Tarea 3

García Ponce José Camilo

1) Diseña una maquina de Turing que acepte el conjunto

$$\{\alpha \in \{a,b,c\}^* \mid \alpha \text{ tiene el mismo numero de } a, b \text{ y } c\}$$

paso 1) el alfabeto de entrada es $\Sigma = \{a,b,c\}$

el alfabeto de la cinta sera $\Gamma = \{a,b,c,\sqcup,X,\vdash\}$ ← inicio de la cinta

los estados seran $Q = \{s, t, q_1, q_2, q_3, q_4, q_5, q_6, q_7$

paso 2) la estrategia es marcar el primer simbolo que encontremos, luego buscar un simbolo diferente para marcar, despues buscamos el otro tipo de simbolo por marcar, si marcamos los tres tipos de simbolos regresamos al inicio y volvemos a empezar (marcamos con X)

paso 3) el criterio de aceptacion sera cuando todos los simbolos esten marcados

paso 4) la funcion de transicion

$$\delta(s,\vdash) = (s,\vdash,\rightarrow)$$
$$\delta(s,a) = (q_1,X,\rightarrow)$$
$$\delta(s,b) = (q_2,X,\rightarrow)$$
$$\delta(s,c) = (q_3,X,\rightarrow)$$
$$\delta(s,\sqcup) = (t,\sqcup,\rightarrow)$$
$$\delta(s,X) = (s,X,\rightarrow)$$
$$\delta(q_1,a) = (q_1,a,\rightarrow)$$
$$\delta(q_1,b) = (q_4,X,\rightarrow)$$
$$\delta(q_1,c) = (q_5,X,\rightarrow)$$
$$\delta(q_1,X) = (q_1,X,\rightarrow)$$
$$\delta(q_2,a) = (q_4,X,\rightarrow)$$
$$\delta(q_2,b) = (q_2,b,\rightarrow)$$
$$\delta(q_2,c) = (q_6,X,\rightarrow)$$
$$\delta(q_2,X) = (q_2,X,\rightarrow)$$
$$\delta(q_3,a) = (q_5,X,\rightarrow)$$
$$\delta(q_3,b) = (q_6,X,\rightarrow)$$
$$\delta(q_3,c) = (q_3,c,\rightarrow)$$
$$\delta(q_3,X) = (q_3,X,\rightarrow)$$
$$\delta(q_4,a) = (q_4,a,\rightarrow)$$
$$\delta(q_4,b) = (q_4,b,\rightarrow)$$
$$\delta(q_4,c) = (q_7,X,\leftarrow)$$
$$\delta(q_4,X) = (q_4,X,\rightarrow)$$
$$\delta(q_5,a) = (q_5,a,\rightarrow)$$
$$\delta(q_5,b) = (q_7,X,\leftarrow)$$
$$\delta(q_5,c) = (q_5,c,\rightarrow)$$
$$\delta(q_5,X) = (q_5,X,\rightarrow)$$
$$\delta(q_6,a) = (q_7,X,\leftarrow)$$
$$\delta(q_6,b) = (q_6,b,\rightarrow)$$
$$\delta(q_6,c) = (q_6,c,\rightarrow)$$
$$\delta(q_6,X) = (q_6,X,\rightarrow)$$
$$\delta(q_7,a) = (q_7,a,\leftarrow)$$
$$\delta(q_7,b) = (q_7,b,\leftarrow)$$
$$\delta(q_7,c) = (q_7,c,\leftarrow)$$
$$\delta(q_7,X) = (q_7,X,\leftarrow)$$
$$\delta(q_7,\vdash) = (s,\vdash,\rightarrow)$$

estados:

s	inicial
t	aceptacion
q_1	marcada a
q_2	marcada b
q_3	marcada c
q_4	marcada a y b
q_5	marcada a y c
q_6	marcada b y c
q_7	marcada a, b y c

2) Describe una maquina de Turing que acepte el conjunto $\{a^n b^{2n}\}$

paso 1) el alfabeto de entrada es $\Sigma = \{a, b\}$
el alfabeto de la cinta seria $\Gamma = \{a, b, \sqcup\}$

los estados seran $Q = \{s, t, q_1,$

paso 2) la estrategia es marcar una a, luego buscar la ultima b, despues marcamos dos bs, luego regresamos a la primera a sin marcar, volvemos a empezar (marcamos con $\sqcup$)

paso 3) el criterio de aceptacion sera cuando todos los simbolos esten marcados

paso 4) funcion de transicion

estados
s inicial
t aceptacion
q_1 marcada a
q_2 por marcar 2 b
q_3 por marcar 1 b
q_4 regresando

$\delta(s, a) = (q_1, \sqcup, \rightarrow)$
$\delta(s, \sqcup) = (t, \sqcup, \rightarrow)$
$\delta(q_1, a) = (q_1, a, \rightarrow)$
$\delta(q_1, b) = (q_1, b, \rightarrow)$
$\delta(q_1, \sqcup) = (q_2, \sqcup, \leftarrow)$
$\delta(q_2, b) = (q_3, \sqcup, \leftarrow)$
$\delta(q_3, b) = (q_4, \sqcup, \leftarrow)$
$\delta(q_4, a) = (q_4, a, \leftarrow)$
$\delta(q_4, b) = (q_4, b, \leftarrow)$
$\delta(q_4, \sqcup) = (s, \sqcup, \rightarrow)$

3) Demuestra que el conjunto $CFL = \{<M> \mid L(M)$ es un lenguaje indep del contexto$\}$ no es recursivo sin usar el teorema de Rice, recuerda que $\emptyset$ y Σ^* son CFL

Estrategia reducir, sabemos que HP no es recursivo, vamos a reducir a HP, entonces demostraremos $HP \leq_m CFL$

necesitamos definir una función σ computable tal que

$$<M>\#<\alpha> \in HP \iff \sigma(<M>\#<\alpha>) = <M'> \in CFL$$

es decir M se detiene con α $\iff$ $L(M')$ es un lenguaje independiente del contexto

solución, veamos las características de M' que es producida por σ con $<M>\#<\alpha>$

- M y α están codificados en el programa de M'
- M' recibe como entrada una cadena de la forma $a^n b^n c^n$, con $n \in \mathbb{N}$
- M' se detendrá en los siguientes casos:

M' recibe la entrada β, M' simula a M con la entrada α,

si M se detiene entonces rechazamos β, pero si M no se detiene aceptamos

De esta manera si M se detiene con α, M' reconoce el lenguaje $\emptyset$ (es CFL)

y si M no se detiene con α, M' reconoce el lenguaje $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ (no es CFL)

σ es computable, ya que podemos hacer una maquina total k que para toda entrada $\langle M\rangle\#\langle\alpha\rangle$ se detenga produciendo $\langle M'\rangle$, y siempre se detiene ya que k no simula a ninguna maquina

4) Demuestra ahora que CFL no es ni siquiera recursivamente enumerable por medio de uno de los teoremas de Rice

Usamos el Teorema de Rice II

Sea P la propiedad sobre los lenguajes r.e. tal que

$P(L) = V \iff L$ es lenguaje independiente del contexto, por lo tanto

$CFL = \{\langle M\rangle \mid P(L(M)) = V\}$

Por def, que CFL no sea r.e es equivalente a que P no sea semidecidible, para esto veremos que P es no monotona debido a que: sabemos que $\{\varepsilon\}$ es independiente del contexto y $\{a^n b^n c^n \mid n \geq 0\}$ no es independiente del contexto, notemos que $\{\varepsilon\} \subseteq \{a^n b^n c^n \mid n \geq 0\}$ y además $P(\{\varepsilon\}) = V$ y $P(\{a^n b^n c^n \mid n \geq 0\}) = F$, por lo tanto se cumple que $\{\varepsilon\} \subseteq \{a^n b^n c^n \mid n \geq 0\} \wedge P(\{\varepsilon\}) = V \not\Rightarrow P(\{a^n b^n c^n \mid n \geq 0\}) = V$

obteniendo que P no es monotona, usamos Teorema de Rice II, P no es semidecidible, entonces CFL no es r.e.

5) Demostrar la siguiente extensión del teorema de Rice: toda propiedad no trivial de pares de conjuntos r.e es indecidible

Sea P una propiedad no trivial de conjuntos de r.e.

probaremos que el conjunto $A = \{ \langle M \rangle \# \langle N \rangle \mid P(L(M), L(N)) = V \}$

no es recursivo, para esto reduciremos HP a A, es decir $HP \leq_m A$

Supongamos, sin pérdida de generalidad, que $P(\emptyset, \emptyset) = F$, si esto no es cierto, entonces aplicamos a la negación de la propiedad.

Ahora, como P es no trivial, existen B, C lenguajes r.e, tal que $P(B, C) = V$, sea M_B y M_C con $L(M_B) = B$ y $L(M_C) = C$

ahora veremos lo de la reducción $HP \leq_m A$, definimos la transformación σ tal que $\langle M \rangle \# \langle \alpha \rangle \in HP \iff \sigma(\langle M \rangle \# \langle \alpha \rangle) = \langle M_1 \rangle, \langle M_2 \rangle \in A$

es decir, M se detiene con α $\iff P(L(M_1), L(M_2)) = V$

Veamos las características de M_1 y M_2

- M, a, M_B y M_C están codificados en el programa de M_1 y M_2
- M_1 recibe una cadena $\beta_1 \in \Sigma^*$, M_2 recibe una cadena $\beta_2 \in \Sigma^*$
- M_1 recibe β_1, luego simula M con α, si esta ejecución se detiene simula M_B con β_1 y aceptamos β_1 si y solo si M_B acepta β_1, si no M_1 no se detiene

- M_2 recibe β_2, luego simula M con α, si esta ejecucion se detiene simula M_C con β_2 y aceptamos β_2 si y solo si M_C acepta β_2, si no M_2 no se detiene

Si M se detiene con α, M_1 y M_2 simulan a M_B y M_C respectivamente entonces $L(M_1) = L(M_B) = B$ y $L(M_2) = L(M_C) = C$, entonces $P(B, C) = V$

Si no se detiene con α, M_1 y M_2 no simulan a M_B y M_C, entonces $L(M_1) = L(M_2) = \varnothing$, entonces $P(\varnothing, \varnothing) = F$

Podemos construir una maquina total K que compute $\sigma(\langle M \rangle \# \langle \alpha \rangle)$,

K no simula ninguna maquina solo produce la codificacion de M_1 y M_2

Utilizalo para demostrar que, dadas dos maquinas $N, M \in MT$, los siguiente problemas son indecidibles

(a) ¿$L(M) = L(N)$?

necesitamos ver que esta propiedad es no trivial

sea $P(M, N) = V \iff L(M) = L(N)$, veamos que P es no trivial

sean S, Q, R tal que y $L(S) = \emptyset$, $L(Q) = \emptyset$, y $L(R) = \Sigma^*$

notemos que todos son c.e., ahora veamos si se cumple P o no

$P(S, Q) = V$ ya qe $L(S) = \emptyset = \emptyset = L(Q)$

pero $P(S, R) = F$ ya qe $L(S) = \emptyset \neq \Sigma^* = L(R)$

por lo tanto P es no trivial, por lo tanto usamos la extension del teorema de Rice y concluimos qe P es indecidible

entonces $L(M) = L(N)$ es indecidible

(b) ¿$L(M) \cap L(N) = \emptyset$?

Sea $P(M, N) = V \iff L(M) \cap L(N) = \emptyset$, veamos qe P es no trivial

sean S, Q, R tal qe $L(S) = \emptyset$, $L(Q) = \Sigma^*$ y $L(R) = \Sigma^*$

notemos qe todos son c.e., ahora veamos si se cumple P o no

$P(S,Q) = V$ ya qe $L(S) \cap L(Q) = \emptyset \cap \Sigma^* = \emptyset$

pero $P(Q,R) = F$ ya qe $L(Q) \cap L(R) = \Sigma^* \cap \Sigma^* = \Sigma^*$

por lo tanto P es no trivial, por lo tanto usamos la extension del teorema de Rice y conclumos qe P es indecidible

entonces $L(M) \cap L(N) = \emptyset$ es indecidible

(c) ¿$L(M) \subseteq L(N)$?

Sea $P(M,N) = V \iff L(M) \subseteq L(N)$, veamos que P es no trivial

sean S y Q tales qe $L(S) = \emptyset$ y $L(Q) = \Sigma^*$, notemos qe son r.e, ahora veamos si se cumple P o no

$P(S,Q) = V$ ya que $L(S) = \emptyset \subseteq \Sigma^* = L(Q)$

pero $P(Q,S) = F$ ya qe $L(Q) = \Sigma^* \not\subseteq \emptyset = L(S)$

por lo tanto P es no trivial, entones usamos la extension del teorema de Rice y conclumos qe P se indecidible

entones $L(M) \subseteq L(N)$ es indecidible

(d) ¿$L(M) \cap L(N)$ es recursivo?

Sea $P(M,N) = V \iff L(M) \cap L(N)$ es recursivo, veamos que P es no trivial

sean S, Q, R tal que $L(S) = \emptyset$, $L(Q) = HP$ y $L(S) = HP$,

ya vimos que HP es re pero no rec,

notemos que todos son r.e., ahora veamos si se cumple P o no

$P(S,Q) = V$ ya que $\emptyset \cap HP = \emptyset$ y sabemos que es recursivo

pero $P(Q,S) = F$ ya que $HP \cap HP = HP$ y sabemos que no es recursivo

por lo tanto P es no trivial, entonces usamos la extensión del teorema de Rice y concluimos que P es indecidible

entonces $L(M) \cap L(N)$ es recursivo, es indecidible

6) Demuestra que los lenguajes decidibles son cerrados bajo intersección y complemento

a) intersección

Sean L_1 y L_2 lenguajes decidibles cualquiera, por definición como son decidibles entonces son recursivos y por lo tanto existen máquinas de Turing totales que los reconoce, sean M_1 y M_2 estas máquinas (M_1 reconoce a L_1 y M_2 reconoce a L_2)

ahora sea L_3 la intersección de L_1 y L_2, ahora veamos una máquina de Turing total que lo acepta M_3

Cuando M_3 reciba la entrada x, M_3 va a simular M_1 con la entrada x y M_2 con la entrada x, entonces M_3 aceptará si y solo si M_1 y M_2 aceptan a la entrada x, en otro caso va a rechazar

por lo tanto M_3 es total y entonces $L_1 \cap L_2$ es recursivo y por lo tanto es decidible

Concluimos que los lenguajes decidibles son cerrados bajo intersección

b) complemento

Sea L un lenguaje decidible cualquiera, por definicion L es recursivo y por lo tanto existe M, una maquina de Turing total que reconoce a L

ahora ~L es el complemento de L, ahora veamos como seria una maquina de Turing que lo reconoce, sea ~M esta maquina.

Cuando ~M reciba la entrada x, va a simular a M y si M acepta entonces ~M rechaza y si M rechaza entonces ~M acepta por lo tanto ~M es total y por lo tanto ~L es recursivo y por lo tanto decidible

Concluimos que los lenguajes decidibles son cerrados bajo complemento