



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Complejidad Computacional - IIC3242

Tarea 1

Entrega: Miércoles 30 de abril

1. En esta pregunta demostraremos que dada una MT determinista M con alfabeto de entrada Σ , k cintas de trabajo y que funciona en tiempo $T(n)$, existe una MT determinista M_3 con alfabeto de entrada Σ , dos cintas de trabajo, que funciona en tiempo $O(T(n) \log_2 T(n))$ y que acepta el mismo lenguaje que M . Para esto, nos basaremos en la demostración presentada en [1]. A diferencia del modelo visto en clases, en esta demostración se utiliza un modelo de máquinas de Turing en el cual cada cinta es de lectura y escritura, y es infinita en ambas direcciones. En esta pregunta primero deberás modificar esta demostración para el caso de máquinas de Turing deterministas con una cinta de lectura y cintas de trabajo infinitas en ambas direcciones, para luego restringirse al modelo visto en clases.

La base de la demostración es lograr simular en la tercera cinta de M_3 cada cinta de trabajo de M , pero manteniendo en una posición marcada (denominada *home column*) el próximo elemento a modificar de cada cinta de trabajo de M . De esta forma cada vez que la cabeza lectora necesite hacer un cambio debe estar en esa posición. Para esto realizaremos la demostración primero para una cinta de trabajo, por lo que a partir de ahora consideraremos $k = 1$. Esto puede sonar extraño en principio (pues estamos simulando una máquina con una cinta de trabajo), pero más adelante entenderás por qué la idea se extiende a k cintas de trabajo para cualquier k . Una primera observación que podrás notar es que si en la última cinta M_3 guardas exactamente lo mismo que en la cinta de trabajo de M , para mantener siempre en una posición fija de esta cinta el próximo elemento a cambiar, sería necesario desplazar todos los símbolos. Esto acotaría por debajo el tiempo de ejecución de M_3 a $T(n)^2$, pues tras cada transición habría que mover todos los elementos de la cinta. Para evitar este problema, el alfabeto de trabajo de M_3 debe ser distinto al de M .

- a) [0.2 puntos] En esta pregunta debes definir el alfabeto de trabajo de la máquina M_3 y explicar en detalle qué representa cada símbolo de dicho alfabeto. Te recomendamos partir definiendo el alfabeto para la máquina M_3 basándote en la idea presentada en [1].
- b) [0.4 puntos] Una vez que tengas definido el alfabeto, debes explicar cómo se ven las posibles configuraciones de la tercera cinta de M_3 y qué representa cada una de estas configuraciones en la cinta de trabajo de M .
- c) [0.4 puntos] En este ítem debes explicar cómo simular las transiciones de M en la máquina M_3 , de forma de mantener la propiedad mencionada (el próximo símbolo a modificar debe estar siempre en una posición fija de la tercera cinta de M_3). En particular, debes definir lo que se denomina un *cleanup de orden i* en [1]. Tendrás que explicar por qué tras cada transición, la tercera cinta de M_3 representa la configuración correspondiente de la máquina M .
- d) [0.2 puntos] Demuestra que la cantidad de pasos necesarios para realizar un cleanup de orden i en M_3 está acotado por $c \cdot 2^i$ para una constante c . Para esto aprovecha la segunda cinta de M_3 para ‘desplazar’ partes de la palabra escrita sobre la tercera cinta.

- e) [0.2 puntos] Demuestra que para todo $i > 0$, antes de realizar el primer cleanup de orden mayor a i , y entre todo par de cleanups de orden mayor a i , necesariamente se debe realizar un cleanup de orden i .
- f) [0.3 puntos] Demuestra que para todo $i > 0$, antes de realizar el primer cleanup de orden mayor a i , y entre todo par de cleanups de orden mayor a i , necesariamente se debe realizar al menos $2^{i-1} - 1$ cleanups de orden menor o igual a i .
- g) [0.2 puntos] Demuestra que dado $i > 0$, la cantidad de cleanups de orden i al simular los primeros n pasos de M es a lo más $n/2^{i-2}$.
- h) [0.2 puntos] Demuestra que la cantidad de cleanups de orden i al simular los primeros n pasos de M es igual a cero para cada $i > 2 + \log_2 n$.
- i) [0.3 puntos] Demuestra que la cantidad total de pasos que necesita realizar M_3 para simular los primeros n pasos de M está acotado por $d \cdot (n + n \log_2 n)$, donde d es una constante.
- j) [0.3 puntos] Explica cómo se puede extender la idea anterior para simular una máquina M con k cintas de trabajo que funciona en tiempo $T(n)$ en una máquina con dos cintas de trabajo que funciona en tiempo $O(T(n) \log_2 T(n))$. Para esto especifica el alfabeto de trabajo de esta máquina de tres cintas y explica su funcionamiento.
- k) [0.3 puntos] Extiende la idea anterior para el modelo de máquinas de Turing en el que las cintas son infinitas hacia la derecha.

2. [3 puntos] Sea AUTOMATA el siguiente lenguaje:

$$\text{AUTOMATA} = \{(N, w) \mid N \text{ es un autómata finito no determinista y } w \in L(N)\}.$$

En esta pregunta debes resolver los siguientes problemas.

- a) [0.5 puntos] Demuestra que AUTOMATA está en NLOGSPACE.
- b) [1.0 puntos] Demuestra que CAMINO $\leq_m^{\log}$ AUTOMATA, de lo cual se concluye que AUTOMATA es NLOGSPACE-hard.
- c) [1.5 puntos] Programa ahora la reducción anterior en una máquina de Turing de acuerdo al lenguaje definido en `db.ing.puc.cl/turingmachine`. Para esto debes definir una representación para los grafos y sus nodos, y otra para los autómatas. Te recomendamos pensar bien en las representaciones, ya que en caso de escogerlas mal el tiempo que tomaría programar la reducción puede aumentar notablemente.

Referencias

- [1] F. C. Hennie and R. E. Stearns. Two-tape simulation of multitape turing machines. *J. ACM*, 13(4):533–546, Oct. 1966.