Complejidad Computacional
Tarea 1

Karla Adriana Esquivel Guzmán
Andrea Itzel González Vargas
Luis Pablo Mayo Vega
Carlos Gerardo Acosta Hernández

Entrega: 06/03/17
Facultad de Ciencias UNAM

Ejercicios

1.

Comencemos con la codificación de la máquina de Turing como una cadena binaria.
Sea $M = (Q, \Sigma, \Gamma, \delta, s, r, t, \vdash, \sqcup)$ una máquina de Turing, con

- $Q = i$
- $\Sigma = j$
- $\Gamma = k$
- s el m-ésimo estado
- t el n-ésimo estado
- r el p-ésimo estado
- $\sqcup$ el q-ésimo caracter de Σ
- $\vdash$ el r-ésimo caracter de Σ

Dicha especificación será codificada por la siguiente cadena binaria:

$$0^i 10^j 10^k 10^m 10^n 10^p 10^q 10^r 1. \tag{1}$$

Continuamos con la codificación de una transición en delta[1] de la forma $\delta(s_u, a_v) = (s_w, a_x, \rightarrow)$ como (2), mientras que una transición de la forma $\delta(s_u, a_v) = (s_w, a_x, \leftarrow)$ como (3):

$$0^u 1 0^v 1 0^w 1 0^x 1 0 \tag{2}$$

$$0^u 1 0^v 1 0^w 1 0^x 1 0 0 \tag{3}$$

De esta manera, podemos codificar completamente la tabla original de M.
Finalmente, una cadena de entrada en Γ^* de la forma $a_{n_1} \dots a_{n_z}$, quedará codificada por la cadena binaria:

$$0^{n_1} 1 0^{n_2} 1 \dots 0^{n_z} \tag{4}$$

Sólo resta decir para completar la codificación que podemos designar con $\langle M \rangle$ a la concatenación de (1) junto con las codificaciones de la forma (2) y (3), mientras que a la entrada codificada como en (4), la designaremos $\langle \alpha \rangle$. Combinadas en una sola cadena, podemos usar un símbolo de separación predefinido como # para obtener: $\langle M \rangle \# \langle \alpha \rangle$.

Recordemos que la Máquina Universal de Turing U que simulará a M, consta de tres cintas, además de la de entrada y salida. La primera de ellas será utilizada para la descripción de M, es decir, la tabla que quedó codificada para cada transición δ como en (3); la segunda, contendrá lo que corresponde con la cinta original de M; y la tercera el estado en el que la M se encontraría.

Al iniciar U, en la primera cinta ya debe tener la tabla de M codificada, mientras que en la tercera cinta aparecerá un 0, indicando que se encuentra en el estado inicial de la máquina original.

Separados por unos, podemos considerar una cadena binaria proveniente de una codificación de transición en δ, como una quinteta, por los 5 elementos en total que componen una transición. Para emular a M, la Máquina Universal de Turing U, deberá hacer lo siguiente por cada transición de la TM original:

1. Localiza en la primera cinta aquella quinteta en la que el primer componente sea igual al que se encuentra en la tercera cinta y el segundo componente sea igual al que se encuentra en la segunda cinta.

2. Copia a la tercera cinta el tercer componente de la primera cinta (el estado al que se transfiere).

3. Sustituye la segunda cinta con lo que tiene el cuarto componente.

4. Interpetra el movimiento de la cabeza que se encuentra en el quinto componente, moviendo acorde la cabeza de la segunda cinta.

[1] Con las cadenas finales, después del último 1 separador $0 := \rightarrow$ y $00 := \leftarrow$.

2.

Demuestra que los siguientes problemas sobre gráficas (es decir, los lenguajes respectivos) están en P (elige la representación que prefieras para las gráficas):

- CONNECTED: el conjunto de todas las gráficas conexas.

  Por el ejemplo de la pagina 25 del libro Arora sabemos que decidir si dos vértices son conexos toma tiempo polinomial. Por lo que podemos esbosar un algoritmo para decidir la coneccidad de una grafica de la siguiente manera:

  Checamos para cada par de vértices conectados por una arista dentro de G si son conexos. Si esto se cumple entonces la gráfica es Conexa y está en P. Esta en P por que a lo mas checamos n*n pares de vertices y cada ves que checamos nos toma tiempo polinomial (por lo mencionado anteriormente), entonces el tiempo que tarda nuestro algoritmo es polinomial.

- BIPARTITE: el conjunto de todas las gráficas bipartitas, es decir, aquellas cuyos vértices puedan ser divididos en dos conjuntos A y B tales que todas las aristas en la gráfica tengan en un extremo un vértice de A y en el otro uno de B.

  Sea G una gráfica.

  Tomamos todos los vértices que no tengan adyacencias entre sí y los colocamos en un mismo conjunto al que denotaremos A. Este paso toma a lo mas (n^2).

  Checamos los vértices sobrantes y comprobamos que no haya adyacencias entre ellos si hay alguna adyacencia entre ellos entonces no hacemos nada pues G no es bipartita, si no hay adayacencias entonces los colocamos en un segundo conjunto que denotaremos como B, si por alguna razón no sobraron vértices entonces simplemente no hacemos nada porque entonces G no es bipartita. Este paso toma a lo mas (n^2) pues revisas por segunda vez los nodos para checar las adyacencias.

  Una vez que tengamos nuestros conjuntos A y B checamos que en cada uno de los vértices exista al menos una adyacencia que vaya de cada uno de los vértices del conjunto A a los vértices del conjunto B y viceversa. Este paso toma a lo mas (n^2) pues se tienen que revisar las adyacencias entre cada uno de los vértices dentro de nuestros conjuntos A y B.

  Este algoritmo nos tomaria a lo mas $3(n^2)$ y esto es tiempo polinomial.

3.

Demostración:

Primero supongamos que $L_S^2 \in P$. Esto quiere decir que la representación binaria en base 2 toma un tiempo polinomial. Además, sabemos que convertir de base 2 a otra con codificación binaria nos lleva un tiempo $O(n^2)$ con algoritmos como "Double Dabble" que sigue siendo polinomial.
Por otro lado, suponiendo que $L_S^b \in P$, para convertir de la representación binaria de una base a la base 2 se puede pasar el número a su lenguaje original ($\{0, \ldots, b-1\}$) en tiempo $O(n)$ y después

convertir a la representación en L_S^2 en $O(\log n)$ que es polinomial.
Por lo tanto, la base no tiene ningún efecto sobre la pertenencia en P.

4.

Demuestra que los siguientes lenguajes están en NP:

- 2COL = {<G> | <G> codifica una gráfica que tiene una 2-coloración}.

  Probar que 2COL $\in$ NP.

  Entrada para $TM =< A, w, c >$.

  Donde $A = G$ y G es una gráfica.

  c es Una Lista de nodos coloreados.

  w se desccribe de la siguiente manera.

  1. Se checa que todos los nodos de G se encuentran en la Lista.
  2. Se checa que todos los nodos de la Lista se encuentren en G.
  3. Checar que para cualesquiera dos nodos conectados por una arista (adyacentes) serán de colores distintos, respectivamente C1 y C2.
  4. Acepta si 1,2 y 3 se cumplen.

- 3COL = {<G> | <G> codifica una gráfica que tiene una 3-coloración}.

  Probar que 3COL $\in$ NP.

  Entrada para $TM =< A, w, c >$.

  Donde $A = G$ y G es una gráfica.

  c es Una Lista de nodos coloreados.

  w se desccribe de la siguiente manera.

  1. Se checa que todos los nodos de G se encuentran en la Lista.
  2. Se checa que todos los nodos de la Lista se encuentren en G.
  3. Checar que para cualesquiera dos nodos conectados por una arista (adyacentes) serán de colores distintos, respectivamente C1 y C2.
  4. Acepta si 1,2 y 3 se cumplen.

- ¿Cuáles están en P?.

  2COL $\in$ P porque una grafica es 2-coloreable si es una gráfica bipartita y por el ejercico 2 sabemos que esto esta en P.

  3COL $\in$ NP-Duro según la literatura.