Finite Automata

$(Q, \Sigma, \delta, q_0, F)$

$|y| > 0$

$|xy| \leq p$

$xy^iz \in A$ ทุก $i \geq 0$

$|vy| > 0$

$|vxy| \leq p$

$uv^ixy^iz \in A$ ทุก $i \geq 0$

CFG

(V, Σ, R, S)

$(\{S\}, \{0,1\}, R, S)$

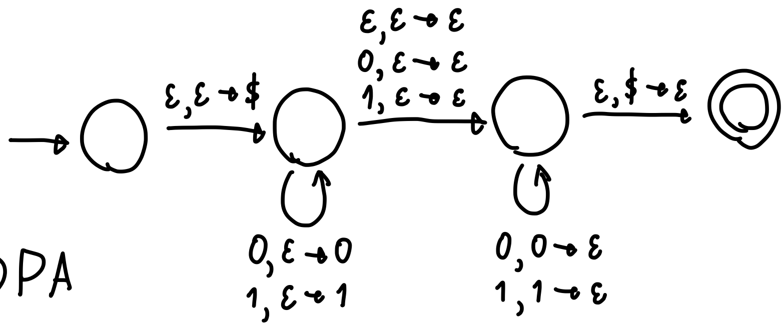
$R: S \to 0S1 \mid \varepsilon$

Pushdown Automaton

$(Q, \Sigma, \Gamma, \delta, q_0, F)$

$\delta: Q \times \Sigma_\varepsilon \times \underset{\text{read}}{\Gamma_\varepsilon} \mapsto P(Q \times \underset{\text{write}}{\Gamma_\varepsilon})$

- Palindrome :



- NPA > DPA

Turing Machine

$(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$

$\delta: Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R\}$

$\Sigma \subseteq \Gamma$, $\sqcup \in \Gamma$, $\sqcup \notin \Sigma$

- A is reg and A' is reg then both are decidable

Machine	Acceptance Problem (A)	Emptiness Testing (E)	Equivalence Testing (EQ)
DFA	✓	✓	✓
CFG	✓	✓	TR reg ✗
TM	TR reg ✗	co-TR reg ✗	non TR ✗

Operation	regular	CFL	decidable	Turing-recognizable
union	✓	✓	✓	✓
concatenation	✓	✓	✓	✓
star	✓	✓	✓	✓
intersection	✓	✗	✓	✓
complement	✓	✗	✓	✗

Q Let $BAL_{\mathsf{DFA}} = \{\langle M\rangle |\ M$ is a DFA that accepts some string containing an equal number of 0s and 1s$\}$. Show that BAL_{DFA} is decidable. (Hint: Theorems about CFLs are helpful here.)

Ans The language of all strings with an equal number of 0s and 1s is a context-free language, generated by the grammar $S \rightarrow 1S0S \mid 0S1S \mid \varepsilon$. Let P be the PDA that recognizes this language. Build a TM M for BAL_{DFA}, which operates as follows. On input $\langle B\rangle$, where B is a DFA, use B and P to construct a new PDA R that recognizes the intersection of the languages of B and P. Then test whether R's language is empty. If its language is empty, *reject*; otherwise, *accept*.

Q For languages A and B, let the ***perfect shuffle*** of A and B be the language

$$\{w|\ w = a_1b_1 \cdots a_kb_k,\ \text{where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B,\ \text{each } a_i, b_i \in \Sigma\}.$$

Refer to Problem 1.41 for the definition of the perfect shuffle operation. Show that the class of context-free languages is not closed under perfect shuffle.

Ans Let A be the language $\{0^k1^k |\ k \geq 0\}$ and let B be the language $\{a^kb^{3k} |\ k \geq 0\}$. The perfect shuffle of A and B is the language $C = \{(0a)^k(0b)^k(1b)^{2k} |\ k \geq 0\}$. Languages A and B are easily seen to be CFLs, but C is not a CFL, as follows. If C were a CFL, let p be the pumping length given by the pumping lemma, and let s be the string $(0a)^p(0b)^p(1b)^{2p}$. Because s is longer than p and $s \in C$, we can divide $s = uvxyz$ satisfying the pumping lemma's three conditions. Strings in C contain twice as many 1s as a's. In order for uv^2xy^2z to have that property, the string vxy must contain both 1s and a's. But that is impossible, because they are separated by $2p$ symbols yet the third condition says that $|vxy| \leq p$. Hence C is not context free.

- **ข้อ 1.11:** จงพิสูจน์ว่า ทุก NFA จะมี NFA ที่สมมูลกัน ซึ่งมี accept state แค่ state เดียว
- **ข้อ 1.29a:** จงพิสูจน์ว่า $A = \{0^n1^n2^n\}$ ไม่เป็น regular language
- **ข้อ 1.29c:** จงพิสูจน์ว่า $A = \{a^{2^n} | n \geq 0\}$ (มี a เรียงติดกันอยู่ 2^n ตัว) ไม่เป็น regular language
- **ข้อ 1.31:** ให้ w^R คือการเอา w มาเขียนเรียงย้อนกลับจากหลังมาหน้า จงพิสูจน์ว่า ถ้า A เป็น regular language แล้ว $A^R = \{w^R | w \in A\}$ ก็เป็น regular language ด้วย

1.11) let $N = (Q, \Sigma, \delta, q_0, F)$

then $N' = (Q \cup \{q_F\}, \Sigma, \delta', \{q_F\})$

where $\delta' = \begin{cases} \delta(q,a) & ; \ a \neq \varepsilon \text{ or } q \notin F \\ \delta(q,a) \cup \{q_F\} & ; \ a = \varepsilon \text{ and } q \in F \end{cases}$

1.29a.) let s be $0^p1^p2^p$ then we can split $s = xyz$ and $xy^iz \in A$ $\forall_{i \geq 0}$

case 1 : if y just only 0 or 1 or 2 then $xyyz$ don't have same number of 0s 1s and 2s

case 2 : if y consist more than one kind of 0,1,2 $xyyz$ is out of order

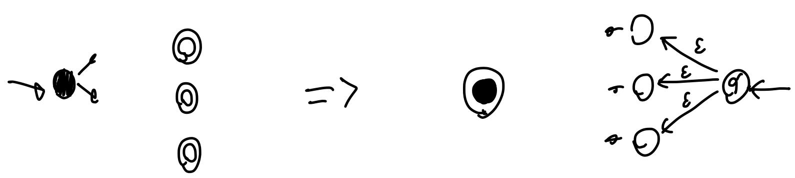
$\therefore$ $0^n1^n2^n$ isn't a regular language

1.29c.) let s be a^{2^p} by pumping lemma, we can split $s = xyz$

$|xy| \leq p$ then $|y| \leq$ and $|xyz| = 2^p$; $|xy^2z| \leq 2^p + p$

but $2^p < 2^p + p < 2^{p+1}$ $\forall_{p \geq 1}$ then $xy^2z \notin A$

1.31) let D be DFA that accept A then we can construct NFA N that accept A^R by let accept state of D be start state of N, start state of D also become accept state of N, and reverse all transition function. With this definition, N accepts A^R.



- ข้อ 2.4d: จงหา CFG ที่สร้างภาษา $\{w \in \{0,1\}^* |$ w มีความยาวเป็นจำนวนคี่ และมีอักษรตรงกลางเป็น $0\}$
- ข้อ 2.6a: จงหา CFG ที่สร้างภาษา $\{w \in \{a,b\}^* |$ w มีตัว a อยู่มากกว่าตัว $b\}$
- ข้อ 2.30b: จงพิสูจน์ว่าภาษา $\{0^n\#0^{2n}\#0^{3n} | n \geq 0\}$ ไม่เป็น CFL
- ข้อ 2.31: จงพิสูจน์ว่าภาษา $\{w \in \{0,1\}^* | w$ เป็นพาลินโดรม และ w มีจำนวน 0 กับ 1 เท่ากัน$\}$ ไม่เป็น CFL (พาลินโดรม คืออ่านจากซ้ายไปขวาแล้วเหมือนกับขวาไปซ้าย)

2.4 d.) $S \to 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0$

2.6 a.) $S \to TaT$
$T \to TT \mid aTb \mid bTa \mid a \mid \varepsilon$

2.6 c.)

$S \to S0 \mid S1 \mid X$	$S \to RT$
$X \to X0 \mid 1X1 \mid \#Y$	$R \to 0R0 \mid 1R1 \mid \#T$
$Y \to 0Y \mid 1Y \mid \varepsilon$	$T \to 0T \mid 1T \mid \varepsilon$

2.30 b.) let pumping length be p and $s = 0^p\#0^{2p}\#0^{3p} \in L$ by pumping lemma

we can split to be $s = uvxyz$, for $|vxy| \leq p$

case 1: v or y contain $\#$ then uv^2xy^2z will have exceeding $\#$ imply $uv^2xy^2z \notin L$

case 2: v and y consist of 0s. In this case we have 3 blocks of 0s which are $0^p, 0^{2p}, 0^{3p}$.

Observed that vxy lay on only one block or two contiguous blocks of 0s

to hold $|vy| > 0$ and $|vxy| \leq p$, they cannot lay on 3 blocks of 0s since

x must contain at least $\#0^{2p}\#$ which violate $|vxy| \leq p$.

$\therefore$ This language isn't CFL $\#$

2.31.) Let p be pumping length and $s = 0^p1^{2p}0^p \in L$ by pumping lemma

we can split $s = uvxyz$, consider choices to choose vxy to let $\forall_{i \geq 1}\ uv^ixy^iz$ still palindrome.

case 1: v and y consist of only 1s, by $|vy| > 0$; there're one or more 1s in vy so uxz will

have 1s at most $2p-1$ but number of 0s is $2p$ cause $s \notin L$

case 2: V and y contain 0, for clarify; $v = 0^n1^n$, $y = 1^n0^n$ for some $n \geq 1$ in $\mathbb{R}$ to preserve palindrome.

In this case, vxy must cover entire 1^{2p} in s which cause $|vxy| \geq 2p+2$ imply this case can't pump.

$\therefore$ This language isn't CFL. $\#$

- ข้อ 3.8a: จงอธิบายขั้นตอนการทำงานของ TM ที่ decides ภาษา $\{w \in \{0,1\}^* | w$ มี 0 และ 1 อยู่จำนวนเท่ากัน$\}$
- ข้อ 3.10: พิจารณา TM ที่สามารถเขียนเทปได้ช่องละไม่เกิน 1 ครั้ง จงพิสูจน์ว่าเครื่องนี้สมมูลกับ TM แบบปกติ
- ข้อ 3.18: จงพิสูจน์ว่า ภาษา A เป็น decidable language ก็ต่อเมื่อมี enumerator ที่สามารถ enumerate A เรียงตาม lexicographic order ได้

3.8a) Iteratively; 1.) find 0 from left to right, if found, overwrite as x and goto step 2.).
Stop iteration if can't find any 0.
2.) find 1 from left to right, if found, overwrite as x and goto step 1.).
reject if can't found any 1.
After iterations, check if tape now all of "x", if yes then accept, otherwise, reject.

3.10.) Instead of overwrite, we store every version of string on tape and seperate with new unused ex. $\not\forall$

let say we have input $T_0 = w_1 w_2 \dots w_n$ we will have $T_1 \not\forall$ at first and want to write w_2 to w_2' we will have

$T_2 = w_1 w_2' \dots w_n$ and write on tape as $T_1 \not\forall T_2 \not\forall$. This procedure need write 2 times one to mark copying

position, another one for write a new version. We can change this by left one space before each character

to mark if it's copied.

3.18) Firstly, Proof that if their exist lexicological enumerator E_l that enumerate A then, we can construct

touring machine M that decide A by run E_l and compare each output of E_l to input w

since output of E_l are lexicological order, we wait until time that E_l should print out w

if E_l skip w reject then else accept. By this procedure, M will decide in finite time.

In converse, let there exist touring machine M that decide A, then we can construct

enumerator E_l by generate string Σ^* in lexicological order and for each string,

let M decide if M accept then print out that string, ortherwise skip and check another string.

This procedure will cover all finite string in A since A is decidable which exist no infinite loop.

- ข้อ 4.09: ให้ $INF_{\text{DFA}} = \{\langle A\rangle | A$ เป็น DFA ที่ $L(A)$ เป็นเซตอนันต์$\}$ จงพิสูจน์ว่า INF_{DFA} เป็น decidable language
  - Thm: ใน Regular Pumping Lemma ใช้ pumping length เป็นจำนวน state ของ DFA ที่ recognize ภาษานั้นได้
- ข้อ 4.11: ให้ $A = \{\langle M\rangle | M$ เป็น DFA ที่ไม่ accept สตริงที่มี 1 อยู่คี่ตัวเลย$\}$ จงพิสูจน์ว่า A เป็น decidable language
- ข้อ 4.13: ให้ $\Sigma = \{0,1\}$ และ $A = \{\langle G\rangle | G$ เป็น CFG ที่สร้างบางสตริงในรูป 1^* ได้$\}$ จงพิสูจน์ว่า A เป็น decidable language

4.9) construct TM I to decide INF_{DFA} by on input $\langle A \rangle$:

1.) Let p be number of state of A.

2.) Let M be DFA that accept all string that length are p or more.

3.) Let T be DFA such that $L(T) = L(A) \cap L(M)$.

4.) Test for $L(T) = \emptyset$ using E_{DFA} decider D. 5.) If D accepts, reject ; if D rejects, accept.

4.11) construct TM decide A by, on input $\langle M \rangle$:

1.) construct DFA N that accepts all string containing odd number of 1s.

2.) construct DFA O such that $L(O) = L(M) \cap L(N)$ 3.) test whether $L(O) = \emptyset$ using E_{DFA} decider T

4.) if T accepts, accept ; if T rejects, reject.

4.13) Firstly, proof that CFG ∩ Reg is CFG, the idea is as same as proof of DFA ∩ DFA = DFA.

Let PDA P and DFA D, construct P′ which $L(P') = L(P) \cap L(D)$, set of state become $Q_P \times Q_D$

P′ have action of PDA like P but each state take care of D's action like a sub-state.

Accept state is $q' \in F_P \times F_D$

Now, we can construct TM decide A by, on input $\langle G \rangle$:

1.) construct CFG C such that $L(C) = L(G) \cap 1^*$

2.) test for $L(C) = \emptyset$ using E_{CFG} decider T

3.) if T accepts, reject ; if T rejects, accept.