18CS10062

RADHIKA PATWARI

Assignment 1 Theo

3/10/2021

1. $P \rightarrow$ property over set of partial recursive functions

$P : RE \longrightarrow \{T, F\}$

$P = \{ i \mid f_i \in RE \text{ and } P(i) \}$

index of partial recursive function (encoding of M that computes f)

Let x_1, x_2 be 2 such indices, $x_1, x_2 \in \mathbb{N}$

$P(x_1) = T$
$P(x_2) = F$

Let P be decidable.

$\exists$ a total computable function (σ) that computes P.

$$\therefore \sigma_1(y) = \begin{cases} x_2 & P(y) = T \\ x_1 & \text{otherwise} \end{cases}$$

18CS10062

Since P is decidable, σ_1 is a total computable function.

By fixed point theorem,

$$\exists\, x_0 \quad f_{x_0}(y) = f_{\sigma(x_0)}(y)$$

If $P(x_0) = T$, f_{x_0} satisfies P.

$f_{\sigma(x_0)}$ satisfies P

f_{x_2} satisfies P

$$\left.\begin{array}{r} P(x_2) = T \\ \text{But } P(x_2) = F \end{array}\right\} \text{Contradiction}$$

$\therefore$ Assumption is wrong

$\Rightarrow$ P is undecidable

$\therefore$ Non trivial property of set of partial recursive functions is undecidable.

18CS10062

2. $L = \{ M \mid \Sigma \in \{0,1\}, M$ writes symbol 1 at some point on its tape $\}$

L is undecidable

Proof. $HP \leq L$ → Reduction from Halting Problem

Let (M, x) be HP instance

$$(M, x) \mapsto N$$

- Construct N with input alphabet $\Sigma = \{0,1\}$
- On input x' to N,

Run M on x and use these tape alphabets $\{K - \Sigma\}$ to do computation

If M halts on x, write 1 on the tape & halt

If M does not halt, 1 could not be written to the tape

$\therefore$ $L = \mathcal{L}(N) = \begin{cases} \Sigma^* & \text{if } M \text{ halts on } x \\ \phi & \text{if } M \text{ does not halt on } x \end{cases}$

$\therefore$ Since HP is undecidable, L is undecidable

18CS10062

4. Let PCP' = all strings of length exactly 5

$A = \{w_1, w_2, w_3 \dots w_k\}$

$B = \{x_1, x_2, x_3 \dots x_k\}$

$\forall w_i \in A$ & $\forall x_i \in B$ ($w_i, x_i \in \Sigma^*$)

& $|w_i| = |x_i| = 5$

Let PCP' = variant of P as defined above

PCP' = (A, B) $\in$ PCP' $\iff \exists$ sequence $i_1, i_2, i_3 \dots i_m$

such that $w_{i_1}, w_{i_2} \dots w_{i_m} = x_{i_1}, x_{i_2} \dots x_{i_m}$

PCP' is decidable = C (Assumption)

Let C be a total TM that decides PCP'

On input (A, B, C - decides if $\exists$ any index i_1 to i_k such that $w_i = x_i$

- if such index exists, solution to C is found (C halts on input)
- else C rejects & halts

→ Checking single index is sufficient as w_i & x_i are strings of constant length 5.

$\therefore w_{i_1} w_{i_2} \dots w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$

$\Rightarrow w_{i_1} = x_{i_1}$ && $w_{i_2} = x_{i_2}$ ... till $w_{i_m} = x_{i_m}$

So if it matches for multiple i's, it matches for one of i's.

→ Its sufficient to check 1 Index → UnDecidable PCP.

Thus there exist PCP $\leq$ PCP' ⟶ Hence PCP' is undecidable.

5. Given $L(G) \Rightarrow P = \{ G \mid L(G) = L(G)^R \}$

We know $S = \{ (G_1, G_2) \mid G_1 = G_2 \}$ is undecidable

$\therefore$ Reduction from S to P is used to show P is undecidable

18CS10062.

5. Proof. $S \leq P$

Let G_1, G_2 be CFGs $\in P$. We can construct a context free Grammar G such that

$$L(G) = \# L(G_1) \cup L(G_2)^R \#$$

where $\#$ = new symbol

So $L(G) = L(G)^R$ iff $L(G_1) = L(G_2)$

Hence Reduction is possible.

S is undecidable $\Leftrightarrow$ P is undecidable.

18CS10062

8. Th+(N) is decidable

- Use a finite automata capable of doing addition if input is presented in a special form. The special form is 3 bit value that represents 8 diff. characters.

We use the below form for $i>0$

$$\Sigma_i = \left\{ \begin{bmatrix}0\\ \vdots\\ 0\\ 0\end{bmatrix}, \begin{bmatrix}0\\ \vdots\\ 0\\ 1\end{bmatrix}, \begin{bmatrix}0\\ \vdots\\ 1\\ 0\end{bmatrix}, \begin{bmatrix}0\\ \vdots\\ 1\\ 1\end{bmatrix} \cdots \begin{bmatrix}1\\ \vdots\\ 1\\ 1\end{bmatrix} \right\}$$

- Let $\phi = Q_1 x_1 Q_2 x_2 \dots Q_l x_l [\psi]$

Q's → quantifiers

ψ → No quantifiers, has variables $x_1 \dots x_l$

For each i from $(0, L)$ define

$$\phi_i = Q_{i+1} x_{i+1} Q_{i+2} x_{i+2} \dots Q_l x_l [\psi]$$

Thus $\phi_0 = \phi$, $\phi_l = \psi$

- ϕ_i has i free variables and each ϕ can take arguments $a_1 \dots a_n$ $\phi_i(a_1 \dots a_i)$ and substitute them for respective x_i's.

- For each $i \in [0, L]$, create a finite automata A_i that recognize strings that represent i-tuples of numbers that make ϕ_i = true

- First construct $A(L)$ directly, iterate i from L to 0.

- Construct $A(i-1)$ using $A(i)$

- Check $A(0)$ accepts empty string

  - if it does, ϕ = true & algorithm accepts.

Machine A_i

18CS10062

We have 2 cases

•) A_L : since $\phi_L = \varphi$ is a boolean combination of additions, we can build a finite automata to compute single additions and regular languages are closed under $\vee, \wedge, \neg$, we can built A_L.

•) A_i from A_{i+1} :

• if $\phi_i = \exists x_{x+1} . \phi_{i+1}$: A_i operates as A_{i+1}, but it nondeterministically guesses value of a_{i+1}, such that A_i accepts input $(a_1 \ldots a_i)$ if some a_{i+1} exists such that A_{i+1} accepts $(a_1 \ldots a_{i+1})$

• if $\phi_i = \forall x_{x+1} \cdot \phi_{i+1}$, it is equivalent to $\neg \forall x_{i+1} . \neg \phi_{i+1}$. Thus we can build a finite automata that recognizes $\neg A_{i+1}$, then apply preceding construction to the $\exists$ quantifier, then account for the complement again to obtain A_i

•) A_0 accepts any input iff ϕ_0 is true. ∴ if A_0 accepts ϵ, ϕ = true and A_0 accepts, otherwise it rejects.

∴ $Th(N, +)$ is decidable.