Logic and Computation II
Part 6. Automata on infinite objects

Kazuyuki Tanaka

BIMSA

April 22, 2025

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

§6.3. S1S and Büchi automata

Let Ω be a finite set (alphabet) and Ω^ω be the set of ω-words $a_0 a_1 a_2 \cdots$ on Ω.

Definition 6.9

A **nondeterministic Büchi automaton** (NBA) is a 5-tuple $M = (Q, \Omega, \delta, Q_0, F)$,

(1) Q is a non-empty finite set, whose elements are called **states**.

(2) Ω is a non-empty finite set, whose elements are called **symbols**.

(3) $\delta : Q \times \Omega \to \mathcal{P}(Q)$ is a **transition relation**, where $\mathcal{P}(Q)$ is the power set of Q.

(4) $Q_0 \subset Q$ is a set of **initial states**.

(5) $F \subset Q$ is a set of **final states**.

$(p, a, q) \in \delta$ represents that M can make a transition from state p to state q for input a. M is **deterministic** (DBA) if δ is a single-valued function (i.e., $\delta : Q \times \Omega \to Q$) and Q_0 is a singleton set.

An infinite run and its acceptance condition

- A **run** of M on an input ω-word $\alpha = a_0 a_1 a_2 \cdots \in \Omega^\omega$ is an infinite sequence of states
$$q_0 q_1 q_2 \cdots \in Q^\omega$$
such that $q_0 \in Q_0$ and $(q_i, a_i, q_{i+1}) \in \delta \ (i \geq 0)$.
- For an infinite run σ, the set of states that appear infinitely in σ is denoted by $\mathrm{Inf}(\sigma)$. In other words, if $\sigma = q_0 q_1 q_2 \cdots$,
$$\mathrm{Inf}(\sigma) = \bigcap_{n \geq 0} \{q_i \mid i \geq n\}.$$
- An infinite run σ is said to be **accepted by** NBA M if $\mathrm{Inf}(\sigma) \cap F \neq \varnothing$, that is, if a state of F occurs infinitely many times in σ.
- An input word α is **accepted by** NBA M if there is an accepted run on α.
- Thus, the **ω-language** $L(M) \subset \Omega^\omega$ **accepted by** by M is defined as
$$L(M) = \{\alpha \in \Omega^\omega \mid \text{there is a run } \sigma \text{ of } M \text{ on } \alpha \text{ such that } \mathrm{Inf}(\sigma) \cap F \neq \varnothing\}.$$

Definition 6.10

A language accepted by an NBA is called an **ω-regular language**.

Theorem 6.11

The following are equivalent.

- L is an ω-regular language.
- $L = \bigcup_{i<n} U_i V_i^\omega$ for some regular languages $U_i (\subset \Omega^*), V_i \subset (\Omega^+)$.

Theorem 6.12

The emptiness problem for ω-regular languages is decidable in polynomial time.

- It is easy to see the class of ω-regular languages is closed under $\cup$ and $\cap$. The difficulty lies in the closure under complement.
- If an ω-regular language were accepted by a DBA, so is its complement. But, not all ω-regular languages are accepted by some DBA. Therefore, we will consider Muller and Rabin automata, whose deterministic machines can imitate NBA.

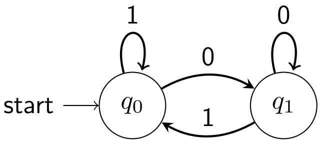
Muller condition

- The acceptance condition of a **Muller automaton** is given by $\mathcal{F} \subseteq \mathcal{P}(Q)$, and a run is accepted iff $\mathrm{Inf}(\sigma) \in \mathcal{F}$.
- Büchi condition ($\mathrm{Inf}(\sigma) \cap F \neq \varnothing$) can be expressed in terms of the Muller condition

$$\mathcal{F} = \{A \subseteq Q \mid A \cap F \neq \varnothing\}.$$

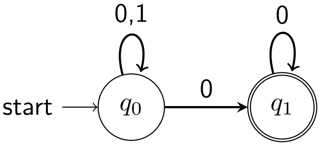
- Non-deterministic / deterministic Muller automata are abbreviated as NMA / DMA.

An DMA accepting $L = (0+1)^*0^\omega$



where $\mathcal{F} = \{\{q_1\}\}$.

An equivalent NBA

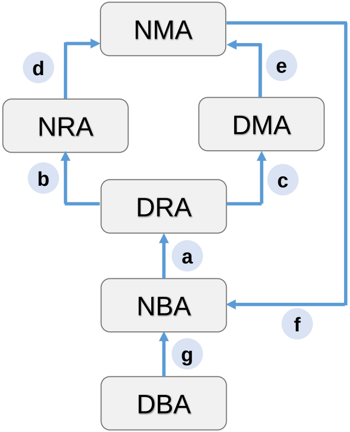


where $F = \{q_1\}$.

Rabin condition

- The acceptance condition of a **Rabin automaton** is given by
$$\mathcal{F} = \{(G_i, R_i) \mid (1 \le i \le k)\},$$
where $G_i, R_i \subset Q$.
- A run σ is **accepted**, if there exists i such that $\mathrm{Inf}(\sigma) \cap G_i \neq \varnothing$ and $\mathrm{Inf}(\sigma) \cap R_i = \varnothing$.
- Non-deterministic / deterministic Rabin automata are abbreviated as NRA / DRA.
- When a G_i/R_i state is visited, we say that the i-th green/red signal is on. A green signal is expected to turn on infinitely many times but a red signal only finitely many.
- A Büchi automaton can be simulated by a Rabin automaton with
$$k = 1,\ G_1 = F,\ R_1 = \varnothing.$$
- A Rabin automaton turns into a Muller automaton if
$$\mathcal{F} = \{A \subseteq Q \mid \bigvee_i (A \cap G_i \neq \varnothing \wedge A \cap R_i = \varnothing)\}$$

- (b), (e) and (g) are obvious. (c) and (d) have been explained above. Now, we are going to show (f).
  - Let M be an NMA with an accepting set $\mathcal{F}$. Goal: construct an NBA N to simulate M.
  - For input x, N mimics M by nondeterministically guessing a run σ of M on x.
  - At some point, N nondeterministically predicts that all states of M not in $\mathrm{Inf}(\sigma)$ have appeared and also guesses that $\mathrm{Inf}(\sigma)$ is a certain set $A \in \mathcal{F}$.
  - Then check if A is indeed $\mathrm{Inf}(\sigma)$ as follows:
    - Any state of σ (from that point) is in A, and
    - Let s be the state of N representing that every state of A appeared at least once. Then N accepts the input if s appears inf. many times.
- (a):NBA$\to$ DRA is the most difficult to prove.



In the figure, "XXA $\to$ YYA" means "for any XXA M_1, there exists a YYA M_2 such that $L(M_1) = L(M_2)$".

NBA→ DRA

- It was first prove by McNaughton in 1966, but his construction was doubly exponential. Safra propose a more efficient exponential construction in 1988.

NBA

Given $B = (Q, \Omega, \delta, Q_0, F)$ with $\mid Q \mid = n$

DRA

We want to construct a deterministic Rabin automaton

$$R = (S, \Omega, \delta', S_0, \{(G_1, R_1), (G_2, R_2) \cdots (G_n, R_n)\})$$

that accepts the same language.

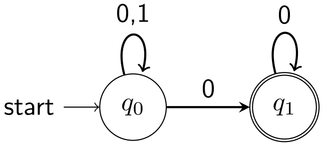
Goal (Safra's Theorem)

Any NBA with n states can be simulated with a DRA consisting of $2^{O(n \log n)}$ states and n pairs of acceptance conditions. Therefore, it can also be simulated with a DMA with the same number of states.

Example 1

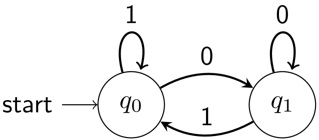
Consider $L = (0+1)^*0^\omega$, where 1 appears finitely times.

An NBA accepting L



This language cannot accepted by any DBA.

A DRA accepting L



where $\mathcal{F} = \{(G_1, R_1)\} = \{(\{q_1\}, \{q_0\})\}$

Two ideas of simimulating an NBA by a DRA

Strategy 1: Determinizing the non-determinism.

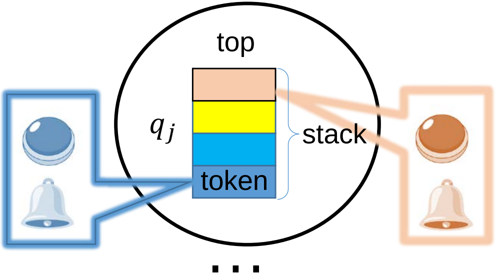
- We express the state transition of an NBA B by moving tokens from some nodes (called **spots**) to another on the diagram of B (called **board**).
- A state of a DRA R is a board with a token on some spots.
- The simulation starts with a board with a token at each initial state of NBA B.
- For an input a, move a token at a spot q to every spot $p \in \delta(q, a)$. Note that the change of the board is deterministic even if $\delta(q, a)$ is non-deterministic.

Strategy 2: Last visiting record to treat the Büchi accepting condition

- Each token should have some partial history of visiting final positions. Indeed, such a token is expressed as a pile of colored tokens, which we call a **stack**.
- A stack is not only moved according to the transition of B, but it may get new color token on the top at a final state. Also, an upper part of a stack may be removed by a certain rule explained below.
- If a color disappear from the boards, sound the buzzer (red signal) of that color. If an invisible color (not on the top) becomes visible, ring the bell (green signal) of the color.
- If a run of B visits a state q infinitely often, R rings the bell of the corresponding color infinite times and sounds the buzzer of that color finite times.

Definition of Stacks

- A **stack** is a pile of colored **tokens**, which is placed on **spots** of a **board**, namely, nodes of the diagram of NBA B. A stack moves from one spot to another along the edges, sometimes changes its contents, and sometimes gets removed.
- A board with some stacks on some spots is a **state** of DRA R.
- The board is connected to different **bells** and **buzzers** for each color.
- The **height** of the stack σ is written as $|\sigma|$.



Buzzer and bell for each color

- A token or its color on the board at time t is said to be **in play** at time t.
- A color is **visible** at time t if it is the color of the token on top of some stack at time t.
- The colors in play at time t are ordered by their **age**, namely the last time they appeared in play. Tokens of the same color in play come into play at the same time and are all of the same age. When adding new tokens, use colors that are not currently in play (reusable) and put them on top of the stack.
- When removing tokens, remove all tokens above a token of a certain color.
- When moving a token, move an entire stack (possibly by making multiple duplicates) to next spots.
- Therefore, in any stack at any time, the tokens are ordered from bottom to top as oldest to youngest.
- At time t, the stacks are linearly ordered by the following reverse lexicographic order $\sigma \ll_t \tau$.
  - σ is a proper extension of τ (τ is obtained by removing the top of σ), or
  - Neither σ nor τ is an extension of the other, and at the lowest position where σ and τ differ in color, the color of σ is older.

The age of colors and order $\ll_t$ change depending on the time, as colors can be added and removed as play progresses.

Simulation

- The simulation starts with a board with one white token at each initial state spot of B.
- Suppose we have a stack of colored tokens at some spot at time t.
- At time $t+1$, the following three steps are all executed in this order:
  Move, Cover, Remove
- It should be noted that we will make a DRA R deterministic. Within a construction step, there may be several processes whose execution order is not essential. Formally, we must clarify the order of such execution. However, since a detailed description would make the whole construction less visible, I leave the details to the reader.
- Thus, we also note that R would use many intermediate states to execute a combination of processes, which are not introduced here, but those do not change the order of the size of R.

Move

- Suppose B reads an input symbol a. For each $q \in Q$, (a copy of) the stack at spot q is moved to each $p \in \delta(q, a)$.
- If there are multiple stacks to put in p, put the smallest stack with respect to $\ll_t$.
- If a certain color disappears in this process, sound the buzzer of that color.

Cover

- For each accepting state $q \in F$, put a token of a new color on the top of a stack at spot q so that stacks with the same visible color are covered with tokens of the same new color, and two with different colors are covered with different new colors.
- New colors enter only in this process. Thus, if color c is placed directly above color d in a stack, then all tokens of color c in play are placed directly above tokens of color d.

Remove

- For any invisible color c in play, remove all tokens above tokens of color c, sound the buzzer of the removed color, and ring the bell of the visible color c.
- Note that when a token is removed in this process, all tokens of that color are removed. The order of removal is not important.

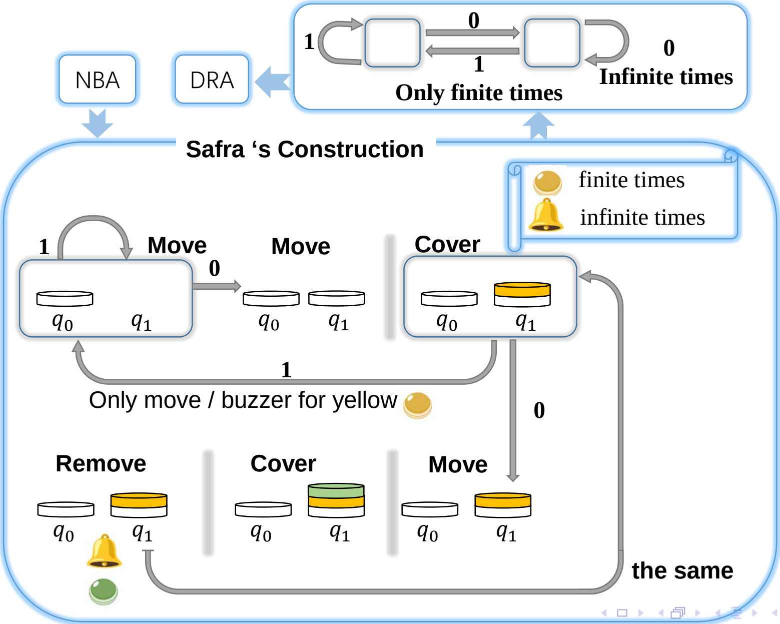
After performing these three steps, there are at most n ($=$ the number of the states) colors left in play. Otherwise, there must be at least one invisible color, then repeat the remove step.

Lemma 6.13

The following are equivalent

(1) An NBA B accepts an input x.

(2) In the DRA R thus constructed, there is a color that rings the bell infinitely many times but sounds the buzzer only finitely many times.


NBA
DRA
1
0
1
0
Only finite times
Infinite times
Safra 's Construction
finite times
infinite times
1
Move
0
Move
Cover
q_0 q_1
1
Only move / buzzer for yellow
0
Remove
Cover
Move
the same

Proof.

To show $(2) \Rightarrow (1)$

- Suppose that there exists a color, say yellow, that rings the bell infinitely many times but the buzzer a finitely many times.
- Let $t_0, t_1, \ldots$ be the time when the yellow bell rings after the last buzzer.
- From time t_0, yellow continues to be in play. Otherwise the buzzer will sound.
- To get a yellow bell at each time t_i, all yellow tokens must be invisible just before and some yellow tokens become visible. In other words, no matter how to move a stack with a yellow token on top from t_i to t_{i+1}, it visits some spot of F.
- Therefore, there exists a run where the state of F appears infinitely.

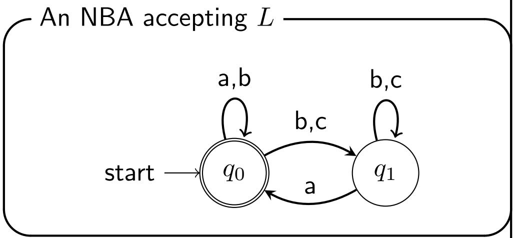
Proof.

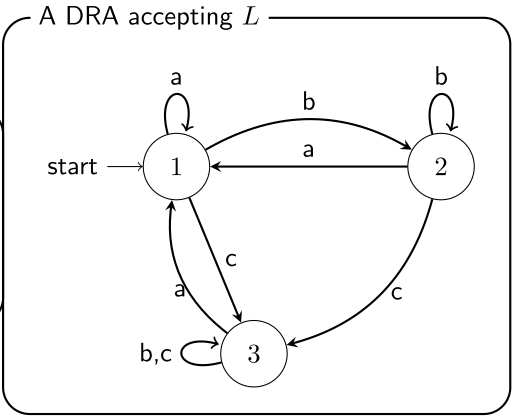
To show $(1) \Rightarrow (2)$

- Conversely, suppose that there is an accepted run ρ of B for x.
- Let σ_t be the stack following ρ at time t.
- Then, set $m = \liminf |\sigma_t|$. In other words, after a certain time t_0, the minimum stack height is m, and it reaches the height m infinitely many times. White (the oldest color) is always in play, so $m \geq 1$, and there are at most n colors in play, so $m \leq n$.
- After time t_0, the color tokens at height $\leq m$ may be replaced by $\ll_t$-smaller ones, which however happens only finite times by the definition of $\ll_t$.
- So, from a certain time t_1, the colors in the stack below m can be assumed to remain unchanged. We assume that the color at the height m is black.
- Since this sequence of actions is an accepted run, a state of F is visited infinitely many times. Although the stack gets a new token each time, eventually the stack height reduces to m again, which rings a black bell.
- Therefore, the black bell rings infinite times and the buzzer sounds only finite times. $\square$

Example 2

Consider $L = \{\{b,c\}^* a \cup b\}^\omega$.





A state of the Rabin automaton R consists of the states of B and the stacks (which may be empty). The number of combinations of stacks is roughly n^n. The treatment of bells and buzzers and auxiliary machineries needs n^n at most. So, the states of R roughly $n^{kn} = 2^{O(n \log n)}$. The acceptance condition consists of n pairs, one for each color.

Therefore, we have

Theorem 6.14 (Safra)

Any NBA with n states can be simulated with a DRA consisting of $2^{O(n \log n)}$ states and n pairs of acceptance conditions. Therefore, it can also be simulated with a DMA with the same number of states.

Corollary 6.15

The class of ω-regular languages is closed with Boolean operations.

Proof.

- We already known that the class of languages accepted by NBA is closed with $\cup$ and $\cap$.
- The closure of complement follows from the above theorem, classes of languages accepted by NBA and DMA are the same.
- In fact, a DMA that accepts the complement of the language of a DMA $M = (Q, \Omega, \delta, q_0, \mathcal{F})$ by replacing the acceptance condition $\mathcal{F}$ of M with $\mathcal{P}(Q) - \mathcal{F}$.

$\square$

Homework

Prove that $L = \{u^\omega : u \in \{0,1\}^+\}$ is not an ω-regular language.

Thank you for your attention!