Omega Automata

December 2023

Ming Gong
Bard College at Simon's Rock

Introduction

- ω-language,
- Büchi automata,
- Properties,
- Application and limitation.

ω-string

Definition

An ω-string over an alphabet Σ is a function $\alpha : \mathbb{N} \to \Sigma$, where $\alpha(n)$ the nth character.

- Example:

$$001001001\ldots$$

$$\alpha(n) = \begin{cases} 1 & \text{if } 3 \mid n, \\ 0 & \text{else.} \end{cases}$$

- An ω-string must be infinite.

ω-Language

Definition

An ω-language is a set of ω-strings.

Definition

The ω-iteration of a language, L^ω, is the set of all infinite strings of the form

$$s_0 s_1 s_2 \ldots$$

where each $s_i \in L$.

- $1010010001\ldots \in (0^*1)^\omega$.
- $\{\epsilon\}^\omega = \emptyset$.

ω-automata

- Variation of finite automata that run on infinite strings.

Definition

An automaton's *run* on an input $s = a_0 a_1 a_2 \ldots$ is a sequence of states

$$q_0 q_1 q_2 \ldots$$

such that $q_{i+1} \in \delta(q_i, a_i)$.

Definition

An ω-automaton is a 5-tuple $M = (Q, \Sigma, \delta, q_0, Acc)$ such that:

- $Acc \subseteq Q^\omega$, a set of runs.

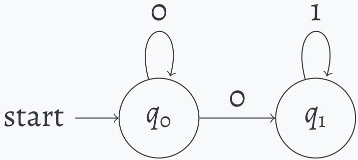
- Encoding $Acc \rightarrow$ different types of ω-automata.

Infinite Set

Definition

The *infinite set* of a run $\mathbf{r}$, $\mathrm{Inf}(\mathbf{r})$ is the set of states that appear infinitely often in $\mathbf{r}$.

Example: On input 001111...



$$\mathbf{r} = q_0 q_0 q_1 q_1 q_1 q_1 q_1 \ldots$$

$$\mathrm{Inf}(\mathbf{r}) = \{q_1\}.$$

Acceptance Conditions

- Büchi: Let $F \subseteq Q$ be a set of *final states*.
  $\mathbf{r} \in Acc$ if $\mathsf{Inf}(\mathbf{r}) \cap F \neq \emptyset$.
  A final state is visited infinitely often.
- Muller: Let $\mathcal{F} \subseteq \mathcal{P}(Q)$ be the *acceptance set*.
  $\mathbf{r} \in Acc$ if $\mathsf{Inf}(\mathbf{r}) \in \mathcal{F}$.
  A specific set of states is visited infinitely often.
- Rabin: Let $\Omega \subseteq \mathcal{P}(Q \times Q)$ be a set of *accepting pairs*.
  $\Omega = \{(E_1, F_1)...(E_k, F_k)\}$.
  $\mathbf{r} \in Acc$ if $\exists (E_i, F_i) \in \Omega$ such that $\mathsf{Inf}(\mathbf{r}) \cap E_i = \emptyset$ but $\mathsf{Inf}(\mathbf{r}) \cap F_i \neq \emptyset$.
  A specific state E_i is visited finitely often, but F_i is visited infinitely often.

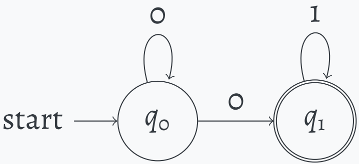
Büchi Automata (NBA)

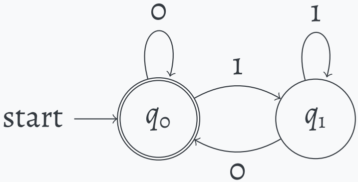
5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$.
- $F \subseteq Q$.

An NBA accepts an input if exists a run such that $\mathsf{Inf}(\mathbf{r}) \cap F \neq \emptyset$.

Examples





Closure Properties

Are the languages of Büchi automata closed under concatenation, iteration, union, intersection, and complement?

Theorem

Büchi-recognizable languages are closed under union.

Proof Idea

Nondeterministically simulate both machines in parallel. If one branch has a successful run, accept.

Intersection

Theorem

Büchi-recognizable languages are closed under intersection.

Proof Idea

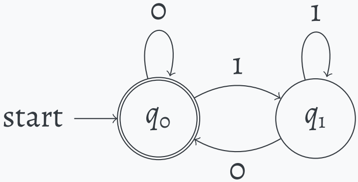
- Goal: accept iff some state in F_1 and some state in F_2 are visited infinitely often.
- Keep track of the states in M_1 and M_2.
- Needn't visit final states simultaneously.
- Two modes: alternatively search for final states.

Proof

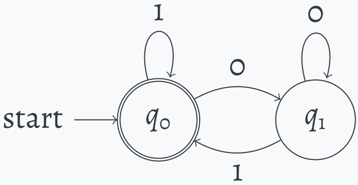
Construct a Büchi automaton $M = (Q, \Sigma, \delta, q_0, F)$, where

- $Q = Q_1 \times Q_2 \times \{1, 2\}$.
- $\delta((q_1, q_2, m), a) = (\delta_1(q_1), \delta_2(q_2), t(m))$.
  $$t(m) = \begin{cases} 1 & (m = 1 \text{ and } q_1 \notin F_1) \text{ or } (m = 2 \text{ and } q_2 \in F_2), \\ 2 & (m = 2 \text{ and } q_2 \notin F_2) \text{ or } (m = 1 \text{ and } q_1 \in F_1). \end{cases}$$
- $q_0 = (q_{01}, q_{02}, 1)$.
- $F = F_1 \times Q_2 \times \{1\} \cup Q_1 \times F_2 \times \{2\}$.

Example

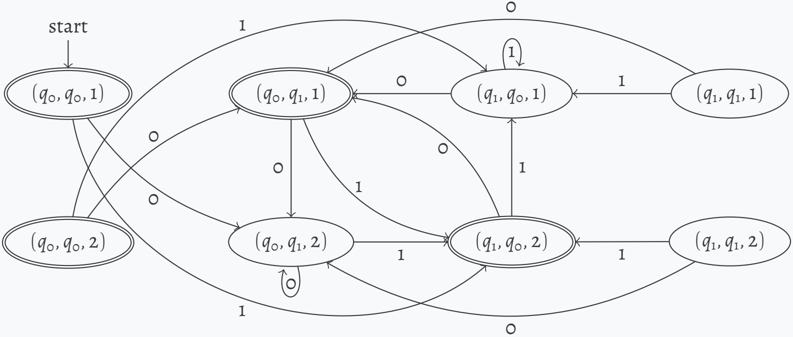


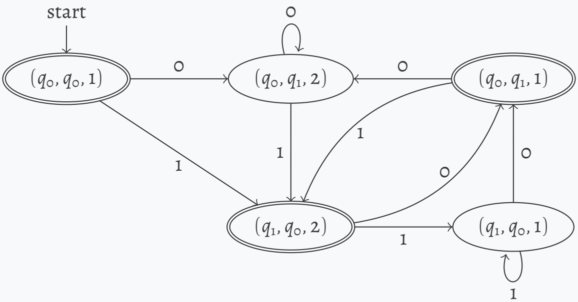
M_1 accepts strings with an infinite number of 0s.



M_2 accepts strings with an infinite number of 1s.

Example





- Mode 1: 0-seeking,
- Mode 2: 1-seeking.

Once a target symbol is read, transition to a "success" state and switch mode.

If not, keep looping.

Büchi's Theorem

Theorem

A language $L \subseteq \Sigma^\omega$ is Büchi recognizable iff L is a finite union of sets RS^ω, where R, S are regular languages.

Proof

$\Rightarrow$ There are two stages in an accepting run:

1. Going from q_0 to q_f,
2. Looping around q_f infinitely often.

$$q_0 \ldots q_f \ldots q_f \ldots q_f \ldots$$

$$L(M) = \bigcup_{q_f \in F} R_{q_0,q_f} \left(R_{q_f,q_f}\right)^\omega ,$$

where $R_{q_a,q_b} = L((Q, \Sigma, \delta, q_a, q_b))$. $\square$

Büchi's Theorem

Theorem

An ω- language is Büchi recognizable iff L is a finite union of sets RS^{ω}, where R, S are regular languages.

Proof

$\Leftarrow$ We can show Büchi recognizable languages are closed under finite union, concatenation, and ω-iteration. □

- Example: $\{(0^n 1^n)^{\omega}; n \geqslant 0\}$ is not Büchi-recognizable.
- Use the pumping lemma to show loop branch cannot be done by a finite number of states.

Deterministic Büchi Automata

Unfortunately, DBAs are less expressive than NBAs.

Proposition

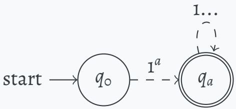
No DBA recognizes the language

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

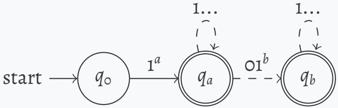
(strings with only finitely many 0s).

Proof

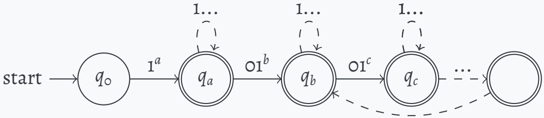
Assume some DBA D recognizes L.



$1^\omega \in L$.



$1^a01^\omega \in L$.



$1^a01^b01^c0\ldots \in L$.

Set of final states is finite, so some must be visited infinitely.

Applications

- Provide a finite way to represent infinite runs.
- Find long-term behaviors for systems not expected to terminate:
  - Network infrastructure,
  - Operating system,
  - Control system.
- Prove decidability problems in mathematical logic.

Works Cited I

Meghyn Bienvenu.

Automata on infinite words and trees, Jan 2010.

Javier Esparza.

Automata theory, Aug 2017.

K Narayan Kumar.

Büchi automata, 2006.

Amaldev Manuel and R. Ramanujam.

Automata over infinite alphabets, Jul 2009.

Works Cited II

Guillaume Sadegh.

Complementing büchi automata, May 2009.

Michael Sipser.

Introduction to the theory of Computation.

Cengage Learning, 2021.

Ivan Zuzak.

Fsm2regex.