hw6

Graded

Student

Giancarlos Marte

Total Points

29 / 40 pts

Question 1

pda formal description

14 / 15 pts

- ✔ **– 0.5 pts** Incomplete/incorrect table (minor infraction)
- ✔ **– 0.5 pts** CFG has minor error
- b and c never appear on stack, shouldn't be on table. Also, the string abcc can never be created with your implementation. Very close though.

Question 2

regular language is context-free

10 / 15 pts

- ✔ **– 2 pts** proof using NFA: incomplete / missing delta
- ✔ **– 3 pts** proof using DFA: conversion to PDA or CFG is not equivalent or is not general

Question 3

non-cfl

4 / 9 pts

- ✔ **– 1 pt** counterexample must be a specific string (e.g., cannot include something like "indent1")
- ✔ **– 2 pts** counterexample not in the language
- ✔ **– 2 pts** didnt explain why each possible splitting of counterexample cannot be pumped
- in counterexample, did you forget a p on indent3?

Question 4

readme

1 / 1 pt

- ✔ **– 0 pts** Correct

Question assigned to the following page: 1

1. Formal Descriptions for PDA's

1.1)

A formal description of this PDA is a 6 tuple (Q, Σ, Γ, δ, q0, F} specified below:

Q= {q1, q2, q3, q4, q5, q6, q7}

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

$\Gamma = \{a,\ \$\}$

δ is given by the following table, where blank cells signify ∅.

Input:	a					b					c					ε				
Stack:	a	b	c	\$	ε	a	b	c	\$	ε	a	b	c	\$	ε	a	b	c	\$	ε
q1																				$(q_2, \$)$
q2					$\{(q_2, a)\}$															$\{(q_3, \varepsilon), (q_5, \varepsilon)\}$
q3						$\{(q_3, \varepsilon)\}$													$\{(q_4, \varepsilon)\}$	
q4															$\{(q_4, \varepsilon)\}$					
q5										$\{(q_5, \varepsilon)\}$										$\{(q_6, \varepsilon)\}$
q6											$\{(q_6, \varepsilon)\}$								$\{(q_7, \varepsilon)\}$	
q7																				

q0 = q1

F = {q4, q7}

Question assigned to the following page: 1

1.2)

string 1: aabb

(q1, aabb, ε) ⊢ (q2, aabb, $)
⊢ (q2, abb, a$)
⊢ (q2, bb, aa$)
⊢ (q3, bb, aa$)
⊢ (q3, b, a$)
⊢ (q3, ε, $)
⊢ (q4, ε, ε)

string 2: aabcc

(q1, aabcc, ε) ⊢ (q2, aabcc, $)
⊢ (q2, abcc, a$)
⊢ (q2, bcc, aa$)
⊢ (q5, bcc, aa$)
⊢ (q5, cc, aa$)
⊢ (q6, cc, aa$)
⊢ (q6, c, a$)
⊢ (q6, ε, $)
⊢ (q7, ε, ε)

1.3)

string 1: aaa

(q1, baa, ε) ⊢ (q2, aaa, $)
⊢ (q2, aa, a$)
⊢ (q2, a, aa$)
⊢ (q2, ε, aaa$)

→ can go no further than q3 and q6, therefore it is not a string in the language

string 2: ccab

(q1, cbba, ε) ⊢ (q2, cbba, $)

→ can reach accept states, but pda will not have finished reading the input, therefore it is not a string in the language

Question assigned to the following page: 1

1.4)

G → aXc | cC | bB | aFb
X → ε | b | bX | aXc
C → ε | c | cC
B → ε | b | bB
F → ε | aFb

Let P be the cfg that represents this pda.
P is a 4 tuple (V, Σ, R, S), where:
V = {G, X, C, B, F}
Σ = {a, b, c, ε}
R = {
G → aXc | cC | bB | aFb,
X → ε | b | bX | aXc,
C → ε | c | cC,
B → ε | b | bB,
F → ε | aFb
}
S = G

Question assigned to the following page: 2

2) A Regular Language is a Context-Free

proof using dfa

step	statement	justification
1	Every regular language is a context free language	given statement to prove
2	A language is regular if it can be recognized by a dfa	definition of a regular language
3	A context free language can be described by a context free grammar	definition of context free language
4	A dfa M = (Q, Σ, δ, q0, F) recognizes the regular language L	(2)
5	M can be converted into a CFG named X = (V, Σ, R, S) where: V = Q Σ = Σ R = $\delta_q q$, where δ_q is the alphabet from the transition to the next state and q is the next state S = q0, where q0 now has a ε as a terminal as well as its R rule.	computation
6	The regular language L can be described by X, therefore it is a context free language	(5) and (3)

■

proof using nfa

step	statement	justification
1	Every regular language is a context free language	given statement to prove
2	A language is regular if it can be recognized by a nfa	definition of a regular language
3	A context free language can be described by a pda	definition of context free language
4	A nfa N = (Q, Σ, δ, q0, F) recognizes the language L	(2)

Question assigned to the following page: 2

5	N can be converted into a pda named X: $Q = Q$ $\Sigma = \Sigma$ $\Gamma = \varepsilon$ $\delta = Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$ $q0 = q0$ $F = F$	computation and (4)
6	The regular language L can be described by X, therefore it is a context free language	(5) and (3)

■

proof using regular expression

step	statement	justification
1	Every regular language is a context free language	given statement to prove
2	A language is regular if it can be recognized by a regular expression	definition of a regular language
3	A regular expression can be converted into an equivalent nfa	theorem (backwards part): a lang is reg iff some regex describes it
4	A context free language can be described by a pda	definition of context free language
5	A regular expression R recognizes the language L	(2)
6	R can be converted into an nfa $N = (Q, \Sigma, \delta, q0, F)$	(3)
7	N can be converted into a pda named X: $Q = Q$ $\Sigma = \Sigma$ $\Gamma = \varepsilon$ $\delta = Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$ $q0 = q0$ $F = F$	computation
8	The regular language L can be described by X, therefore it is a context free language	(4) and (7)

■

Question assigned to the following page: 3

3) Non-CFL Whitespace Checking

step	statement	justification
1	prove that W is not a CFL W = {w | w = if_b_1: $indent_1 n_1 elif\_$ b_2: $indent_2 n_2 else$: $indent_3 n_3$} where: 1. b1, b2 is in {true, false} 2. n1, n2, n3 are in {0, 1, …, 9} 3. indent1, indent2, indent3 are in _ _* 4. |indent1| = |indent2| = |indent3|	given statement to prove
2	proof by contradiction on W	method of proof
3	assume that W is a CFL	(2)
4	The pumping lemma for CFL says that for a CFL there is a pumping length where any string in the language of length at least p, then that string may be divided into five pieces satisfying three conditions. s = string p = pumping length s = uvxyz → 5 pieces conditions: 1. for i >= 0, uv^ixy^iz in A, 2. |vy| > 0, and 3. |vxy| <= p	definition of pumping lemma for CFL
5	counterexample for assumption: $if\_b_1: indent_1^{\ p} n_1 elif\_b_2: indent_2^{\ p} n_2 else: indent_3 n_3$	(4)
6	pumping the first two indents will always cause the condition of W, |indent1| = |indent2| = |indent3| to be false. This means that no matter how you split it the string will not be in the language, therefore the pumping lemma does not work on W u = if_b_1: v = $indent_1^{\ p}$	(1), (4) and (5)

Question assigned to the following page: 3

	x = $n_1 elif\_b_2$: y = $indent_2^{p}$ z = $n_2 else: indent_3 n_3$	
7	W is not pumpable, which means it is not a CFL	(6)
8	statement 7 and 3 contradict, therefore W is not a CFL due to the contrapositive of the CFL pumping lemma.	(3), (4) and (7)

∎

Question assigned to the following page: 4

README
other students: none
websites/books: class slides, lecture,
https://www.youtube.com/watch?v=pMvvnTHenHE&t=184s for #2 using dfa proof, trying to convert dfa transition function to R for CFG.
hours spent: 5 hours