Title Lecture 1

Date 2023 · 9 · 20

Review

总纲

complexity theory
hardness complexity class

automato theory
问题定义、计算模型
finite automata
push down automata
图灵机

- church - Turning theis
图灵机可能是终级的计算模型

computability theory
问题是否可计算

问题的定义

- 优化问题：最小生成树问题
- 查找问题：找到 weight 不超过 k 的生成树
- 判定问题：是否存在生成树 weight 至多 k
- 计数问题：有几棵生成树 weight 至多 k

其中判定问题相较其它几种问题简单

通常将问题化为判定问题确定困难与否

Decision Problem

给一个问题输入，得
Yes → Yes_instance
No → No_instance

对于计算机而言，信息需编码

Graph G，is there a spanning tree < weight k ?

{ $\boxed{\text{encode}}$ (G, k) : (G, k) is a yes-instance } → language

- Alphabet 字符集，有限长 $\Sigma = \{0, \ldots\ldots, 9\}$
- String 字符串，从 Alphabet 中取出 symbols 组序列

! 空集，空字符串存在

- Σ^i：Σ 上所有长度为 i 的 string 集合。

$\Sigma = \{0, 1\}$
$\Sigma^2 = \{00, \cdots, 11\}$

$\Sigma^* = \bigcup_{i \geq 0} \Sigma^i \quad \Sigma^+ = \bigcup_{i \geq 1} \Sigma^i$

- 拼接 $u = 123$ $v = 56$ $uv = 12356$ concatenation

重复 $w = 01$ $w^0 = e$ $w^1 = 01$ $w^2 = 0101$ exponentiation

反向 $w = 01$ $w^R = 10$ reversal

- language : a subset $L \subseteq \Sigma^*$

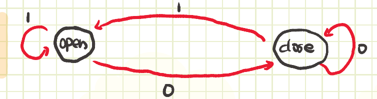
! decision problem 与 language 一一对应

decision problem ⇒ {encodings of yes-instance}

可以将 decision problem 抽象为 language

Title L1

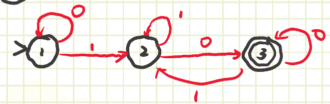
finite automata



⇒ state diagram

→○ ⇒ 表示初始状态，有且唯一

◎ ⇒ 表示结束状态，可无且可不唯一~



(可为空) final state $\subseteq K$

A finite automata $M = (K, \Sigma, \delta, s, F)$

K: 有限状态集合

Σ: input Alphabet

δ: transition func

s: 初始状态 $\in K$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta : K \times \Sigma \to K$ 决定输入后的走向

→ yields in one step

$(q_0, 1010) \vdash_M (q_1, 010) \vdash_M (q_2, 10) \vdash_M (q_1, 0) \vdash_M (q_2, e)$

curr state　未读输入

↳ a configuration, a element of $K \times \Sigma^*$

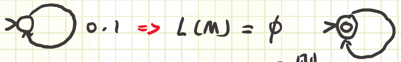
$(q, w) \vdash_M^* (q', w')$　$\vdash_M^*$ 不限 step num

若 $(q, w) = (q', w')$　$(q, w) \vdash_M \cdots \vdash_M (q', w')$

M accepts $w \in \Sigma^*$ if $(s, w) \vdash_M^* (q, e)$ $q \in F$

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

M accepts $L(M)$ ← $w \in L$, M accepts w; $w \notin L$, M doesn't accept w

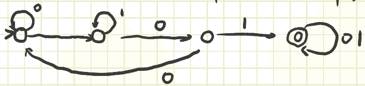


⇒ $L(M) = \phi$　⇒ $L(M) = \{0,1\}^*$

正则

A language is regular if it's accept by some finite automata

证明 $\{w \in \{0,1\}^*, w$ 中含"101"子串$\}$ 正则



Language Regular Operation

- union $A \cup B = \{w : w \in A \text{ or } w \in B\}$
- concatenation $A \cdot B = \{ab : a \in A \text{ and } b \in B\}$
- star $A^* = \{w_1 w_2 \cdots w_k : w_i \in A \text{ and } k \geq 0\}$

正则的 language 在上述操作下封闭

if A · B regular, $A \cup B$ regular (证明)

设 $\Sigma_A = \Sigma_B$ $\exists M_A = (K_A, \Sigma, \delta_A, s_A, F_A)$ $M_B = (K_B, \Sigma, \delta_B, s_B, F_B)$

$K_U = K_A \times K_B$　$s_U = (s_A, s_B)$　$F_U = \{(q_A, q_B) \in K_A \times K_B, q_A \in F_A \text{ or } q_B \in F_B\}$

$\delta_U((q_A, q_B), a) = (\delta_A(q_A, a), \delta_B(q_B, b))$

Title Lecture 2

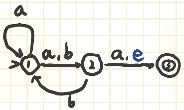
Date 2023 · 9 · 27 Review

若 A 和 B 均正则，证明 A·B 正则

- 非确定性有限状态机

deterministic finite auto. DFA

Non-deterministic ... NFA ⇒



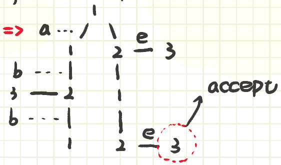
下一状态有多种选择，允许 e-transition

NFA : $(K, \Sigma, \Delta, s, F)$ $\Delta \subseteq K \times (\Sigma \cup \{e\}) \times K$

$(s, w) \vdash_M^* (q, e)$ for some $q \in F$ (只要某种情况可行)

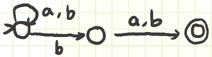
$L(M) = \{w \in \Sigma^* : M \text{ accepts } w\}$ M accepts L(M)

如上图中，给定 (1, abb) ⇒



NFA always make the right guess

$L = \{w \in \{a, b\}^* ;$ 倒数第二个 symbol 为 b $\}$

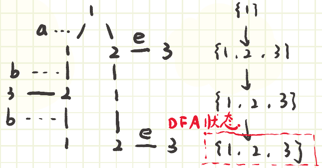


$\forall$ DFA M ⇒ $\exists$ NFA M' st L(M) = L(M') ①

$\forall$ NFA M ⇒ $\exists$ DFA M' st L(M) = L(M') ②

① ⇒ DFA 是 NFA 的一种特殊情况

证明 ② 设 DFA M' 可以模拟树状的计算



$M' = (K', \Sigma, \delta, s', F')$

$K' = 2^K = \{Q : Q \subseteq K\}$

$F' = \{Q \subseteq K : Q \cap F \neq \phi\}$

- $\forall q \in K$ $E(q) = \{p \in K : (q, e) \vdash_M^* (p, e)\}$

$s' = E(s)$ 可能有 e-transition

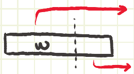
$\delta(Q, a) = \bigcup_{q \in Q} \bigcup_{p:(q,a,p) \in \Delta} E(p)$

claim $\forall p, q \in K$, $w \in \Sigma^*$ $\exists (p, w) \vdash_M^* (q, e)$

可用归纳法证明 iff $(E(p), w) \vdash_{M'}^* (Q, e)$ for some $q \in Q$

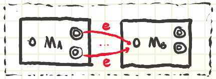
能被某一 NFA 接受的 Language 正则

证明 A·B 正则



$\exists$ DFA M_A, M_B

NFA ⇒



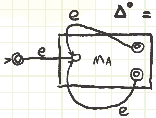
$M_A = \{K_A, \Sigma, \Delta_A, S_A, F_A\}$

$M_B = \{K_B, \Sigma, \Delta_B, S_B, F_B\}$

$M^\circ = \{K_A \cup K_B, \Sigma, \Delta^\circ, S_A, F_B\}$

$\Delta^\circ = \Delta_A \cup \Delta_B \cup \{(q, e, S_B) : q \in F_A\}$

若 A 正则，A^* 正则



Title Lecture 3

Date 2023 · 10 · 11 Review

Regular Expression. (REX)

$R=(a\cup b)^*a$ $L(R)=(\{a\}\cup\{b\})^*\circ\{a\}$

- $L(\phi)=\phi$ $a\in\Sigma$ $L(a)=\{a\}$
- $L(R_1\cup R_2)=L(R_1)\cup L(R_2)$ $L(R_1R_2)=L(R_1)\circ L(R_2)$

  $L(R^*)=(L(R))^*$ ← $a^*b\cup b^*a=((a^*)b)\cup((b^*)a)$

$e \to \phi^*$

$\{w\in\{a\cup b\}^*$; w 开头 a 结尾 b$\}$ → $a(a\cup b)^*b$

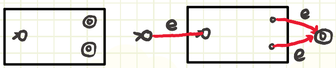
$\{w\in\{a\cup b\}^*$; w 至少两 a$\}$ → $(a\cup b)^*a(a\cup b)^*a(a\cup b)^*$

a language B is regular iff ∃ regular expression R $L(R)=B$

- 将 NFA M 转化为正则表达式 R

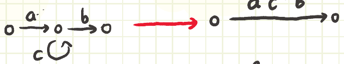
1 ① 简化 M:

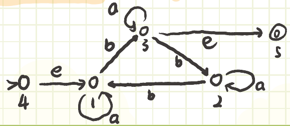
  a) 无状态指向 s



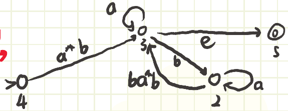
  b) $|F|=1$，不指向任何 state

② 剔除状态

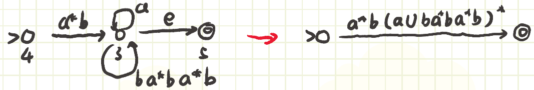




删 q_1



删 q_1



2

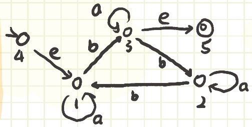
① NFA $M=(K,\Sigma,\delta,s,F)$ $k=\{q_1,\cdots q_n\}$ $s=q_{n-1}$ $F=\{q_n\}$

② $(p,a,q_{n-1})\notin\Delta$ $\forall p,a$ ③ $(q_n,a,p)\notin\Delta$ $\forall a,p$

④ DP → subproblem : $i,j\in[1,n]$ $k\in[1,n]$ $z>k$

$L(R_{ij}^k)=\{w\in\Sigma^*$: w 使 M 从 $q_i\to q_j$，不经过 $q_z\}$

希望求出 $R_{(n-1)n}^{n-2}$



$R_{11}^0=a$

$R_{13}^0=b$

....

base case :

for $i,j\in[1,n]$

if $i==j$: $R_{ij}^0=\phi^*\cup a_1\cdots a_m$

else : $R_{ij}^0=a_1\cup\cdots\cup a_m$ → recurrence $R_{ij}^k=R_{ij}^{k-1}\cup(R_{ik}^{k-1}(R_{kk}^{k-1})^*R_{kj}^{k-1})$

- pumping theorem → pumping length

regular language L. $\exists p\in N^*$ $w\in L$ $|w|\geq p$ 可分为 $w=xyz$

满足 ① $\forall i\geq 0$ $xy^iz\in L$ ② $|y|>0$ ③ $|xy|\leq p$

L 有限 $p=\max_{w\in L}|w|+1$ → 可验证性质

L 无限 $p=|k|$ → 证明：此情况下 w 必有重复的状态.

$\{0^n1^n : n\geq 0\}$ 非正则，证明

Title L3

假定正则，有 pumping length · p

$w = 0^p 1^p \Rightarrow w = xyz$，根据性质有

若 $y = 0^k$ 则 $w = xy^0z = 0^{p-k}1^p$，非正则

（需要 $|xy| \le p$，$|z| \ge p$）

Title Lecture 4

Date 2023·10·18 Review

Context-free language

- context-free schema

S: start symbol

$S \to aSb$ ⇢ rule
$S \to A$ 大写 → non-terminal
$A \to c$ 小写 → terminal
$A \to e$ ⇢ 空

希望将所有 non-terminal → terminal

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aacbb$

- context-free grammer $G = (V, \Sigma, S, R)$ ↱ start symbol

V: 有限 symbol 集合 Σ: 一个 terminal symbol 集合

$S \subseteq V-\Sigma$ R: 规则的集合(有限) $\Sigma \in V$

$(A, w) \longleftrightarrow A \to w$ $R \subseteq (V-\Sigma) \times V^*$

$\forall x, y, z \in V^*$ $\forall A \in V-\Sigma$ $xAy \Rightarrow_G xuy$ if $(A, u) \in R$

↳ derive

a derivation from w to u of length n

$\forall w, u \in V^*$ $w \Rightarrow_G^* u$ if $w = u$ or $w \Rightarrow_G \cdots \Rightarrow_G u$

∴ G 生成 $w \in \Sigma^*$ if $S \Rightarrow_G^* w$ $L(G) = \{w \in \Sigma^* \mid G 生成的w\}$

证明 $\{w \in \{a, b\}^* : w = w^{Reverse}\}$ is context-free

$S \to e$ $S \to a$ $S \to b$ $S \to aSa$ $S \to bSb$

↳ $S \to e \mid a \mid b \mid aSa \mid bSb$

后续严谨来说需证明 $\begin{cases} \text{if } w \in L(G),\ w^R = w \\ \text{if } w = w^R,\ w \in L(G) \end{cases}$

$S \to SS$ $S \to (S)$ $S \to e$

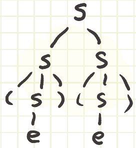
$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()()$ left-most

$S \Rightarrow SS \Rightarrow S(S) \Rightarrow S() \Rightarrow (S)() \Rightarrow ()()$ right-most

二者等价，对应了同一棵语法树 parse tree

- parse tree: internal node: non-terminal

leaf node: e 必是其父节点唯一子节点



( ) ( ) ⇢ yield of parse tree

语法树可能存在歧义，如:

$E \to E+E$ $E \to E \times E$ $E \to (E)$ $E \to e$ (可构造不同语法树)

一些 language 不可能由不歧义的 Grammar 生成

- inherently ambiguous $\{a^i b^j c^k : i = j \text{ or } j = k\}$
- CFG is in CNF if every rule is: $a \in \Sigma$

1. $S \to e$ 2. $A \to BC$ 3. $A \to a$ $B, C \subseteq V - \Sigma - \{S\}$

若 G is a CFG in CNF, G 生成 w $|w| = n \geq 1$

derivation length 为 $2n-1$

Title L4

任意的 CFG 均有一个等价的 CFG 满足 CNF

证明：对于任意 CFG 可能有（使不满足 CNF）

1. S 于 rule 右侧　2. $A \to e$ 当 $A \neq S$
3. $A \to B$. $\exists B \in V-\Sigma$　4. $A \to u_1 \cdots u_k$　$k \geq 3$　$u_i \in V$
5. $A \to u_1 u_2$　$\exists u \in \{u_1, u_2\}$　$u \in \Sigma$

- 修复 1. $S_0 \to S$
- 修复 3. $A \to B$　$\exists B \in V-\Sigma$
  - $A \neq S$，与删 $A \to e$ 同理
  - $A = S$，若删，则需连上 B 之后的所有 rule
- 修复 5　　$A \to BC$
  如 $A \to bC$　　$B \to b$
- 若 $A \to e$　$B \to ACA$
  删去 $A \to e$ 后补上
  $B \to AC$　$B \to CA$　$B \to C$
- 修复 4，将长度变为 2
  $A \to u_1 V_2$　$V_2 \to u_2 \cdots u_k$
  $V_2 \to u_2 V_3$　$V_3 \to u_3 \cdots u_k$
  ....

Pushdown Automata (PDA)

PDA <=> CFG　　PDA = NFA + stack

一个 PDA　$P = (K, \Gamma, \Sigma, \Delta, s, F)$

Γ : stack alphabet

Δ : 有限集合 $\Delta \subseteq (K \times (\Sigma \cup \{e\}) \times \Gamma^*) \times (K \times \Gamma^*)$

当前状态　读到 symbol　旧栈顶　新栈顶 push　该栈状态 (pop)

$((p, a, 123), (q, 45))$

$((p, a, e), (q, \beta))$

未读的 symbols

configuration of PDA　$K \times \Sigma^* \times \Gamma^*$

$(p, x, \alpha) \vdash_P (q, y, \beta)$　$\exists\ ((p, a, r), (q, \tau)) \in \Delta$

有 $x = ay$　$\alpha = r\eta$　$\beta = \tau\eta$　　同理有 $\vdash_P^*$

p accepts $w \in \Sigma^*$ when $(s, w, e) \vdash_P^* (q, e, e)$　$q \in F$

处于 final state　读完 w　栈清空

$L(P)$ 为 P 能接受的 w 的集合，语言

表示数量

设计一个 PDA，接受 $\{w \in \{0,1\}^* ;\ \#1's = \#0's\}$

① $K = \{s, q, f\}$　$\Sigma = \{0, 1\}$　$\Gamma = \{1, 0, \$\}$

$\Delta = \{((s, e, e), (q, \$)), ((q, 0, 0), (q, 00)),$
$((q, 0, \$), (q, 0\$)), ((q, 0, 1), (q, e)),$
$\ldots$ 为 1 类似 $\ldots$ $((q, e, \$), (f, e))\}$

② $K = \{s\}$　$F = \{s\}$，$\Sigma = \Gamma = \{0, 1\}$　"猜测"

$\Delta = \{((s, 0, 1), (s, e)),\ ((s, 0, e), (s, 0)),$
$((s, 1, 0), (s, e)),\ ((s, 1, e), (s, 1))\}$

Title Lecture 5

Date 2023.10.25 Review

CFG ⟷ PDA

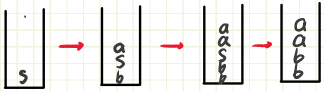
- $G=(V,\Sigma,S,R) \Rightarrow M$ st. $L(M)=L(G)$

① 在 stack 中，非确定地从 S 生成一个字符串

② 与输入比对，相符的话接受.

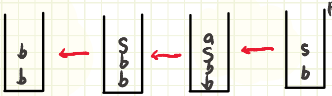
输入 $S\to aSb$

aabb $S\to e$



↓ 匹配. 替换 同时进行

为使 nonterminal 处于栈顶



$\begin{cases} G=(V,\Sigma,S,R) \\ M=(K,\Sigma,\Gamma,\Delta,s,F) \end{cases}$

$\Gamma=V$ $K=\{s,f\}$ $F=\{F\}$ for $(A,u)\in R$

$\Delta=\{((s,e,e),(f,S))$ $((f,e,A),(f,u))$

$((f,a,a),(f,e))$ for $a\in\Sigma$

- simple PDA: ① $|F|=1$ ② $\forall ((p,a,\alpha),(q,\beta))\in\Delta$

  $\alpha.\beta$ 只有一个非空且长为 1

- PDA → simple PDA

① $|F|>1$: 创造新 final state f' $\forall q\in F$ $\exists((q,e,e),(f',e))$

② 2.1 同时 push, pop

$((p,a,\alpha),(q,\beta))$ $|\alpha|\geq 1$ $|\beta|\geq 1$

↳ $((p,a,\alpha),(r,e))$ $((r,e,e),(q,\beta))$

2.2 pop 多个 symbol

$((p,a,\alpha),(q,e))$ $|\alpha|\geq 1$ $\alpha=C_1\cdots C_k$ → $((p,a,C_1),(r_1,e))$ $((r_1,e,C_2),(r_2,e))$ ⋮ $((r_{k-1},e,C_k),(q,e))$

new states $r_1 \cdots r_{k-1}$

2.3 push 多个 symbol 同 2.2

2.4 不 push 不 pop: $((p,a,e),(q,e))$

↳ $((p,a,e),(r,a))$ $((r,e,a),(q,e))$

- 将 simple PDA 转为 CFG

$P=(K,\Sigma,\Gamma,\Delta,s,F) \Rightarrow G=(V,\Sigma,S,R)$

构建 A_{pq} $(p,q\in K)$，使 $w\in\Sigma^*$ 且 $w\in\{u\in\Sigma^*: (p,u,e)\vdash_P^* (q,e)\}$ A_{pq} 可推为 w

$S=A_{sf}$ 对于 R 的构建：

① $A_{pp}\to e$ ②' $A_{pq}\to A_{pr}\cdot A_{rq}$ 在 r 时 stack 为空

②² 若中途 stack 非空

p 开始时 push α，q 时 pop α (simple PDA 的性质)

故 $A_{pq}\to aA_{p'q'}b$ $((p,a,e),(p',\alpha))$ $((q',b,\alpha),(q,e))$

PDA 可定义 context-free language.

所有正则语言均是 context-free 的

Title L5

context-free language 性质

- closure properties $\begin{cases} \cup\ .\ \circ\ .\ *\ . \rightarrow \text{闭包} \\ \cap\ ,\ \bar{A} \rightarrow \text{不闭包} \end{cases}$

$G_A = (V_A, \Sigma, S_A, R_A)$ $G_B = (V_B, \Sigma, S_B, R_B)$

$G_{A\cup B}: S \rightarrow S_A \mid S_B$ $G_{AB}: S \rightarrow S_A S_B$ $G_{A^*}: S \rightarrow SS_A$

$\cap$ 的反例 $A=\{a^i b^j c^k;\ i=j\}$ $B=\{a^i b^j c^k;\ j=k\}$

A. B. 均 context-free $A\cap B = \{a^n b^n c^n\}$ 非 context-free

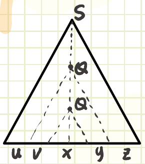
$\bar{A} \Rightarrow A\cap B = \overline{A\cup B}$ 反证可得.

- Pumping theorem for CFL

$\forall$ L is CFL $\exists p \in N^*$ st $\forall w\in L$ $|w|\geq p$ 可被分为

$w=uvxyz$ 满足 ① $uv^i x y^i z \in L$ $(i\geq 0)$

② $|v|+|y|>0$ ③ $|vxy| \leq p$



从S出发，对于w可形成一颗树 树的非叶节点均为non-terminal

只要足够长，必有Q出现2次

$S \rightarrow^* uQz$ $Q \rightarrow^* vQy$

$Q \rightarrow^* x$ $\therefore S \rightarrow^* uv^i x y^i z$

$\forall L\in CFL$ $\exists G\in CFG$ st $L(G)=L$ 使

$b=\max\{|u|;(A,u)\in R\}$ 树的 fanout $\leq b$ 有n个叶节点

则树高 $\geq \log_b n$ 使 $p=b^{|V-\Sigma|}+1$

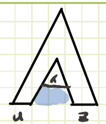
$w\in L$ $|w|\geq p$ w 的 parse tree 的高 $\geq |V-\Sigma|+1$

一定有重复的 non-terminal 状态.

- 证明 $|v|+|y|>0$ 若 $v=y=e$ → $w=uxz$

设 w 的语法树T本为 min 小于T，矛盾

- 证明 $|vxy| \leq p$，则为 证 Q 的子树



高度小于 $|V-\Sigma|+1$ 则 # leaf node $< b^{|V-\Sigma|+1} = p$

子树中不可能再有重复 non-terminal (取 min 后)

对于 $\{a^n b^n c^n\}$ 取 $a^p b^p c^p = uvxyz$

需要有 $uv^i x y^i z \in L$ $|v|+|y|>0$ $|vxy| \leq p$

中段中不可能同时有 a c $(|vxy|\leq p)$

则 $uv^0 x y^0 z$ b 与 a 或 c 的数量必不等.

故 $\{a^n b^n c^n\} \notin CFL$