Logic and Computation II

Part 5. Automata on infinite objects

Kazuyuki Tanaka

BIMSA

April 6, 2023

北京雁栖湖
应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF
MATHEMATICAL SCIENCES AND APPLICATIONS

Logic and Computation II

- **Part 4. Formal arithmetic and Gödel's incompleteness theorems**
- **Part 5. Automata on infinite objects**
- **Part 6. Recursion-theoretic hierarchies**
- **Part 7. Admissible ordinals and second order arithmetic**

Part 4. Schedule

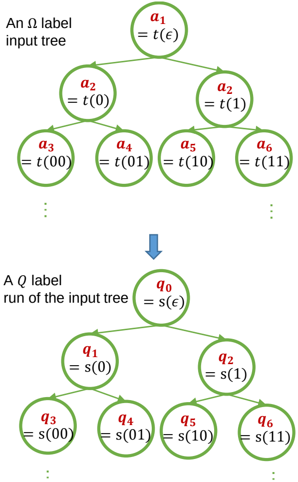
- Mar.28, (1) Automata on infinite strings
- Mar.30, (2) The decidability of S1S
- Apr. 4, (3) Tree automata
- Apr. 6, (4) The decidability of S2S
- Apr.11, (5) Finite model theory
- Apr.13, (6) Parity games

Today's topics

1. Recap
2. Parity games
3. Closed under complement
4. S2S and MTA
5. Decidability of S2S

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.

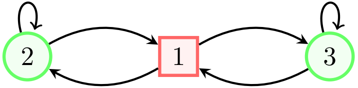
Parity games

A parity game $G = (V_{\text{I}}, V_{\text{II}}, E, \pi)$ is a game on a directed graph $(V_{\text{I}} \cup V_{\text{II}}, E)$ with a priority function $\pi : V_{\text{I}} \cup V_{\text{II}} \to \{0, 1, \cdots, k\}$:

- The set of vertices is partitioned into V_{I} and V_{II} ($V_{\text{I}} \cap V_{\text{II}} = \varnothing$).
- 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 $v \in V_{\text{I}}$ (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 $\sigma : (V_{\text{I}} \cup V_{\text{II}})^{<\omega} V_{\text{I}} \to V_{\text{I}} \cup V_{\text{II}}$.
  A strategy for player II is a mapping $\tau : (V_{\text{I}} \cup V_{\text{II}})^{<\omega} V_{\text{II}} \to V_{\text{I}} \cup V_{\text{II}}$.
- 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 $i = 1, 2, 3$.



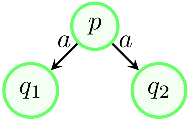
$q_2 \in V_{\mathsf{I}}$ $\quad q_1 \in V_{\mathsf{II}}$ $\quad q_3 \in V_{\mathsf{I}}$

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 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 has a memoryless winning strategy."

Now we present the main lemma.

Lemma

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

PTA M does not accept t $\Leftrightarrow$ 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. $\Leftrightarrow$ $L(t,\sigma) \cap L(A) = \emptyset$. ← 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'.

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."
- Sets V_1, V_2 of vertices (positions) of player I, II ($V_1 \cap V_2 = \varnothing$) and a set of edges (legal moves) $E \subset (V_1 \times V_2) \cup (V_2 \times V_1)$ are defined as follows:
  - $V_1 = \{0,1\}^* \times Q$,
  - $V_2 = \{(d,(q,q_0,q_1)) \in \{0,1\}^* \times Q^3 : \delta(q, t(d), q_0, q_1)\}$,
  - $E = \{(d,(q,q_0,q_1)),(d\hat{\ }i, q_i)) \in V_2 \times V_1 : i = 0,1\} \cup \{((d,q),(d,(q,q_0,q_1))) \in V_1 \times V_2\}$.
- The game starts with II 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 $f : V_2 \to V_1$ be a memoryless strategy for II (not necessarily a winning strategy).
- Let S_{II} be the set of total functions from Q^3 to $\{0, 1\}$.
- Since II's memoryless strategy can be viewed as $\sigma : \{0, 1\}^* \to S_{\mathsf{II}}$, it can also be viewed as a S_{II}-labeled tree.
- So, for a path $d_0 d_1 d_2 \cdots$ of $\{0, 1\}^*$, let $(a_0, s_0)(a_1, s_1)(a_2, s_2) \cdots$ be a path of a $\Omega \times S_{\mathsf{II}}$-labeled tree, where $a_n = t(d_0 d_1 \cdots d_{n-1}), s_n = \sigma(d_0 d_1 \cdots d_{n-1}) (n \geq 0)$.
- Moreover, we treat this path 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 $\Omega' = \Omega \times S_{\mathsf{II}} \times \{0, 1\}$. Let $L(t, \sigma)$ denote the set of all such words.
- We can define an NPA A which accepts an ω word $\alpha = (a_0, s_0, d_0)(a_1, s_1, d_1)(a_2, s_2, d_2) \cdots$ iff an infinite sequence $q_0, q_1, q_2 \cdots$ can be chosen consistently with α to satisfy the parity condition.
- In fact, for $A = (Q, \Omega', \delta', Q_0, \pi)$, Q, Q_0, π are the same as the PTA M, and $\Omega' = \Omega \times S_{\mathsf{II}} \times \{0, 1\}$, and

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

Then the following holds.

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.

- To show $\Leftarrow$, B.W.O.C., suppose strategy σ is not a winning strategy for II.
- Then, if player I chooses $(q, a, q_0, q_1) \in \delta$ appropriately, there exists $\alpha = (a_0, s_0, d_0)(a_1, s_1, d_1)(a_2, s_2, d_2) \cdots$ 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, $M' = (P, \Omega, \eta', P_0, \pi')$,

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

Claim 2

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

- $(\Rightarrow)$ For a $t \in L(M')$, we fix an accepting run-tree r.
- For each node $d \in \{0,1\}^*$ in r, there exists $s_d \in S_{\mathrm{II}}$ satisfying η'.
- Then we merge them to define a memoryless strategy $\sigma : \{0,1\}^* \to S_{\mathrm{II}}$.
- 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. □

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

Lemma (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

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. □

Conversely, let $\{P_a : a \in \Omega\}$ $(P_a = t^{-1}(a))$ be the partition of $\{0,1\}^*$ determined by the Ω- labeled tree t. If an S2S formula φ holds in the structure

$$(\{0,1\}^* \cup \mathcal{P}(\{0,1\}^*), S_0(x), S_1(x), \in, P_a)_{a\in\Omega},$$

φ is said to holds in t. Then,

Lemma

Given an MTA M on Ω, there exists an S2S formula φ_M containing $P_a(a \in \Omega)$ as a set constant such that

$$t \in L(M) \Leftrightarrow \varphi_M \text{ holds in } t.$$

Proof. The idea of constructing the S2S formula φ_M from MTA M is almost the same as the proof of the lemma for S1S. First, the basic predicates of S1S can be used in S2S. For example, "$x = y$", "$X \subseteq Y$", "$X = Y$" etc. can be used. In addition, we define

- "$x = \epsilon$" : $\neg\exists y(S_0 y = x \vee S_1 y = x)$.
- "$\mathrm{Path}(X)$" : $\exists x \in X(x = \epsilon) \wedge \forall x \in X(x \neq \epsilon \to \exists y \in X(S_0 y = x \vee S_1 y = s))\wedge$
  $\forall x \in X \exists! y(S_0 x = y \vee S_1 x = y)$.

Now, let $M = (Q, \Omega, \delta, Q_0, \mathcal{F})$ be a *complete* (no dead ends in state transitions) MTA . Then, if the input tree is represented by $\{P_a : a \in \Omega\}$, the run-tree $\vec{Y} = \{Y_q\}$ (Y_q is the set of vertices with label q) is expressed as follows.

$$\begin{aligned}\mathrm{run}(\vec{Y}) \;=\; & \bigvee_{q \in Q_0} \epsilon \in Y_q \\ & \wedge \forall x \bigvee_{(q,a,q_0,q_1) \in \delta} (x \in Y_q \wedge P_a(x) \wedge x0 \in Y_{q_0} \wedge x1 \in Y_{q_1}) \\ & \wedge \forall x \bigwedge_{p \neq q} \neg(x \in Y_p \wedge x \in Y_q)\end{aligned}$$

Furthermore, the Muller acceptance condition is expressed as

$$\begin{aligned}\varphi_M \;=\; & \exists \vec{Y}(\mathrm{run}(\vec{Y}) \\ & \wedge \forall X(\mathrm{Path}(X) \to \bigvee_{F \in \mathcal{F}} (\bigwedge_{q \in F} Y_q \cap X \text{ is infinite} \wedge \bigwedge_{q \notin F} Y_q \cap X \text{ is finite})\end{aligned}$$

Obviously, this satisfies the lemma. □

Corollary

S2S is decidable.

Proof. Let σ be an S2S sentence. Its truth can be determined by checking whether or not the emptiness problem of the MTA language equivalent to $\sigma \wedge (X = X)$. This problem is decidable by the lemma in Page 16 of this slides. □

Homework

Let $\Omega = \{a, b\}$.
(1) Construct a PTA M_1 that accepts Ω-labeled trees in which a appears finitely.
(2) Construct a a PTA M_2 that accepts Ω-labeled trees in which a appears infinitely many times only in one path.

Bonus Homework

By SωS, we denote the monadic second-order theory of $\mathcal{T}_\omega = (\mathbb{N}^*, \{S_i(x)\}_{i \in \mathbb{N}}, \subset, \preccurlyeq)$, where $S_i(w) = w\hat{\ }i$ $(i \in \mathbb{N})$, $\subset$ is the prefix and $\preccurlyeq$ is the lexicographic order.
Now let $f : \mathbb{N}^* \to \{0,1\}^*$ be

$$f(n_0 n_1 \dots n_{k-1}) = 0^{n_0} 1 0^{n_1} 1 \cdots 1 0^{n_{k-1}} 1, \quad \text{and } f(\epsilon) = \epsilon.$$

Letting D be the range of f, we have $\mathcal{D} = (D, \{S_i^D(x)\}_{i \in \mathbb{N}}, \subset^D, \preccurlyeq^D) \cong \mathcal{T}_\omega$.
Then show that $\mathcal{D}$ is S2S-definable (Note: $\subset$ and $\preccurlyeq$ cannot be defined in $(\mathbb{N}^*, \{S_i(x)\}_{i \in \mathbb{N}})$). From this, derive that SωS is decidable.

Thank you for your attention!