MA50B
Clase Auxiliar

24 de abril 2008

Ejercicio 1

Sea A un lenguaje reconocible sobre $\Sigma = \{0,1\}^*$ cuyas palabras son representaciones de máquinas de Turing M_L que deciden L. Dicho de otra manera:

$$A = \{< M_1 >, < M_2 >, ...\}$$

Donde M_i son mT que deciden. Mostrar que existe D, lenguaje decidible que no se puede decidir por ninguna mT M, con $< M > \in A$.

Solución:

A es reconocible, entonces sea E un enumerador de A. E produce como salidas la secuencia de palabras

$$< T_1 >, < T_2 >, ...$$

donde $\forall i \geq 1 :< T_i > \in A$.

Construimos la mT M de la manera siguiente:

$M(w)$:

1. Si w no representa una mT rechazar.
2. Calcular $i = (w)_2$ (i es el número representado por w en base 2)
3. Lanzar el enumerador E, recuperar la $i-esima$ palabra, que vamos a notar $< M_w >$.
4. Correr M_w sobre w.
   (a) Si M_w acepta, rechazar.
   (b) Si M_w rechaza, aceptar.

M es un decididor pués siempre se puede alcanzar el paso 4 y dado que M_w es por hipotesis un decididor, M siempre para.

Ahora sea $D = L_M$ (lenguaje reconocido por M). Si D es decidido por un M' tal que $< M' > \in A$ entonces existe w tal que M y $M_w = M'$ deciden el mismo lenguaje. Sin embargo, si M_w acepta w entonces M rechaza w (por construcción de M) y de misma manera, si M_w rechaza w entonces M acepta w. Contradicción.

Ejercicio 2

Sea A un lenguaje reconocible cuyas palabras son representaciones de mT que deciden:

$$A = \{< M_1 >, < M_2 >, ...\}$$

con $M_i, i \geq 1$ decididor.

Construir un lenguaje C decidible tal que cada máquina representada por una palabra de A tenga un máquina representada por una palabra de C equivalente [1] y vice-versa.

Solución:

A es reconocible. Entonces sea E un enumerador de A que produce como salidas: $< M_1 >, < M_2 >, ...$ Sea C el lenguaje: $C = \{< T_i >: i \geq 1\}$ donde cada T_i se construye como sigue:

Se toma la máquina M_i (mT representada por la $i-esima$ palabra que produce E). Se obtiene T_i tomando M_i y sacando los posibles estados aislados (estados que no se pueden alcanzar por ningún estado y distintos del estado inicial, estado de aceptación y estado de rechazo). Luego se agrega arbitrariamente i estados aislados para obtener T_i.

Nota: Esa construcción puede parecer extraña. La idea es solamente obtener una máquina T_i que se comporta igual que la máquina M_i pero que tenga codificado el número i en su representación. Se podría pensar en otras formas de "estampillar" la máquina T_i con el número i.

Por construcción toda máquina representada por una palabra de A tiene su equivalente en C y vice-versa pués cada máquina T_i tiene mismo comportamiento que la máquina M_i (los estados aislados nunca van a estar involucrados en un computo, dado que no se pueden alcanzar).

Falta mostrar que C es decidible. Vamos a construir la mT D que decide C:

$D(w)$:

1. Si w no es una representación de una mT rechazar.
2. $w =< M >$. Contar los estados aislados de M. Se obtiene i.
3. Ejecutar el enumerador E hasta obtener la $i-esima$ palabra $< M_i >$.
4. Sacar los estados aislados de M y M_i. Se obtiene M' y M'_i.
5. Si $< M' > = < M'_i >$ aceptar sino rechazar.

Claramente D siempre para. Luego si $w \in C$, D acepta. Si $w \notin C$, D rechaza, pués por construcción de C una máquina que tiene i estados aislados tiene que ser igual (sacando los estados aislados) a la máquina M_i (sin estados aislados).

En resumen: D acepta $w \Leftrightarrow w \in C$.

[1] Dos mT M y M' son equivalentes ssi $\forall w \in \Sigma^*$:

- M y M' acepta w, o
- M y M' rechazan w, o
- M y M' no paran con w de entrada.

Ejercicio 3

Mostrar que $HALT_{mT} = \{< M, w >: M$ es una mT y M se detiene en la entrada $w \in \Sigma^*\}$ es indecidible.

Solución:

Mostraremos que $A_{mT} = \{< M, w >: M$ es una mT y M acepta $w \in \Sigma^*\}$ se reduce a $HALT_{mT}$. Vamos a construir $f : A_{mT} \to HALT_{mT}$ una función de reducción. f es calculada por la mT R:

$R(w)$:

1. Si w no es de la forma $< M, w >$ donde M es una mT, rechazar.
2. Construir M' tal que:

   $M'(x)$:

   (a) Simula M en x.

   (b) Si M acepta, aceptar sino entrar en un bucle infinito.
3. Retornar $< M', w >$.

Tenemos $w \in A_{mT} \Rightarrow f(w) \in HALT_{mT}$. También $w \notin A_{mT} \Rightarrow f(w) \notin HALT_{mT}$ (pués, si w es malformada, rechaza y si $w \notin A_{mT}$, M' nunca va a parar (se queda “colgada” en (a) o en (b))).

Otras reducciones que siguen un patrón similar se pueden encontrar en el “Sipser”. Ver quizás en particular:

- $A_{mT} \leq_m REG_{mT}$, donde $REG_{mT} = \{< M > M$ es una mT y L_M, lenguaje reconocido por M, es regular $\}$.
- $A_{mT} \leq_m EQ_{mT}$ y $A_{mT} \leq_m EQ^c_{mT}$ donde $EQ_{mT} = \{< M_1, M_2 >: M_1, M_2$ son mT y $L_{M_1} = L_{M_2}\}$ y EQ^c_{mT} es el complemento de EQ_{mT}. Esas dos reducciones tienen como consecuencia interesante que EQ_{mT} no es reconocible y tampoco co-reconocible.