# Title Lecture 8

Date 2023·11·22 Review

$A_{CFG} = \{$"G" "w" : G is a CFG that generate w$\}$

$M_{C1}$ = on input "G" "w" 的派生

1. G ⟶ 等价 CFG G' in CNF 2. 列举所有长度为 $2|w|-1$

3. if any of them generate w accept. 4. else, reject.

$A_{PDA} = \{$"P" "w" : P is a PDA that accept w$\}$

$M_{C2}$ = on input "P" "w"

1. P 转化为等价 CFG G 2. run $M_{C1}$ on "G" "w"

3. return $M_{C1}$ 的结果

$E_{CFG} = \{$"G" : G is a CFG that $L(G) = \phi\}$

$M_{C3}$ = on input G

sol. 通过不断替换即可.若可被替换为全 terminal
则不接受.否则拒绝

$E_{PDA} = \{$"P" : P is a PDA with $L(P) = \phi\}$

$M_{C4}$ 利用 $M_{C3}$ 即可

## reduction

A·B 是在 $\Sigma_A$ . $\Sigma_B$ 上的语言

从 A 到 B 的规约是一个 recursive function $f: \Sigma_A \to \Sigma_B$ (computable)

for $\forall x \in \Sigma_A^*$ $x \in A$ iff $f(x) \in B$

存在从 A 到 B 的规约 f，若 B recursive 则 A recursive.

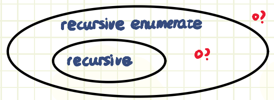
证明: B recursive ⟶ $\exists M_B$ decides B

$M_A$ = on input x 1. 计算 $f(x)$ 2. run $M_B$ on $f(x)$

3. return the result of $M_B$

存在从 A 到 B 的规约，则难度 $A \le B$

若 A non-recursive 则 B non-recursive. 后 - 负




是否为真包含关系 H

是否存在不可判定的语言. Hd

- A B equinumerous 若 ∃ bijection $f: A \to B$ (双射). 若 set 可数 若其有限或 equinumerous with $\mathbb{N}$, 否则不可数.

A 可数 iff ∃ injection $f: A \to \mathbb{N}$ → 任何可数集的子集可数

$\Sigma$ 为 alphabet, 则 $\Sigma^*$ 可数 → 所有 TM 的集合可数

$\Sigma$ 为非空 alphabet, $\mathcal{L}$ 是 $\Sigma$ 上所有语言的集合. $\mathcal{L}$ 不可数

证明: 反证，假设 $\mathcal{L}$ 可数. 则其中的语言为 $L_1 L_2 L_3 \cdots$

since $\Sigma^*$ 可数 $\Sigma^*$ 可表示为 $S_1 S_2 S_3 \cdots$

设 $D = \{s_i : s_i \notin L_i\}$ $s_i \in D$ iff $s_i \notin L_i$ 故 $D \ne L_i$ 故 $D \notin \mathcal{L}$

$$\begin{array}{c|cccc} & S_1 & S_2 & S_3 & \cdots \\ \hline L_1 & 0 & 1 & 0 & \\ L_2 & 1 & 1 & 1 & \\ L_3 & 0 & 0 & 1 & \to D \\ \vdots & & & & \end{array}$$

$S_i$ 为某一串

$L_i$ 为某一语言

D 为新构造的语言

$\mathcal{L}$ 被假设含所有语言

矛盾

↳ 存在不可判定的语言

Title L8

- Halting problem $H = \{\text{"M" "w"} : M \text{ halts on } w\}$

H是 recursive enumerable

→ universal TM：可编程

证明：U = on input "M" "w" 1. run M on w

U halts on "M" "w" <=> M halts on w <=> "w" "M" ∈ H

H is not recursive.

TM的编码 ↓

证明：Hd = {"M" : M is a TM not halt on "M"}

① if H recursive then Hd recursive.

∃ $M_H$ decides H  Md = on input "M" 1. run $M_H$ on "M" "M"

2. if $M_H$ accept "M" "M" reject  3. else accept.

② Hd 不可被半判定

假定 Hd 可半判定，∃D 半判定 Hd

D on input "M" { halt if M not halt on "M"
looping if M halts on "M"

若将 "D" 输入 D，则出现悖论，矛盾.

可将不可判定的 $L_0$ 规约到某一L，以证明L不可判定

$L_1 = \{\text{"M"} : M \text{ is a TM halts on } e\}$ 证明不可判定

sol. 构造 M halts on w iff $M^*$ halts on e

$M^*$ = on input u  1. run M on w

$M^*$ halts on e ⟷ $M^*$ halts on some input

M halts on w ⟷ $M^*$ halts on every input.

若1停机，$M^*$ 可接受所有u

若不停机，$M^*$ 不接受任何u

$L_2 = \{\text{"M"} : M \text{ is a TM that halts on some input}\}$

sol. 同样有 $H \le L_2$

$L_3 = \{\text{"M"} : M \text{ is a TM halts on every input}\}$

sol. 同样有 $H \le L_3$

$L_4 = \{\text{"}M_1\text{" "}M_2\text{"} : \text{TMs}, L(M_1) = L(M_2)\}$

sol. 构造 M halts on every input iff $L(M_1) = L(M_2)$

令 $M_2$ = on input x 1. halt.  $M_1 := M$

then $L(M_1) = L(M_2)$ → $L(M) = \Sigma^*$

$M_A$ 可判 $L_3$  $M_B$ 可判 $L_4$  $M_A$ = on input "M"

$M_A \to M_B$ 归约

1. 构造 $M^*$ 任意输入停机  2. run $M_B$ on "M" "$M^*$"

3. return $M_B$ result