Theory of Computation

Fall 2021

1) Consider $\Sigma_1 = \{a, b, c\}$.

a) (5pts) State the pumping Lemma for regular languages.

⇒ Pumping Lemma:

If A is a regular language, then there is a number p (the pumping length) where if s is any string in A of length at least p, then s may be divided into three pieces, $s = xyz$, satisifying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$,
2. $|y| > 0$, and
3. $|xy| \leq p$

b) (5 pts) Is the following language regular or not?

$$L1 = \{a^m b^n c^p : m \geq n \geq p \geq 0\}$$

⇒ L1 is not regular language.

c) (10pts) Prove your answer to question b. You may ~~yet~~ use Pumping Lemma if needed.

⇒ Proof by contradiction:

Let's assume $L = \{a^m b^n c^p : m \geq n \geq p \geq 0\}$ is regular language.

Let l be pumping length, $|s| \geq l$ where $s = xyz$ and $s \in L$.

Let $s = a^l b^l c^l$

(i) $s = xyz$ where $x = a^k$, $y = a^t$, $z = a^u b^l c^l$

- $|y| > 0$
- $|xy| \leq l$.
- xy^iz $(\forall i \geq 0)$

consider $i=0 \Rightarrow s_0 = a^{r+u}\ b^l\ c^l$

$l = r+t+u$;

$r+u \neq l$

$r+u \neq r+t+u$

$m \neq n$

$\therefore s_0 \notin L$

$\therefore$ It is contradiction to our assumption;

$\therefore$ L is not regular.

2) Consider the content-free language over $\Sigma_2 = \{x, y\}$:

$L_2 = \{x^n y^n ; n \geq 0\}$

a) (10 pts) Give a content-free grammar for this language L2.

$\Rightarrow$ CFG for L_2: $G = (V, \Sigma, R, S)$

$V = \{S\}$

$\Sigma = \{x, y, \varepsilon\}$

$S = \{q_0\}$

$R = \{S \rightarrow xSy \mid \varepsilon\}$

$x^2y^2 = xSy$;

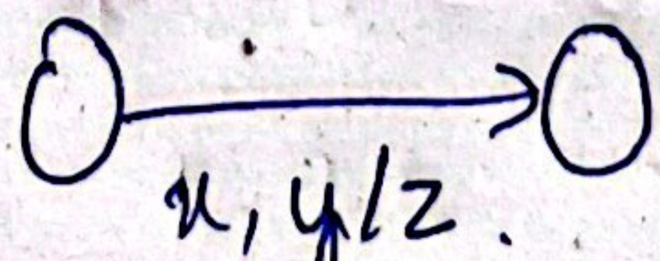
$xxSyy$

$xx\varepsilon yy$

$xxyy$

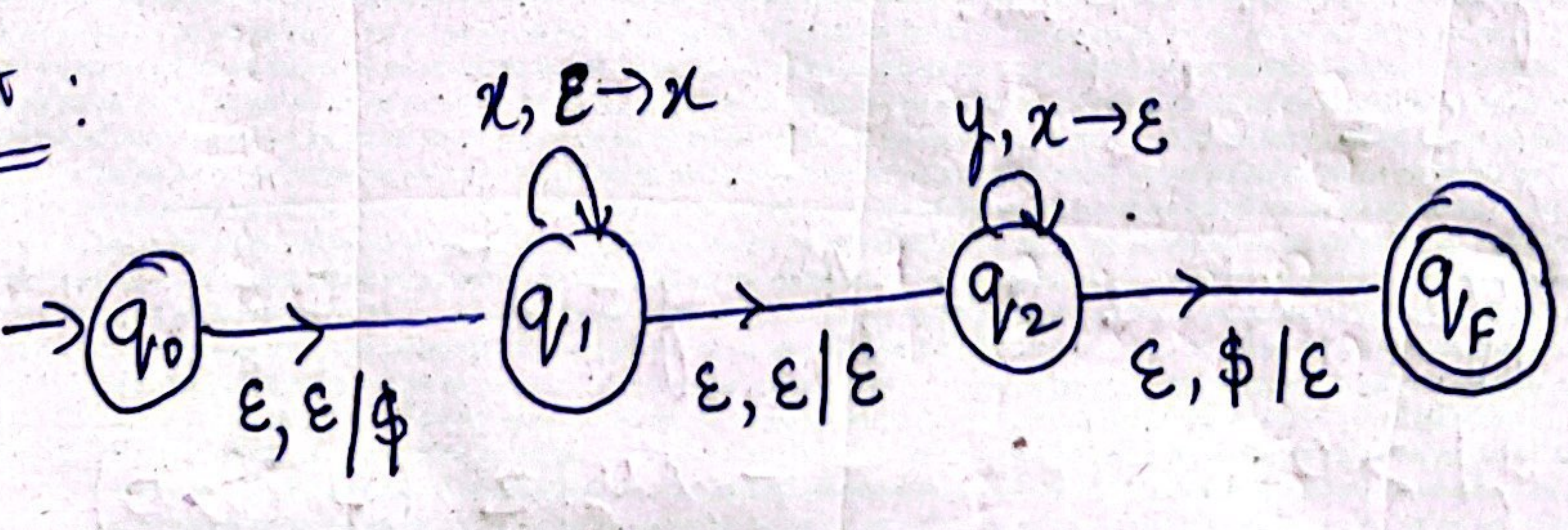
x^2y^2

b) (10 pts) Draw the state diagram of a pushdown automaton to recognize this language. You may use the following notation to label your machine's transitions.



(read I/P symbol x, stack top is y, push symbol z)

$\Rightarrow$ PDA :



3) The SUBSET-SUM Problem takes as input a set S of integers and an integer T, the question is whether there exists a non-empty subset R that sums to T.

a. (5pts) Define polynomial-time reducibility $A \leq_p B$.

⇒ A language A is polynomial time reducible to a another language B if there is a polynomial time computable function $f: \Sigma^* \to \Sigma^*$ such that: $w \in A \Longleftrightarrow f(w) \in B$ for every $w \in \Sigma^*$.

Terminology /notation:

- $A \leq_p B$
- f is the polynomial time reduction of A to B
- "polynomial time many-one reducible".

b) (5 pts) In general, how do you prove that a given problem X is NP-complete? Please list the steps.

⇒ A language Y is NP-complete if:

1) Y is in NP, and
2) For every language $X \in NP$ we have $X \leq_p Y$.

c) (10pts) Prove that 3-CNF-SAT $\leq_p$ SUBSET-SUM.

(3-CNF-SAT problem: Given a formula in 3-CNF, is there an assignment of the variables such that the formula evaluates to true?

For example, $(x \vee \neg y \vee \neg z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee y \vee \neg z)$ is a 3-CNF formula).

⇒ Let ϕ be a Boolean formula with variables $x_1, \ldots, x_l$ and clauses $c_1, \ldots, c_k$. The reduction converts ϕ to an instance of the SUBSET-SUM problem $\langle S, t \rangle$, wherein the elements of S and the number t are the rows on the table, expressed in ordinary decimal notation. The rows above the double line are labeled.

$y_1, z_1, y_2, z_2, \ldots, y_l, z_l$ and $g_1, h_1, g_2, h_2, \ldots, g_k, h_k$

and constitute the elements of S. The row below the double line is t.

Thus, S contains one pair of numbers, y_i, z_i, for each variable x_i in ϕ. The decimal representation of these numbers is in two parts, as indicated in the table. The left-hand part comprises a 1 followed by $l-i$ 0s. The right-hand part contains one digit for each clause, where the digit of y_i in column c_j is 1 if clause c_j contains literal x_i, and the digit of z_i in column c_j is 1 if clause c_j contains literal $\overline{x_i}$. Digits not specified to be 1 are 0.

The table is partially filled in to illustrate sample clauses, c_1, c_2, and c_k:

$$(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3 \vee \ldots) \wedge \ldots \wedge (\overline{x_3} \vee \ldots \vee \ldots)$$

Additionally, S contains one pair of numbers, g_j, h_j for each clause c_j. These two numbers are equal and consist of a 1 followed by $k-j$ 0s.

Finally, the target number t, the bottom row of the table, consists of l 1s followed by k 3s.

	1 2 3 4 … l	c_1 c_2 … c_k
y_1	1 0 0 0 … 0	1 0 … 0
z_1	1 0 0 0 … 0	0 0 … 0
y_2	1 0 0 … 0	0 1 … 0
z_2	1 0 0 … 0	1 0 … 0
y_3	1 0 …	1 1 … 0
z_3	1 0 …	0 0 … 1
⋮	⋮	
y_l	1	0 0 … 0
z_l	1	0 0 … 0
g_1		1 0 … 0
h_1		1 0 … 0
g_2		1 … 0
h_2		1 … 0
⋮		
g_k		… 1
h_k		… 1
		3 3 … 3

x_1 = True
y = True
z = False.

ϕ = True

Yes