GATE

CS & IT

Theory of Computation

DPP: 1

Turing Machine & Undecidability

Q1 Consider the following two regular expressions R_1 and R_2 over $\Sigma = \{a,b\}$.
$R_1 = a^* (ba^*)^*$
$R_2 = (a^* + b^*)^*$
Which of the following is true?
(A) $R_1 \subset R_2$
(B) $R_2 \subset R_1$
(C) $R_1 = R_2$
(D) $R_1 \cap R_2 = a^*b^*$

Q2 Consider the following statements:
I. A regular grammar can generate some CFL language.
II. There are some non-regular languages for which PDA not exist.
III. Context sensitive grammar exists for every CFL.
IV. If L is a regular and M is not a regular language then L.M is necessarily not regular
Number of correct statement is/are______?

Q3 Which of the following languages are not decidable?
(A) L = {<G>} | G is CFG and R is a regular set such that L (G) = R}
(B) L = {<G>} | G is CFG such that $L(G) = \phi$}
(C) L = {<G_1, G_2 >} | G_i is CFG such that $L(G_1) \cap L(G_2) = \sum^*$ }
(D) L = {< G_1, G_2>} | G_i is CFG such that $L(G_1) = L(G_2)$}

Q4 Consider the following problems:
P_1: Does a given program ever produce an output?
P_2: Given DFAs D_1 and D_2 is $L(D_1) \cap L(D_2) = \phi$?
P_3: For Context sensitive grammar G and string w, is $w \in L(G)$?
How many languages is/are undecidable? ______.

Q5 If $L_1 \cap L_2$ is regular and L_1 is non-regular then, L_2 must be
(A) Regular but not finite
(B) Non-regular
(C) Finite
(D) None of these

Q6 If P_1 is reducible to P_2 then, which of the following is/are correct?
(A) If P_1 is decidable then, P_2 is undecidable.
(B) If P_2 is undecidable then, P_1 is decidable.
(C) If P_2 is decidable then, P_1 must be decidable.
(D) If P_1 is undecidable then, P_2 must be undecidable.

Q7 Let,
L_1 = CFL
L_2 = DCFL
L_3 = Regular
Then, which of the following is/are correct?
(A) $L_3 - L_1$ is CSL
(B) $L_1 \cup (L_2 \cap \overline{L}_3)$ is CFL
(C) $L_2 \cdot L_3$ is Regular
(D) $L_2 \cup L_3$ is DCFL

Q8 Consider the following statements:
S_1: Complement of finite language may be finite.
S_2: Kleene star of finite language may be finite.
S_3: Subset of finite language is always finite.

S_4: Complement of infinite language may be finite.

Number of incorrect statements is/are______.

Q9 Which of the following problems are undecidable?

(A) Membership problem in context-free languages.

(B) Whether a given context-free language is regular.

(C) Whether a finite state automation halts on all inputs.

(D) Membership problem for type 0 languages.

Q10 Which one of the following is **UNDECIDABLE**?

(A) Given a Turing machine M, a string s and an integer k, M accepts s within k steps

(B) Equivalence of two given Turing machines

(C) Language accepted by a given finite state machine is not empty

(D) Language generated by a context free grammar is non empty

Q11 Consider the following decision problems:

(P_1): Does a given finite state machine accept a given string?

(P_2): Does a given context free grammar generate an infinite number of strings

Which of the following statements is true?

(A) Both (P_1) and (P_2) are decidable

(B) Neither (P_1) nor (P_2) are decidable

(C) Only (P_1) is decidable

(D) Only (P_2) is decidable

Q12 Which of the following problems is undecidable?

(A) Membership problem for Context free grammar.

(B) Ambiguity problem for Context free grammar.

(C) Finiteness problem for Finite automaton.

(D) Equivalence problem for Finite automaton.

Q13 Recursive languages are:

(A) a proper superset of context free languages

(B) always recognized by a pushdown automata

(C) also called type-0 languages

(D) recognizable by Turing machines

Answer Key

Q1	(C)
Q2	3~3
Q3	(A, C, D)
Q4	1~1
Q5	(D)
Q6	(C, D)
Q7	(A, B, D)
Q8	1~1
Q9	(B, D)
Q10	(B)
Q11	(A)
Q12	(B)
Q13	(A, D)

Hints & Solutions

Q1 Text Solution:

$L(R_1) = a^*(ba^*)^*$

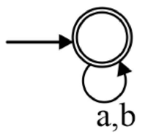
$= \{\epsilon, a, b, aa, ab, ba, bb, \ldots\ldots\}$

$L(R_1) = (a+b)^*$

$L(R_2) = (a^* + b^*)^*$

$= \{\epsilon, a, b, aa, ab, ba, bb\ldots\}$

DFA for R_2:



$= L(R_2) = (a+b)^*$

So, both languages are equivalent.

Q2 Text Solution:

- Every regular is CFL. So, regular can generate some CFL but not all CFL **(TRUE)**
- $L = \{a^n b^n c^n \mid n \geq 0\}$ it is non- regular language, for L we cannot generate PDA. **(TRUE)**
- CSG can generate all CFG and regular. Because Every CFG and regular are CSG. **(TRUE)**
- $L = \phi$ and $M = \{a^n b^n \mid n \geq 1\}$

$= L \times M$

$= \phi \times \{a^n b^n \mid n \geq 1\}$

$= \phi$ Regular **(FALSE)**

Q3 Text Solution:

(b) Emptiness Problem for CFG is decidable.
Equivalence and completeness problems are Undecidable for CFG.

Q4 Text Solution:

- Problem P_1 is non-trivial so, P_1 is undecidable.
- Disjointness problem for DFA/NFA are decidable.
- Membership problem for context sensitive grammar is decidable.

Q5 Text Solution:

Given $L_1 \cap L_2$ = Regular

L_1 = Non - regular

Then, L_2 need not be regular

So, option (a, b, c) are false

Therefore, option (d) is correct.

Q6 Text Solution:

Given $P_1 \leq P_2$

- If P_1 is decidable then P_2 may/may not be decidable.
- If P_2 is undecidable then P_1 may/may not be undecidable.
- If P_2 is decidable then, P_1 must be decidable.
- If P_1 is undecidable then, P_2 must be undecidable.

Q7 Text Solution:

(a) CFL Regular $\cup$ $\overline{\text{CFL}}$

Regular $\cap$ CSL

= CSL **(True)**

(b) CFL $\cup$ (DCFL $\cap$ $\overline{\text{Regular}}$)

CFL $\cup$ DCFL

= CFL **(True)**

(c) DCFL $\cdot$ Regular

Need not be Regular **(False)**

(d) DCFL $\cup$ Regular = DCFL **(True)**

Q8 Text Solution:

- $\overline{\text{Finite}}$ = Always Infinite

Incorrect

- $\phi^* = \epsilon$ **Correct**
- Subset of finite language always finite.

Correct

- $\overline{(a+b)^*} = \phi$

Correct

Q9 Text Solution:

Option (A), Membership problem in context-free languages. is **DECIDABLE.**

Option (B), Whether a given context-free language is regular is **undecidable.** *Regularity is*

decidable till DCFL.
Option (C),Whether a finite state automation halts on all inputs. **DECIDABLE**
Option (D), Membership problem for type 0 languages. **undecidable** (undecidable for RE or semidicidable)

Q10 Text Solution:

Option (A) is **DECIDABLE**; since numbers of steps are finite . After k steps we can check whether final state reached or not. If we reach final state input accepted otherwise rejected.

Option (B) is **undecidable;** to check for equivalence of two Turing Machine we need to check

$$(L(G_1) - L(G_2)) \cup (L(G_2) - L(G_1)) = \emptyset$$

We can't check emptiness for Turing Machine.

Option (C) is **DECIDABLE**; emptiness of the finite state machine can be checked by reachability of the final state from initial state in transition graph. If the final state is reachable then non empty otherwise empty.

Option (D) is **DECIDABLE;** there is an algorithm to find emptiness by checking usefulness of start symbol of the grammar. If the start symbol is useful then grammar generated the language non-empty otherwise empty.

Q11 Text Solution:

P_1 is **DECIDABLE**; we can simply input a string to finite state machine and after processing the string we can check the state of the machine. If the machine reaches the final state then accepted otherwise rejected.

P_2 is **DECIDABLE**; to test whether L(G) is finite, we use following algorithm

I. Convert context free grammar G' = (V', T, P', S) in **Chomsky Normal Form** and with no useless symbols. II. A simple test for finiteness of a CNF grammar with no useless symbols is to draw a directed graph with a vertex for each variable and an edge from A to B. There is a production of the form $A \to BC$ or $A \to CB$ for any C. Then the language generated is finite if and only if this graph has no cycles.

Example

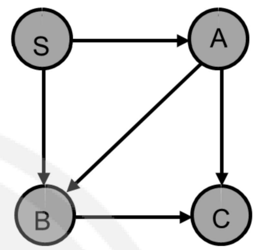
Consider the following grammar

$S \to AB$

$A \to BC|a$

$B \to CC|b$

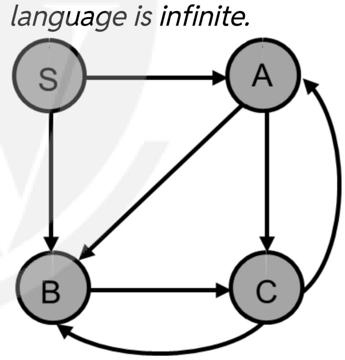
$C \to a$



The corresponding graph is, this graph has no cycle, and hence the language is finite.

If we add a production rule C→AB,

we get the graph containing cycle and hence language is infinite.



Q12 Text Solution:

Option (A); Membership problem for CFGs is **DECIDABLE**; we can use CYK algorithm to test membership of a string in $L(G)$.

Option (B); Ambiguity problem for CFGs is **UNDECIDABLE**. Post correspondence problem (PCP) problem reduces to ambiguity problem.

Option (C); Finiteness problem for FSAs is **DECIDABLE**; we can test for loop in transition

graph.
Option (D); Equivalence problem for FSAs is **DECIDABLE**; regular languages are closed under complement and intersection and hence; $(L(G_1) - L(G_2)) \cup (L(G_2) - L(G_1))$ will be a regular language and can be tested for the emptiness.

Q13 Text Solution:

Option (A) is **TRUE;** In Chomsky hierarchy recursive language includes context free languages.
Option (B) is **FALSE**; Recursive languages are recognized by halting Turing machine.
Option (C) is **FALSE;** type-0 languages also include recursive enumerable languages.
Option (D) is **TRUE**; Recursive languages are recognize as well as decides by the Turing machines.