Logic and Computation I

Part 3. First order logic and decision problems

Kazuyuki Tanaka

BIMSA

November 21, 2024

北京雁栖湖
应用数学研究院
Beijing Institute
of Mathematical
Sciences and Applications

Logic and Computation I

- **Part 1. Introduction to Theory of Computation**
- **Part 2. Propositional Logic and Computational Complexity**
- **Part 3. First Order Logic and Decision Problems**
- **Part 4. Modal logic** (shifted to next semester)

Part 3. A new schedule (subject to change)

- Nov.14, (4) Ehrenfeucht-Fraïssé's theorem
- Nov.19, (5) Ehrenfeucht-Fraïssé's theorem II
- Nov.21, (6) Presburger arithmetic
- Nov.26, (7) Peano arithmetic
- Nov.28, (8) Gödel's first incompleteness theorem
- Dec. 3, (9) Gödel's second incompleteness theorem
- Dec. 5, (10) Second order logic
- Dec.10, (11) Second order arithmetic

Recap

- We consider a language of finitely many relation symbols and constants.
- The (quantifier) rank of a formula measures the entanglement of quantifiers appearing in it. For example, the rank of $\forall y(\forall x \exists y(x = y) \wedge \forall z(z > 0))$ is 3.
- By $\mathcal{A} \equiv_n \mathcal{B}$, we mean that structures $\mathcal{A}, \mathcal{B}$ satisfy the same formulas with rank $\leq n$.
- Given an $\mathcal{A}$ and n, there is the **Scott-Hintikka sentence** $\varphi^n_{\mathcal{A}}$ of rank n such that $\mathcal{B} \models \varphi^n_{\mathcal{A}} \Leftrightarrow \mathcal{B} \equiv_n \mathcal{A}$.
- By $\mathcal{A} \simeq^n \mathcal{B}$, we mean that player II has a winning strategy in $\mathrm{EF}_n(\mathcal{A}, \mathcal{B})$, where n is the round of the game.
- **EF theorem** For all $n \geq 0$, $\mathcal{A} \equiv_n \mathcal{B}$ iff $\mathcal{A} \simeq^n \mathcal{B}$.
- **Corollary** The following are equivalent.
  (1) For any n, there exist $\mathcal{A} \in K$ and $\mathcal{B} \notin K$ such that $\mathcal{A} \equiv_n \mathcal{B}$.

  (2) K is not an elementary class (K cannot be defined by a first-order formula).
- By the EF theorem, DLO is decidable.
- DLO is PSPACE-complete. TQBF is polynomial-time reducible to DLO.

We next apply the EF theorem to the problem of length of finite linear orders.

Lemma 3.35 (Gurevich)

Fix any $m > 0$. If L_1, L_2 are two finite linear orders with length $\geq 2^m$, $L_1 \equiv_m L_2$.

Proof.

- By $[n] = (n, <)$, we denote a finite linear order on n, where n is identified with $\{0, 1, \ldots, n-1\}$.
- For each k, we define a threshold function $|x|_k$ by $|x|_k = |x|$ if $|x| < 2^k$; $|x|_k = \infty$, otherwise.
- Consider a partial isomorphism $\vec{a}(\subset [n]) \mapsto \vec{b}(\subset [n'])$ that satisfies the following conditions: if $\vec{a} = (a_1, a_2, \ldots, a_l)$ and $\vec{b} = (b_1, b_2, \ldots, b_l)$ are arranged in ascending order, and $a_0 = b_0 = 0$, $a_{l+1} = n$, $b_{l+1} = n'$, then

  for any $i \leq l$, $|a_{i+1} - a_i|_k = |b_{i+1} - b_i|_k$ holds.

  Then, let I_k be the set of such partial isomorphisms.
- By $\varnothing \in I_k$ we mean $|n|_k = |n'|_k$. Thus, if $n, n' \geq 2^m$, then $\varnothing \in I_m$

- Take any $\vec{a} \mapsto \vec{b} \in I_k$. We can show that for any $a \in n$, there exists a $b \in n'$ such that $\vec{a}a \mapsto \vec{b}b \in I_{k-1}$ holds. Here, $\vec{a}a$ and $\vec{b}b$ are rearranged in order.
- First consider the case $|a_{i+1} - a_i|_k = |b_{i+1} - b_i|_k < \infty$ and $a_{i+1} > a > a_i$. Then, $|a_{i+1} - a|_{k-1} < \infty$ or $|a - a_i|_{k-1} < \infty$ hold. For instance, if $|a - a_i|_{k-1} = d < \infty$, then $a = a_i + d$ and we may take $b = b_i + d$.
- Next consider the case $|a_{i+1} - a_i|_k = |b_{i+1} - b_i|_k = \infty$ and $a_{i+1} > a > a_i$. Then $|a_{i+1} - a|_{k-1} = \infty$ or $|a - a_i|_{k-1} = \infty$ holds. If one is $< \infty$, then b is determined in the same way as above. If both are ∞, b can be taken so that $|b_{i+1} - b|_{k-1} = \infty$ and $|b - b_i|_{k-1} = \infty$.
- Therefore, we have $I_0 \neq \varnothing$. More strictly, we obtain $[n] \simeq^m [n']$.
- Thus, by the EF theorem, for $n, n' \geq 2^m$, $[n] \equiv_m [n']$. □

Theorem 3.36

There is no first-order formula expressing the parity of length of a finite linear order.

Proof Assume we have such a formula φ. Let $\mathrm{qr}(\varphi) = m$. Then by the above lemma, linear orders longer than 2^m cannot be separated by φ, a contradiction. □

The connectivity of graphs cannot be defined by a first-order formula.

- We show this by reducing the parity problem of linear orders to it. We first make a special graph from a linear order.
- Given a linear order $<$, let $\mathrm{succ}(x,y) \equiv (x<y) \land \forall z(z \le x \lor y \le z)$ and $\mathrm{succ2}(x,y) \equiv \exists z(\mathrm{succ}(x,z) \land \mathrm{succ}(z,y))$.
  Also let $\mathrm{first}(x) \equiv \neg\exists y\ \mathrm{succ}(y,x)$ and $\mathrm{last}(x) \equiv \neg\exists y\ \mathrm{succ}(x,y)$
- Finally, we make a graph on $V=n$ by defining $\mathrm{edge}(x,y)$ as follows.
  $$\mathrm{edge}(x,y) \equiv \mathrm{succ2}(x,y) \lor$$
  $$((\exists z(\mathrm{succ}(x,z) \land \mathrm{last}(z)) \land \mathrm{first}(y))) \lor (\mathrm{last}(x) \land (\exists z(\mathrm{first}(z) \land \mathrm{succ}(z,y))))$$
  In this graph, every other points in a line are connected by an edge, and the first point is connected from the second last point, and also the second point is from the last point.
- If a linear order has even number of points, the graph becomes two cycles (disconnected), and if odd number, it results in a single cycle.
- In other words, if the connectivity of a graph can be defined, then the parity of the length of a linear order can be defined, a contradiction.

Homework 3.5.1

Given a finitely connected graph, the existence of an Eulerian cycle in it cannot be described in first-order logic.

- To expand the scope of application of the EF theorem, we would like to consider structures with functions.
- Rewriting functions as relations requires the use of extra quantifiers for function composition, and the need to use more complicated formulas for atomic formulas involving functions.
- However, there are no big problems when dealing with arbitrary ranks. For example, the following argument is possible for groups.
- $G_1 \equiv G_2 \Rightarrow G_1 \times H \equiv G_2 \times H$ for three groups G_1, G_2, H. For this proof, we observe that II's winning play $\vec{g_1} \leftrightarrow \vec{g_2}$ in EF$_n(G_1, G_2)$ can be modified as II's winning play $(\vec{g_1}, \vec{h}) \leftrightarrow (\vec{g_2}, \vec{h})$ in EF$_n(G_1 \times H, G_2 \times H)$.

§3.5 Presburger arithmetic

- **Presburger arithmetic** is a first order theory for structure $\mathcal{N} = (\mathbb{N}, 0, 1, +)$ in the language $\mathcal{L}_{\mathrm{P}} = \{0, 1, +\}$.
- We want to find a method to determine whether or not $\mathcal{N} \models \sigma$ holds for a sentence σ in the language $\mathcal{L}_{\mathrm{P}}$.
- The method we adopt here is an application of computational models such as automata. This technique will be extended to second-order logic in the next semester.
- Note that in Presburger arithmetic, $<$ is defined as $x < y \leftrightarrow \exists z(x + z + 1 = y)$. The congruence relation $\equiv_k$ is also defined. Then Presburger arithmetic with $<$ and $\equiv_k$ admits the elimination of quantifiers, which is another method of solving the decision problem.

- First, let us consider how to express the sequence of natural numbers $(n_1, n_2, \ldots, n_s)$ (where $s > 0$) in terms of a word that recognized by the automaton.

- We first consider a vertical vectors of length s with elements $0, 1$ as a symbol handled by an automaton. So, the alphabet Ω_s consists of 2^s symbols defined by

$$\vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix} \quad \text{where } b_1, b_2, \ldots, b_s = 0 \text{ or } 1.$$

We may also write $\vec{b} = {}^t[b_1, b_2, \ldots, b_s]$.

- A word $\vec{b}_1\vec{b}_2 \ldots \vec{b}_t$ over Ω_s can be expressed as

$$\begin{bmatrix} b_{11} \\ b_{12} \\ \vdots \\ b_{1s} \end{bmatrix} \begin{bmatrix} b_{21} \\ b_{22} \\ \vdots \\ b_{2s} \end{bmatrix} \cdots \begin{bmatrix} b_{t1} \\ b_{t2} \\ \vdots \\ b_{ts} \end{bmatrix} = \begin{bmatrix} b_{11}b_{21} \ldots b_{t1} \\ b_{12}b_{22} \ldots b_{t2} \\ \vdots \\ b_{1s}b_{2s} \ldots b_{ts} \end{bmatrix}$$

- An s-tuple $(n_1, n_2, \ldots, n_s)$ of natural numbers is represented by $\vec{b}_1\vec{b}_2\ldots\vec{b}_t$ as follows.

$$\begin{gathered}
n_1 = b_{11} + b_{21}\cdot 2 + \cdots + b_{t1}\cdot 2^{t-1}\\
n_2 = b_{12} + b_{22}\cdot 2 + \cdots + b_{t2}\cdot 2^{t-1}\\
\vdots\\
n_s = b_{1s} + b_{2s}\cdot 2 + \cdots + b_{ts}\cdot 2^{t-1}
\end{gathered}$$

- In other words, the binary representation of natural number n_i is $b_{ti}b_{(t-1)i}\ldots b_{1i}$.

- So, if we add the zero vector $\vec{0}$ to the right of the word $\vec{b}_1\vec{b}_2\ldots\vec{b}_t$, the resulting sequence $\vec{b}_1\vec{b}_2\ldots\vec{b}_t\vec{0}$ represents the same sequence $(n_1, n_2, \ldots, n_t)$ of natural numbers. But if we add $\vec{0}$ to the left of $\vec{b}_1\vec{b}_2\ldots\vec{b}_t$, the resulting sequence $\vec{0}\ldots\vec{b}_t$ represents $(2n_1, 2n_2, \ldots, 2n_s)$.

- Note that the zero vector $\vec{0}$ is different from the empty string

$$\varepsilon = \begin{bmatrix} \\ \\ \\ \end{bmatrix}.$$

- Regarding an s-tuple of natural numbers $(n_1, n_2, \dots, n_s)$ (where $s > 0$) as a word over Ω_s, we next consider the set of $(n_1, n_2, \dots, n_s)$ that satisfies a given formula $\varphi(x_1, x_2, \dots, x_s)$. Then we will construct an automaton that can accept such a language.
- First, an atomic formula in Presburger arithmetic is expressed as follows.

$$a_1x_1 + a_2x_2 + \cdots + a_sx_s = b, \qquad \cdots \qquad (\star)$$

where a_ix_i is short for $\pm\underbrace{(x_i + x_i + \cdots + x_i)}_{|a_i| \text{ copies}}$ and b for $\pm\underbrace{(1 + 1 + \cdots + 1)}_{|b| \text{ copies}}$.

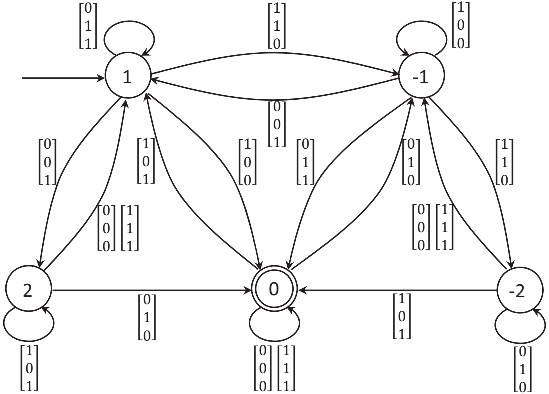
Note that a_i's and b may be negative because terms are transposed to express a formula as $(\star)$.

- Also, we may assume $s > 0$, since by setting $a_i = 0$, you can add the variable x_i meaninglessly.

- Let $\vec{c} = {}^t[c_1, c_2, \ldots, c_s]$ be the first letter of the word representing the solution $(n_1, n_2, \ldots, n_s)$ of Equation $(\star)$.
- Then, let $(n'_1, n'_2, \ldots, n'_s)$ be the sequence of numbers represented by the remaining strings excluding $\vec{c}$. Then for each i,
$$n_i = c_i + 2n'_i.$$
Hence,
$$a_1 n'_1 + a_2 n'_2 + \cdots + a_s n'_s = \frac{b - \Sigma_i a_i c_i}{2}.$$
- Let $M = |b| + \Sigma_i |a_i|$. For any ${}^t[c_1, c_2, \ldots, c_s] \in \Omega$, $|\sum_i a_i c_i| \le \Sigma_i |a_i| \le M$. Then for any $b' \in [-M, M]$, $\frac{b' - \Sigma_i a_i c_i}{2} \in [-M, M]$.
- Now define an automaton $\mathcal{M} = (Q, \Omega_s, \delta, q_0, F)$ for Equation $(\star)$ by:
  - the set of states Q are {the integers in $[-M, M]\} \cup \{\perp\}$.
  - transition function $\delta : Q \times \Omega \to Q$ is defined as follows: for any q $(\neq \perp)$,
  $$\delta(q, \vec{c}) = \frac{q - \Sigma_i a_i c_i}{2} \text{(if it is an integer)}; \ = \perp \text{(otherwise)}; \ \delta(\perp, \vec{c}) = \perp.$$
  - the initial state $q_0 = b$,
  - the set of final states $F = \{0\}$.

Example 6

The transition of an automaton for $x_1 + 2x_2 - 3x_3 = 1$ is shown as follows. The arrows entering the deadlock state $\perp$ are omitted in the picture.


1
-1
2
0
-2
$\begin{bmatrix}0\\1\\1\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}1\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\1\end{bmatrix}$
$\begin{bmatrix}0\\0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\\1\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$

- At state 1, if the input symbol is ${}^t[0,0,0]$, it enters the deadlock. For such an input, n_1, n_2, and n_3 are all multiples of 2, and so they can not be a solution of $x_1 + 2x_2 - 3x_3 = 1$.
- On the other hand, it accepts the word ${}^t[1,1,0]^t[0,1,1]$, which represents $(n_1, n_2, n_3) = (1, 3, 2)$.

- An automaton thus defined accepts the language of words representing s-tuples $(n_1, n_2, \ldots, n_s)$ that satisfy the atomic formula $\varphi(x_1, x_2, \ldots, x_s)$.

- It is also easy to extend an automaton expressing an atomic formula to that for a Boolean combination of them, since the class of regular languages is closed under Boolean operations.

- It is also easy to add quantifiers. If $\mathcal{M} = (Q, \Omega_s, \delta, q_0, F)$ is a deterministic automaton corresponding to a formula $\varphi(x_1, x_2, \ldots, x_s)$, then a nondeterministic automaton $\mathcal{M}' = (Q, \Omega_{s-1}, \delta', \{q_0\}, F)$ corresponding to $\exists x_1 \varphi(x_1, x_2, \ldots, x_s)$ can be constructed as follows.

  $$\delta'(q, {}^t[c_2, \ldots, c_s]) = \{\delta(q, {}^t[b, c_2, \ldots, c_s]) : b = 0, 1\}$$

  Then $\mathcal{M}'$ accepts a word representing $(n_2, \ldots, n_s)$ iff $\mathcal{M}$ accepts a word representing $(n_1, n_2, \ldots, n_s)$ for some n_1. Note that a nondeterministic automaton can always be transformed into a deterministic automaton.

- The universal quantifier $\forall x$ can be rewritten as $\neg\exists x\neg$.
- Thus, for every formula $\varphi(x_1, x_2, \ldots, x_s)$ in Pressburger arithmetic, we can construct an automaton accepting the language of words representing s-tuples $(n_1, n_2, \ldots, n_s)$ that satisfy the formula $\varphi(x_1, x_2, \ldots, x_s)$.
- For a sentence σ, it can be treated by adding a meaningless variable, and the truth of the sentence can be determined by whether the language accepted by automaton is empty or Ω_1^*.
- Therefore, we obtain the following theorem.

Theorem 3.37

Presburger arithmetic is decidable.

Summary

- By the EF theorem, DLO is decidable.
- DLO is PSPACE-complete. TQBF is polynomial-time reducible to DLO.
- (Gurevich) For any $m > 0$, for any two finite linear sequences L_1, L_2 of length 2^m or greater, $L_1 \equiv_m L_2$.
- For finite linear orders, there is no first-order formula expressing the parity of its length.
- The connectivity of a graph cannot be defined by a first-order formula.
- For every formula $\varphi(x_1, x_2, \ldots, x_s)$ in Presburger arithmetic, we can construct an automaton accepting the language of words representing s-tuples $(n_1, n_2, \ldots, n_s)$ that satisfy the formula $\varphi(x_1, x_2, \ldots, x_s)$.
- Presburger arithmetic is decidable.

Thank you for your attention!

§3.6 Peano arithmetic: Introduction

- So-called "**Peano's postulates**" (1889) is famous as an axiomatic treatment of the natural numbers. However, it is not a formal system in the sense of modern logic, since its underlying logic is ambiguous. Moreover, we should also notice previous advanced studies by C.S. Peirce (1881) and R. Dedekind (1888).
- It was Hilbert who began to consider natural number theory as a formal theory in first-order logic.
- The decision problem must be considered the main problem of mathematical logic. (Hilbert-Ackermann, 1928).
- FO$(\mathbb{N}, +)$ is decidable. (Presburger, 1929)
- FO$(\mathbb{N}, \cdot)$ is decidable. (Skolem, 1930)
- Ramsey, On a problem of formal logic, 1930.
- Gödel's undecidable arithmetical propositions, 1931.

G. Peano

C.S. Peirce

R. Dedekind

Peano arithmetic is a first-order theory in the language of ordered rings $\mathcal{L}_{\mathrm{OR}} = \{+, \cdot, 0, 1, <\}$, consists of the following mathematical axioms.

Definition 4.1

Peano arithmetic (PA) has the following formulas in $\mathcal{L}_{\mathrm{OR}}$ as a mathematical axiom.

Successor:	$\neg(x+1=0)$,	$x+1=y+1 \rightarrow x=y$.
Addition:	$x+0=x$,	$x+(y+1)=(x+y)+1$.
Multiplication:	$x \cdot 0 = 0$,	$x \cdot (y+1) = x \cdot y + x$.
Inequality	$\neg(x<0)$,	$x<y+1 \leftrightarrow x<y \vee x=y$.
Induction:	$\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1)) \rightarrow \forall x \varphi(x)$.	

- Induction is not a single formula, but an axiom schema that collects the formulas for all the $\varphi(x)$ in $\mathcal{L}_{\mathrm{OR}}$. Note that $\varphi(x)$ may include free variables other than x.
- In "Peano's postulates", induction is expressed in terms of sets, but Peano arithmetic does not presuppose set theory.

- In a modern formal system, to add a new function, it must be defined explicitly so that the extended system is a conservative extension.
- The primitive recursive definition is not an explicit definition. In fact, if we add the primitive recursive definition of multiplication to Presburger arithmetic (a system of only addition), the resulting system loses completeness and decidability, and it is not a conservative extension.
- In other words, multiplication is not definable from addition.
- On the other hand, the inequality $x < y$ can be defined from addition as abbreviation for $\exists z(y = (x + z) + 1)$. However, we prefer to include the inequality as a primitive symbol, because it allows us to define the hierarchy of formulas simply.
- Similarly, in the following, we assume that $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\forall$, $\exists$, etc. are all pre-set.