Theory of Computation
Chapter 4: Undecidability

Lectures by Monalisa

MonalisaCS

- **Section 6: Theory of Computation($\cong$ 10mark)**
  Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context free languages, pumping lemma. Turing machines and undecidability.

- Chapter 1:Regular Language [RL,FA,RE ,Pumping lemma]
- Chapter 2: Context free Language [Grammar(RG,CFG),CFL,PDA, Pumping lemma]
- Chapter 3: Recursive enumerable Language [CSL, LBA ,RS,RES,TM]
- Chapter 4: Undecidability(Undecidable, Rice Theorem ,Reducibility, PCP)

MonalisaCS

Undecidability

- Problems for which no algorithm exist is called as undecidable & if algorithm exist is called as decidable.
- Undecidable Example

1. Ambiguity of CFG.
2. Regularity of CFG
3. Equality of CFG
4. Completeness problem of CFG[L(G) = Σ*]
5. Conversion of NPDA to DPDA
6. Conversion of ambiguous to unambiguous grammar.
7. Halting problem of TM

❖ Following property of TM are undecidable

1. Emptiness
2. Finiteness
3. Equalness
4. Membership
5. Regularity
6. Context freedom & dependency
7. Recursiveness
8. Completeness or Totality problem

MonalisaCS

- **Rice Theorem:**
- Every non trivial properties of REL is Undecidable.
- Formal Definition
- If P is a non-trivial property, and the language holding the property, L_p , is recognized by Turing machine M, then $L_p = \{<M> \mid L(M) \in P\}$ is undecidable.
- Property of languages, P, is simply a set of languages. If any language belongs to P ($L \in P$), it is said that L satisfies the property P.
- A property is called to be trivial if either it is not satisfied by any recursively enumerable languages, or if it is satisfied by all recursively enumerable languages.
- A non-trivial property is satisfied by some recursively enumerable languages and are not satisfied by others. Formally speaking, in a non-trivial property, where $L \in P$, both the following properties hold:
- **Property 1** − There exists Turing Machines, M1 and M2 that recognize the same language, i.e. either ($<M1>, <M2> \in L$) or ($<M1>,<M2> \notin L$)
- **Property 2** − There exists Turing Machines M1 and M2, where M1 recognizes the language while M2 does not, i.e. $<M1> \in L$ and $<M2> \notin L$

MonalisaCS

Reducibility

- P_1 is solvable
- $P_1 \leq P_2$ reducible
- Solution of P_2 is solution of P_1.
- If P_1 is undecidable then P_2 also undecidable
- EX : $P_1 = n^4 - 1$, $P_2 = n^2 - 1$, $P_3 = n^2 + 1$
- Solution of P_2 & P_3 are solution of P_1
- $A \leq B$, (A is reducible to B) , i. e, solving A cannot be "harder" than solving B.
- 1.If A is reducible to B, and B is decidable, then A is decidable.
- i) if A is reducible to B, and B is recursive, then A is recursive.
- 2.If A is undecidable and reducible to B, then B is undecidable.
- i) if B is recursively enumerable, and A is reducible to B, then A is recursively enumerable.
- ii) if A is not recursively enumerable, and reducible to B, then B is not recursively enumerable.

MonalisaCS

https://monalisacs.com/

- Rule 1: If B is recursive then A is recursive
- Rule 2: If B is recursively enumerable then A is recursively enumerable
- Rule 3: If A is not recursively enumerable then B is not recursively enumerable
- **Post Correspondence Problem**
- The Post Correspondence Problem (PCP) is an undecidable decision problem.
- Given the following two lists, **X** and Y of non-empty strings over ∑
- $X = (x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$
- $Y = (y_1, y_2, y_3, \ldots\ldots\ldots, y_n)$
- There is a Post Correspondence Solution, if for some $i_1, i_2, \ldots\ldots\ldots\ldots i_k$, where $1 \leq i_j \leq n$, the condition $x_{i1} \ldots\ldots\ldots x_{ik} = y_{i1} \ldots\ldots\ldots y_{ik}$ satisfies.
- Example 1
- X = (a, ba, aa) and Y = (ab, aa, a)
- = (1, 2 ,3) and = (4 , 5 ,6)
- w=abaaa ,123=456 or a ba aa= ab aa a
- w=aabaa ,321=645 or aa ba a= a ab aa

MonalisaCS

https://www.youtube.com/@MonalisaCS