Logic and Computation II

Part 6. Automata on infinite objects

Kazuyuki Tanaka

BIMSA

May 8, 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**

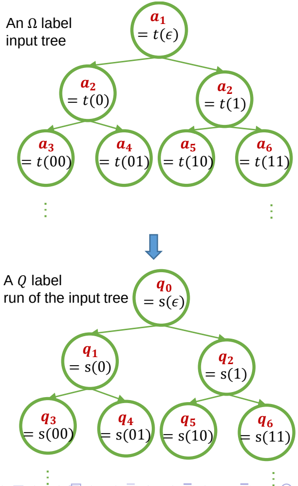
Part 6. Schedule (tentative)

- Apr.15, (1) Second-order arithmetic and analytical hierarchy
- Apr.17, (2) Büchi automata
- Apr.22, (3) Safra's theorem
- Apr.24, (4) The decidability of S1S
- May 6, (5) Tree automata
- May 8, (6) The decidability of S2S
- May 13, (7) Finite model theory
- May 15, (8) Parity games

Parity trees
Parity games
Closed under complement
S2S and MTA

Recap

- An (Ω-)**labeled tree** is the complete binary tree $\{0,1\}^*$ with each vertex labeled by a symbol in Ω. It can be viewed as a function $t : \{0,1\}^* \to \Omega$.
- The **tree automaton** $M = (Q, \Omega, \delta, Q_0, Acc)$:
  - Q: a set of states,
  - $\delta \subseteq Q \times \Omega \times Q^2$: a transition relation,
  - $Q_0 \subseteq Q$: a set of initial states, and
  - Acc: an acceptance conditions.
- For an input Ω-labeled tree $t : \{0,1\}^* \to \Omega$, a run-tree of M is a Q-labeled tree $s : \{0,1\}^* \to Q$ such that
  - $s(\epsilon) \in Q_0$, where ϵ is empty and represents the root of the binary tree.
  - for any $u \in \{0,1\}^*$, $(s(u), t(u), s(u0), s(u1)) \in \delta$.
- To simplify the discussion, assume that for any input, a run-tree can be constructed. (Such an automaton is said to be **complete**).



- For a Q-labeled tree s and an infinite path $\alpha : \mathbb{N} \to \{0,1\}^*$, $s(\alpha)$ denotes the ω-sequence of states (labels) on a path α in s. $\mathsf{Inf}(s(\alpha))$ denotes the set of states which appears infinitely often on $s(\alpha)$.
- An input tree t is accepted by a tree automaton M ($t \in L(M)$) iff there is a run-tree s in which all its infinite paths $s(\alpha)$ satisfy the following condition.
  - For a **Büchi tree automaton** (BTA) M, the acceptance condition Acc is $F(\subseteq Q)$: $\mathsf{Inf}(s(\alpha)) \cap F \neq \varnothing$.
  - For a **Muller tree automaton** (MTA) M, Acc is $\mathcal{F}(\subseteq \mathcal{P}(Q))$: $\mathsf{Inf}(s(\alpha)) \in \mathcal{F}$.
  - For a **Rabin tree automata**(RTA) M, Acc is $\mathcal{F} = \{(G_i, R_i) \mid 1 \leq i \leq k\}$, where $G_i, R_i \subset Q$: there exists i satisfying $\mathsf{Inf}(s(\alpha)) \cap G_i \neq \varnothing$ and $\mathsf{Inf}(s(\alpha)) \cap R_i = \varnothing$.
  - For a **parity tree automaton** (PTA) M, Acc is a priority function $\pi : Q \to \{0, 1, \ldots, k\}$: $\min\{\pi(q) : q \in \mathsf{Inf}(s(\alpha))\}$ is even.
- Even with nondeterminism, BTA has less expressive power than the other three. PTA $\to$ RTA $\to$ NMA is easy, and NMA $\to$ PTA was shown in the last lecture.

Parity condition of PTA

Theorem 6.20

PTA and MTA accept the same languages.

Proof. A parity condition can be easily expressed as a Muller condition: $F \in \mathcal{F}$ iff F is a set of states whose smallest priority is even.
Conversely, given an MTA $M = (Q, \Omega, \delta, Q_0, \mathcal{F})$, we want to construct a PTA $M' = (Q', \Omega, \delta', Q'_0, \pi)$ which accepts the same language.

- Let Q' be the set of permutations of $Q \cup \{\natural\}$ (where $\natural \notin Q$). An element of Q' denotes a **Last Appearing Record** of the states so that the rightmost q corresponds to the current state of M, and $\natural$ represents the place where such q appeared just before now. If $\delta(p, a, r_1, r_2)$ in M and $q_1 \dots q_m \natural q_{m+1} \dots q_n \in Q'$ and $q_n = p, q_i = r_1, q_j = r_2$,
$$\delta'(q_1 \dots q_m \natural q_{m+1} \dots q_n, a, q_1 \dots q_{i-1} \natural q_{i+1} \dots q_n q_i,\ q_1 \dots q_{j-1} \natural q_{j+1} \dots q_n q_j).$$
- A priority function $\pi : Q' \to \{0, 1, \dots, 2|Q| + 1\}$ is defined as follows: For $u \natural v \in Q'$,
$$\pi(u \natural v) = 2|u| \text{ if } \{q \in Q : v \text{ contains } q\} \in \mathcal{F}; \quad = 2|u| + 1 \text{ otherwise.}$$
- Q'_0 can be Q', but a more efficient choice is the set of sequences in Q' with the rightmost belonging to Q_0.

- We compare the run-trees of MTA M and PTA M' for the same input tree.
- A state q that appears finitely (infinitely) many times in a path of the run-tree of M also occurs finitely (infinitely) many times to the right of $\natural$ in the corresponding path of the run-tree of M'.
- Therefore, from a certain time onwards, the states that appear finitely are fixed in a sequence u on the left side of $\natural$, and the states that appears infinitely and $\natural$ are permuted repeatedly. We fix such a sequence u and let V be the set of states not in u.
- If $\natural$ comes to the leftmost in the sequence, that is, if it comes immediately after u, it has the lowest priority. Such cases always occur infinitely. So, V is the set of states appearing infinitely many times. Hence, if a path satisfies the acceptance of M, i.e., $V \in \mathcal{F}$, the lowest priority of states of M' appearing infinitely many time is even, and so it also satisfies the acceptance condition for M'.
- Conversely, consider a path satisfying the acceptance condition of M'. Since the states appearing infinitely with the lowest priority is $u\natural v$ for a sequence v from V, the path also satisfies the acceptance condition of M because the lowest priority must be even.
- Therefore, the accepted tree languages of M and M' are the same. □

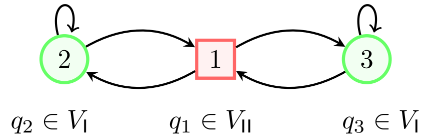
Parity games

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 $\pi : V_{\mathsf{I}} \cup V_{\mathsf{II}} \to \{0, 1, \cdots, k\}$:

- The set of vertices is partitioned into V_{I} and V_{II} ($V_{\mathsf{I}} \cap V_{\mathsf{II}} = \varnothing$).
- Two players, player I and II, move a token along the edges of the graph, which results in a path $\rho = v_0 v_1 \cdots$, called a **play**.
- At a vertex $v \in V_{\mathsf{I}}$ (V_{II}), it is player I (II)'s turn to choose some v' such that $(v, v') \in E$. Note that the choice of v' may depend on the past moves.
- A strategy for player I is a mapping $\sigma : (V_{\mathsf{I}} \cup V_{\mathsf{II}})^{<\omega} V_{\mathsf{I}} \to V_{\mathsf{I}} \cup V_{\mathsf{II}}$.
  A strategy for player II is a mapping $\tau : (V_{\mathsf{I}} \cup V_{\mathsf{II}})^{<\omega} V_{\mathsf{II}} \to V_{\mathsf{I}} \cup V_{\mathsf{II}}$.
- The winner of a finite play is the player whose opponent is unable to move.
- Parity winning condition: Player I wins with an infinite play if the smallest priority that occurs infinitely often in the play is even. II wins otherwise
- σ is a **winning strategy for player I** if whenever he follows σ the resulting play satisfies the parity condition.

Example

Consider the following parity game $G = (V_{\mathsf{I}}, V_{\mathsf{II}}, E, \pi)$, where $V_{\mathsf{I}} = \{q_2, q_3\}$ and $V_{\mathsf{II}} = \{q_1\}$, $\pi(q_i) = i$ for $i = 1, 2, 3$.

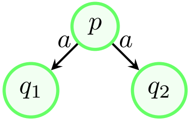


Assume the game starts from q_1, player II has a winning strategy.

- A game G is said to be **determined** if one of the two players has a winning strategy.
- A game G is said to be **positionally determined** if one of the two players has a memoryless winning strategy.
- A **memoryless** strategy for player I is a mapping $\sigma : V_{\mathsf{I}} \to V_{\mathsf{I}} \cup V_{\mathsf{II}}$.
  A **memoryless** strategy for player II is a mapping $\tau : V_{\mathsf{II}} \to V_{\mathsf{I}} \cup V_{\mathsf{II}}$.
- As we'll show later, parity games are positionally determined.

Express a PTA as an infinite game

- Given a PTA $M = (Q, \Omega, \delta, Q_0, \pi)$ and an input tree t, we construct an infinite game $G(M,t)$ in which two players alternately move as follows:
  (1) Player I (Automaton) chooses a next pair of states (q_1, q_2) from $\delta(p, a)$.
  (2) Player II (Path Finder) chooses either 0 or 1 for the next direction.



- The **goal of the Path Finder** is to find a path $\alpha \subseteq \{0,1\}^*$ in the run-tree s that does not satisfy the acceptance condition, whereas the **goal of the Automaton** is to find the Q labels of the run-tree so that the label sequence satisfies the acceptance conditions.
- Player I (automaton) wins in $G(M,t)$ if the label string $s(\alpha)$ produced by the two players satisfies the acceptance condition of M.
- Thus "M accepts $t \Leftrightarrow$ The automaton has a winning strategy in $G(M,t)$."
- Assume the determinacy of this game (either player has a winning strategy),
  "M does not accept $t \Leftrightarrow$ The path finder has a winning strategy in $G(M,t)$."
- For the moment, we also assume the following (which we will prove in next week).
  "The parity game is positionally determined."

Now we present the main lemma.

Lemma 6.21

For any PTA M, there is a PTA M' that accepts the complement of $L(M)$.

Proof.

- Let $M = (Q, \Omega, \delta, Q_0, \pi)$ be a PTA and L^c the complement of $L(M)$. First, we will define a parity game $G(M, t)$ such that
  "an input tree t belongs to $L^c \Leftrightarrow$ player II has a winning strategy."
- $G(M, t) = (V_{\mathsf{I}}, V_{\mathsf{II}}, E, \pi)$ is defined as follows:
  $V_1 = \{0,1\}^* \times Q, \quad V_2 = \{(d, (q, q_0, q_1)) \in \{0,1\}^* \times Q^3 : \delta(q, t(d), q_0, q_1)\},$
  $E = \{(d, (q, q_0, q_1)), (d\hat{\ }i, q_i)) \in V_2 \times V_1 : i = 0, 1\} \cup \{((d, q), (d, (q, q_0, q_1))) \in V_1 \times V_2\}.$
- The game starts with I by choosing an element from $\{\epsilon\} \times Q_0$.
- The priority function of the games follows π of PTA M, i.e., the priority for $(d, (q, q_0, q_1)) \in V_2$ and $(d, q) \in V_1$ are both $\pi(q)$. Then, the same $\pi(q)$ always appears twice consecutively, but it does not matter with the parity condition. Player I wins when the smallest priority appearing infinitely often is even.

- Let S_{II} be the set of total functions from Q^3 to $\{0,1\}$. Then, II's memoryless strategy can be viewed as $\sigma : \{0,1\}^* \to S_{\mathrm{II}}$. Hence, it can also be viewed as a S_{II}-labelled tree.
- So, given an input tree t and a memoryless strategy σ for II (not necessarily a winning strategy), we have a $\Omega \times S_{\mathrm{II}}$-labelled tree of $(a_0, s_0)(a_1, s_1)(a_2, s_2)\cdots(a_n, s_n)$ such that $a_i = t(d_0 d_1 \cdots d_{i-1}), s_i = \sigma(d_0 d_1 \cdots d_{i-1})$ ($d_i \in \{0,1\}$, $0 \leq i \leq n$).
- Moreover, we treat an infinite path $(a_0, s_0)(a_1, s_1)(a_2, s_2)\cdots$ through this tree as an ω-word $\alpha = (a_0, s_0, d_0)(a_1, s_1, d_1)(a_2, s_2, d_2)\cdots$ on $\Omega' = \Omega \times S_{\mathrm{II}} \times \{0,1\}$. Let $L(t,\sigma)$ denote the set of all such words.
- We can define an NPA A which accepts an ω-word $\alpha = (a_0, s_0, d_0)(a_1, s_1, d_1)\cdots$ iff a sequence $q_0, q_1, q_2\cdots$ can be chosen consistently with α to satisfy the parity condition.
- Now, we set $A = (Q, \Omega', \delta', Q_0, \pi)$, where Q, Q_0, π are the same as the PTA M, and $\Omega' = \Omega \times S_{\mathrm{II}} \times \{0,1\}$, and

$$\delta' = \{(q, (a, s, i), q_i) : \text{there exists } (q, a, q_0, q_1) \in \delta \text{ s.t. } s(q, q_0, q_1) = i\}.$$

Note that this definition does not depend on II's strategy σ.

Claim 1

II's memoryless strategy σ is the winning strategy $\Leftrightarrow L(t,\sigma) \cap L(A) = \varnothing$.

$(\Rightarrow)$ By way of contradiction, let $\alpha \in L(t,\sigma) \cap L(A)$.

- Then there exists a run $q_0 q_1 q_2 \cdots$ of A on input α satisfying the parity condition.
- On the other hand, for II's strategy σ, if player I chooses $(q, a, q_0, q_1) \in \delta$ following δ', then they produce a play $q_0 q_1 q_2 \cdots$ in which I wins. So, σ is not a winning strategy.

$(\Leftarrow)$ By way of contradiction, suppose strategy σ is not a winning strategy for II.

- If player I chooses $(q, a, q_0, q_1) \in \delta$ appropriately, there exists $\alpha = (a_0, s_0, d_0)(a_1, s_1, d_1) \cdots$ such that its corresponding $q_0, q_1, q_2 \cdots$ satisfies the parity condition.
- Thus $\alpha \in L(t,\sigma) \cap L(A)$.

Now, since $L(A)$ is an ω-regular language, there exists a DPA $A' = (P, \Omega', \eta, q_0, \pi')$ that accepts the complement of $L(A)$ on Ω'.

- Then we construct a desired PTA M' from a DPA A'. That is, $M' = (P, \Omega, \eta', P_0, \pi')$,

$$\eta' = \{(p, a, p_0, p_1) : \exists s \in S_{\text{II}}\ ((p, (a, s, 0), p_0) \in \eta \wedge (p, (a, s, 1), p_1) \in \eta)\}.$$

Claim 2

$t \in L(M') \Leftrightarrow t \notin L(M)$.

($\Rightarrow$) Suppose $t \in L(M')$, and fix an accepting run-tree r.

- For each node $d \in \{0,1\}^*$ in r, there exists $s_d \in S_{\text{II}}$ satisfying η'.
- Then we merge them to define a memoryless strategy $\sigma : \{0,1\}^* \to S_{\text{II}}$.
- Next, consider a run of DPA A' for an ω-word α in $L(t, \sigma)$. It is the sequence of labels of the tree r for the $\{0,1\}^\omega$ components of α and satisfies the parity condition. So $\alpha \in L(A')$, which means $\alpha \notin L(A)$.
- Thus, $L(t, \sigma) \cap L(A) = \varnothing$. By Claim 1, σ is a memoryless winning strategy for II in $G(M, t)$. Therefore, $t \notin L(M)$.

- ($\Leftarrow$) Suppose $t \notin L(M)$.
- Then player II has a memoryless winning strategy σ in $G(M,t)$, which can be viewed as a S_{II}-labeled tree. From claim 1, $L(t,\sigma) \cap L(A) = \varnothing$, so $L(t,\sigma) \subset L(A')$.
- A P-labeled sequence of DPA A' for a finite subsequence of ω-word α in $L(t,\sigma)$ is uniquely determined. Based on them, there exists a P-labeled tree r which is a run-tree of M' for t.
- Since each P-labeled path of the tree r satisfies the parity condition, r satisfies the acceptance condition of M' and so M' accepts the input tree t. □

The outline of the proof is shown in the following diagram.

PTA M does not accept t $\iff$ II has a winning strategy σ for t in the game $G(M,t)$.

$\Updownarrow$

The $\Omega \times S_{\mathrm{II}}$ labeled $T^{t,\sigma}$ has no path satisfying the parity conditions.

$\Updownarrow$

The ω-language $L(t,\sigma)$ on $\Omega' = \Omega \times S_{\mathrm{II}}$ has no ω-sequence of states satisfying the parity condition.

$\Updownarrow$

PTA M' accepts t. $\iff$ $L(t,\sigma) \cap L(A) = \emptyset$. ← Let A be an NPA which accepts all ω-words on Ω'.

↑ Let A' be a DPA which accepts the complement of $L(A)$. Let M' be a PTA constructed form A'.

Using a parity game similar to $G(M,t)$ above, it is easy to show the following.

Lemma 6.22 (PTA emptiness problem)

It is decidable whether the accepted language of PTA is empty or not.

Proof. Given PTA $M = (Q, \Omega, \delta, Q_0, \pi)$, consider the following parity game $G(M) = (V_1, V_2, E, Q_0, \pi')$.

- $V_1 = Q, \quad V_2 = \delta,$
- $E = \{(q,(q,a,q_0,q_1)) \in V_1 \times V_2\} \cup \{((q,a,q_0,q_1),q_i) \in V_2 \times V_1 : i = 0,1\},$
- $\pi'(q) = \pi(q), \quad \pi'((q,a,q_0,q_1)) = \pi(q).$

This is like removing the position information $d \in \{0,1\}^*$ from the above $G(M,t)$. Therefore,

Player I has a winning strategy in $G(M)$ starting from a state in $Q_0 \Leftrightarrow L(M) \neq \varnothing$

And if player I has a winning strategy in $G(M)$, he has a memoryless winning strategy. Since V_1, V_2 are finite sets, it is decidable in finite steps that player I has a winning strategy. □

S2S and MTA

- Now we will show the equivalence of S2S and MTA.
- First, to translate an S2S formula $\varphi(\vec{x}, \vec{X})$ into a tree language, we need something like the characteristic sequence we defined to translate S1S.
- For simplicity, we replace the first-order variable x with second-order variable X representing the singleton set, and consider the translation of the formula $\varphi(\vec{X})$ with no free occurrences of first-order variables.
- Let $\vec{T} = (T_1, \ldots, T_n)$ be an n-tuple of subsets of $\{0,1\}^*$. Letting $\Omega = \{0,1\}^n$, we express $\vec{T}$ by an Ω-labeled tree $t : \{0,1\}^* \to \{0,1\}^n$ such that for each $i = 1, \ldots, n$,

$$T_i = \{d \in \{0,1\}^* : i\text{th element of } t(d) \text{ is } 1\}$$

  Then, such a t is called the **characteristic representation tree** (representation tree, in short) of $\vec{T}$.

Lemma 6.23

Given an S2S formula $\varphi(\vec{X})$, there exists an MTA M_φ on $\Omega = \{0,1\}^n$ such that,

$$L(M_\varphi) = \{ \text{ The representation tree of } \vec{T} : \varphi(\vec{T}) \text{ holds}\}.$$

Proof. The atomic formula of S2S has a form

$$S_{b_1} S_{b_2} \dots S_{b_k} x \in X \ (\text{where } b_i = 0, 1).$$

Then (d, T) satisfies the above relation iff the word $db_k \dots b_2 b_1$ belongs to T. So, it is easy to construct a PTA M that accepts the set of the representation trees of such (d, T)'s. Furthermore, since the class of languages accepted by MTA's is closed under Boolean operations and projections, any S2S formula has an equivalent MTA. □

Thank you for your attention!