Alejandro Manzanares Lemus

Ejercicios Relación 2

Ej. 1 | * Supongamos que L' es r.e

$L' = \{0w \mid w \in L\} \cup \{1w \mid w \notin L\}$ se puede expresar como:

$L' = \{0w \mid w \in L\} \cup \{1w \mid w \in \overline{L}\}$ porque los w que no pertenecen a L, pertenecen a $\overline{L}$

Sabemos que L es un lenguaje r.e pero no recursivo, por tanto, podemos inferir que $\overline{L}$ no es ni recursivo, ni r.e, porque de serlo L seria recursivo, pero no lo es.

* Supongamos las máquinas de turing M, M_L, M_{NL} tal que $L(M) = L'$, $L(M_L) = L$, $L(M_{NL}) = \overline{L}$

Dada una palabra w, M añade un 0 delante y realiza los mismos pasos que M_L. Si M_L acepta la palabra, tambien la acepta M y viceversa, si M_L no acepta, tampoco lo hace M. También, dada una palabra w, M añade un 1 delante y realiza los mismos pasos que M_{NL}. Si M_{NL} acepta la palabra, También lo hace M y si M_{NL} no acepta, tampoco acepta M.

El problema es que como $\overline{L}$ no es ni r.e ni recursivo, no puede existir M_{NL} tal que $L(M_{NL}) = \overline{L}$ y por tanto, no puede existir M tal que $L(M) = L'$, por lo que L' no es r.e.

Al ser L' no r.e, no podemos afirmar si $\overline{L'}$ es r.e o recursivo.

Ej 2

a) Dada una MT, determinar si acepta al menos dos palabras distintas

La propiedad de "aceptar dos palabras distintas" es una propiedad asociada al lenguaje del problema y además es una propiedad no trivial, porque no todos los lenguajes r.e la cumplen. Por ello, nos valemos del teorema de Rice para afirmar que este problema no es decidible, puesto que debemos satisfacer una propiedad no trivial de los lenguajes r.e.

Para demostrar que es semidecidible, bastará con encontrar una máquina de turing que acepte dos palabras distintas.

Máquina de turing no determinista:

Seleccionamos de forma aleatoria u y v ∈ L

Si u y v son distintas

MT para en un estado de aceptación

Si no

no se acepta.

b) Dada una MT, determinar si acepta un número finito de palabras

Este problema no es ni decidible ni semidecidible, para demostrarlo, se puede establecer una reducción desde un problema que sabemos que no es semidecidible

El problema EMPTY se define como: Dada una MT M, determinar si $L(M) = \emptyset$. El lenguaje asociado al problema es L_e. Como $\overline{L_e}$ es r.e pero no recursivo, L_e es no r.e.

El problema EMPTY por tanto es no semidecidible.

El problema EMPTY se reduce al problema 2b, por tanto 2b no es semidecidible

Un caso positivo para el problema EMPTY es un caso positivo para 2b, porque si $L(M) = \emptyset$ se acepta un número finito de palabras, es decir, se aceptan 0 palabras.

Un caso negativo para el problema EMPTY es un caso negativo para 2b, porque si $L(M) \neq \emptyset$ significa que se aceptan 1 o varias palabras, al ser el conjunto de palabras a comprobar infinito no se puede asegurar que solo se acepte un número finito de estas, porque no podemos comprobar el conjunto de palabras al completo.

c) Dada una MT, determinar si el lenguaje aceptado es independiente del contexto.

Utilizaremos la propiedad del lenguaje vacío de ser independiente del contexto para demostrar que este problema es no semidecidible.

Para esto podemos establecer una reducción desde el problema C-UNIVERSAL.

Este problema se define como: Dada una MT M y una entrada w, M no acepta w.

El problema C-UNIVERSAL se reduce al problema Zc, por tanto Zc no es semidecidible

Maquina de Turing F(M,w)

Entrada: una palabra $v \in \{0,1\}^*$

Simulamos los pasos de M con la entrada w

Si M no acepta w:

rechazamos la entrada

Si no:

aceptamos la entrada.

F(M,w) es un proceso algoritmico que convierte casos positivos y negativos de C-UNIVERSAL a casos del problema Zc

Para un caso positivo de C-UNIVERSAL, M no acepta w y por tanto se rechaza cualquier entrada. Por esto el lenguaje aceptado es $\emptyset$, que es un lenguaje independiente del contexto.

Para un caso negativo de C-UNIVERSAL, M acepta w y por tanto se acepte la entrada, obteniendo como resultado que el lenguaje aceptado pueda no ser independiente del contexto.

d) Dada una MT, saber si para la entrada 0011 no va a usar más de 10 casillas de la cinta.

Este problema es decidible, ya que existe un numero finito de pasos que el algoritmo puede realizar en 10 casillas.

En una máquina de turing, para un número de 10 casillas, solo existen un numero $n = 10 \cdot S^{10} \cdot Q$ posibles configuraciones. Siendo S el conjunto de simbolos (0, 1, #) y Q el conjunto de posibles estados. Si ejecutamos una máquina de turing sobre la entrada 0011, cuando alcance el paso n+1, sabemos que si el cabezal de la MT se encuentra dentro de las 10 casillas, como ya ha pasado por todas las posibles transiciones, podemos afirmar que esta MT no utiliza más de 10 casillas

e) Dadas dos MT, saber si aceptan el mismo lenguaje

Este problema es no semidecidible puesto que dadas dos MT, M_1 y M_2, debemos comprobar que $L(M_1) = L(M_2)$ y para esto debemos poder afirmar que TODA palabra de L_1 se encuentra en L_2 $L(M_1) \subseteq L(M_2)$ y además que toda palabra de L_2 se encuentra en L_1, es decir, $L(M_2) \subseteq L(M_1)$

Veamos entonces que al ser el problema 2e1: Dadas dos MT M_1 y M_2 determinar si: $L(M_1) \subseteq L(M_2)$ un problema no semidecidible, el problema 2e también es no semidecidible

Para demostrar que 2e1 es no semidecidible, hacemos una reducción desde el problema EMPTY, ya enunciado previamente.

Sea R una MT que no acepte ninguna palabra ($L(R) = \emptyset$) y M una instancia del problema EMPTY. Asociamos el par $F(M) = (M, R)$ siendo $F(M)$ un proceso algorítmico

Si M es un caso positivo para EMPTY: $L(M) = \emptyset$, es un caso positivo para 2e1 ya que $L(M) \subseteq L(R) = \emptyset$.

Si M es un caso negativo para EMPTY: $L(M) \neq \emptyset$, es un caso negativo para 2e1 ya que $L(M) \not\subseteq L(R) = \emptyset$

Por tanto EMPTY se reduce a 2e1, y al ser 2e1 un problema no semidecidible, tampoco es semidecidible 2e

f) Dada una MT M y una palabra u, saber si la MT acepta la palabra u en número de pasos menor o igual a $|u|$

Este problema es decidible, puesto que existe una MT que resuelve este problema en un número finito de pasos

Maquina de Turing M_1

Entrada: palabra u

Ejecutar M un numero $|u|$ de pasos

Si M acepta u

  M_1 acepta

Si no

  M_1 rechaza