بسم الله الرحمن الرحیم

نظریه زبان‌ها و ماشین‌ها

جلسه ۲۸

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

تصمیم‌ناپذیری

- با ماشین تورینگ به عنوان قدرتمندترین ماشین محاسباتی آشنا شدیم.
- چندین مثال برای مسائل قابل حل بیان کردیم.
- یک مثال که قابل حل نیست گفتیم.
- مثالهای دیگر؟

کاهش‌پذیری

- با ماشین تورینگ به عنوان قدرتمندترین ماشین محاسباتی آشنا شدیم.
- چندین مثال برای مسائل قابل حل بیان کردیم.
- یک مثال که قابل حل نیست گفتیم.
- اکنون قصد داریم تکنیک جدیدی برای اثبات حل‌ناپذیری برخی مسائل معرفی کنیم.

A *reduction* is a way of converting one problem to another problem in such a way that a solution to the second problem can be used to solve the first problem.

کاهش‌پذیری

- یک کاهش از A به B، الگوریتمی است که مسئله A را به کمک الگوریتم حل‌کننده مسئله B به عنوان زیربرنامه حل کند.

When A is reducible to B, solving A cannot be harder than solving B because a solution to B gives a solution to A. In terms of computability theory, if A is reducible to B and B is decidable, A also is decidable. Equivalently, if A is undecidable and reducible to B, B is undecidable.

- یادآوری مسائل تصمیم‌پذیر NFA و DFA

مثال

$$HALT_{\mathsf{TM}} = \{\langle M, w\rangle |\ M \text{ is a TM and } M \text{ halts on input } w\}.$$

THEOREM 5.1

$HALT_{\mathsf{TM}}$ is undecidable.

- این زبان، تورینگ تشخیص‌پذیر است.

مثال

$$HALT_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}.$$

THEOREM 5.1

$HALT_{\mathsf{TM}}$ is undecidable.

- اثبات با تناقض
- فرض تصمیم‌پذیر بودن $HALT_{TM}$
- سپس یافتن یک کاهش از A_{TM} به $HALT_{TM}$
- A_{TM} تصمیم‌ناپذیر بود
- پس $HALT_{TM}$ نمیتواند تصمیم‌پذیر باشد

مثال

PROOF Let's assume for the purpose of obtaining a contradiction that TM R decides $HALT_{\mathsf{TM}}$. We construct TM S to decide A_{TM}, with S operating as follows.

$S =$ "On input $\langle M, w\rangle$, an encoding of a TM M and a string w:

1. Run TM R on input $\langle M, w\rangle$.
2. If R rejects, *reject*.
3. If R accepts, simulate M on w until it halts.
4. If M has accepted, *accept*; if M has rejected, *reject*."

Clearly, if R decides $HALT_{\mathsf{TM}}$, then S decides A_{TM}. Because A_{TM} is undecidable, $HALT_{\mathsf{TM}}$ also must be undecidable.

زبان‌های تصمیم‌ناپذیر

$$A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a } \mathsf{TM} \text{ and } M \text{ accepts } w\}.$$

$$HALT_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a } \mathsf{TM} \text{ and } M \text{ halts on input } w\}.$$

$$E_{\mathsf{TM}} = \{\langle M\rangle \mid M \text{ is a } \mathsf{TM} \text{ and } L(M) = \emptyset\}.$$

$$REGULAR_{\mathsf{TM}} = \{\langle M\rangle \mid M \text{ is a } \mathsf{TM} \text{ and } L(M) \text{ is a regular language}\}.$$

$$EQ_{\mathsf{TM}} = \{\langle M_1, M_2\rangle \mid M_1 \text{ and } M_2 \text{ are } \mathsf{TM}\text{s and } L(M_1) = L(M_2)\}.$$

$$ALL_{\mathsf{CFG}} = \{\langle G\rangle \mid G \text{ is a } \mathsf{CFG} \text{ and } L(G) = \Sigma^*\}.$$

مثال

$$E_{\mathsf{TM}} = \{\langle M \rangle |\ M \text{ is a TM and } L(M) = \emptyset\}.$$

THEOREM 5.2

E_{TM} is undecidable.

مثال

PROOF IDEA We follow the pattern adopted in Theorem 5.1. We assume that E_{TM} is decidable and then show that A_{TM} is decidable—a contradiction. Let R be a TM that decides E_{TM}. We use R to construct TM S that decides A_{TM}. How will S work when it receives input $\langle M, w\rangle$?

مثال

One idea is for S to run R on input $\langle M \rangle$ and see whether it accepts. If it does, we know that $L(M)$ is empty and therefore that M does not accept w. But if R rejects $\langle M \rangle$, all we know is that $L(M)$ is not empty and therefore that M accepts some string—but we still do not know whether M accepts the particular string w. So we need to use a different idea.

مثال

Instead of running R on $\langle M \rangle$, we run R on a modification of $\langle M \rangle$. We modify $\langle M \rangle$ to guarantee that M rejects all strings except w, but on input w it works as usual. Then we use R to determine whether the modified machine recognizes the empty language. The only string the machine can now accept is w, so its language will be nonempty iff it accepts w. If R accepts when it is fed a description of the modified machine, we know that the modified machine doesn't accept anything and that M doesn't accept w.

M_1 = "On input x:
1. If $x \neq w$, *reject*.
2. If $x = w$, run M on input w and *accept* if M does."

مثال

IUT-ECE

Putting all this together, we assume that TM R decides E_{TM} and construct TM S that decides A_{TM} as follows.

$S =$ "On input $\langle M, w\rangle$, an encoding of a TM M and a string w:

1. Use the description of M and w to construct the TM M_1 just described.
2. Run R on input $\langle M_1\rangle$.
3. If R accepts, *reject*; if R rejects, *accept*."

مثال

Note that S must actually be able to compute a description of M_1 from a description of M and w. It is able to do so because it only needs to add extra states to M that perform the $x = w$ test.

If R were a decider for E_{TM}, S would be a decider for A_{TM}. A decider for A_{TM} cannot exist, so we know that E_{TM} must be undecidable.

قضیه رایس

- آیا زبان زیر تصمیم‌پذیر است؟

$$REGULAR_{\mathsf{TM}} = \{\langle M\rangle |\ M \text{ is a } \mathsf{TM} \text{ and } L(M) \text{ is a regular language}\}.$$

قضیه رایس

IUT-ECE

Similarly, the problems of testing whether the language of a Turing machine is a context-free language, a decidable language, or even a finite language can be shown to be undecidable with similar proofs. In fact, a general result, called Rice's theorem, states that determining *any property* of the languages recognized by Turing machines is undecidable.

$$P_{TM} = \{< M >|\ M \text{ is a } TM \text{ and } L(M) \text{ has a property } P\}$$

قضیه رایس

Rice's theorem. Let P be any nontrivial property of the language of a Turing machine. Prove that the problem of determining whether a given Turing machine's language has property P is undecidable.

In more formal terms, let P be a language consisting of Turing machine descriptions where P fulfills two conditions. First, P is nontrivial—it contains some, but not all, TM descriptions. Second, P is a property of the TM's language—whenever $L(M_1) = L(M_2)$, we have $\langle M_1 \rangle \in P$ iff $\langle M_2 \rangle \in P$. Here, M_1 and M_2 are any TMs.

قضیه رایس

- آیا زبان زیر تصمیم‌پذیر است؟

$$REGULAR_{\mathsf{TM}} = \{\langle M\rangle |\ M \text{ is a } \mathsf{TM} \text{ and } L(M) \text{ is a regular language}\}.$$

- دیگر مثالها:
  - آیا L(M) یک CFL است؟
  - آیا L(M) شامل palindromes است؟
  - آیا L(M) تهی است؟

خلاصه

IUT-ECE

	Regular	*Context-Free*	*Context-Sensitive*	*D*	RE
Automaton	FSM	PDA	LBA		TM
Grammar(s)	Regular Regular expressions	Context-free	Context-sensitive		Unrestricted
ND = D?	Yes	No	unknown		Yes
Closed under:					
Concatenation	Yes	Yes	Yes	Yes	Yes
Union	Yes	Yes	Yes	Yes	Yes
Kleene star	Yes	Yes	Yes	Yes	Yes
Complement	Yes	No	Yes	Yes	No
Intersection	Yes	No	Yes	Yes	Yes
$\cap$ with Regular	Yes	Yes	Yes	Yes	Yes
Decidable:					
Membership	Yes	Yes	Yes		No
Emptiness	Yes	Yes	No		No
Finiteness	Yes	Yes	No		No
$= \Sigma^*$	Yes	No	No		No
Equivalence	Yes	No	No		No