28. Gauri Visputi
TCS, 5th sem

Assignment 1

Q.1) Define the following terms:

i) Undecidability:
A problem is said to be decidable if there exist a Turning Machine that gives the correct answer for every statement in the domain of the problem.

ii Otherwise, the class of problems is said to be un-decidable.

iii These two statement are equivalent:
a 1. A class of problem is un-decidable.
2. A class of problem is un-solvable.

iv) A language can be proved to be undecidable through a method of reduction.

8. 1 To show that a problem A is un-decidable, It must reduce another problem that is known to be undecidable ~~to~~A.

2. Having proved that the halting problem is undecidable. It is use problem reduction to show that other problem.

v First of all, we will provide proof for the undecidability of some standard problems. Subsequently, these problems will be used to show that other problems are undecidable. Some standard un-decidable problems are:

1. Halting problem of a Turing machine.
2. Digonalization language.
3. The post correspondance problem.
4. The Universal langudge.

(ii) Pumping lemma.

1. In pumping lemma we use pigeon Hole principle.
2. If languge 'L' is infinite regular language then their exist some positive integer 'n' such that, any string $w \in L$ whose length is m or greater than m can be decompose'd into three ~~must~~ ~~Conditions~~ parts: $w = xyz$ where.
3. Their are three must conditions:
   i) The length of $|xy|$ is less than or equal to m.
   ii) $|y| > 0$
   iii) $w^i = xy^iz$ is also in L for all $i = 0, 1, 2, 3 \ldots$
4. It is use to prove that given set is not regular.

Q.2) Define TM and give its variants.

Ans i) A Turning machine. M is a 7-tuple given by

$$M = (Q, \Sigma, \Gamma, \delta, \square, q_0, F)$$

where,

1. Q is finite set of states.
2. Σ is finite set of input.
3. Γ is a finite set of symbol (the tape alphabet).
4. ~~$q_0 \in Q$ is the initial symbol.~~
4. δ is the transition function.
   $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L \times R\}$.
5. $\square \in \Gamma$ is a special symbol called blank.
   $\Sigma = \Gamma - \{\square\}$
6. q_0 is the initial state.
7. F is the set of final (accepting) states.

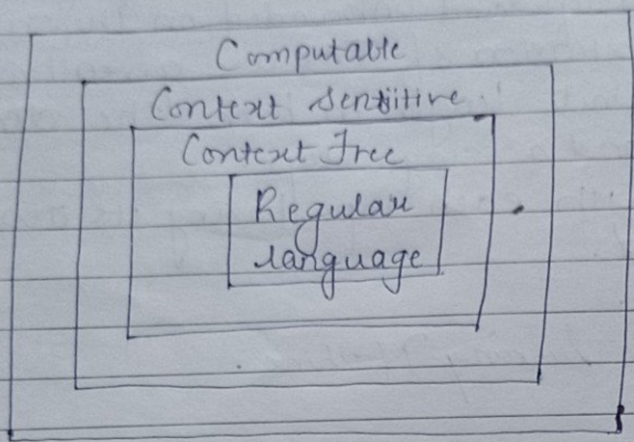
In a standard Turing machine, the tap is semi-infinite. It is bounded on left and unbounded on the right side.
Some of the extension of TM are given below.

1. Tape is of infinite length in both the directions.
2. Multiple heads and a single tape.
3. Multiple tapes with each tape having its own independente head.
4. k-dimension tape.
5. Non-deterministic turning Machine.

Q.3. Explain Chomsky Hierarchy for formal languages.

Ans Chomsky Hierarchy represents the class of language that are excepted by different machine.

language Class	Language	Grammer	Machines	Example
Type 3	Regular	Regular	FSM, DFA/NFA	a^*b^*
Type 2	Context free	Context free	PDA	a^nb^n
Type 1	decidable	Context Sentsitive	Linear bounded automaton	$a^nb^nc^n$
Type 0	Computable	Unrestricted	TM	all



Every language of type 3 is also type 2, 1, 0
Every language of type 2 is also type 1, 0

Q.4 What is Myhill Nerode Theorem? Explain necessity of it?

Ans i) Given a language L, two strings x and y are said to be in the same class if for all possible strings z either both ~~are~~ ~~xy~~ xz and yz are in L or both are not.

ii) The Myhill-Nerode theorem says:

1. A language L divides the set of all possible strings into mutually exclusive classes.
2. If L is regular, the number of classes created by L is finite.
3. If the number of classes L, creates is finite, then L is regular.

iii) In finite automata, each state can be thought of as creating a class of strings. Two strings are said to be in the same class if they both trace a path from starting state q_0 to some state q_i (say)

Q.5) Explain CNF and GNF with example?

Ans. Definition of CNF

The CFG without ϵ production is said to be Chomsky Normal Form. If every production is of the form.

$$A \rightarrow BC$$
$$A \rightarrow a$$

where $A, B, C \in \{V\}$

$A \in \{V\}$

$a \in \{T\}$

Find the CNF Example of CNF.

$S \rightarrow aAbB$

$A \rightarrow aA$

$B \rightarrow bB \mid b$

Step 1: Simplification of Grammar.

i) ϵ production.

ii) Unit production

iii) Useless production.

$\therefore$ The given grammar is with three production.

Step 2: Every symbol in α in the production of the form, $A \rightarrow \alpha$ where $|\alpha| \geq 2$ should be a variable.

This can be achieved with the help of by adding two production.

$$C_a \rightarrow a$$
$$C_b \rightarrow b$$

Then the production will be

$S \to C_a A C_b B$

$A \to C_a A$

$B \to C_b B \mid b$

$C_a \to a$

$C_b \to b$

where $S \to C_a A C_b B$

so, $S \to C_a C_1 \quad \{C_1 \to A C_b B\}$.

$C_1 \to A C_2 \quad \{C_2 \to C_b B\}$

$C_2 \to C_b B$

$A \to C_a A$

$B \to C_b B \mid b$

$C_a \to a$

$C_b \to b$

This is Equivalent form.

GNF Defination:-

- CFG $G = \langle V, T, P, S \rangle$ is said to be in GNF.
- The Context free Grammer, If every production is said to be in form of $A \to a\alpha$ where $a \in T$ and $\alpha \in V^*$

ie. GNF Should Start with terminal at RHS

$A \to A\alpha$

Example of GNF.

$G = \langle V, T, P, S \rangle$

$A_1 \to A_2 A_3$

$G = (\{A_1, A_2, A_3\}, \{a, b\}, P, A_1\}$
where P is
$A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 | b$
$A_3 \to A_1 A_2 | a$

Step: Since the RHS of the productions of A_1 and A_2 start with terminal or higher number variable.
We begin with produrion
$A_3 \to A_1 A_2$
and substring the string
$A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 | b$
$A_3 \to A_2 A_3 A_2 | a$.

Since the R.H.S of production
$A_3 \to A_2 A_3 A_2$ begins which lowered numbered variable. We subtitute for 1^{st} occurrance of the A_2
$A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 | b$
$A_3 \to A_3 A_1 A_3 A_2 | b A_3 A_2 | a$.

Apply the lemma.
$A_1 \to A_2 A_3$
$A_2 \to A_3 A_1 | b$

$A_3 \to b A_3 A_2 B_3 | a B_3 | b A_3 | a$.
$B_3 \to A_1 A_3 A_2 | A_1 A_3 A_2 B_3$

Step2: Now all the production with A_3 on the left have right hand sides that start with terminal or higher numbered variables. Those which starts with terminal are used to replace A_3 in the production $A_3 \rightarrow A_3 A_1$

$A_3 \rightarrow bA_3A_2B_3$
$A_3 \rightarrow aB_3$
$A_3 \rightarrow bA_3A_2$
$A_3 \rightarrow a$.

- $A_2 \rightarrow A_3 A_1$

replace A_3 by A_1 production.

$A_2 \rightarrow bA_3A_2B_3A_1$
$A_2 \rightarrow aB_3A_1$
$A_2 \rightarrow bA_3A_2A_1$
$A_2 \rightarrow aA_1$
$A_2 \rightarrow b$

- $A_1 \rightarrow A_2A_3$

replace A_2 by its production at above.

$A_1 \rightarrow bA_3A_2B_3A_1A_3$
$A_1 \rightarrow aB_3A_1A_3$
$A_1 \rightarrow bA_3A_2A_1A_3$
$A_1 \rightarrow aA_1A_3$
$A_1 \rightarrow bA_3$

- $B_3 \rightarrow A_1A_3A_2$

$B_3 \rightarrow A_1A_3A_2B_3$

The B_3 production has to be converted in normal form by replacement of production.

$B_3 \rightarrow A_1 A_3 A_2$

$B_3 \rightarrow b A_3 A_2 B_3 A_1 A_3 A_3 A_2$
$B_3 \rightarrow a B_3 A_1 A_3 A_3 A_2$
$B_3 \rightarrow b A_3 A_2 A_1 A_3 A_3 A_2$
$B_3 \rightarrow a A_1 A_3 A_3 A_2$
$B_3 \rightarrow b A_3 A_3 A_2$

$B_3 \rightarrow A_1 A_3 A_2 B_3$
A_1 replace by A_1 production's to get terminal at first in RHS.

$B_3 \rightarrow b A_3 A_2 B_3 A_1 A_3 A_3 A_2 B_3$
$B_3 \rightarrow a B_3 A_1 A_3 A_3 A_2 B_3$
$B_3 \rightarrow b A_3 A_2 A_1 A_3 A_3 A_2 B_3$
$B_3 \rightarrow a A_1 A_3 A_3 A_2 B_3$
$B_3 \rightarrow b A_3 A_3 A_2 B_3$

Q.6 Explain DPDA and NPDA with languages of them.
Ans. In a DPDA there is only one move in every situation. A DPDA is less powerful than NPDA. Every context free language cannot be accepted by a DPDA.. For Example, a string of the form ww^R cannot be processed by a DPDA. The class of a language a DPDA can accept lies in between a regular language and CFL.

A DPDA is defined as:
$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where
$\delta(q, a, x)$ has one move for any $q \in Q$, $x \in \Gamma$ & $a \in \Sigma$

A DPDA can be designed for regular language into two steps:

Step1 Construct an equivalent DFA for the given regular language.

Step2: For every transition $\delta(q_i, a) = q_j$ in FA (where $q_i, q_j \in Q$ and $a \in \Sigma$) we can write an equivalent transition for DPDA.

$$\delta(q_i, a, Z_0) = \{(q_j, Z_0)\}$$

The move for PDA involves neither a push nor a pop operation.

NPDA:

- A NPDA provides non-determinism to PDA.
- In a DPDA there is only one move in every situation. Where as, in case of NPDA there could be multiple moves under a situation.
- DPDA is less powerful than NPA.
- Every context free language can not be recognized by a DPDA. ~~but it can be recog.~~
- The class of language a DPDA can accept lies in between a regular language and CFL.
- A palindrome can be accepted by a NPDA but it cannot be accepted by a DPDA.

Q.7) Write Short notes on:

a) Recursive and Recursively Enumerable language.

1 There is a different between recursively enumerable (Turing Acceptable). and recursive (Turing Decidable) language.

2 Following Statement are equivalent:

i. The language L is Turing acceptable.

ii. The language L is recursively enumerable.

3 Following Statement are equivalent

i The language L is Turing decidable.

ii The language L is recursive

iii There is an algorithm for recognizing L.

4. Every Turing acceptable language need not be Turing decidable.

b) Intractable Problems.

i) P denotes the class of problems for each of which, there is at least one known polynomial time deterministic TM solving it.

ii) NP denotes the class of all problems, for each of which, there is at least one known non-deterministic polynomial time solution. However. this solution may not be reducible to a polynomial time deterministic TM.

iii A problem which does not have any known polynomial time algorithm is called an intractable problem otherwise it is called tractable.

iv A solution by deterministic TM is called an algorithm. A solution by a Non-deterministic TM may not be algorithm

v For every non-deterministic TM solution, there is a deterministic TM solution of a problem. But there is no computation equivalence between deterministic TM and non-deterministic TM.

vi) A problem is NP-complete if it is in NP and for which no polynomial time deterministic TM solution is known so far.

Eg of NP-complete are
Satisfiability problem (SAT)
Travelling Salesman Problem (TSP)

vii) A Boolean expression is said to be satisfiable if at least one truth assignment makes the Boolean expression true.

viii) Eg The truth value of a Boolean expression depends on the truth value of its variables.

ix) Satisfiability problem is NP-Complete.

x) Polynomial time reduction plays an important role in defining NP-Completeness. A polynomial time reduction is a polynomial-time algorithm which constructs instances of a problem P_2 from the instance of some other problem P_1.

xi) A problem L is said to be NP-hard if for any problem L_1 in NP, there is a polynomial-time reduction of L_1 to L.

c) Rice's Theorem:

i) "Every ~~problem~~ ~~problem~~ property that is satisfied by some but not all recursively enumerable language is un-decidable".

ii) Any property that is satisfied by some recursively enumerable language but not all is known as nontrivial property.

iii) It have seen many properties of R.E. languages that are un-decidable.

iv) The property includes.
1. Given a TM, M is L(M) nonempty?
2. Given a TM, M is L(M) finite?
3. Given a TM, M is L(M) regular?
4. Given a TM, M is L(M) recursive?

v) The Rice's Theorem can be proved by reducing some other unsolvable problem to nontrivial property of recursively enumerable language.

d) Post Correspondence Problem: (PCP)

i) Let A and B ~~to~~ be nonempty lists of string over Σ. A and B are given as below:

$A = \{x_1, x_2, x_3, \ldots\ldots, x_k\}$

$B = \{y_1, y_2, y_3, \ldots\ldots, y_k\}$

ii) There is a post correspondence A and B if there is a sequence of one or more integers i, j, k ... m. Such that.

iii) The string $x_i x_j x_k \ldots x_m$ is Equal to $y_i y_j \ldots y_m$.

for Eg:

$A = \{a, aba^3, ab\}$

$B = \{a^3, ab, b\}$

When the elements of A and B are listed, will produced identical strings.

The required sequence is (2,1,1,3)

$A_2 A_1 A_1 A_3 = aba^3 a a ab = aba^6 b.$

$B_2 B_1 B_1 B_3 = aba^3 a^3 b = aba^6 b.$

Thus the PCP has solution.

Thus we accepting the un-decidability of post Correspondence problem without proof.