Started on	Monday, 8 November 2021, 2:34 PM
State	Finished
Completed on	Monday, 8 November 2021, 4:00 PM
Time taken	1 hour 25 mins
Grade	Not yet graded

Information

Curly brackets are taken care in programming languages. It checks with opening and closing of braces. When there is a mismatch in opening and closing of curly braces the compiler checks for it and throws an error if there is missing parenthesis or can also be said to be as unbalanced parentheses. Thus for every statement in the programming have its own definitions in Context Free Grammars (CFGs). Based on the concept of CFG used in the above scenario answer the below questions.

Question 1
Complete
Marked out of 4

Let L = language of binary palindromes s.t., substitutions for 0 and 1 are defined as follows:
$s(0) = \{a^n b^n \mid n \geq 1\}, s(1) = \{xx, yy\}$

Prove that s(L) is also a CFL

CFG for L:

S=> 0S0|1S1|ε

CFG for s(0):

S0=> aS0b | ab

CFG for s(1):

S1=> xx | yy

Therefore, CFG for s(L):

S=> S0SS0 | S1 S S1 |ε

S0=> aS0b | ab

S1=> xx | yy

Question 2
Complete
Marked out of 6

Show that the language L={$a^n b^n c^n$ | n>=1} is not a CFL.

Let L is context free. Then, L must satisfy pumping lemma.

At first, choose a number n of the pumping lemma. Then, take z as 0n1n2n.

Break z into uvwxy, where

|vwx| ≤ n and vx ≠ ε.

Hence vwx cannot involve both 0s and 2s, since the last 0 and the first 2 are at least (n+1) positions apart. There are two cases −

Case 1 − vwx has no 2s. Then vx has only 0s and 1s. Then uwy, which would have to be in L, has n 2s, but fewer than n 0s or 1s.

Case 2 − vwx has no 0s.

Here contradiction occurs.

Hence, L is not a context-free language.

Information

Assume you're looking for the result of an arithmetic statement on your Smartphone calculator.
To evaluate it, it employs a push down automata implementation.

- How will it be able to determine which of the computations should be completed first?
- How does your Smartphone assess the nested expression?

It's all predicated on the operators' order of precedence. The Push down Automata uses the Context-Free Grammar to verify this precedence, which is known as syntax.

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T*F \mid T/F \mid F$

$F \rightarrow x \mid y \mid (E) \mid -F$

The variables E (Expression), T (Term) and F (Factor) produce more complex expressions or terminal symbols (numbers) in an arithmetic expression

Question 3

Complete

Marked out of 3

“Pushdown Automata can act as a comparator between two entities only” - Justify this statement with reasons

“Pushdown Automata can act as a comparator between two entities only” -

Reason:

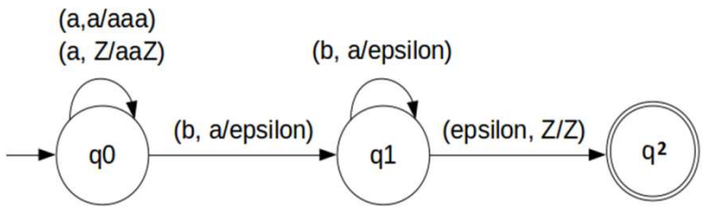
- Push-down automata have limited memory.
- Since it has only a single stack, comparison can be made only between 2 Entities.
- Stack is the data structures used by pushdown automata.
- Since, push and pop are the only two operations performed in stack,comparison is done only between two entities
- Therefore, pushdown automata cannot act a s a comparator for more than two entities.

eg:

for eg: {an bn| n>=0} can be done using PDA where single comparison is used but {an bn cn| n>=0} cannot be solved using PDA where more than 1 comparison is used , so PDA can acts as a comparator between two entities alone

Question **4**
Complete
Marked out of 4

Consider the transition diagram for a PDA given below:



i) Based on the above transition diagram, Complete the input string verification process in the following table and verify whether the input string is accepted or not by the PDA

State	Input	Stack	Transition used
q0	aabbbb	z	
q0	aabbbb	aaz	δ(q0, a, Z) = (q0, aaZ)
q0	...	...	...
...	...	...	...
...	...	...	...
			Accepted/Rejected

ii) Identify the language accepted by the above PDA

- **STATE input stack transition**
  q0 aabbbb z

STATE	**input**	**stack**	**transition**
q0	**aabbbb**	**aaz**	**δ (q0,a,Z)=(q0,aaZ**)
q0	abbbb	aaa	δ (q0,a,a)=(q0,aaa)
q0	bbbb	ε	δ(q0,b,a)=(q1,ε)
q1	bbb	ε	δ(q1,b,a)=(q1,ε)
q1	bb	ε	δ (q1,b,a)=(q1,ε)
q1	b	ε	δ (q1,b,a)=(q1,ε)
q1	ε	Z	**Accepted/ Rejected**

Context free language (CFL).

Question 5
Complete
Marked out of 3

Consider the CFG G:

S à0S1 | A A à 1A0 | ε

Assume that the above CFG is converted into its equivalent PDA transitions.

Complete the missing entries for the following symbols involved in this grammar:

1) δ(q, ε, S) = ----------

2) δ(q, ε, A) = { (q,1A0), (q, ε)}

3) δ(q, 0,0) = ----------

4) δ(q, 1,1) = ----------

1. **δ(q, ε, S) = {(q, 0SX) | (q, 1SY) | (q, ε)} (or) {(q,0S1), (q, ε)}**
2. δ(q, ε, A) = { (q,1A0), (q, ε)}
3. **δ(q, 0, 0) = {(q, ε)}**
4. **δ(q, 1, 1) = {(q, ε)}**

Question **6**

Complete

Marked out of 3

There are almost three types of machines in Theory of Computation, Finite Automata, Pushdown Automata and Turing Machines. Which one is the most powerful machine? Justify your answer.

Turing Machine. It manipulates symbols on a strip of tape which has higher storage when compared to Finite or Pushdown Automata.

Question 7
Complete
Marked out of 4

Consider the transition table of a Turing machine M.

Assume that the input and output are in unary representation

State	Input (0)	Input (Blank - B)
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$
q_1		$(q_2, 0, R)$
q_2		Halt

i) Trace the acceptance of the input w = 000BBB as per the above transition table

ii) Identify the function of the Turing machine M.

i) At the end of the computation, the final result will be 00000B

(q0, 0, R) - 000BBB No changes are performed since it is zero
(q0, 0, R) - 000BBB No changes are performed since it is zero
(q0, 0, R) - 000BBB No changes are performed since it is zero
(q1, 0, R) - 000BBB will be changed as 0000BB
(q2, 0, R) - 0000BB will be changed as 00000B
Halt

ii) The function of above Turing Machine is to add 2 to any given number.

For example when the number stored in the tape is 3 (000), then 2 will be added to it and the final result becomes 5 (00000).

Question **8**

Complete

Marked out of 3

Match the Following:

Automata	Languages	Examples
1. Finite Automata	(i) Context Free Languages	(A) L = {$a^n b^n c^n$ | n≥ 1}
2. Push Down Automata	(ii) Recursively Enumerable languages	(B) L = { w | w starts with 10 over {0,1}}
3. Turing Machine	(iii) Regular Languages	(C) L = {wcw^R | w Є{a,b}*}

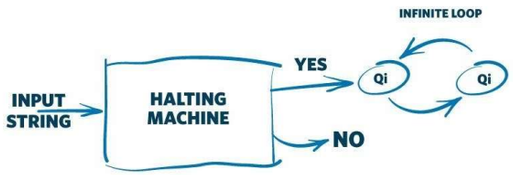
1-iii-b

2-i-c

3-ii-a

Information

Consider the scenario of the undecidable halting problem. The halting problem is a decision problem about properties of computer programs on a fixed Turing-complete model of computation, i.e., all programs that can be written in some given programming language that is general enough to be equivalent to a Turing machine. The problem is to determine, given a program and an input to the program, whether the program will eventually halt when run with that input.



In this abstract framework, there is no resource limitations on the amount of memory or time required for the program's execution; it can take arbitrarily long and use an arbitrary amount of storage space before halting. The question is simply whether the given program will ever halt on a particular input. Based on this scenario answer the following questions.

Question 9
Complete
Marked out of 4

"Recursive languages are those in which TMs that *always* halt, no matter accepting or non-accepting" – Justify the statement with an example.

Halting TM : (Accepts Recursive languages) : TMs that always halt, no matter accepting or non no matter accepting or non-accepting (called as decidable problems)

TM : (Accepts Recursively enumerable): TMs that are guaranteed to halt are guaranteed to halt only on acceptance only on acceptance. If non-accepting, it may or may not halt (i.e., could loop forever). (Either decidable or partially decidable)

Decidable Problem

If there is a Turing machine that decides the problem, called as **Decidable problem.**

A decision problem can be solved by an algorithm that halts on all inputs in a finite number of steps.

A problem is decidable if there is an algorithm that can answer either yes or no.

A language for which membership can be decided by an algorithm that halts on all inputs in a finite number of steps.

Decidable problem is also called a totally decidable problem, algorithmically solvable, recursively solvable.

Undecidable Problem:

A problem that cannot be solved for all cases by any algorithm whatsoever.

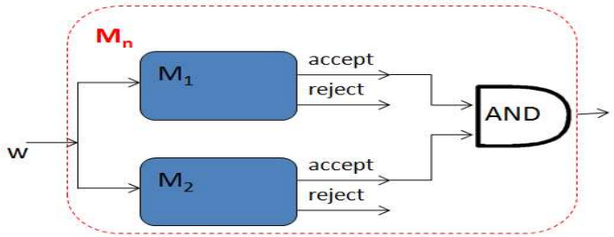
Equivalent Language cannot be recognized by a Turing machine that halts for all inputs.

The following problems are undecidable problems:

- Halting Problem
- Emptyness Problem
- Finiteness Problem
- Equivalence Problem

Question 10
Complete
Marked out of 2

Consider the following Turing Machine



Identify the closure property that is depicted in the above diagram.

The closure property in **INTERSECTION** where it uses and operation between two machines.

Question 11

Complete

Marked out of 2

Spot the error statements regarding PCP with respect to the tiling problem from the following:

PCP is an undecidable problem

PCP is a decidable problem

PCP is a deterministic problem

PCP is a non-deterministic problem

The error statements are:

1. PCP is a decidable problem
2. PCP is a deterministic problem

Question 12
Complete
Marked out of 2

Assume that the top and bottom side of the tile is represented as List A and List B.

List A ={a, c, ba, acb}
List B = { ac, ba, a,b}.

Identify the instance order for obtaining the string“ acbaaacb “ as one of the solution using PCP

12314

Question 13

Complete

Marked out of 2

"Post Correspondence Problem is decidable over the unary alphabet Σ={1}" – Justify the above statement with reasons.

- The **comparison takes a lot of time** and may be infinite. To **reduce the time complexity** of the comparisons we start with the first.
- We will count the number of 1s in the top strings (t) and the number of 1s in the bottom string (b).
- If t = b, we accept. Any way we place the pairs, it will always be that the top is the same as the bottom.
- If t != b, we reject.

Question **14**

Complete

Marked out of 2

Assume that the top and bottom side of the tile is represented as Lists.

Let

List A= {001,0011,11,101}

List B= {01,111,111,010}

Similarly,

Let List C={00,001,1000}

List D= {0,11,011}

Which one of the above pairs has a post correspondence solution?

Only pair (A, B)

List A= {001,0011,11,101}

List B= {01,111,111,010} has a PCP solution

Question 15
Complete
Marked out of 4

Check whether the lists

M = (abb, aa, aaa) and N = (bba, aaa, aa)

have a Post Correspondence Solution? If yes, derive the solution

Yes.

Here,

x2x1x3 = 'aaabbaaa'

and
y2y1y3 = 'aaabbaaa'

We can see that

x2x1x3 = y2y1y3

Hence, the solution is **i = 2, j = 1, and k = 3.**

Question 16
Complete
Marked out of 2

Which one of the following problem(s) is undecidable?

Deciding if a given string is generated by a given context-free grammar.

Deciding if the language generated by a given context-free grammar is finite.

Deciding if a given context-free grammar is ambiguous

Deciding if the language generated by a given context-free grammar is empty.

Deciding if a given **context-free grammar (CFG)** is ambiguous.

A context-free grammar is not closed under ambiguity .A set is closed under an operation means when we operate an element of that set with that operator we get an element from that set.

Here, context-free grammar generates a context-free language and set of all context-free languages is also a set. But, ambiguity is not an operation and hence we can never say that CFG is closed under ambiguity.

« PREVIOUS ACTIVITY
18CS501 -THEORY OF COMPUTATION - PT1 - 20.09.2021