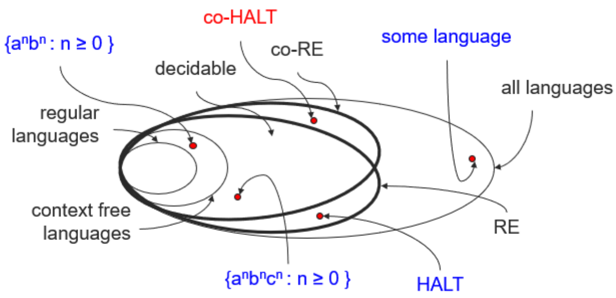
第9章 可判定性和复杂度

语言之间的包含关系



停机问题不可判定

Theorem: HALT is not decidable (undecidable).

Proof:

- Suppose TM H decides HALT
  - if M halts on x, H accept
  - if M does not halt on x, H reject
- Define new TM H': on input <M>
  - if H accepts <M, <M>>, then loop
  - if H rejects <M, <M>>, then halt
- consider H' on input <H'>:
  - if it halts, then H rejects <H', <H'>>, which implies it cannot halt
  - if it loops, then H accepts <H', <H'>>, which implies it must halt
- contradiction. Thus neither H nor H' can exist

a language L is decidable if and only if L is RE and L is co-RE

A natural non-RE Language：the complement of HALT is not recursively enumerable

复杂度

注意：现在开始讨论的都是可判定的问题！

大O表示法

语言是串的集合，而 **complexity class** 是语言的集合；我们已经见到的complexity classes有：

Regular Languages, Context-Free Languages, Decidable Languages, RE Languages, co-RE languages

时间复杂度

Definition: Time complexity class

TIME(t(n)) = {L | there exists a TM M that decides L in time O(t(n))}

单带的时间复杂度和多带的时间复杂度不等价：

Theorem: Let t(n) satisfy t(n)≥n. Every t(n) multitape TM has an equivalent $O(t(n)^2)$ single-tape TM.

多项式时间 P

Definition: TIME(t(n)) = {L | there exists a TM M that decides L in time O(t(n))}

Definition: "P" or "polynomial-time" is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing Machine.

$$P = \cup_{k \geq 1} \text{TIME}(n^k)$$

非确定多项式时间 NP

Definition: NTIME(t(n)) = {L | there exists a NTM M that decides L in time O(t(n))}

$$NP = \cup_{k \geq 1} \text{NTIME}(n^k)$$

多项式时间可验证

Poly-Time Verifiers

"certificate" or "proof"

- NP = {L | L is decided by some poly-time NTM}
- Very useful alternate definition of NP:

Theorem: language L is in NP if and only if it is expressible as:

$$L = \{ x \mid \exists y, |y| \leq |x|^k, \langle x, y\rangle \in R \}$$

efficiently verifiable

where R is a language in P.

- poly-time TM M_R deciding R is a "verifier"

L ∈ NP iff. L = { x | ∃ y, |y| ≤ |x|^k, <x, y> ∈ R }

证明一个问题是NP的：构造NPC，或者用以上定理

归约

将问题A归约到问题B：可以用问题B解决问题A，证明B至少比A要难

形式定义

Definition: A is mapping reducible to B, written $A \leq_m B$, if there is a computable function f such that for all w

$$w \in A \Leftrightarrow f(w) \in B$$

- "yes maps to yes and no maps to no" means:

  $w \in A$ maps to $f(w) \in B$

  & $w \notin A$ maps to $f(w) \notin B$
- f is called the reduction of A to B

归约函数必须是可计算的

Definition: $f : \Sigma^* \to \Sigma^*$ is computable if there exists a TM M_f such that on every $w \in \Sigma^*$ M_f halts on w with f(w) written on its tape.

常见的不可判定问题

$$A_{TM} = \{<M, w> : M \text{ accepts input } w\}$$

$$HALT = \{ <M, x> \mid TM\ M \text{ halts on input } x \}$$

$$E_{TM} = \{<M> : L(M) = Ø\}$$

对于接收/不停机/拒绝的再理解:

1）如果该串w在该语言中，则图灵机必定会停机，并表示接收

2）如果该串w不在该语言中，图灵机或不停机，或停机并告诉你 Reject

所以对于任意图灵机，我们都可以互换它的Accept和Reject，不仅限于Recursive。因为如果他表明了Accept或者Reject，则必定已停机；不停机的串虽然也不在该语言中，但是和Reject有所区别。