hw5

Graded

23 Hours, 55 Minutes Late

Student

Giancarlos Marte

Total Points

26 / 30 pts

Question 1

CFG derivations and parse trees

9 / 9 pts

✔ **– 0 pts** Correct

Question 2

design a CFG

10 / 10 pts

✔ **– 0 pts** Correct

Question 3

design a PDA

6 / 10 pts

✔ **– 1.5 pts** PDA still has content left at end

✔ **– 1.5 pts** Used incorrect cfg->pda conversion

✔ **– 1 pt** Combined too many transitions (minor infraction)

You never popped the start state, and your implementation of the algorithm is a bit off. Separating into multiple rules would've made this problem easier for you, because with one rule, you only needed a single push and pop. You seem to be repeating yourself with your pushes and pops

Question 4

readme

1 / 1 pt

✔ **– 0 pts** Correct

Question assigned to the following page: 1

1) Context-free grammars and string derivations

1.1

String 1:

i = 3; if i == 0: print((1, i)) else: print(i + 7)

Derivation of string 1:

step	derivation	rule
1	<STMTS>	start state
2	<STMT>; <STMTS>	<STMTS> → <STMT>; <STMTS>
3	<ID> = <EXPR>; <STMT>	<STMT> → <ID> = <EXPR> <STMTS> → <STMT>
4	i = <NUM>; if <EXPR>: <STMT> else: <STMT>	<ID> → i <EXPR> → <NUM> <STMT> → if <EXPR>: <STMT> else: <STMT>
5	i = 3; if <EXPR> == <EXPR>: print(<EXPR>) else: print(<EXPR>)	<NUM> → 3 <EXPR> → <EXPR> == <EXPR> <STMT> → print(<EXPR>) <STMT> → print(<EXPR>)
6	i = 3; if <ID> == <NUM>: print(<TUP>) else: print(<EXPR> + <EXPR>)	<EXPR> → <NUM> <EXPR> → <ID> <EXPR> → <TUP> <EXPR> → <EXPR> + <EXPR>
7	i = 3; if i == 3: print((<EXPRS>)) else: print(<ID> + <NUM>)	<ID> → i <NUM> → 3 <TUP> → (<EXPRS>) <EXPR> → <NUM> <EXPR> → <ID>
8	i = 3; if i == 3: print((<EXPR>, <EXPRS>)) else: print(i + 7)	<ID> → i <NUM> → 7 <EXPRS> → <EXPR>, <EXPRS>
9	i = 3; if i == 3: print((<NUM>, <EXPR>)) else: print(i + 7)	<EXPRS> → <EXPR> <EXPR> → <NUM>

Question assigned to the following page: 1

10	i = 3; if i == 3: print((1, <ID>)) else: print(i + 7)	<EXPR> → <ID> <NUM> → 1
11	i = 3; if i == 3: print((1, i)) else: print(i + 7)	<ID> → i <NUM> → 1

String 2:

x = 4; y = lambda(z): z + 10; print(y(x))

Derivation of string 2:

step	derivation	rule
1	<STMTS>	start state
2	<STMT>; <STMTS>	<STMTS> → <STMT>; <STMTS>
3	<ID> = <EXPR>; <STMT>; <STMTS>	<STMT> → <ID> = <EXPR> <STMTS> → <STMT>; <STMTS>
4	x = <NUM>; <ID> = <EXPR>; <STMT>	<ID> → x <EXPR> → <NUM> <STMT> → <ID> = <EXPR> <STMTS> → <STMT>
5	x = 4; y = lambda(<IDS>) : <EXPR>; print(<EXPR>)	<NUM> → 4 <ID> → y <EXPR> → lambda(<IDS>) : <EXPR> <STMT> → print(<EXPR>)
6	x = 4; y = lambda(<ID>) : <EXPR> + <EXPR>; print(<ID> (<EXPR>))	<IDS> → <ID> <EXPR> → <EXPR> + <EXPR> <EXPR> → <ID> (<EXPR>)
7	x = 4; y = lambda(z) : <ID> + <NUM>; print(y (<ID>))	<ID> → z <EXPR> → <ID> <EXPR> → <NUM> <ID> → y <EXPR> → <ID>

8	x = 4;
	y = lambda(z) : z + 10;
	print(y (x))
		<ID> → z
		<NUM> → 10
		<ID> → x

Question assigned to the following page: 1

1.2

Formal description of the grammar PY

A context free grammar for PY is a 4-tuple (V, Σ, R, S), where:

V = {<STMTS>, <STMT>, <EXPRS>, <EXPR>, <TUP>, <NUM>, <IDS>, <ID>}

Σ = {;, =, if, :, else, print, (,), ε, ==, +, lambda, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}

R = RULES, where RULES is a set with the given rules

S = <STMTS>

1.3a

String:

if x == 9: print((a, b, c)) else: print(0)

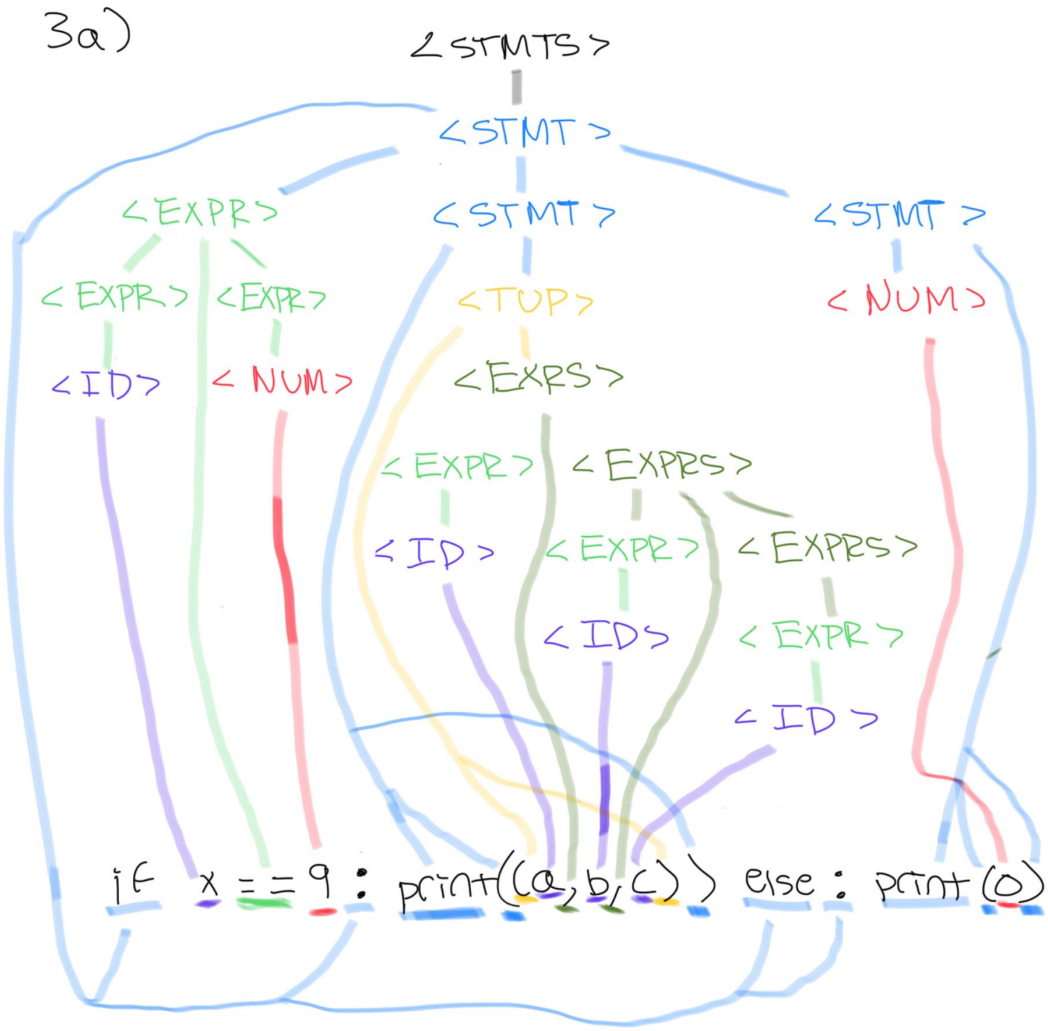
Derivation:

step	derivation	rule
1	<STMTS>	start state
2	<STMT>	<STMTS> → <STMT>
3	if <EXPR> : <STMT> else: <STMT>	<STMT> → if <EXPR> : <STMT> else: <STMT>
4	if <EXPR> == <EXPR>: print(<EXPR>) else: print(<EXPR>)	<EXPR> → <EXPR> == <EXPR> <STMT> → print(<EXPR>) <STMT> → print(<EXPR>)
5	if <ID> == <NUM>: print(<TUP>) else: print(<NUM>)	<EXPR> → <ID> <EXPR> → <NUM> <EXPR> → <TUP> <EXPR> → <NUM>
6	if x == 9: print((<EXPRS>)) else: print(0)	<ID> → x <NUM> → 9 <TUP> → (<EXPRS>) <NUM> → 0
7	if x == 9: print((<EXPR>, <EXPRS>)) else: print(0)	<EXPRS> → <EXPR>, <EXPRS>
8	if x == 9: print((<ID>, <EXPR>, <EXPRS>)) else: print(0)	<EXPR> → <ID> <EXPRS> → <EXPR>, <EXPRS>

Question assigned to the following page: 1

9	if x == 9: print((a, <ID>, <EXPR>)) else: print(0)	<ID> → a <EXPR> → <ID> <EXPRS> → <EXPR>
10	if x == 9: print((a, b, <ID>)) else: print(0)	<ID> → b <EXPR> → <ID>
11	if x == 9: print((a, b, c)) else: print(0)	<ID> → c

Parse tree:



Question assigned to the following page: 1

1.3b.

String: a = 1; f = lambda(x): x + a; f(5)

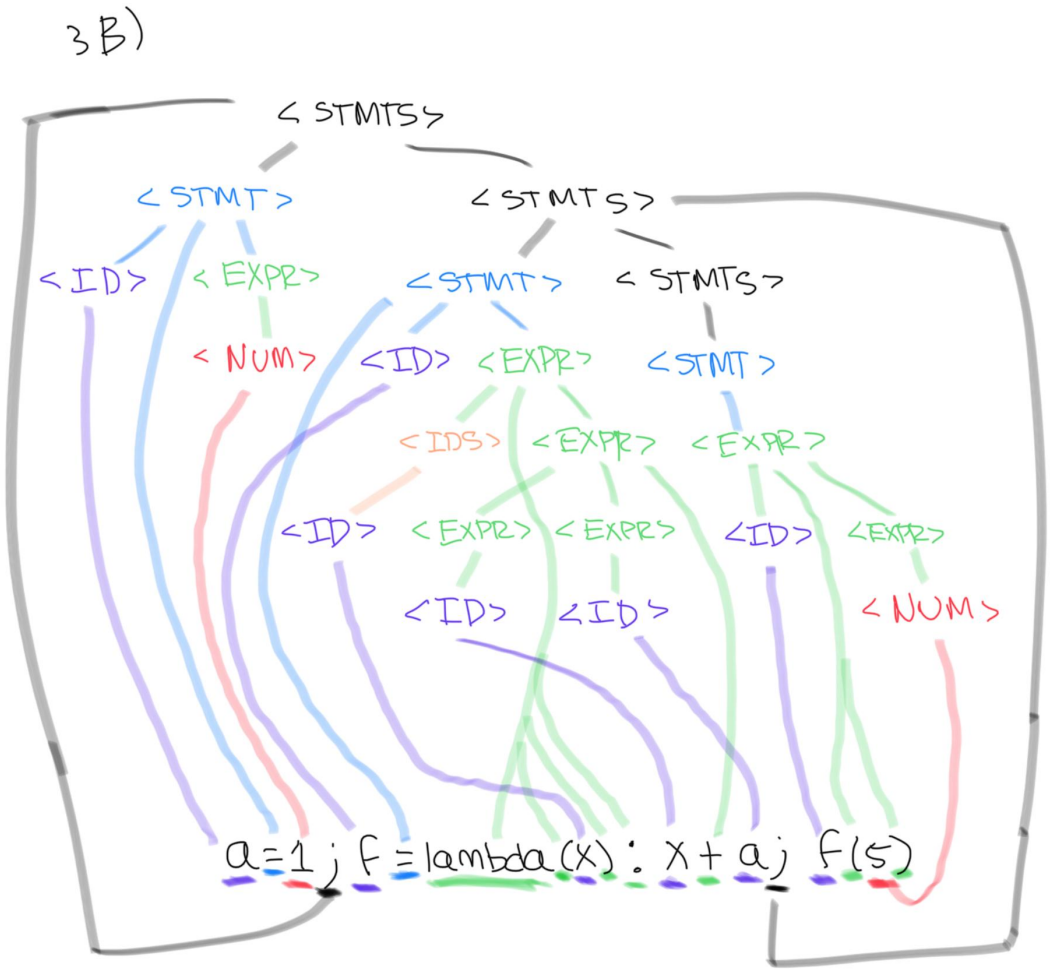
Derivation:

step	derivation	rule
1	<STMTS>	start state
2	<STMT>; <STMTS>	<STMTS> → <STMT>; <STMTS>
3	<ID> = <EXPR>; <STMT>; <STMTS>	<STMT> → <ID> = <EXPR> <STMTS> → <STMT>; <STMTS>
4	a = <NUM>; <ID> = <EXPR>; <STMT>	<ID> → a <EXPR> → <NUM> <STMT> → <ID> = <EXPR> <STMTS> → <STMT>
5	a = 1; f = lambda(<IDS>) : <EXPR>; <EXPR>	<NUM> → 1 <ID> → f <EXPR> → lambda(<IDS>) : <EXPR> <STMT> → <EXPR>
6	a = 1; f = lambda(<ID>) : <EXPR> + <EXPR>; <ID> (<EXPR>)	<IDS> → <ID> <EXPR> → <EXPR> + <EXPR> <EXPR> → <ID> (<EXPR>)
7	y = 1; f = lambda(x) : <ID> + <ID>; f (<NUM>)	<ID> → x <EXPR> → <ID> <EXPR> → <ID> <ID> → f <EXPR> → <NUM>

8	y = 1;
	f = lambda(x) : x + a;
	f (5)
		<ID> → x
		<ID> → a
		<NUM> → 5

Question assigned to the following page: 1

Parse tree:


3B)
<STMTS>
<STMT>
<STMTS>
<ID>
<EXPR>
<STMT>
<STMTS>
<NUM>
<ID>
<EXPR>
<STMT>
<IDS>
<EXPR>
<EXPR>
<ID>
<EXPR>
<EXPR>
<ID>
<EXPR>
<ID>
<ID>
<NUM>
a=1; f=lambda(x): x+a; f(5)

Question assigned to the following page: 2

2) Design a CFG, including whitespace?

step	statement	justification
1	prove that the following language B is a CFL: B = {w | if_*var*==*num*:$indent_1 num_1$else:$indent_2 num_2$} B is a language over the alphabet Σ = {a, …, z, 0, 1…, 9, _, =, :}	given
2	A language is a CFL if a CFG describes it.	theorem and given
3	A CFG is a 4-tuple (V, Σ, R, S), where V represents a set of variables, Σ is a set of terminals, R is a set of rules and S is the start variable.	definition of a CFG
4	V = {var, num, num_1, num_2, $indent_1$, $indent_2$}	(3) and (1)
5	Σ = {a, b, …, z, 0, 1, …, 9, _, =, :}	(1) and (3)

6	R = {
	start → if_var==num:$indent_1 num_1$else:$indent_2 num_2$,
	var → a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z,
	num→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9,
	num_1→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9,
	num_2→ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9,
	$indent_1$→ _ _*,
	$indent_2$→ $indent_1$
	}
		(4), (5) and (3)

7	S = start	(3) and (6)
8	A CFG (V, Σ, R, S) describes the language B, therefore it is a CFL.	(4), (5), (6) and (7)

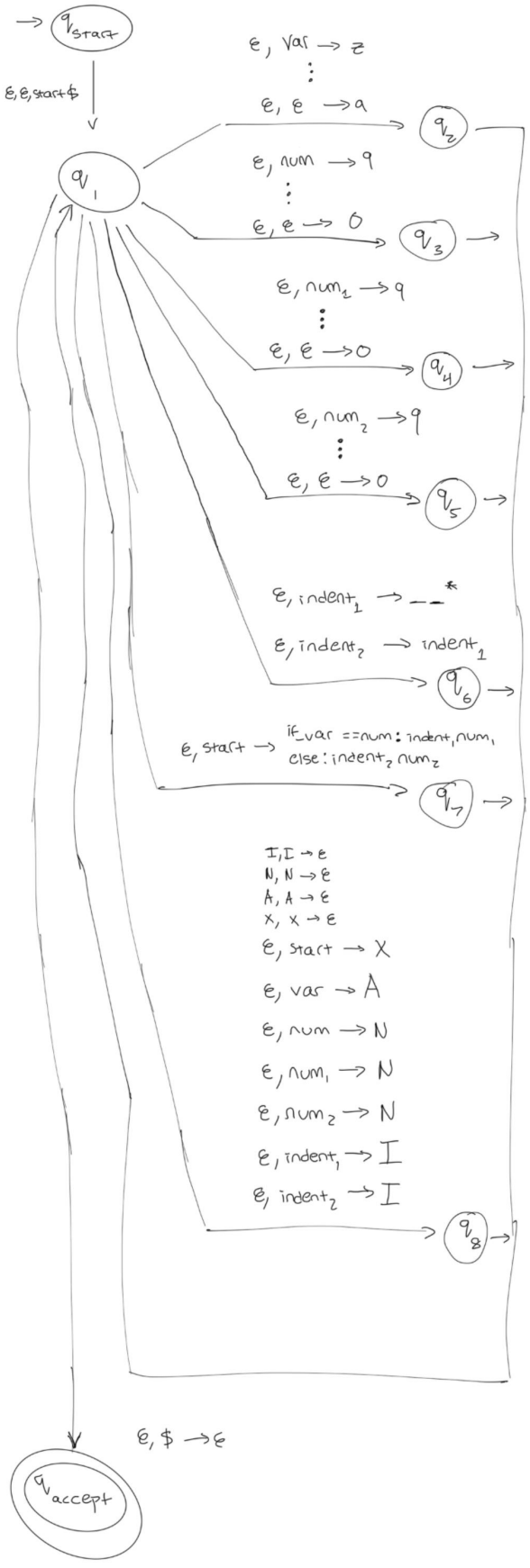
Question assigned to the following page: 3

3) Design of a PDA, including whitespace?

Let:
X = if_var==$num{:}indent_1 num_1$else:$indent_2 num_2$,
A = a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z,
N = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9,
I = _ _*

(graph is on the next page)

Question assigned to the following page: 3


q_{start}
ε, ε, start$
q_1
ε, var → z
ε, ε → a
q_2
ε, num → 9
ε, ε → 0
q_3
ε, num_1 → 9
ε, ε → 0
q_4
ε, num_2 → 9
ε, ε → 0
q_5
ε, $indent_1$ → __*
ε, $indent_2$ → $indent_1$
q_6
ε, start → if_var ==num: $indent_1$ num_1
else: $indent_2$ num_2
q_7
I, I → ε
N, N → ε
A, A → ε
X, X → ε
ε, start → X
ε, var → A
ε, num → N
ε, num_1 → N
ε, num_2 → N
ε, $indent_1$ → I
ε, $indent_2$ → I
q_8
ε, $ → ε
q_{accept}

Question assigned to the following page: 4

README
names of others: none
books/websites used: class slides
time spent: 5 hours