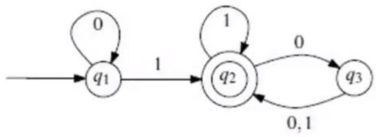
1. DFA -> Language

Lecture 2 P15-example

We have a machine state diagram as below:



q_1 is the start state and q_2 is an accept state.

Question.

Can the input string 0101010 be accepted by this machine?

Which kind of strings can be accepted by this machine?

① accept every binary string ends with 1
② accept every binary string that there are even numbers of 0, following the rightmost 1

①②

2. Language -> DFA

Assignment 1 Q1-b

Question 1

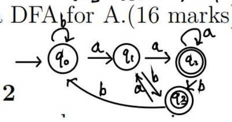
Let $\Sigma = \{a, b\}$. Define $A = \{w \in \Sigma^* \mid |w| \geq 2$, second-to-last symbol of w is $a\}$ (20 marks)

(a) List the first 4 strings in A in lexicographic order. (4 marks)

Answer: They are $w_1 = aa$, $w_2 = aaa$, $w_3 = aaaa$, $w_4 = aaaaa$.

(b) Draw a DFA for A.(16 marks)

Answer:



Question 2

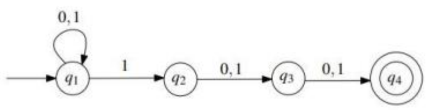
3. 判断 DFA 能否接受字符串 w\table 与 diagram 转换

4. Language -> NFA

Lecture 3 P16-example

Example

Let A be the language $A = \{w \in \{0, 1\}^* :$ w has a 1 in the third position from the right$\}$, design $M : L(M)$.



5. NFA -> Language\NFA 能否接受字符串 w\ possible path 与 accepting path

6. DFA -> NFA

答案不唯一，加一个 q0，由空输入指向原 q0 即可。

7. NFA -> DFA

Assignment 1 Q3

Question 3

Convert the given NFA to its corresponding DFA. (20 marks)

DFA M = ({q0, q1}, {0,1}, δ, q0, {q0})

Answer: ① The δ' of DFA is shown in the table:

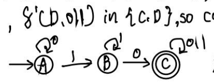
		0	1
A	q_1	q_1	q_2
B	q_2	$\{q_2, q_3\}$	q_2
C	q_3	q_3	$\{q_2, q_3\}$
D	$\{q_2, q_3\}$	$\{q_2, q_3\}$	$\{q_2, q_3\}$

② Suppose the Q' of DFA is $\{A = q_1, B = q_2, C = q_3, D = \{q_2, q_3\}\}$

XJTLU/21-22/S1 ③ Since the accepting state of NFA is q_3, Page 1 of 2
the accepting state of DFA $F' = \{C, D\}$.
Then we can divide Q' into {A, B} and {C, D} according to whether $q \in Q'$ is in F'.

④ Since $\delta'(A, 0|1)$ in {A, B}, $\delta'(B, 0)$ not in {A, B}, so $A \in Q'$, $B \in Q'$.
$\delta'(C, 0|1)$ in {C, D}, $\delta'(D, 0|1)$ in {C, D}, so combine {C, D} as C.

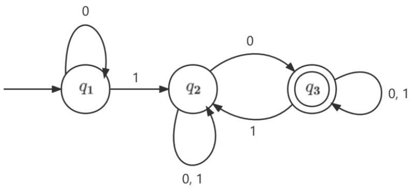
⑤ So $F' = \{C\}$. We get DFA:



$M = (\{A, B, C\}, \{0,1\}, \delta', A, \{C\})$

Xi'an Jiaotong-Liverpool University 西交利物浦大学

NFA



8. Proof of closed of RL[The NFA after the operation\RE -> NFA]

(1) Union(DFA perspective)

[1] Union

Regular Languages Closed Under Union DFA perspective ② Regula

① Proof

Given M_1 and M_2 that $A = L(M_1)$ and $B = L(M_2)$, we can define $M = (Q, \Sigma, \delta, q, F)$:

- $Q = Q_1 \times Q_2 = \{(q_1, q_2): q_1 \in Q_1 \text{ and } q_2 \in Q_2\}$
- Σ is same as the alphabet of A and B
- $q = (q_1, q_2)$
- $F = \{(q_1, q_2): q_1 \in Q_1 \text{ or } q_2 \in Q_2\}$
- $\delta : Q \times \Sigma \to Q$

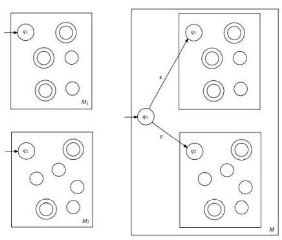
  $\delta((q_1, q_2), a) = (\delta(q_1, a), \delta(q_2, a)), a \in \Sigma$

① Since $\delta^*((q_1, q_2), w) = (\delta^*(q_1, w), \delta^*(q_2, w))$
② If $\delta^*((q_1, q_2), w) \in F$, then $\delta^*(q_1, w) \in F$ or $\delta^*(q_2, w) \in F$.
③ And M accepts w $\Leftrightarrow$ $\delta^*(q_1, w) \in F$ or $\delta^*(q_2, w) \in F$
④ So M accepts w $\Leftrightarrow$ M_1 accepts w or M_2 accepts w
which means $A \cup B$ is still a RL.

Proof

- $Q =$
- q_0 is
- $F =$
- $\delta : Q$

Then $L(R_1) = L(R_1) \cup L(R_2)$ has NFA as:



(2) Concatenation

[2] Concatenation

Regular Languages Closed Under Concatenation

The concatenation of A_1 and A_2 is defined as:

$$A_1 A_2 = \{ww' : w \in A_1 \text{ and } w' \in A_2\}$$

Proof

Consider the following NFAs:

NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes $A_1 = L(M_1)$

NFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognizes $A_2 = L(M_2)$

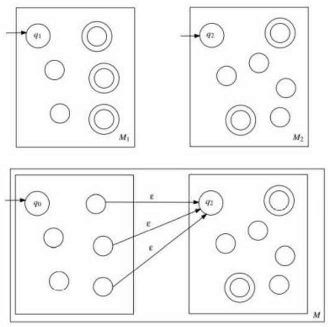
We will construct an NFA $M = (Q, \Sigma, \delta, q, F)$ for $A_1 A_2$

Proof

- $Q = Q_1 \cup Q_2$
- M has the same start state as M_1 : q_1
- Set of accept states of M is same as M_2 : F_2
- $\delta : Q \times \Sigma_\epsilon \to P(Q)$ is defined as: For any $r \in Q$ and for any $a \in \Sigma_\epsilon$

$$\delta(r,a) = \begin{cases} \delta_1(r,a) & \text{if } r \in Q_1 \text{ and } r \notin F_1 \\ \delta_1(r,a) & \text{if } r \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(r,a) \cup \{q_2\} & \text{if } r \in F_1 \text{ and } a = \varepsilon \\ \delta_2(r,a) & \text{if } r \in Q_2 \end{cases}$$

Then $L(R_1) = L(R_1)\, L(R_2)$ has NFA as:



(3) Star

[3] Star

Regular Languages Closed Under Kleene star

The star of A is defined as:

$$A^* = \{u_1 u_2 \ldots u_k : k \geq 0 \text{ and } u_i \in A \text{ for all } i = 1, 2, \ldots, k\}$$

Proof

Consider the following NFA:

NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes $A = L(M_1)$

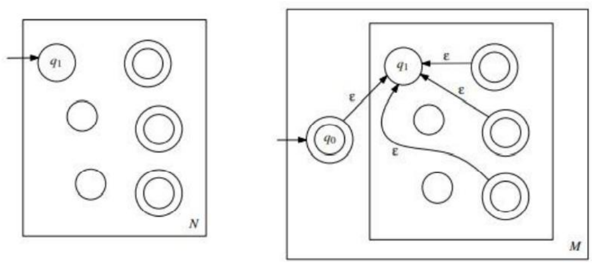
We will construct an NFA $M = (Q, \Sigma, \delta, q, F)$ for A^*

Proof

- $Q = \{q_0\} \cup Q_1$
- q_0 is the start state of M
- $F = \{q_0\} \cup F_1$
- $\delta : Q \times \Sigma_\epsilon \to P(Q)$ is defined as: For any $r \in Q$ and for any $a \in \Sigma_\epsilon$

$$\delta(r,a) = \begin{cases} \delta_1(r,a) & \text{if } r \in Q_1 \text{ and } r \notin F_1 \\ \delta_1(r,a) & \text{if } r \in F \text{ and } a \neq \varepsilon \\ \delta_1(r,a) \cup \{q_1\} & \text{if } r \in F_1 \text{ and } a = \varepsilon \\ \{q_1\} & \text{if } r = q_0 \text{ and } a = \varepsilon \end{cases}$$

6th case (Kleene star). If $R = (R_1)^*$ and $L(R_1)$ has NFA N, then $L(R) = (\ L(R_1)\ )^*$ has NFA M as:



9. RL -> Regular expression

Lecture 4 P19-example

The language $\{w : w \text{ contains exactly two } 0\text{s}\}$ is described by the expression:

$$1^*01^*01^*$$

The language $\{w : w \text{ contains at least two } 0\text{s}\}$ is described by the expression:

$$(0\cup1)^*0(0\cup1)^*0(0\cup1)^*$$

The language $\{w : 1011 \text{ is a substring of } w\}$ is described by the expression:

$$(0\cup1)^*1011(0\cup1)^*$$

10. RE -> RL

Lecture 4 P23-example

11. 证明一个 expression 是 regular 的

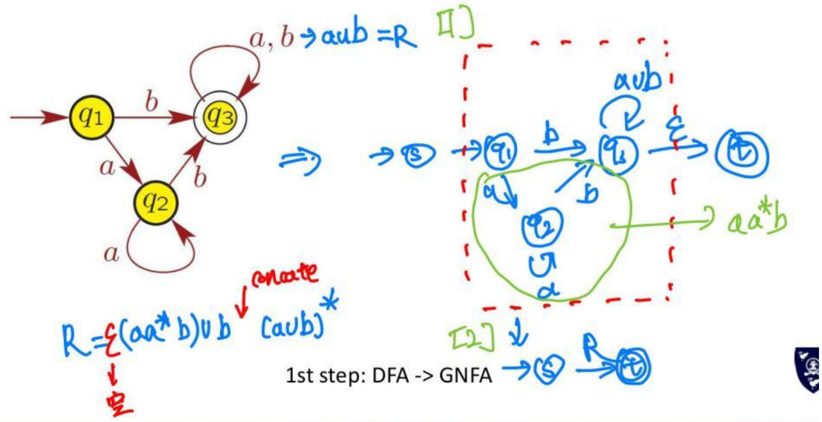
Lecture 4 P21-example

12. DFA -> RE [Iterative procedure: DFA -> GNFA -> RE]

Lecture 5 P12/14

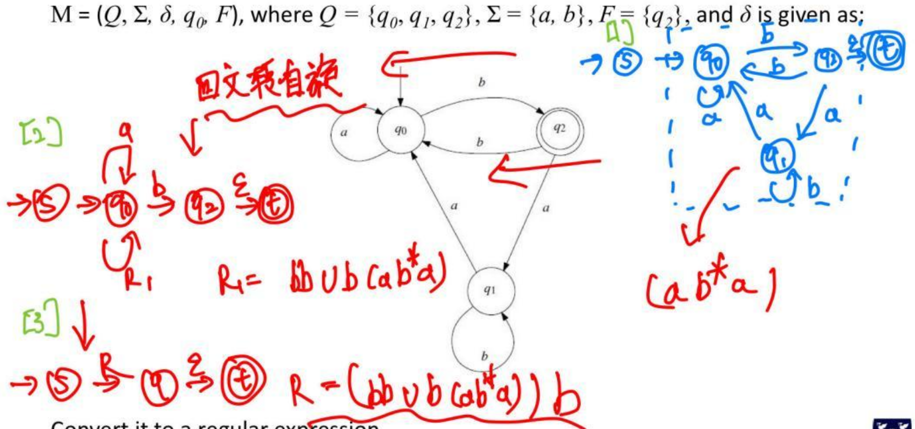
Example

Convert the given DFA into regular expression



Exercise

M = (Q, Σ, δ, q_0, F), where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$, and δ is given as;



Convert it to a regular expression.

13. Use pumping lemma prove RL

Assignment 1 Q5-b

Question 5

$S_2 \to bS_2 \mid \varepsilon \}$ $\overset{*}{\Rightarrow} a \cdots ab \cdots b c \cdots c e^2 = a^i b^j c^i e^2 \; (i, j \geq 0)$

Let $\Sigma = \{a, b\}$, and consider the language $A = \{w \in \Sigma^* \mid w$ has more b's than a's$\}$. (20 marks)

(a) Describe pumping lemma for regular languages. (4 marks)

Answer: Suppose A is a regular language. Then there exist an integer $p \geq 1$, called the pumping length, such that the following holds: Every string s in A, with $|s| \geq p$, can be written as $s = xyz$, such that $y \neq \varepsilon$ (i.e., $|y| \geq 1$), $|xy| \leq p$, and for all $i \geq 0$, $xy^iz \in A$.

(b) Prove that A is not a regular language. (16 marks)

Answer: Proof by contradiction: Assume A is a regular language, then A holds the pumping lemma, which means there exist a pumping length p, and for a string $s = a^p b^{p+1}$ $(p \geq 0)$, $s = xyz$

Since $|s| = 2p+1 \geq p$, pumping lemma holds. By pumping lemma, $|xy| \leq p$, $|y| \geq 1$, so x, y only contain a.

XJTLU/21-22/S1 Page 2 of 2

Suppose $x = a^m$, $y = a^n$, $z = a^k b^{p+1}$, for some $m \geq 0$, $n \geq 1$, $k \geq 0$ and $m+n+k = p$, $(m+n) \leq p$

If $i = 2$ which means $n = 2n$. Then the number of a in s is $m + 2n + k = p + n$

Since $n \geq 1$, $p + n \geq p + 1$, which means we get more or equal a's than b's, $xy^2z \notin A$, which is contradict to the pumping lemma. Therefore, A is not a regular language.

14. Construct CFG\[Derivation of CFG]

Assignment 1 Q4\[Assignment 2 Q1]

Question 4

Design context free grammars for the following languages (only providing rules). (20 marks)

(a) The set $\{a^n \mid n \geq 1\}$ (6 marks)

Answer: $G_1 = \{V, \Sigma, R, S\}$ $V = \{S\}$ $\Sigma = \{a\}$ $R = \{S \to aS \mid a\}$

Proof: $S \Rightarrow aS \Rightarrow aaS \overset{*}{\Rightarrow} aa \cdots aS = a^n \; (n \geq 1)$

(b) The set $\{0^n 1^n \mid n \geq 1\}$ (6 marks)

Answer: $G_2 = \{V, \Sigma, R, S\}$ $V = \{S\}$ $\Sigma = \{0, 1\}$ $R = \{S \to 0S1 \mid 01\}$

Proof: $S \Rightarrow 0S1 \Rightarrow 00S11 \overset{*}{\Rightarrow} 00 \cdots 0S11 \cdots 1 = 00 \cdots 011 \cdots 1 = 0^n 1^n \; (n \geq 1)$

(c) The set $\{a^i b^j c^i e^2 \mid i, j \geq 0\}$ (8 marks)

Answer: $G_3 = \{V, \Sigma, R, S\}$ $V = \{S, S_1, S_2\}$ $\Sigma = \{a, b, c, e\}$ $R = \{S \to S_1 e^2$, $S_1 \to aS_1c \mid S_2$, $S_2 \to bS_2 \mid \varepsilon\}$

Proof: $S \Rightarrow S_1 e^2 \overset{*}{\Rightarrow} a \cdots aS_1c \cdots ce^2 \overset{*}{\Rightarrow} a \cdots ab \cdots bc \cdots ce^2 = a^i b^j c^i e^2 \; (i, j \geq 0)$

Question 5

Let $\Sigma = \{a, b\}$, and consider the language $A = \{w \in \Sigma^* \mid w$ has more b's than a's$\}$. (20 marks)

15. CFG -> CNF

Lecture 6 P19-example\Tutorial 4 Q4\Assignment 2 Q3

Step 1. Eliminate the start variable from the right-hand side of the rules.

- New start variable S_0
- New rule $S_0 \to S$

Step 2. Remove **ε-rules** $A \to \varepsilon$, where $A \in V - \{S\}$.

- Before: $B \to xAy$ and $A \to \varepsilon \mid \cdots$
- After: $B \to xAy \mid xy$ and $A \to \cdots$

When removing $A \to \varepsilon$ rules, insert all new replacements:

- Before: $B \to AbA$ and $A \to \varepsilon \mid \cdots$
- After: $B \to AbA \mid bA \mid Ab \mid b$ and $A \to \cdots$

Step 3. Remove **unit rules** $A \rightarrow B$, where $A \in V$.

- Before: $A \rightarrow B$ and $B \rightarrow xCy$
- After: $A \rightarrow xCy$ and $B \rightarrow xCy$

Step 4. Eliminate all rules having more than two symbols on the right-hand side.

- Before: $A \rightarrow B_1B_2B_3$
- After: $A \rightarrow B_1A_1$, $A_1 \rightarrow B_2B_3$

Step 5. Eliminate all rules of the form $A \rightarrow ab$, where a and b are not both variables.

- Before: $A \rightarrow ab$
- After: $A \rightarrow B_1B_2$, $B_1 \rightarrow a$, $B_2 \rightarrow b$.

16. DFA -> CFG

Lecture 6 P13-example

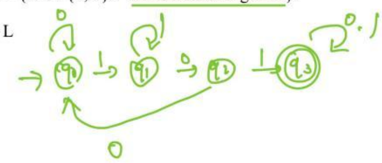
Example

L → DFA → CFG

Let L be the language defined as

$$L = \{w \in \{0, 1\}^* : \text{101 is a substring of } w\}.$$

The DFA M that accepts L



$G = (V, \Sigma, R, S)$

$V = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$

R: $q_0 \rightarrow 0q_0 \mid 1q_1$, $q_2 \rightarrow 0q_0 \mid 1q_3$

How can we convert M to a context-free grammar G whose language is L?

$q_1 \rightarrow 1q_1 \mid 0q_2$, $q_3 \rightarrow 0q_3 \mid 1q_3 \mid \varepsilon$

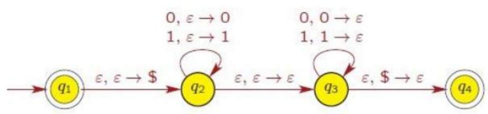
17. CFL -> PDA

Tutorial 5 Q3/Q4

3. PDA for language $\{ww^R \mid w \in \{0, 1\}^*\}$

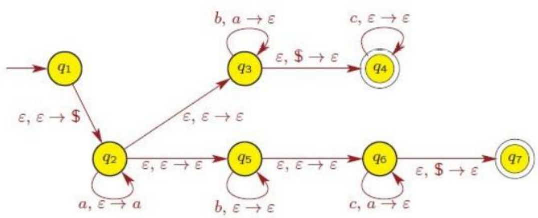
4. PDA for language $\{ a^ib^jc^k \mid i, j, k \geq 0 \text{ and } i = j \text{ or } i = k \}$

3.



- $q_1 \rightarrow q_2$: First pushes \$ on stack to mark bottom
- $q_2 \rightarrow q_2$: Reads in first half w of string, pushing it onto stack
- $q_2 \rightarrow q_3$: Guesses that it has reached middle of string
- $q_3 \rightarrow q_3$: Reads second half w^R of string, matching symbols from first half in reverse order (recall: stack LIFO)
- $q_3 \rightarrow q_4$: Makes sure that no more input symbols on stack

4.



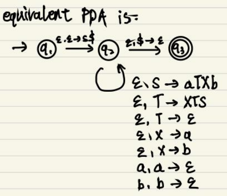
18. CFG -> PDA

Assignment 2 Q2

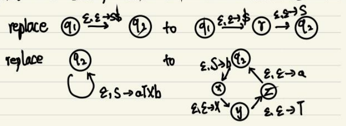
Consider the following CFG $G = (V, \Sigma, R, S)$, where $V = \{S, T, X\}$, $\Sigma = \{a, b\}$, the start variable is S, and the rules R are:
$S \to aTXb$
$T \to XTS|\varepsilon$
$X \to a|b$

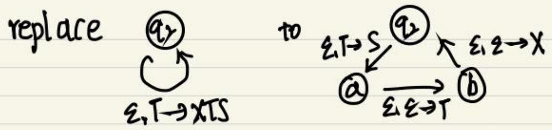
Q2 To convert G to an quivalent PDA, the idea is, to build a PDA to simulate a leftmost derivation of G, which is 'aab'.

① The equivalent PDA is:

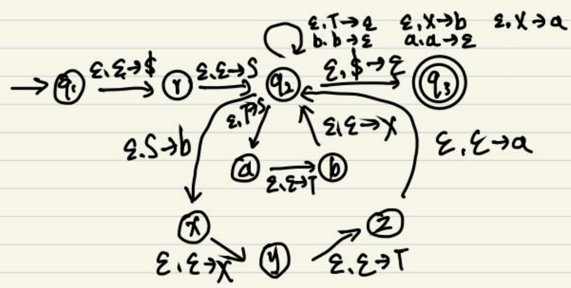


② Considering PDA cannot push strings instead of ≤ 1 symbols onto stack, some extra states should be added:





③ So the equivalent PDA is:



19. CFL must be RL

20. Pumping lemma for CFL

Assignment 2 Q3

Q3 If language $A = \{0^n 1^{3n} 0^n \mid n \geq 0\}$ is not context free, then A does not hold pumping lemma.

Method 1:

Proof by contradiction:

① Suppose A is context free, then there exist an integer $p \geq 1$, called the pumping length, such that the following holds: Every string s in A, with $|s| \geq p$, can be written as $s = uvxyz$, such that
1. $|vy| \geq 1$
2. $|vxy| \leq p$ and
3. $uv^ixy^iz \in L$, for all $i \geq 0$

Suppose we could construct some CFG G for A

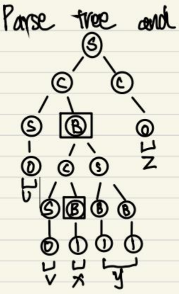
$$\begin{cases} S \to CC \mid BB \mid 0 \\ C \to SB \mid 0 \\ B \to CS \mid 1 \end{cases}$$

(This is the CNF of A. If the rules under CNF do not comply pumping lemma, then all forms of other CFG will not comply pumping lemma.)

② Set string $s = 001110 \in A$

$S \Rightarrow CC \Rightarrow SBC \Rightarrow SCSC \Rightarrow SSBSC \Rightarrow SSBBBC \Rightarrow 0SBBBC \Rightarrow 00BBBC \Rightarrow 001BBC \Rightarrow 0011BC \Rightarrow 00111C \Rightarrow 001110$

③ Parse tree and splitting:



Taking B as repeated variable.

get $s = uv^ixy^iz = 0\,0^i\,1\,(11)^i\,0$

④ S satisfy property 1 and 2 of pumping lemma, but does not satisfy property 3: $00^i1(11)^i0 \notin A$ when $i > 1$, which is a contradiction.

So A is not context free.

21. TM configuration/Detailed process

Tutorial 7 Q2/Q1

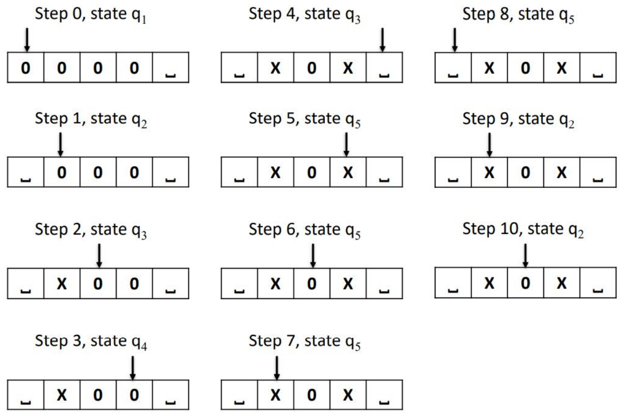
(1) Configuration:

2.

$$q_1 0000, \quad \sqcup q_2 000, \quad \sqcup x q_3 00, \quad \sqcup x 0 q_4 0, \quad \sqcup x 0 x q_3 \sqcup, \quad \sqcup x 0 q_5 x,$$
$$\sqcup x q_5 0 x, \quad \sqcup q_5 x 0 x, \quad q_5 \sqcup x 0 x, \quad \sqcup q_2 x 0 x, \quad \sqcup x q_2 0 x, \quad \sqcup x x q_3 x,$$
$$\sqcup x x x q_3 \sqcup, \quad \sqcup x x q_5 x, \quad \sqcup x q_5 x x, \quad \sqcup q_5 x x x, \quad q_5 \sqcup x x x, \quad \sqcup q_2 x x x,$$
$$\sqcup x q_2 x x, \quad \sqcup x x q_2 x, \quad \sqcup x x x q_2 \sqcup, \quad \sqcup x x x \sqcup q_{\text{accept}} \sqcup .$$

(2) Detailed process

1.



22. Run TM on input w/Language of TM/Address/Encoding

23. Acceptance problem

(1) L_{DFA}, L_{NFA}, L_{REX}, CFGs, CFLs are decidable

(2) L_{TM} is not decidable

24. Countable / Uncontable set: just need to know the basic idea

25. Reducibility: need to know how to use it to prove TM-R and TM-D

Theorem 推导

If $A \leq_m B$ and B is decidable, then A is decidable.

Corollary 同样地

If $A \leq_m B$ and A is undecidable, then B is undecidable also.

Theorem Proof

1. Let M_B is the TM that decides B. (B is decidable)
2. Let f be the reducing function from A to B because $A \leq_m B$
3. Consider M_A = " On input w: ① M_A should compute f(w)
   ② run M_B as a subroutine of A on input f(w), and give the same result

4. Since f is a reducing function, then $w \in A \Leftrightarrow f(w) \in B$.
   ① 若 $w \in A$, then $f(w) \in B$, M_B and M_A accept.
   ② 若 $w \notin A$, then $f(w) \notin B$, M_B and M_A reject.

Thus M_A decides A, A is decidable.

(1)

Theorem

If A $\leq_m$ B and B is Turing-recognizable, then A is Turing-recognizable.

Corollary

If A $\leq_m$ B and B is not Turing-recognizable, then ~~B~~ A is not Turing-recognizable.

Theorem Proof

1. Let M_B be the TM that recognize B
2. Let f be the reducing function from A to B
3. Define a new TM as follows: M_A = " On input w:
   ① Compute f(w)
   ② Run M_B on input f(w) and give the same result "

4. Since f is reducing function, then $w \in A \Leftrightarrow f(w) \in B$
   ① If $w \in A$, then $f(w) \in B$, so M_B and M_A accept
   ② If $w \notin A$, then $f(w) \notin B$ so M_B and M_A reject or loop

Thus M_A recognize A

(2)

26. Minimal length description: 3 Theorems

27. Enumerator

Enumerator

Theorem

A language is enumerable if and only if it has an enumerator.

Proof 1. Assume that L has an enumerator. E, we construct a TM M with an arbitary string w.

TM M(w): ① run E, everytime E enters the print state, let v be the string on the print tape.

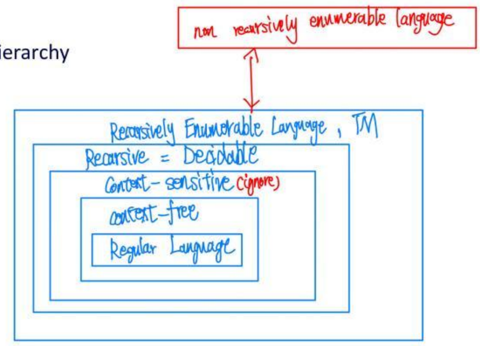
② if w = v, then M accept

2. Then M satisfies properties: if $w \in L$, then w will be sent to the printer, M accepts; if $w \notin L$, then w is never sent to the printer, M will not terminate.

3. Thus, M satisfies the defination, L is enumerable.

28. Hierarchy

Hierarchy



29. Big-O and Little-O

(1) Definition and difference

(2) Is Big-O or Little-O properly?[Count time complexity]

30. 图灵机的概念们：

图灵机接受的语言 L(M) 是指被图灵机 M 接受的所有属于字母表的字符串的集合

1. 图灵可识别(Turing-recognizable): 对于语言 A，如果存在 TM M 使得 A = L(M),即语言 A 与图灵机接受的语言相同。也称为**递归可枚举(Recursively enumerable)**或者**(enumerable language)**

2. Enumerator(枚举器): 通常表示为一个双带图灵机，一个工作带一个打印带。

3. Decider(判定器): 一个对于所有输入都会停机(halt)不会 loop 的图灵机，就是 decider。

(1) 对于语言 L，如果存在一个图灵机 M，对于属于 L 的 w，输入 M 后 M 在接受态停机；对于不属于 L 的 w，输入 M 后 M 在拒绝态停机，那么称 **L 为可判定(decidable)语言，L 对应的问题称为可判定问题。**也叫 **Turing-decidable**，或称**递归语言(recursive language)**。

(2) 图灵可判定与图灵可识别的区别: 对于不属于该语言的字符串 w，图灵可判定语言存在图灵机 M，输入 w 时终止在拒绝态，而图灵可识别则不确定，有可能 loop 也有可能 halt。

(3) 递归可枚举语言里面包括不可判定问题，只有递归语言中一定是可判定问题。

(4) 能保证停机的图灵机=算法

4. 没有图灵机能够接受的语言叫非递归可枚举语言（不递归也不枚举），例如对角化语言与通用语言。所对应的问题叫 **unsolvable problem 不可解问题**。递归可枚举语言对应的问题叫做不可判定问题(undecidable problem)，**是不可解问题的子集。**

5. 多带图灵机与普通图灵机等价：加入井号来代表不同纸带的边界（其实是**用多道图灵机模拟多带图灵机**）。

6. 非确定图灵机(NTM)也与普通(确定)图灵机等价，可以用多带图灵机模拟 NTM

7. 一个数组 A 如果与 N(自然数数组)存在双射(bijection)或其大小有限，则称 A 是 countable set。

(1) Z(整数数组)与 Q(有理数数组)都是 countable 的

(2) R(实数数组)是 uncontable 的