

1. 다음과 같은 production으로 정의되는 CFG에 대해 다음에 답하시오. (25점)

$$S \rightarrow aA|aBB, A \rightarrow aaA|\lambda, B \rightarrow bB|bbC, C \rightarrow B$$

(1) unit-production, useless production, λ -production을 제거하시오. (반드시 적절한 순서를 사용할 것)

$$\textcircled{1} \lambda\text{-production 없애기: } S \rightarrow a(aA|aBB), A \rightarrow aaA|aa, B \rightarrow bB|bbC, C \rightarrow B$$

$$\textcircled{2} \text{unit - " " } S \rightarrow aa(aBB|a), A \rightarrow aaA|aa, B \rightarrow bB|bbC, C \rightarrow bB|bbC$$

$$\textcircled{3} \text{useless 없애기: } S \rightarrow aA|a, A \rightarrow aaA|aa$$

(2) (1)에서 얻어진 최종 결과를 CNF 형태로 바꾸시오. ($D \rightarrow a$ 를 추가로 사용 할 것)

$$\left. \begin{array}{l} S \rightarrow DA|a \\ A \rightarrow E|DD \\ E \rightarrow DA \\ D \rightarrow a \end{array} \right) \text{ 또는 } \left(\begin{array}{l} S \rightarrow DA|a \\ A \rightarrow E|DD \\ E \rightarrow DD \\ D \rightarrow a \end{array} \right)$$

(3) CYK알고리즘을 이용하여 string aaaa의 membership을 판단하려한다. $V_{22}, V_{23}, V_{21}, V_{14}$ 를 각각 구하고 membership을 판단하시오.

$$V_{22} = \{S, D\}, V_{23} = \{A\}, V_{21} = \{S, E\}, V_{14} = \{A\}, aaaa \notin L$$

또는

$$V_{22} = \{S, D\}, V_{23} = \{A, E\}, V_{21} = \{S\}, V_{14} = \{A\}, //$$

(4) (1)에서 얻어진 최종 결과를 GNF 형태로 바꾸시오. ($D \rightarrow a$ 를 추가로 사용 할 것)

$$S \rightarrow aA|a, D \rightarrow aDA|aD, D \rightarrow a$$

(5) (4)의 GNF를 이용하여 이 language를 accept하는 NPDA를 찾으시오.

$$M = (\{q_0, q_1, q_f\}, \{a\}, \{S, A, D, \epsilon\}, \{S, q_0, \epsilon, \{q_f\}\})$$

$$\delta(q_0, \epsilon, \epsilon) = \{(q_1, S)\}$$

$$\delta(q_1, \epsilon, \epsilon) = \{(q_f, \epsilon)\}$$

$$\delta(q_1, a, S) = \{(q_1, A), (q_1, \epsilon)\}$$

$$\delta(q_1, a, A) = \{(q_1, DA), (q_1, D)\}$$

$$\delta(q_1, a, D) = \{(q_1, \epsilon)\}$$

2. 다음 npda $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, z\}, \delta, q_0, z, F)$,

$$\delta(q_0, a, z) = \{(q_1, a), (q_2, \lambda)\}, \delta(q_1, b, a) = \{(q_1, b)\}, \delta(q_1, b, b) = \{(q_1, b)\}, \delta(q_1, a, b) = \{(q_2, a)\}$$

(1) $F = \{q_2\}$ 인 경우 이 npda에 의해 accept되는 language를 집합으로 나타내시오. (5점)

$$L = \{ab^n a \mid n \geq 1\} \cup \{a\}$$

(2) $F = \{q_1, q_2\}$ 인 경우 이 npda에 의해 accept되는 language를 집합으로 나타내시오. (5점)

$$L = \{ab^n a \mid n \geq 1\} \cup \{ab^m \mid m \geq 0\}$$

3. Binary notation으로 표현된 정수 x 를 입력받아 어떤 함수 $f(x)$ 를 계산하는 Turing Machine M 이 다음과 같이 주어졌다. (15점)

$$M = (\{q_0, q_r, q_f\}, \{0, 1\}, \{0, 1, \square\}, \delta, q_0, \square, \{q_f\})$$

$$\delta(q_0, 0) = (q_0, 0, R), \delta(q_0, 1) = (q_0, 1, R), \delta(q_0, \square) = (q_r, 0, L), \delta(q_r, 0) = (q_r, 0, L), \delta(q_r, 1) = (q_r, 1, L), \delta(q_r, \square) = (q_f, \square, R)$$

(1) 아래와 같은 초기 상황이 주어졌을 때, M 의 transition 과정을 기술하시오. (5점)

$$q_0 101 \vdash q_0 1 10 q_0 1 \vdash 10 1 q_0 \square \vdash 10 q_r 10 1 \vdash 1 q_r 0 10 \vdash q_r 10 10 \vdash q_r \square 10 10 \vdash q_f 10 10$$

(2) M 이 계산하는 함수 $f(x)$ 는 무엇인지, 그 수학적식을 쓰시오. (5점)

$$f(x) = 2x$$

(3) 다음은 M 을 수정하여 $f(x) = x \div 2$ (x 를 2로 나눈 몫)를 계산하는 Turing Machine M_{div} 의 일부를 만든 것이다. 완성되기 위해서는 transition이 3개 더 필요하다. 이것이 무엇인지 쓰시오. (5점)

(hint: $\delta(q_0, \square)$ 인 상황에 대한 처리부터 생각)

$$M_{div} = (\{q_0, q_1, q_r, q_f\}, \{0, 1\}, \{0, 1, \square\}, \delta, q_0, \square, \{q_f\}),$$

$$\delta(q_0, 0) = (q_0, 0, R), \delta(q_0, 1) = (q_0, 1, R), \delta(q_r, 0) = (q_r, 0, L), \delta(q_r, 1) = (q_r, 1, L), \delta(q_r, \square) = (q_f, \square, R)$$

$$\delta(q_0, \square) = (q_1, \square, L), \delta(q_1, 1) = (q_r, \square, L), \delta(q_1, 0) = (q_r, \square, L)$$

(\times 입력이 1 또는 0일 경우, 출력은 \square 임)

4. Universal Turing Machine이 무엇인지, 아는 대로 쓰시오. (5점)

현재 컴퓨터 처럼 "programming 이 가능한 범용" turing machine.

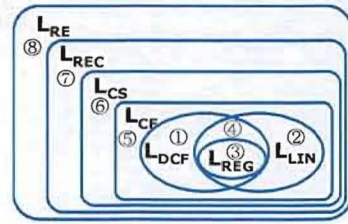
세 개의 tape을 이용하여 구현 가능

$$\left[\begin{array}{l} \text{tape ①: 특정 계산을 수행하는 TM에 대한 description code} \\ \text{②: " " TM에 주어진 입력} \\ \text{③: " " TM의 현재 state를 저장} \end{array} \right]$$

② ③을 확인 후 ①에서 적절한 transition을 찾아 ②, ③을 변경하는 식으로 수행함.

5. 다음 각 Language가 오른쪽 그림의 영역 ①부터 ⑧ 중 어디에 속하는지 말하고, 그 이유를 쓰시오. (20점)

(1) $L = \{a^n b^{n^2} c^m \mid n, m \geq 1\}$ 답: ①



• DCF인 이유: DPDA를 만들수 있음

- a가 들어오면 stack에 1 push

- b가 들어오기 시작하면 state 바뀌어 1 pop.

- stack top에 1이 있으면, b가 들어오면 state 바뀌어 1 push

- c가 들어오기 시작하면 state 바뀌어 1 pop

- stack top에 1이 없고 있을 때 끝나면 final.

• Linear가 아닌 이유: pumping Lemma로 증명.

Assume L is Linear.

$\forall m > 0$, set $w = a^m b^{2m} c^m$

For all decomposition of $w = uvxyz$ ($|uvy| \leq m$, $|vy| \geq 1$),

$w = a^{k_1} a^{k_2} x a^{k_3} c^{k_4}$ ($k_1 + k_2 + k_3 + k_4 \leq m$, $k_2 + k_3 \geq 1$)

Set $x = 0$ $w_0 = a^{m-k_2} b^{2m} c^{m-k_4} \notin L$ ($\because k_2 + k_3 \geq 1$) $\therefore L$ is not linear

(2) $L = \{a^n b^n a^n \mid n \geq 1\}$ 답: ⑥

• CS인 이유: LBA를 만들수 있음

tape에서 처음 a를 x로 바꾸고 오른쪽 이동. b 찾아서 y로 → 오른쪽 이동 a 찾아서 z로, → 오른쪽 이동 b 찾아서 u로 → 다시 처음 x 찾아 같은 길 반복하며 갯수를 check 함. String 바깥을 사용하지 않으므로 LBA.

• CF가 아닌 이유: pumping Lemma로 증명

Assume L is CF.

$\forall m > 0$ set $w = a^m b^m a^m b^m$.

Consider all possible decomposition of $w = uvxyz = a^{\dots} ab^{\dots} ba^{\dots} ab^{\dots}$

①, ①': $vay = a^{k_1} a^{k_2}$ ($k_1 + k_2 \leq m$) set $x = 0$. $w_0 = a^{m-k_2} b^m a^m b^m \notin L$

②, ②': $vay = b^{k_1} a b^{k_2}$ (\cdot) set $x = 0$ $w_0 = a^m b^{m-(k_1+k_2)} a^m b^m \notin L$

③, ③': $vay = a^{k_1} x b^{k_2}$ (\cdot) set $x = 0$ $w_0 = a^{m-k_1} b^{m-k_2} a^m b^m \notin L$

④: $vay = b^{k_1} x a^{k_2}$ ($1 \leq k_1 + k_2 \leq m$) set $x = 0$, $w_0 = a^m b^{m-k_1} a^{m-k_2} b^m \notin L$

$\therefore L$ is not CF.

6. 다음 명제의 참, 거짓을 판단하고 간단히 증명하시오. (20점) (교과서의 theorem으로 증명된 사실은 별도 증명 없이 활용 가능함)

(1) L_1, L_2 가 recursive이면 $L_1 - L_2$ recursive이다. True

L_2 가 recursive이면 L_2^c 도 recursive. L_1 과 L_2^c 를 위한 membership algorithm 있음

$L_1 - L_2 = L_1 \cap L_2^c$ 이므로,

L_1, L_2^c 의 membership 알고리즘을 함께 사용하여 둘다 accept인 경우만 accept 하는 membership 알고리즘 만들수 있음 $\therefore L_1 - L_2$ 도 recursive

(2) L_1 이 context-free이고, L_2 가 regular일 때, L_1 과 L_2 에 동시에 속하는 string이 있는지 판단하는 알고리즘이 존재한다. True

$L_1 \cap L_2$ 는 context free (regular intersection 이므로).

Context free language는 emptiness를 판단하는 algorithm이 없다.

$L_1 \cap L_2 = \emptyset$ 일때 동시에 속하는 string이 없는 것이 될.

(3) $L = \{(ab)^n \mid n \geq 1\}$ 를 생성하는 LL(1) grammar는 존재하지 않는다. False

LL(1) grammar:

$S \rightarrow aA$ $A \rightarrow bB$ $B \rightarrow S \mid \lambda$

(4) 임의의 Unrestricted Grammar G 에 대해, language $(L(G))^c$ 를 생성하는 Unrestricted Grammar가 존재한다. False

Unrestricted Grammar에 의해 생성되는 language는 Recursively Enumerable.

L 이 recursively enumerable 일때, L^c 는 아닐 수도 있음 (판단하지 않음)

L^c 이 recursively enumerable이 아닐 경우 Unrestricted Grammar가 존재할 가능성 있음 (Unrestricted grammar는 Recursively Enumerable language에 대응)

7. 어떤 Turing Machine의 transition function은 현재 head가 위치한 셀의 symbol 뿐 아니라, 그 좌우에 있는 symbol에도 의존하여 다음 state가 결정된다고 한다. 이 Turing machine의 formal definition을 쓰고, 그 power를 standard Turing machine과 비교하시오. (5점)

$M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$

$\delta: Q \times \Gamma \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \Gamma$

power는 같음. (standard turing machine이 M 을 simulate 가능)

STM의 tape alphabet Γ 재정의: $\Gamma_3 = \{abc \mid a, b, c \in \Gamma\}$

M 의 transition이 $\delta(q_i, (a, b, c)) = (q_j, d, L)$ 일때

STM에서는 $\delta_3(q_i, abc) = (q_j, d, L)$ 로 정의.