Morning

Turing Machine

- Enumerating
- Coding of TM
- Recursive language
- Chomsky Grammar

Enumerating Strings

All $w \in \{0,1\}^*$, in order of the length :

ε, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, …

1, 10,11,100,101,110,111,1000,1001,1010,1011, …

To take 1w as a binary integer, where 1w = i, w is called the *i*th string.

Coding of Turing machine

Let TM $M = (Q, \{0,1\}, \Gamma, \delta, q_1, B, \{q_2\})$

Where $Q = \{q_1, q_2, \ldots, q_r\}$, $\Gamma = \{X_1, X_2, X_3, \ldots, X_s\}$

$$X_1 : 0,\ \ X_2 : 1,\ \ X_3 : B,\ \ D_1 : \leftarrow,\ \ D_2 : \rightarrow$$

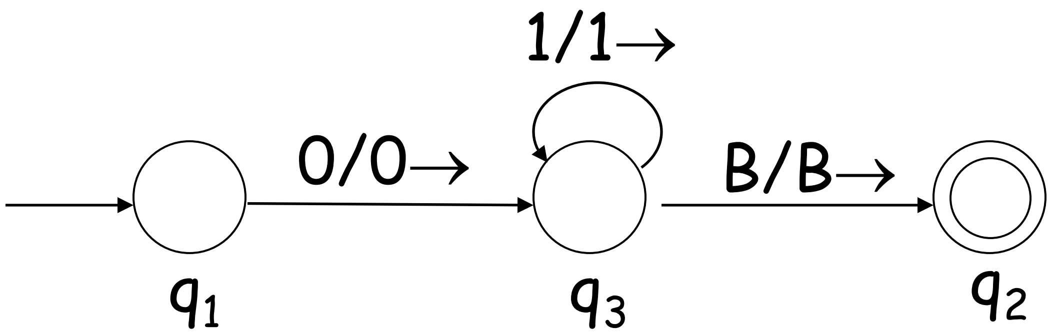
Coding :

$$\delta(q_i, Xj) = (q_k, X_m, D_n)$$

$$\Rightarrow 0^i 1 0^j 1 0^k 1 0^m 1 0^n$$

$$M \Rightarrow C_1 11\ C_2 11\ C_3 11\ \ldots\ C_{n-1} 11\ C_n$$

Example 1 Coding of TM



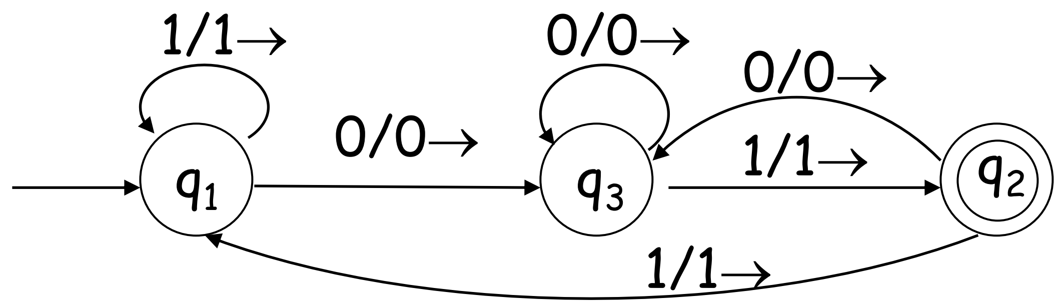
$\delta(q_1, 0) = (q_3, 0, \rightarrow) \Rightarrow 010100010100$

$\delta(q_3, 1) = (q_3, 1, \rightarrow) \Rightarrow 0001001000100100$

$\delta(q_3, B) = (q_2, B, \rightarrow) \Rightarrow 00010001001000100$

TM $\Rightarrow$ 010100010100 11 0001001000100100 11
00010001001000100

Example 2 Coding of TM



$\delta(q_1,0) = (q_3,0,\rightarrow)$

$\Rightarrow$ 010100010100

$\delta(q_1,1) = (q_1,1,\rightarrow)$ $\Rightarrow$ 010010100100

$\delta(q_3,0) = (q_3,0,\rightarrow)$ $\Rightarrow$ 00010100010100

$\delta(q_3,1) = (q_2,1,\rightarrow)$ $\Rightarrow$ 000100100100100

$\delta(q_2,0) = (q_3,0,\rightarrow)$ $\Rightarrow$ 0010100010100

$\delta(q_2,1) = (q_1,1,\rightarrow)$ $\Rightarrow$ 0010010100100

TM $\Rightarrow$ 01010001010011010010100100110001010001010011

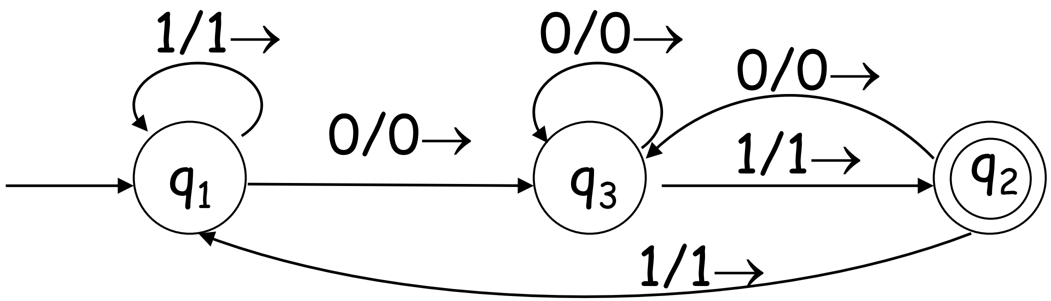
000100100100100110010100010100110010010100100

Not - RecuEnuLang

$$L_d = \{ w_i \mid w_i \notin L(M_i) \}$$

	1	2	3	4	...
1	0	1	1	0	...
2	1	0	0	0	...
3	0	0	0	1	...
4	0	1	0	1	...
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Not - RecuEnuLang



TM ⇒ 01010001010011010010100100110001010001010011

000100100100100110010100010100110010010100100

M_i

w_i

$$w_i \notin L(M_i)$$

L_d is not *RecuEnuLang*

Theorem L_d is not a recursively enumerable language.
That is there is no TM to accept L_d.

Proof : Suppose L_d were L(M) for some TM M.

$\Rightarrow$ There is at least one code for M, say i, that $M=M_i$

Now, ask if w_i is in L_d.

- w_i is in $L_d \Rightarrow M_i$ accepts $w_i \Rightarrow w_i$ is not in L_d
- w_i is not in $L_d \Rightarrow M_i$ does not accept $w_i \Rightarrow w_i$ is in L_d

Recursive languages

L is recursive if $L = L(M)$ for some TM M such that

1. $w \in L \Rightarrow$ M accepts w and halts

2. $w \notin L \Rightarrow$ M eventually halts

Recursive languages

If L is recursive language, so is $\overline{L}$.

Suppose L=$L(M)$, M=$(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Let $\overline{M}$=$(Q \cup \{r\}, \Sigma, \Gamma, \delta, q_0, B, \{r\})$ such that

1. r is a new state which is not in Q
2. if $\delta(q,a) = \phi$ for any $q \in Q-F$ and $a \in \Sigma$

   then $\delta(q,a) = (r, a, \rightarrow)$

Recursive languages

If both L and its complement $\overline{L}$ are RE, then L is recursive.

Suppose $M_1 = (Q_1, \Sigma, \Gamma, \delta_1, q_1, B, F_1)$

$M_2 = (Q_2, \Sigma, \Gamma, \delta_2, q_2, B, F_2)$

$M = (Q_1 \times Q_2, \Sigma, \Gamma, \delta, (q_1, q_2), B, F_1 \times (Q_2 - F_2))$

$$\delta((p,q),(a,b)) = (\delta_1(p,a), \delta_2(q,b))$$

Universal TM

$$L_u = \{ (M,w) \mid w \in L(M) \}$$

Universal language

Let $L(M) = \{0\}\{1\}^*$

Tape 1 : 010100010100 11 0001001000100100 11

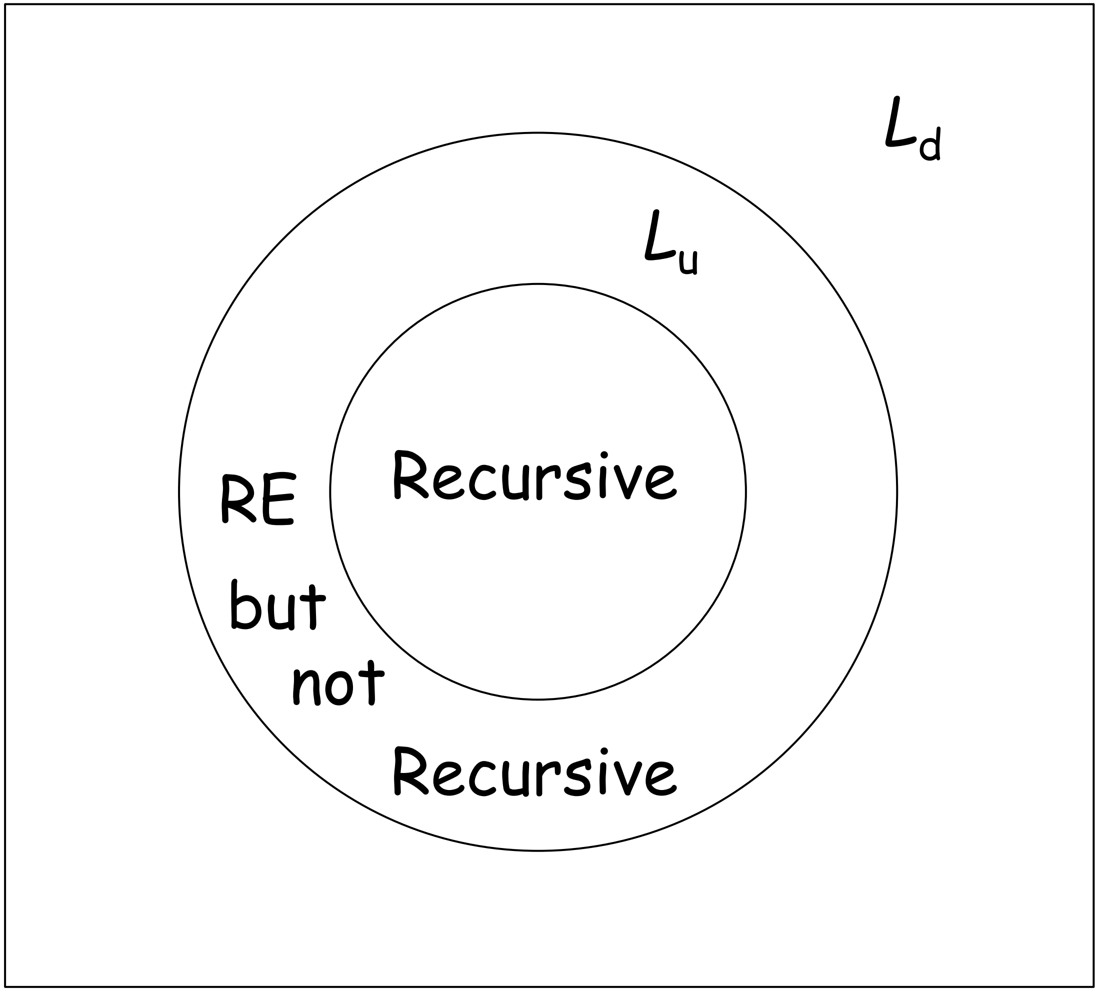
00010001001000100111011

Tape 2 : 10100100

Tape 3 : 0

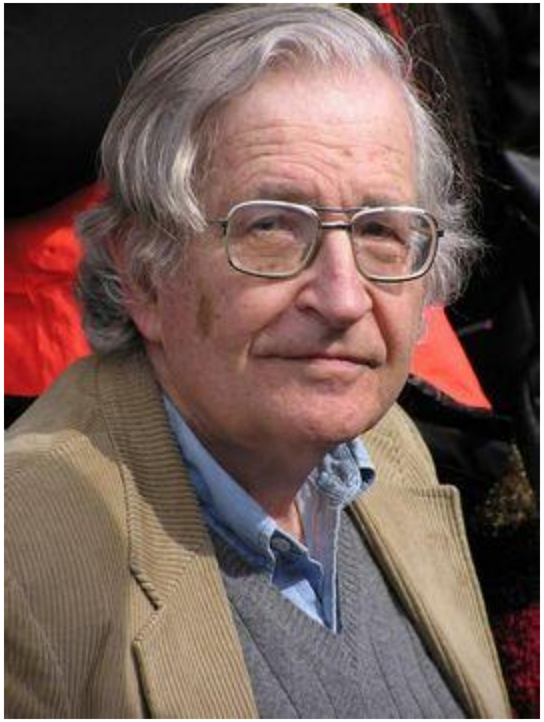
Tape 4 :

Recursive languages



Chomsky Grammar

- Noam Chomsky (1928-)
- Chomsky Grammar (1956)
- Syntactic Structures



Chomsky Grammar

Type 0: phrase structure grammar(PSG)

$\alpha \to \beta$; $\alpha \in (V \cup T)^* V (V \cup T)^*$, $\beta \in (V \cup T)^*$

Type 1: context sensitive grammar(CSG)

$\alpha A \beta \to \alpha \omega \beta$; $A \in V$, $\alpha, \omega, \beta \in (V \cup T)^*$

Type 2: context free grammar(CFG)

$A \to \omega$; $A \in V$, $\omega \in (V \cup T)^*$

Type 3: regular grammar(RG)

$A \to \alpha \mid \alpha B$; $A, B \in V$, $\alpha \in T^*$

Phrase Short Grammar

W=aaabbbccc

$S \rightarrow abc \mid aAbc$

$Ab \rightarrow bA$

$Ac \rightarrow Bbcc$

$bB \rightarrow Bb$

$aB \rightarrow aa \mid aaA$

S
aAbc
abAc
abBbcc
aBbbcc
aaAbbcc
aabAbcc
aabbAcc
aabbBbccc
aabBbbccc
aaBbbbccc
aaabbbccc

Context Sensitive Grammar

$S \to aDc$

$D \to aDE \mid b$

$bEc \to bbcc$

$bEE \to bbFE$

$FE \to FF$

$FFc \to GFc \to Gcc$

$FG \to GG$

$bGc \to bbcc$

$bGG \to bbHG$

$HG \to HH$

$HHc \to EHc \to Ecc$

$HE \to EE$

$\underline{W = aaabbbccc}$

S

$a\underline{D}c$

$aa\underline{D}Ec$

$aaa\underline{D}EEc$

$aaa\underline{b}\underline{E}\underline{E}c$

$aaabb\underline{FE}c$

$aaabb\underline{FFc}$

$aaabb\underline{GFc}$

$aaab\underline{bGcc}$

$aaabbbccc$

Right Linear Grammars

A grammar $G = (V, T, S, P)$ is said to be right linear if all productions are of the form

$$A \to xB$$

$$A \to x$$

where $A, B \in V$, and $x \in T^*$

Left Linear Grammars

A grammar $G = (V, T, S, P)$ is said to be right linear if all productions are of the form

$$A \to Bx$$

$$A \to x$$

where $A, B \in V$, and $x \in T^*$

Example 3

$G = (\{S\}, \{a, b\}, S, P)$

$$S \rightarrow abS \mid a$$

$$S \rightarrow Sba \mid a$$

Example 4

$L=\{w \mid w \in \{0, 1\}^*$ and ending with 01 $\}$

$L= \{0, 1\}^* \{01\}$

$S \rightarrow A\ 01$

$A \rightarrow A0 \mid A1 \mid \varepsilon$

ie, $S \Rightarrow A01$
$\Rightarrow A001$
$\Rightarrow A0001$
$\Rightarrow A10001$
$\Rightarrow 10001$

$G=(\{S, A\}, \{0, 1\}, S, P)$

What is the right linear grammar for L ?

Example 5

$L = \{ w \mid w \in \{0, 1\}^* \text{ and } w \text{ contains } 01 \}$

$L = \{0, 1\}^* \quad \{01\} \quad \{0, 1\}^*$

$S \rightarrow 0S \mid 1S, \quad S \rightarrow 01A, \quad A \rightarrow 0A \mid 1A \mid \varepsilon$

$G = (\{S, A\}, \{0, 1\}, S, P)$

$P: \quad S \rightarrow 0S \mid 1S \mid 01A, \quad A \rightarrow 0A \mid 1A \mid \varepsilon$

Linear Bounded Automata

A linear bounded automata is a nondeterministic Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

that Σ must contain two special symbols [and], such that $\delta(q_i, [)$ can contain only elements of the $(q_j, [, \rightarrow)$, and $\delta(q_i,])$ can contain only elements of the $(q_j,], \leftarrow)$

Good good study

day day up !