Intro to computability

Peter Mayr

Computability Theory, September 28, 2023

What can a computer do in principle?

Hilbert's Tenth Problem (1900)

Given a polynomial $p(x_1, \ldots, x_n)$ with integer coefficients, decide whether it has an integer zero.

Matiyasevich (1970)

No such algorithm exists. Hilbert's Tenth Problem is undecidable.

Other undecidable problems

1. Hilbert's Entscheidungsproblem for first order logic (Church, Turing 1936)
2. Halting Problem for Turing machines
3. Word problem for (semi)groups

What is efficiently computable?

- The computational complexity of an algorithm is usually measured in the time or space (memory) it requires depending on the size of the input.
- This depends on the specific computational model.

Topics of this course

- Models of computation
  - automata, regular languages
  - Turing machines
  - recursive functions
- Undecidability
  - Halting problem
  - Word problem for semigroups
- Degrees of undecidability
  - Turing reductions
  - arithmetical hierarchy
  - Post's problem for Turing degrees
- Computational complexity
  - time and space complexity
  - P vs NP, NP-completeness
  - L, NL, PSPACE

Some classic textbooks we will reference

On computational models:

- Sipser. Introduction to the theory of computation. Thomson Course Technology, Boston, 2nd edition, 2006.

On computability:

- Soare, Robert I. Turing computability : theory and applications. Springer, Berlin, 2016.

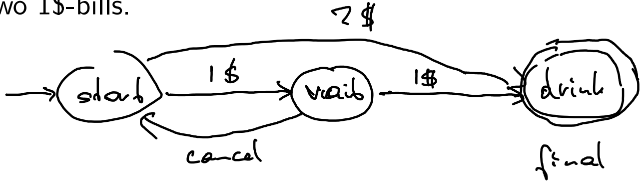
On computational complexity:

- Arora, Sanjeev; Barak, Boaz. Computational complexity: a modern approach. Cambridge University Press, 2007

1. Automata and regular languages

Example

Model a vending machine M_1 that delivers a drink for one 2\$-coin or two 1\$-bills.



Machine accepts inputs

11
2
1C2
1C11

inputs

	δ	1	2	C
states	s:= start	w	d	
	w:= wait	d		s
	d:= drink			

Automata

Definition

A **deterministic finite automaton (DFA)** is a 5-tuple $(Q, \Sigma, \delta, s, F)$ with

- Q a finite set (**states**),
- Σ a finite set (**input alphabet**),
- $\delta\colon Q \times \Sigma \to Q$ the **transition function**,
- $s \in Q$ the **start state**,
- $F \subseteq Q$ the set of **final/accepting states**.

A DFA starts in state s and reads some input string $(a_1, \ldots, a_n)$ for $a_i \in \Sigma$. If it reads a_i in state q_i, it changes to state $\delta(q_i, a_i)$.

Example

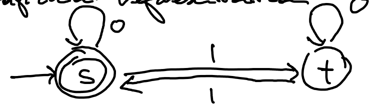
start

$M_2 = (Q, \Sigma, \delta, s, \{s\})$ with $Q = \{s, t\}, \Sigma = \{0, 1\}$.

final

δ	0	1
s	s	t
t	t	s

graphical representation



Every word with an even number of 1s drives M_2 to a final state

Languages

Definition

- $\Sigma^* := \bigcup_{n \in \mathbb{N}} \Sigma^n$ is the set of all **words** over Σ.
  $|w|$ is the **length** of a word.
  $\epsilon \in \Sigma^0$ is the **empty word** (length 0).
- uv is the **concatenation** of words u, v.
- $L \subseteq \Sigma^*$ is a **language**.

$\mathbb{N} = \{0, 1, 2, ..\}$
$|001| = 3$
$\epsilon = ()$

Definition

For a DFA $(Q, \Sigma, \delta, s, F)$ the **extended transition function**

$$\delta^* \colon Q \times \Sigma^* \to Q$$

is defined inductively for $q \in Q, w \in \Sigma^*, a \in \Sigma$ by

$$\begin{aligned} \delta^*(q, \epsilon) &:= q \\ \delta^*(q, wa) &:= \delta(\delta^*(q, w), a) \end{aligned}$$

Definition

Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA, let $w \in \Sigma^*$.

- M **accepts** w if $\delta(s, w) \in F$.
- M **rejects** w otherwise.

The **language of** M is

$$L(M) := \{w \in \Sigma^* \ : \ M \text{ accepts } w\}.$$

Example (continued)

$L(M_2) = \{w \in \{0,1\}^* \ : \ w \text{ contains an even number of 1s } \}$

Example

Is there a DFA M_3 such that
$L(M_3) = \{w \in \{0,1\}^* \ : \ 001 \text{ is a substring of } w\}$?

Nondeterministic finite automata

- **Deterministic:** current state and input symbol uniquely determine next state
- **Nondeterministic:** several choices for next state
  - Interpretation: all possible transitions are done in parallel/the 'right' one is guessed.
  - Not a realistic model of computation but a useful theoretical device for its analysis.

Definition

A **nondeterministic finite automaton with ϵ-transitions (ϵ-NFA)** is a 5-tuple $(Q, \Sigma, \Delta, s, F)$ like a DFA except that

$$\Delta \colon Q \times \Sigma \cup \{\epsilon\} \to P(Q) \qquad (P(Q) \ldots \text{power set of } Q)$$

- Recall ϵ is the empty word, not an element in Σ.
- ϵ-transitions allow the NFA to change from a state q to any state in $\Delta(q, \epsilon)$ without input.
- Wlog, the sets $T := \Delta(q, a)$ for any $a \in \Sigma \cup \{\epsilon\}$ are **ϵ-closed** (i.e. if $t \in T$, then also $\Delta(t, \epsilon) \subseteq T$).

Example

Definition

For an ϵ-NFA $N = (Q, \Sigma, \Delta, s, F)$ the **extended transition function**

$$\Delta^* \colon Q \times \Sigma^* \to P(Q)$$

is defined inductively for $q \in Q, w \in \Sigma^*, a \in \Sigma$ by

$$\begin{array}{ll} \Delta^*(q, \epsilon) & := \Delta(q, \epsilon) \\ \Delta^*(q, wa) & := \bigcup_{r \in \Delta^*(q,w)} \Delta(r, a) \end{array}$$

assuming all $\Delta(q, \epsilon)$ and $\Delta(r, a)$ are ϵ-closed.

- N **accepts** w if $\Delta^*(s, w) \cap F \neq \emptyset$
  (i.e. N accepts w iff $\exists$ some path from s to a state in F that is labelled by w).
- N **rejects** w otherwise.

The **language of** N is

$$L(N) := \{w \in \Sigma^* \ : \ M \text{ accepts } w\}.$$

Note

- Every DFA can be considered as ϵ-NFA with $\Delta(q,a) := \{\delta(q,a)\}$ (singleton) and $\Delta(q,\epsilon) := \emptyset$.
- Every language accepted by a DFA is also accepted by some ϵ-NFA. What about the converse?

Theorem (Subset construction (Rabin, Scott 1959))

Let $N = (Q, \Sigma, \Delta, s, F)$ be an ϵ-NFA with all $\Delta(q,a)$ ϵ-closed. Let $M = (Q', \Sigma, \delta, s', F')$ be the DFA with

- $Q' := P(Q)$,
- $\delta(R,a) := \bigcup_{q \in R} \Delta(q,a)$ for $R \subseteq Q, a \in \Sigma$,
- $s' := \Delta(s,\epsilon)$,
- $F' := \{R \in P(Q) \;:\; R \cap F \neq \emptyset\}$.

Then $L(N) = L(M)$.

Proof

First show for all $w \in \Sigma^*$ that

$$\delta^*(s', w) = \Delta^*(s, w) \qquad (\dagger)$$

by induction on $|w|$.

Base case: For $w = \epsilon$,

$$\delta^*(s', \epsilon) = s' = \Delta(s, \epsilon) = \Delta^*(s, \epsilon).$$

Induction hypothesis: $(\dagger)$ holds for $w \in \Sigma^*$ of length n.

Let $a \in \Sigma$. Then

$$\begin{aligned}
\delta^*(s', wa) &= \delta\big(\,\delta^*(s', w), a\big) && \text{by definition of } \delta^* \\
&= \delta\big(\,\Delta^*(s, w), a\big) && \text{by induction hypothesis} \\
&= \textstyle\bigcup_{q \in \Delta^*(s,w)} \Delta(q, a) && \text{by definition of } \delta \\
&= \Delta^*(s, wa) && \text{by definition of } \Delta^*
\end{aligned}$$

Hence $(\dagger)$ is proved.

Proof, continued

Note: N accepts w iff $\Delta^*(s, w) \cap F \neq \emptyset$

iff $\delta^*(s', w) \in F'$ by (†) and the definition of F'

iff M accepts w. $\square$

Note

The subset construction translates an NFA with $|Q|$ states into a DFA with $2^{|Q|}$ states. Often fewer suffice.

Example, continued

Recall t ϵ-NFA N with 3 states, $L(N) = \{0^\ell 1^m 2^n \ : \ \ell, m, n \in \mathbb{N}\}$. There is a DFA M with $L(M) = L(N)$ and

δ	0	1	2
$\{a, b, c\}$	$\{a, b, c\}$	$\{b, c\}$	$\{c\}$
$\{b, c\}$	$\emptyset$	$\{b, c\}$	$\{c\}$
$\{c\}$	$\emptyset$	$\emptyset$	$\{c\}$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Since the other subsets cannot be reached from the starting state $\{a, b, c\}$, they can be omitted.