LAST NAME:

FIRST NAME: instructor

Problem 1 [**9 points**] Let L be the language over the alphabet $\Sigma = \{a, b, c, d\}$ that contains exactly those strings whose form is:

$$b^k a^m c^k d^m$$

where $k, m \geq 0$ are natural numbers.

If L is context free, then use part (a) of the answer space below to write a complete formal definition of a context free grammar that generates L, and do not write anything in part (b).

If L is not context free, then do not write anything in part (a) of the answer space, but complete the missing parts of the text given in part (b) so as to obtain a proof that L is not context free.

(a) Context free grammar for L:

Answer:

(b) Proof that L is not context free:

Observe that all words of L satisfy the following characteristic property:

#b's = #c's
#a's = #d's

Assume the opposite, that L is context free.
Let ρ be the constant as in the Pumping Lemma for L.
Let $w_0 \in L$ be a string defined as follows:

$w_0 =$ $b^k a^m c^k d^m$, $m > \rho$, $k > \rho$

w_0 belongs to L because

it satisfies definition of L

w_0 must pump because

$|w_0| = 2k + 2m > 2\rho + 2\rho = 4\rho > \rho$

In any "pumping" decomposition of w_0, the pumping window satisfies the following property:

it is contained either within one of the 4 segments or within 2 adjacent ones.

because

it is too short to extend through 3 or more

By pumping 1 times, we obtain a string

which violates the stated characteristic property because at most two adjacent letter are pumped and at least two are not.

and thus does not belong to L. Since L violates the Pumping Lemma, L is not context free.

LAST NAME:

FIRST NAME: instructor

Problem 2 [**9 points**] Let L be the language over the alphabet $\Sigma = \{a, b, c, d\}$ that contains exactly those strings whose form is:

$$b^{3k}a^{2n+1}c^{4n+2}d^{k+3}$$

where $k, n \geq 0$ are natural numbers.

If L is context free, then use part (a) of the answer space below to write a complete formal definition of a context free grammar that generates L, and do not write anything in part (b).

If L is not context free, then do not write anything in part (a) of the answer space, but complete the missing parts of the text given in part (b) so as to obtain a proof that L is not context free.

(a) Context free grammar for L:

Answer:

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

$V = \{S, A\}$

$\Sigma = \{a, b, c, d\}$

P:

$S \to bbbSd \mid Addd$

$A \to aaAcccc \mid acc$

(b) Proof that L is not context free:

Observe that all words of L satisfy the following characteristic property:

Assume the opposite, that L is context free.
Let β be the constant as in the Pumping Lemma for L. Let $w_0 \in L$ be a string defined as follows:

$w_0 =$

w_0 belongs to L because

w_0 must pump because

In any "pumping" decomposition of w_0, the pumping window satisfies the following property:

because

By pumping times, we obtain a string

which violates the stated characteristic property because

and thus does not belong to L. Since L violates the Pumping Lemma, L is not context free.

LAST NAME: ____________________

FIRST NAME: instructor

Problem 3 [**9 points**] Let L be the language over the alphabet $\Sigma = \{a, b, c\}$ that contains exactly those strings whose form is:

$$b^{2k}a^{3n}c^{4k}$$

where $k, n \geq 0$ are natural numbers.

If L is regular, then use part (a) of the answer space below to write a regular expression that generates L, and do not write anything in part (b).

If L is not regular, then do not write anything in part (a) of the answer space, but complete the missing parts of the text given in part (b) so as to obtain a proof that L is not regular.

(a) regular expression for L:

Answer:

(b) Proof that L is not regular:

Observe that all words of L satisfy the following characteristic property:

#c's = twice #b's

Assume the opposite, that L is regular.
Let γ be the constant as in the Pumping Lemma for L. Let $w_0 \in L$ be a string defined as follows:

$w_0 =$ $b^{2k} c^{4k}$, $k > \gamma$

w_0 belongs to L because

obtained from the template by $n=0$

w_0 must pump because

$|w_0| = 6k > 6\gamma > \gamma$

In any "pumping" decomposition of w_0, the pumping window satisfies the following property:

it is contained entirely within the b-segment.

because

its length, say j, is: $j \leq \gamma < k < 2k$

By pumping 1 times, we obtain a string

$b^{2k+j} c^{4k}$

which violates the stated characteristic property because $4k \neq 2(2k+j)$ since $j > 0$.

and thus does not belong to L. Since L violates the Pumping Lemma, L is not regular.

LAST NAME:

FIRST NAME: instructor

Problem 4 [**12 points**] Let L be the language accepted by the pushdown automaton:
$M = (Q, \Sigma, \Gamma, \delta, q, F)$ where: $Q = \{q, p\}$;
$\Sigma = \{a, b, c, d, g, h\}; \Gamma = \{H, O, R, S, T, Y\}; F = \{q\}$ and δ is defined by the following transition set:

$$
\begin{array}{ll}
[q, d, \lambda, p, SHORT] & [p, h, H, p, \lambda] \\
[q, h, \lambda, p, STORY] & [p, a, O, p, \lambda] \\
 & [p, b, R, p, \lambda] \\
 & [p, g, S, q, \lambda] \\
 & [p, d, T, p, \lambda] \\
 & [p, c, Y, p, \lambda]
\end{array}
$$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$ where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol X_1 is pushed first, while the rightmost symbol X_n is pushed last.)

(a) Write a regular expression that represents L. If such a regular expression does not exist, prove it.

Answer:

$(ddbahg \cup hcbadg)^*$

(c) Write a complete formal definition of a context-free grammar that generates L. If such a grammar does not exist, prove it.

Answer:

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

$V = \{S, A, B\}$

$\Sigma = \{a, b, c, d, g, h\}$

P:

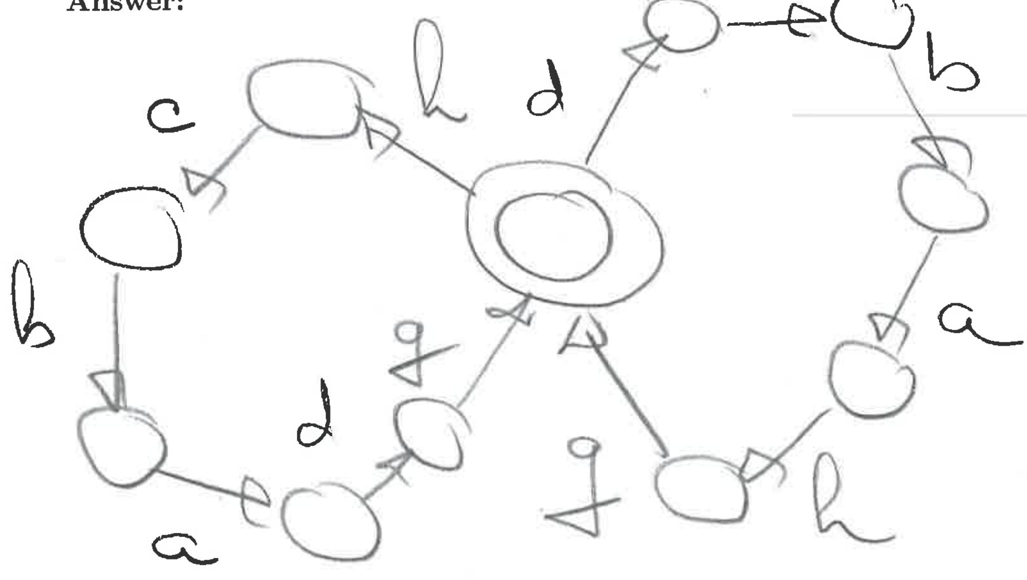
$S \to \lambda \mid SS \mid A \mid B$

$A \to ddbahg$

$B \to hcbadg$

(c) Draw a state-transition graph of a finite-state automaton that accepts L. If such an automaton does not exist, prove it.

Answer:



LAST NAME:

FIRST NAME: instructor

Problem 5 [**12 points**] Let L be the language accepted by the pushdown automaton:
$M = (Q, \Sigma, \Gamma, \delta, q, F)$ where: $Q = \{q, p\}$;
$\Sigma = \{a, b, c, d, g, h\}; \Gamma = \{A, E, G, N, R, S\}; F = \{p\}$
and δ is defined by the following transition set:

$[q, c, \lambda, q, GREEN]$	$[p, a, A, p, \lambda]$
$[q, a, \lambda, q, GRASS]$	$[p, c, E, p, \lambda]$
$[q, \lambda, \lambda, p, \lambda]$	$[p, g, G, p, \lambda]$
	$[p, h, N, p, \lambda]$
	$[p, b, R, p, \lambda]$
	$[p, d, S, p, \lambda]$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \dots X_n \in \Gamma^*$ where $n \geq 2$, is pushed on the stack by an individual transition, then the left-most symbol X_1 is pushed first, while the rightmost symbol X_n is pushed last.)

(a) Write a complete formal definition of a context-free grammar that generates L. If such a grammar does not exist, prove it.

Answer:

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

$V = \{S\}$

$\Sigma = \{a, b, c, d, g, h\}$

P:

$S \to cShccbg \mid aSddabg \mid \lambda$

(b) Explain how to construct an algorithm that solves the following problem:

INPUT: Pushdown automaton P_1 that accepts a language L_1 and a Turing Machine T_2 that accepts a language L_2;

OUTPUT: Turing Machine T_3 that accepts the language $L_1 \cup L_2$;

If this algorithm does not exist, prove it.

Answer:

T_3 simulates P_1 and T_2 and accepts if and only if P_1 and T_2 accept.

LAST NAME:

FIRST NAME: instructor

Problem 6 [**12 points**] Let L be the language accepted by the pushdown automaton: $M = (Q, \Sigma, \Gamma, \delta, q, F)$ where: $Q = \{q, p, s, t\}$; $\Sigma = \{a, b, c, d, g, h\}; \Gamma = \{A, E, H, L, R, T\}; F = \{s\}$ and δ is defined by the following transition set:

$[q, a, \lambda, q, HEAR]$	$[p, a, A, p, \lambda]$
$[t, b, \lambda, t, TELL]$	$[p, c, E, p, \lambda]$
$[q, \lambda, \lambda, p, \lambda]$	$[p, b, R, p, \lambda]$
$[p, c, \lambda, t, \lambda]$	$[p, h, H, p, \lambda]$
$[t, \lambda, \lambda, s, \lambda]$	$[s, g, L, s, \lambda]$
	$[s, c, E, s, \lambda]$
	$[s, d, T, s, \lambda]$

(Recall that M is defined so as to accept by final state and empty stack. Furthermore, if an arbitrary stack string, say $X_1 \ldots X_n \in \Gamma^*$ where $n \geq 2$, is pushed on the stack by an individual transition, then the leftmost symbol X_1 is pushed first, while the rightmost symbol X_n is pushed last.)

(a) Write a complete formal definition of a context-free grammar that generates L. If such a grammar does not exist, prove it.

Answer:

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

$V = \{S, A, B\}$

$\Sigma = \{a, b, c, d, g, h\}$

P:

$S \to AcB$

$A \to aAhac \mid \lambda$

$B \to bBggcd \mid \lambda$

(b) Explain how to construct an algorithm that solves the following problem:

INPUT: Pushdown automaton P_1 that accepts a language L_1.

OUTPUT: Pushdown automaton P_2 that accepts the language $\overline{L_1}$.

If this algorithm does not exist, prove it.

Answer:

This algorithm does not exist: there exist context-free languages whose complement is not context-free.

LAST NAME:

FIRST NAME: instructer

Problem 7 [**12 points**] Consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, q, F)$ such that: $Q = \{q, p, s, x\}$; $\Sigma = \{a, b, c\}$; $\Gamma = \{B, a, b, c\}$; $F = \{x\}$; and δ is defined by the following transition set:

$[q, a, p, a, R]$	$[p, a, s, a, R]$	$[s, a, q, a, R]$
$[q, b, p, b, R]$	$[p, b, s, b, R]$	$[s, b, q, b, R]$
$[q, c, p, c, R]$	$[p, c, s, c, R]$	$[s, c, q, c, R]$
$[q, B, q, B, R]$	$[p, B, x, c, R]$	

(M has an one-way infinite tape (infinite to the right only.) B is the designated blank symbol.) M accepts by final state.)

Let L_A be the set of strings which M accepts.
Let L_R be the set of strings which M rejects.
Let L_∞ be the set of strings on which M diverges.

(a) Write a regular expression that defines L_A. If such a regular expression does not exist, prove it.

Answer:

$((a \cup b \cup c)(a \cup b \cup c)(a \cup b \cup c))^*(a \cup b \cup c)$

(b) Write a regular expression that defines L_R. If such a regular expression does not exist, prove it.

Answer:

$((a \cup b \cup c)(a \cup b \cup c)(a \cup b \cup c))^*(a \cup b \cup c)(a \cup b \cup c)$

(c) Write a regular expression that defines L_∞. If such a regular expression does not exist, prove it.

Answer:

$((a \cup b \cup c)(a \cup b \cup c)(a \cup b \cup c))^*$

(d) Explain how to construct an algorithm that solves the following problem:

INPUT: Turing Machine T_1 that accepts a language L_1;

OUTPUT: Regular expression that defines L_1.

If this algorithm does not exist, prove it.

Answer: This algorithm does not exist: some recursively enumerable languages are not regular.

LAST NAME:

FIRST NAME: instructor

Problem 8 [**14 points**] Consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, q, F)$ such that: $Q = \{q, p, t, x\}$; $\Sigma = \{a, b, c\}$; $\Gamma = \{B, a, b, c, A, E, K\}$; $F = \{x\}$; and δ is defined by the following transition set:

$[q, a, p, A, R]$	$[p, a, p, a, R]$	$[t, A, x, a, R]$
$[q, b, p, E, R]$	$[p, b, p, b, R]$	$[t, E, q, c, R]$
$[q, c, p, K, R]$	$[p, c, p, c, R]$	$[t, K, x, K, R]$
$[q, B, q, B, R]$	$[p, B, t, B, L]$	

(M has an one-way infinite tape (infinite to the right only.) B is the designated blank symbol. M accepts by final state.)

Let L_A be the set of strings which M accepts.
Let L_R be the set of strings which M rejects.
Let L_∞ be the set of strings on which M diverges.

(a) Write a regular expression that defines L_A. If such a regular expression does not exist, prove it.

Answer:

$a \cup c$

(b) Write a regular expression that defines L_R. If such a regular expression does not exist, prove it.

Answer:

$(a \cup b \cup c)(a \cup b \cup c)(a \cup b \cup c)^*$

(c) Write a regular expression that defines L_∞. If such a regular expression does not exist, prove it.

Answer:

$\lambda \cup b$

(e) Explain how to construct an algorithm that solves the following problem:

INPUT: Turing Machine T_1 that accepts a language L_1;

OUTPUT: **yes** if L_1 is regular;
no otherwise.

If this algorithm does not exist, prove it.

Answer:

This algorithm does not exist. If it existed, it would decide the set of Turing Machines whose languages have the nontrivial property "is regular".

The property is nontrivial since it is true for Σ^* and false for the set of palindromes over Σ.

LAST NAME:

FIRST NAME: instructor

Problem 9 [**16 points**] Consider the Turing machine $M = (Q, \Sigma, \Gamma, \delta, q, F)$ such that: $Q = \{q, p, t, s, v, x\}$; $\Sigma = \{a, b, c\}$; $\Gamma = \{B, a, b, c, A, E, K\}$; $F = \{x\}$; and δ is defined by the following transition set:

$[q,a,p,a,R]$	$[p,a,t,A,R]$	$[t,a,t,a,R]$
$[q,b,p,b,R]$	$[p,b,t,E,R]$	$[t,b,t,b,R]$
$[q,c,p,c,R]$	$[p,c,t,K,R]$	$[t,c,t,c,R]$
$[q,B,q,B,R]$	$[p,B,p,B,R]$	$[t,B,s,B,L]$
$[s,a,v,a,L]$	$[v,a,v,a,L]$	
	$[v,b,v,b,L]$	
	$[v,c,v,c,L]$	
	$[v,K,x,K,R]$	

(M has an one-way infinite tape (infinite to the right only.) B is the designated blank symbol. M accepts by final state.)

Let L_A be the set of strings which M accepts.
Let L_R be the set of strings which M rejects.
Let L_∞ be the set of strings on which M diverges.

(a) Write a regular expression that defines L_A. If such a regular expression does not exist, prove it.

Answer:

$(a \cup b \cup c)c(a \cup b \cup c)^*a$

(b) Write a regular expression that defines L_∞. If such a regular expression does not exist, prove it.

Answer:

$\lambda \cup a \cup b \cup c$

(c) Explain how to construct an algorithm that solves the following problem:

INPUT: Turing Machine T_1 that decides a language L_1;

OUTPUT: Turing Machine T_2 that decides the language $\overline{L_1}$;

If this algorithm does not exist, prove it.

Answer:

T_2 is identical to T_1 but its decision is the negation of the decision of T_1.

(d) Explain how to construct an algorithm that solves the following problem:

INPUT: Turing Machine T_3 that accepts a language L_3;

OUTPUT: Turing Machine T_4 that accepts the language $\overline{L_3}$;

If this algorithm does not exist, prove it.

Answer:

This algorithm does not exist: the complement of a recursively enumerable language need not be rec. enum.