Logic and Computation II

Part 7. Recursion-theoretic hierarchies

Kazuyuki Tanaka

BIMSA

May 20, 2025

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

Logic and Computation II

- **Part 4. Modal logic**
- **Part 5. Modal μ-calculus**
- **Part 6. Automata on infinite objects**
- **Part 7. Recursion-theoretic hierarchies**

Part 6. Schedule (tentative)

- May 20, (1) Oracle computation and relativization
- May 22, (2) m-reducibility and simple sets
- May 27, (3) T-reducibility and Post's problem
- May 29, (4) Arithmetical hierarchy

§6.7. Parity games

- A parity game $G = (V_{\mathsf{I}}, V_{\mathsf{II}}, E, \pi)$ is a game on a directed graph $(V_{\mathsf{I}} \cup V_{\mathsf{II}}, E)$ with a priority function $\pi : V_{\mathsf{I}} \cup V_{\mathsf{II}} \to \{0, 1, \cdots, k\}$ and $V_{\mathsf{I}} \cap V_{\mathsf{II}} = \varnothing$. Player I **wins** in an infinite path (play) ρ iff the smallest number appearing infinitely often in $\pi(\rho)$ is even.
- A **(memoryless) strategy** for player I is a mapping $\sigma : V_{\mathsf{I}} \to V_{\mathsf{I}} \cup V_{\mathsf{II}}$. Similar for II's τ.
- A play ρ is **consistent** with such a σ if for all i, $\rho_i \in V_{\mathsf{I}} \Rightarrow \sigma(\rho_i) = \rho_{i+1}$. Similarly for τ.
  σ (τ) is a **winning strategy** if player I (II) wins in any play consistent with σ (τ) .
- Let $W_{\mathsf{I}}(G, \sigma)$ be the set of starting points $\rho_0 \in V$ such that σ is a winning strategy for player I. Let
$$W_{\mathsf{I}}(G) = \bigcup_{\mathsf{I}'s \text{ winning strategy } \sigma} W_{\mathsf{I}}(G, \sigma).$$
- Similarly, $W_{\mathsf{II}}(G, \tau)$ and $W_{\mathsf{II}}(G)$ are defined.
- When $W_{\mathsf{I}}(G) \cup W_{\mathsf{II}}(G) = V$, the game G is said to have **memoryless determinacy**.

Lemma 6.26

In any parity game G, there exists a strategy σ for player I such that $W_{\mathsf{I}}(G,\sigma) = W_{\mathsf{I}}(G)$. Similarly, there exists a II's strategy τ such that $W_{\mathsf{II}}(G,\tau) = W_{\mathsf{II}}(G)$.

If there exist σ and τ such that $W_{\mathsf{I}}(G,\sigma) \cup W_{\mathsf{II}}(G,\tau) = V$, game G is said to have **uniform memoryless determinacy**. From the above lemma, if a parity game has memoryless determinacy, it also has uniform memoryless determinacy.

Before proving that any parity game has (uniform) memoryless determinacy, we introduce some notions.

- We say that $v \in V$ is an **absorbing vertex** if no edges exit from v, i.e., $\{w : (v,w) \in E\} = \{v\}$. Note that we assume that no deadlocks exist.
- We say that $v \in V$ is a **vanishing vertex** if no edges enter v, i.e., $\{w : (w,v) \in E\} = \varnothing$.
- Vertices that are neither absorbing nor vanishing are called **relevant vertices**, and the set of such vertices is denoted by V_{r}.
- $\pi(v)$ for $v \in V_{\mathrm{r}}$ is called a **relevant priority**.

Theorem 6.27

Any parity game $G = (V_{\mathsf{I}}, V_{\mathsf{II}}, E, \pi)$ has uniform memoryless determinacy.

Proof We prove by induction on the number of relevant priorities $\pi(V_{\mathrm{r}})$.

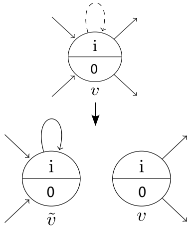
Base case: There are no relevant points, that is, all vertices are absorbing or vanishing.

- From an absorbing vertex v, $v \in W_{\mathsf{I}}(G, \sigma)$ for any σ (if $\pi(v)$ is even), or $v \in W_{\mathsf{II}}(G, \tau)$ for any τ (if it is odd).
- From a vanishing vertex v, each edge goes to an absorbing vertex, and so by selecting an appropriate $\sigma(v)$ or $\tau(v)$, we have $v \in W_{\mathsf{I}}(G, \sigma) \cup W_{\mathsf{II}}(G, \tau)$. Thus, there exist σ and τ such that $W_{\mathsf{I}}(G, \sigma) \cup W_{\mathsf{II}}(G, \tau) = V$.

Induction case: Suppose the number of relevant priorities is $k > 0$.

- We first prove a weak claim $W_{\mathsf{I}}(G) \cup W_{\mathsf{II}}(G) \neq \varnothing$.
- For simplicity, assume that the minimum of the relevant priorities is 0.

- We will modify the game G so that the vertices with priority 0 are changed to non-relevant vertices. Such a modified game is called G^+, to which we will apply the induction hypothesis.
- Let D be the set of relevant vetices with priority 0 in G.
- Make a copy of D and put $\tilde{D} := \{\tilde{v} : v \in D\}$.
- $G^+ = (V_{\text{I}}^+, V_{\text{II}}^+, E^+, \pi^+)$ is defined as follows.
- $V_{\text{I}}^+ := V_{\text{I}} \cup \{\tilde{v} : v \in D \cap V_{\text{I}}\}$,
- $V_{\text{II}}^+ := V_{\text{II}} \cup \{\tilde{v} : v \in D \cap V_{\text{II}}\}$,
- $E^+ := \{(u, v) \in E : v \notin D\} \cup \{(u, \tilde{v}) : (u, v) \in E \wedge v \in D\} \cup \{(\tilde{v}, \tilde{v}) : v \in D\}$
- $\pi^+ := \pi \cup \{(\tilde{v}, 0) : v \in D\}$.

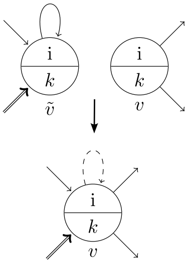


i=I, II

G^+ is obtained by separating each vertex v of D into vanishing vertex v and absorbing vertex $\tilde{v}$.

Therefore, the number of relevant priorities of G^+ is less than that of G.

- By induction hypothesis, there exist σ^+ and τ^+ such that $W_{\mathrm{I}}(G^+, \sigma^+) \cup W_{\mathrm{II}}(G^+, \tau^+) = V^+ = V_{\mathrm{I}}^+ \cup V_{\mathrm{II}}^+$.
- Let $\sigma^\pm : V_{\mathrm{I}} \to V$ and $\tau^\pm : V_{\mathrm{II}} \to V$ strategies in G derived from $\sigma^+ : V_{\mathrm{I}}^+ \to V^+$ and $\tau^+ : V_{\mathrm{II}}^+ \to V^+$, respectively by restricting them to V.
  That is, $\sigma^\pm$ restricts the domain of σ^+ to V_{I}: when $\sigma^+(u) = \tilde{v} \in \tilde{D}$, let $\sigma^\pm(u) = v$. Similarly for $\tau^\pm$.
- Then, we will show that $W_{\mathrm{I}}(G, \sigma^\pm) \cup W_{\mathrm{II}}(G, \tau^\pm) \neq \varnothing$.



- First, consider the case $W_{\mathrm{I}}(G^+, \sigma^+) = V^+$.
- Take any play ρ consistent with $\sigma^\pm$ in G.
- If a vertex of D appears infinitely many times in ρ, then I wins in ρ.
- Otherwise, from some vertex in ρ, its remaining play ρ' does not visit D, and since ρ' also obeys $\sigma^\pm$ in G, ρ' obeys σ^+ in G^+, which means that player I wins in G^+, and thus also wins with ρ' in G, because any finite part of the play makes no difference to the parity condition.
- Therefore, I wins in every play consistent with $\sigma^\pm$ in G. That is, $W_{\mathrm{I}}(G, \sigma^\pm) = V$.

- Next, consider the case $W_{\rm I}(G^+,\sigma^+) \neq V^+$. Since $W_{\rm I}(G^+,\sigma^+) \cup W_{\rm II}(G^+,\tau^+) = V^+$, we have $v \in W_{\rm II}(G^+,\tau^+) = V^+ - W_{\rm I}(G^+,\sigma^+)$.
- Consider a play starting from v consistent with τ^+. If an absorbing vertex $\tilde{v} \in \tilde{D}$ appears in the middle, then after that, it just repeats $\tilde{v}$, and so priority 0 appears infinitely often, which means player I wins, which contradicts with $v \in W_{\rm II}(G^+,\tau^+)$.
- Thus, in such a play of G^+ from v, no vertexes in $D \cup \tilde{D}$ appear except for v as a vanishing vertex.
- Hence, any play of G starting from v and consistent with $\tau^\pm$ does not also enter D in the middle, and so it is also consistent with τ^+, which means player II wins in it. That is, $v \in W_{\rm II}(G,\tau^\pm)$.
- Combining the above two cases, we can say at least $W_{\rm I}(G) \cup W_{\rm II}(G) \neq \varnothing$ for any game G with the number k of relevant priorities.

- Next we show $W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G) = V$. By the way of contradiction, assume $W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G) \neq V$.

- Let $V^- := V - (W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G))$ and consider the game G^- by restricting G to V^-. Note that for every $v \in V^-$ there is a $u \in V^-$ such that $(v, u) \in E$. Because if every u such that $(v, u) \in E$ belongs to $W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G)$, so is v, which contradicts $v \in V^-$. Therefore, the game G^- is a correct parity game.

- In the following, for contradiction, we will show $W_{\mathrm{I}}(G^-) \cup W_{\mathrm{II}}(G^-) = \varnothing$. This contradicts with the previous claim $W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G) \neq \varnothing$, noticing that the number of the relevant priorities of G^- is not larger than that number k of G, and so we can use the induction hypothesis. Therefore, our assumption $W_{\mathrm{I}}(G) \cup W_{\mathrm{II}}(G) \neq V$ is denied.

- First, we assume $W_{\mathrm{I}}(G^-) \neq \varnothing$. Then, let $v \in W_{\mathrm{I}}(G^-)$ and σ^- be a winning strategy for I starting from v in G^-. Consider a play ρ starting at v in G consistent with σ^-. We will show that ρ is also a winning play for I in G, and therefore $v \in W_{\mathrm{I}}(G)$, which contradicts with the choice of $v \in V^-$.

- Now, if ρ is always in G^-, then it is a winning play for I since it is consistent with σ^-. Actually, at $u \in V_{\mathrm{I}} \cap V^-$ in the middle of ρ, the next move is selected within V^- by σ^-.

- At $u \in V_{\mathsf{II}} \cap V^-$ in the middle of ρ, if a vertex of $W_{\mathsf{II}}(G)$ can be chosen as the next move, then u is also in $W_{\mathsf{II}}(G)$, which contradicts with $u \in V^-$.
- At $u \in V_{\mathsf{II}} \cap V^-$ in the middle of ρ, if a vertex of $W_{\mathsf{I}}(G)$ is chosen as the next move, then from the vertex, player I can change strategies to win in G, and thus in sum, we have $v \in W_{\mathrm{I}}(G)$, a contradiction. This shows $W_{\mathrm{I}}(G^-) = \varnothing$.
- Similarly, $W_{\mathsf{II}}(G^-) = \varnothing$. Hence, $W_{\mathrm{I}}(G^-) \cup W_{\mathsf{II}}(G^-) = \varnothing$.
- Since G^- is a parity game with at most k relevant priorities, $W_{\mathrm{I}}(G^-) \cup W_{\mathsf{II}}(G^-) \neq \varnothing$, which denies the assumption of $W_{\mathrm{I}}(G,\sigma) \cup W_{\mathsf{II}}(G,\tau) \neq V$. □

Further readings

The above proof is based on S. Le Roux's paper:
"Memoryless determinacy of infinite parity games: Another simple proof", *Info. Proc. Letters* 143 (2019).
Le Roux's proof also relies on Haddad's paper: " Memoryless determinacy of finite parity games: another simple proof", *Info. Proc. Letters* 132 (2018) 19–21.
which in turn refers to many previous studies.

§7.1. Oracle computation and relativization

- Fix a function $\xi : \mathbb{N} \to \mathbb{N}$. Then, a function $f : \mathbb{N}^n \to \mathbb{N}$ is said to be **computable** in ξ if there exists an algorithm that computes f using ξ as a database.
- Consider a Turing machine as a computational model. Besides the usual input tape and working tapes, it is equipped with an infinite tape storing ξ as data, from which necessary information (values of $\xi(n)$) can be retrieved.
- Such a machine is called an **oracle Turing machine**. A function that can be computed by **oracle** ξ is called **ξ-computable** or **computable in** ξ.
- The three classes of functions defined in part 1 in last semester (primitive recursive functions, recursive functions, and partial recursive functions) are extended as primitive recursive functions in ξ, recursive functions in ξ, and partial recursive functions in ξ, by adding ξ to the initial functions in each definition.

Primitive recursive in ξ

Definition 7.1

Given a function $\xi : \mathbb{N} \to \mathbb{N}$, the functions **primitive recursive in ξ** are defined as below.

1. Constant 0, **successor function** $\mathrm{S}(x) = x + 1$, **projection** $\mathrm{P}^n_i(x_1, x_2, \ldots, x_n) = x_i$ $(1 \leq i \leq n)$ and ξ are primitive recursive in ξ.
2. **Composition**.
   If $g_i : \mathbb{N}^n \to \mathbb{N}$, $h : \mathbb{N}^m \to \mathbb{N}\,(1 \leq i \leq m)$ are primitive recursive in ξ, so is $f = h(g_1, \ldots, g_m) : \mathbb{N}^n \to \mathbb{N}$ defined as below:
   $$f(x_1, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)).$$
3. **Primitive recursion**.
   If $g : \mathbb{N}^n \to \mathbb{N}$, $h : \mathbb{N}^{n+2} \to \mathbb{N}$ are primitive recursive in ξ, so is $f : \mathbb{N}^{n+1} \to \mathbb{N}$ defined as below:
   $$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
   $$f(x_1, \ldots, x_n, y + 1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Recursive in ξ

Definition 7.1

The functions **recursive in ξ** are defined as below.

1. Constant 0,
   Successor function $\mathrm{S}(x) = x + 1$,
   Projection $\mathrm{P}^n_i(x_1, x_2, \cdots, x_n) = x_i$ $(1 \leq i \leq n)$ and ξ are recursive in ξ.
2. **Composition**. Analogous to primitive recursive in ξ.
3. **Primitive recursion**. Analogous to primitive recursive in ξ.
4. **Minimalization** (minimization).
   Let $g : \mathbb{N}^{n+1} \to \mathbb{N}$ be recursive in ξ satisfying that $\forall x_1 \cdots \forall x_n \exists y\ g(x_1, \cdots, x_n, y) = 0$. Then, the function $f : \mathbb{N}^n \to \mathbb{N}$ defined by
   $$f(x_1, \cdots, x_n) = \mu y(g(x_1, \cdots, x_n, y) = 0)$$
   is recursive in ξ, where $\mu y(g(x_1, \cdots, x_n, y) = 0)$ denotes the smallest y such that $g(x_1, \cdots, x_n, y) = 0$.

Partial recursive in ξ (part 1/3)

Definition 7.1

The function **partial recursive in ξ** are defined as follows.

1. Constant 0, **Successor function** $\mathrm{S}(x) = x + 1$, **Projection** $\mathrm{P}_i^n(x_1, x_2, \cdots, x_n) = x_i$ $(1 \leq i \leq n)$ and ξ are partial recursive in ξ.

2. **Composition**. If $g_i : \mathbb{N}^n \to \mathbb{N}, h : \mathbb{N}^m \to \mathbb{N} (1 \leq i \leq m)$ are partial recursive in ξ, the composed function $f = h(g_1, \cdots, g_m) : \mathbb{N}^n \to \mathbb{N}$ defined by

$$f(x_1, \cdots, x_n) \sim h(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n))$$

is partial recursive in ξ, where $h(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n)) = z$ means that each $g_i(x_1, \cdots, x_n) = y_i$ is defined and $h(y_1, \cdots, y_m) = z$.

Note: By $f(x_1, \cdots, x_n) \sim g(x_1, \cdots, x_n)$, we mean that either both functions are undefined or defined with the same value.

Partial recursive in ξ (part 2/3)

Definition 7.1

3. **Primitive recursion.**
   If $g : \mathbb{N}^n \to \mathbb{N}, h : \mathbb{N}^{n+2} \to \mathbb{N}$ are partial recursive in ξ, the function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by

$$\begin{aligned} f(x_1, \cdots, x_n, 0) &\sim g(x_1, \cdots, x_n) \\ f(x_1, \cdots, x_n, y+1) &\sim h(x_1, \cdots, x_n, y, f(x_1, \cdots, x_n, y)) \end{aligned}$$

   is partial recursive in ξ.

Partial recursive in ξ (part 3/3)

Definition 7.1

4. **Minimization.**

- Let $g : \mathbb{N}^{n+1} \to \mathbb{N}$ be partial recursive in ξ.
- If "$g(x_1, \cdots, x_n, c) = 0$, and for each $z < c$, $g(x_1, \cdots, x_n, z)$ is defined with non-zero values", then we put $\mu y(g(x_1, \cdots, x_n, y) = 0) = c$;
  if there is no such c, then $\mu y(g(x_1, \cdots, x_n, y) = 0)$ is undefined.
- Then $f : \mathbb{N}^n \to \mathbb{N}$ satisfying

$$f(x_1, \cdots, x_n) \sim \mu y(g(x_1, \cdots, x_n, y) = 0)$$

is partial recursive in ξ.

Definition 7.1

An n-ary relation $R \subset \mathbb{N}^n$ is called **(primitive) recursive in ξ**, if its characteristic function $\chi_R : \mathbb{N}^n \to \{0, 1\}$ is (primitive) recursive in ξ;

$$\chi_R(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } R(x_1, \ldots, x_n) \\ 0 & \text{otherwise} \end{cases}$$

- All the theorems of recursion theory mentioned in part 1 of the last semester can be extended to statements with oracles, which are called **relativizations** of the original theorems. We will show some examples of relativization in the following slides.
- The (partial) recursive functions in ξ also match the (partial) computable functions in ξ, and the domain of a partial recursive function in ξ is called compututably enumerable in ξ (ξ-CE).

Theorem 7.2 (Relativized Kleene normal form theorem)

There are a primitive recursive function $U(y)$ and a primitive recursive relation in ξ $T^\xi(e, x_1, \cdots, x_n, y)$ such that if $f(x_1, \cdots, x_n)$ is partial recursive in ξ , then there exists e such that

$$f(x_1, \cdots, x_n) \sim U(\mu y T^\xi(e, x_1, \cdots, x_n, y)),$$

where $\mu y T^\xi(e, x_1, \cdots, x_n, y)$ takes the smallest value y satisfying $T^\xi(e, x_1, \cdots, x_n, y)$; if there is no such y, it is undefined.

Proof.

- We define a relation $T^\xi(e, x_1, \cdots, x_n, y)$ as follows:
  $T^\xi(e, x_1, \cdots, x_n, y) \Leftrightarrow$ "y is the Gödel number (code) of the whole computation process γ of TM of index e with input $(x_1, \cdots, x_n)$ and oracle ξ"
- The whole computation process γ is a sequence of configurations $\alpha_0 \triangleright \alpha_1 \triangleright \cdots \triangleright \alpha_n$ with an initial α_0 and an accepting α_n, which can regarded as a word over $\Omega \cup Q \cup \{\triangleright\}$.
- In general, it is not decidable whether a whole computation process γ exists or not. But for a given γ, we can easily check that for each $i < n$, $\alpha_i \triangleright \alpha_{i+1}$ is a valid transition of a TM, as well as α_0 and α_n are an initial and accepting configurations.

Some remarks on the proof

- A primitive recursive function $U(y)$ that extracts the output from the code of the computational process does not depend on ξ. $\square$
- We call $U(\mu y T^\xi(e, x_1, \cdots, x_n, y))$ a **partial recursive function in** ξ of **index** e, denoted as $\{e\}^\xi(x_1, \cdots, x_n)$.
- If ξ in $\{e\}^\xi(x_1, \cdots, x_n)$ is regarded as an argument, it can be rewritten as $\{e\}(x_1, \cdots, x_n, \xi)$.
- Notice that to evaluate $\{e\}(x_1, \cdots, x_n, \xi)$, at most the initial segment $\xi \restriction y$ is used in the calculation, where y is the code of the whole calculation process γ. Furthermore, if the finite sequence $\xi \restriction y$ is identified with its code, $\{e\}(x_1, \cdots, x_n, \xi \restriction y)$ becomes an ordinary partial recursive function.

Definition

Let $U(y)$ and T be primitive recursive functions defined in and after the relativized Kleene normal form theorem. The following function $F : \mathbb{N}^n \times (\mathbb{N}^\mathbb{N})^k \to \mathbb{N}$ is called a **partial recursive functional** with **index** e,

$$F(x_1, \cdots, x_n, \xi_1, \cdots, \xi_k) = U(\mu y T(e, x_1, \cdots, x_n, y, \xi_1 \restriction y, \cdots, \xi_k \restriction y)).$$

- Here $\mathbb{N}^\mathbb{N}$ is the set of total functions from $\mathbb{N}$ to $\mathbb{N}$. The domain D of a partial recursive functional $F : \mathbb{N}^n \times (\mathbb{N}^\mathbb{N})^k \to \mathbb{N}$ is

$$(x_1, \cdots, x_n, \xi_1, \cdots, \xi_k) \in D \Leftrightarrow \exists y T(e, x_1, \cdots, x_n, y, \xi_1 \restriction y, \cdots, \xi_k \restriction y),$$

  which is called a CE set (in a broad sense) or Σ^0_1 set.
- Such general classes will be treated in the following lectures.

Theorem 7.3 (Relativized enumeration theorem)

$\{e\}^\xi(x_1, \cdots, x_n)$ is partial recursive in ξ on $e, x_1, \cdots, x_n$, and it is also a partial recursive functional on $e, x_1, \cdots, x_n, \xi$.

Theorem 7.4 (Relativized parameter theorem)

For any $m, n \geq 1$, there exists a primitive recursive function S_n^m:$\mathbb{N}^{m+1} \to \mathbb{N}$ such that

$$\{e\}^\xi(x_1, \cdots, x_n, y_1, \cdots, y_m) \sim \{S_n^m(e, y_1, \cdots, y_m)\}^\xi(x_1, \cdots, x_n).$$

Theorem 7.5 (Relativized recursion theorem)

Let $f(x_1, \cdots, x_n, y)$ be partial recursive in ξ. There exists e such that

$$\{e\}^\xi(x_1, \cdots, x_n) \sim f(x_1, \cdots, x_n, e).$$

Further Reading

- Kozen, D. C. (2006). *Theory of computation* (Vol. 170). Heidelberg: Springer.
- Soare, R. I. (2016). *Turing computability. Theory and Applications of Computability*. Springer.

Thank you for your attention!