{Th}(\mathcal{M}_2)$.
- Consider $\mathcal{M}_2' = (\mathbb{Z}, PLUS)$.
- $\mathcal{M}_2'$ is not a model of $\psi$.

# Automatic Mathematics

- Let $\mathcal{M}$ be a model.
- Th($\mathcal{M}$) is a language.
  - It is a set consisting of true sentences in $\mathcal{M}$.
- Define a 3-ary relation $TIMES = \{(a, b, c) : a \times b = c\}$.
- Define a 3-ary relation $EXP = \{(a, b, c) : a^b = c\}$.
- Consider the model $(\mathbb{N}, >, PLUS, TIMES, EXP)$.
- Let
  - $\phi_1$ be $\forall q \exists p \forall x \forall y [p > q \wedge (x > 1 \wedge y > 1 \rightarrow \neg TIMES(x, y, p))]$.
  - $\phi_2$ be $\forall a \forall b \forall c \forall n \forall p \forall q \forall r [a > 0 \wedge b > 0 \wedge c > 0 \wedge n > 2 \wedge EXP(a, n, p) \wedge EXP(b, n, q) \wedge EXP(c, n, r) \rightarrow \neg PLUS(p, q, r)]$
  - $\phi_3$ be $\forall q \exists p \forall x \forall y \forall z [p > q \wedge x > 1 \wedge y > 1 \wedge TIMES(x, y, z) \rightarrow (\neg(z = p) \wedge \neg PLUS(p, 2, z))]$
- We know $\phi_1, \phi_2 \in \mathrm{Th}(\mathbb{N}, >, PLUS, TIMES, EXP)$.
- If the membership problem for $\mathrm{Th}(\mathbb{N}, >, PLUS, TIMES, EXP)$ is decidable, we can solve the twin prime conjecture automatically!

# Addition with Finite Automata

- Consider the alphabet

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

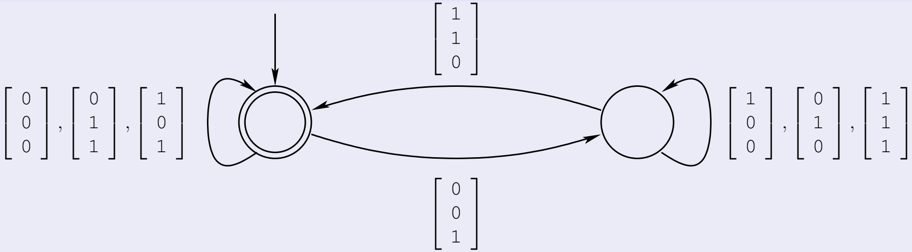
- A string over $\Sigma_3$ represents a triple of natural numbers.
  - $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ represents $(1, 3, 5)$.
- A language in $\Sigma_3^*$ therefore represents a relation with arity 3.
- We now show *PLUS* is represented by a regular language over $\Sigma_3^*$.
  - Finite automata can count after all!

# Addition with Finite Automata

## Lemma 1

*PLUS is regular.*

## Proof.




We first represent binary numbers in the reverse order, construct the finite automaton, then reverse its transitions. □

$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ represents $(3, 11, 14) \in \mathit{PLUS}$!

# Th($\mathbb{N}$, +) is Decidable

## Theorem 2

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

## Proof.

Let $\phi = \mathsf{Q}_1 x_1 \mathsf{Q}_2 x_2 \cdots \mathsf{Q}_l x_l[\psi]$ be a sentence where $\mathsf{Q}_i$ represents $\exists$ or $\forall$ ($i = 1, \ldots, l$) and $\psi$ is a formula without quantifiers. Define $\phi_i = \mathsf{Q}_{i+1} x_{i+1} \mathsf{Q}_{i+2} x_{i+2} \cdots \mathsf{Q}_l x_l[\psi]$. Note that $\phi_0 = \phi$, $\phi_l = \psi$ and $\phi_i$ has $i$ free variables. For each $i$, consider column vectors of size $i$:

$$\Sigma_i = \left\{ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \right\}$$

We construct a finite automaton $A_i$ which recognizes an $i$-ary relation such that $(x_1, x_2, \ldots, x_i) \in L(A_i)$ iff $\phi_i(x_1, x_2, \ldots, x_i)$ is true.
$A_l$ is easy. In Th($\mathbb{N}$, +), atomic formulae are generalized *PLUS* in Lemma 1. $A_l$ is obtained through Boolean operations.

# Th$(\mathbb{N}, +)$ is Decidable

## Proof (cont'd).

Assume $A_{i+1} = (\Sigma_{i+1}, Q, \delta, q, F)$ for $\phi_{i+1}(x_1, x_2, \ldots, x_l)$ is available. Consider $\phi_i = \exists x_{i+1}\phi_{i+1}$. Let $A_i = (\Sigma_i, Q \cup \{q'\}, \delta', q', F)$ where

$$\begin{array}{rcll}
\delta'(r, \begin{bmatrix} b_1 \\ \vdots \\ b_i \end{bmatrix}) & = & \delta(r, \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ 0 \end{bmatrix}) \cup \delta(r, \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ 1 \end{bmatrix}) & \text{if } r, s \in Q \text{ (guess the quantified bit)} \\
\delta'(q', \epsilon) & = & \delta(q, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}) \cup \delta(q, \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}) & \text{(guess the leading bit)}
\end{array}$$

Clearly, $(a_1, \ldots, a_i) \in L(A_i)$ iff there is an $a_{i+1}$ such that $(a_1, \ldots, a_i, a_{i+1}) \in L(A_{i+1})$.
For $\phi_i = \forall x_{i+1}\phi_{i+1}$, we construct $A_i$ for $\neg\exists x_{i+1}\neg\phi_{i+1}$.
To check if $\phi$ is true, we check if $\epsilon \in L(A_0)$. If $\epsilon \in L(A_0)$, the algorithm accepts $\phi$; if $\epsilon \notin L(A_0)$, the algorithm rejects $\phi$. □

# Th$(\mathbb{N}, +, \times)$ is Undecidable

## Lemma 3

*Let $M$ be a Turing machine and $w$ a string. We construct a formula $\phi_{M,w}(x)$ in the language of $(\mathbb{N}, +, \times)$ such that $\exists x \phi_{M,w}(x)$ is true iff $M$ accepts $w$.*

## Proof (sketch).

$\phi_{M,w}(x)$ denotes that $x$ is an accepting computation history of $M$ on $w$. We use a (very) large natural number to represent a configuration. For instance, $u_1 u_2 \cdots u_k q_i v_1 v_2 \cdots v_l$ is represented by $p_1^{u_1} \cdots p_k^{u_k} p_{k+1}^{|\Sigma|+i} p_{k+2}^{v_1} \cdots p_{k+l+1}^{v_l}$ where $p_i$ is the $i$-th prime number. □

## Theorem 4

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

## Proof.

Recall

$$A_{\mathrm{TM}} = \{\langle M, w\rangle : M \text{ is a TM and } M \text{ accepts } w\}$$

is undecidable. We give a reduction from $A_{\mathrm{TM}}$ to Th$(\mathbb{N}, +, \times)$. On input $\langle M, w\rangle$, the reduction outputs $\exists x \phi_{M,w}(x)$. Then $\langle M, w\rangle \in A_{\mathrm{TM}}$ iff $\exists x \phi_{M,w}(x)$. □

# Philosophical Consequences

- Since $\text{Th}(\mathbb{N}, +)$ is decidable, one can check any formula in the language of $(\mathbb{N}, +)$ is true automatically.
  - Whenever we have a conjecture in the language of $(\mathbb{N}, +)$, we just run a program to see whether the conjecture is true of not.
  - Doing mathematics cannot be easier.
- Unfortunately, $\text{Th}(\mathbb{N}, +, \times)$ is undecidable. We cannot prove or disprove a conjecture fully automatically.
  - Doing mathematics needs intelligence.

# Formal Proofs

- A formal proof $\pi$ of a statement $\phi$ is a sequence of statements $S_1, S_2, \ldots, S_l = \phi$ such that each $S_i$ "follows" from $S_1, S_2, \ldots, S_{i-1}$ and axioms about numbers.
  - We can give a mathematical definition of formal proofs.
  - To learn more about it, take a logic course or go to FLOLAC summer school.
- For our purposes, it suffices to know the following properties about formal proofs:
  1. The correctness of a proof of a statement can be checked by a machine.
     - Formally, $\{\langle \phi, \pi \rangle : \pi \text{ is a proof of } \phi\}$ is decidable.
  2. The system of proofs is sound.
     - That is, if a statement is provable, it is true.

# Gödel's Incompleteness Theorem

**Theorem 5**

*The collection of provable statements in* $Th(\mathbb{N}, +, \times)$ *is Turing-recognizable.*

**Proof.**

Consider

$P =$ "On input $\phi$ :

1. $s \leftarrow \epsilon$.
2. Check if $s$ is a proof of $\phi$ by the first property of formal proofs.
   1. If yes, accept $\phi$;
   2. If no, $s \leftarrow$ the next string.
3. Go to step 2."

□

# Gödel's Incompleteness Theorem

**Theorem 6**

*Some true statement in $Th(\mathbb{N}, +, \times)$ is not provable.*

**Proof.**

Suppose not. The following TM decides $\text{Th}(\mathbb{N}, +, \times)$:
$G =$ "On input $\phi$:

1. Run $P$ (Theorem 5) on $\phi$ and $\neg\phi$ in parallel.
2. If $P$ accepts $\phi$, accept.
3. If $P$ accepts $\neg\phi$, reject."

Note that either $\phi$ or $\neg\phi$ is true. Hence either $\phi$ or $\neg\phi$ is provable by assumption. Thus $P$ will accept either $\phi$ or $\neg\phi$. If $P$ accepts $\phi$, $\phi$ is true; if $P$ accepts $\neg\phi$, $\phi$ is false (the second property of formal proofs). Thus $G$ decides $\text{Th}(\mathbb{N}, +, \times)$. A contradiction to Theorem 4. □

# An Example

Assume a TM can obtain a copy of its own description (via recursion theorem).

**Theorem 7**

*The sentence $\psi_{unprovable}$ as described in the proof, is unprovable.*

**Proof.**

Let $S$ be a TM that operates as follows.
$S =$ "On any input:

1. Obtain own description $\langle S \rangle$ via the recursion theorem.
2. Construct the sentence $\psi = \neg \exists x[\phi_{S,0}(x)]$, using Lemma 3.
3. Run algorithm $P$ from the proof of Theorem 5.
4. If stage 3 accepts, accept."

□