Autómatas y Lenguajes Formales 2017-1
Facultad de Ciencias UNAM
Nota de Clase 14

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

23 de noviembre de 2016

1. Introducción

Las máquinas de Turing como modelos de cómputo que tienen varios usos, como aceptadoras de lenguajes, como generadoras de lenguajes y como calculadoras de funciones, tienen una generalización para realizar cómputos de forma "universal".

La idea de proporcionar una máquina capaz de realizar cualquier cómputo utilizando una memoria para almacenar datos fue propuesta por A. Turing en 1936 en un artículo donde describe estas máquinas pero sobre todo resalta la idea de que las máquinas pueden realizar los cálculos de cualquier algoritmo concebido.

La tesis de Church-Turing afirma que la noción intuitiva de algoritmo es capturada de manera exacta por la noción matemática de máquina de Turing:

Tesis de Church-Turing: Un problema es soluble algorítmicamente si y sólo si es soluble mediante una máquina de Turing.

Es decir, las máquinas de Turing implementan a cualquier algoritmo. Equivalentemente, una función es computable si y sólo si es soluble mediante una máquina de Turing.

Estudiaremos en esta nota las máquinas universales así como sus codificiaciones.

1.1. La Máquina Universal

Pensar en la existencia de una máquina de Turing que se comporte de la misma forma que una computadora real es posible. Es decir, una máquina que sea útil para múltiples propósitos y que sea también capaz de programar y ejecutar máquinas de Turing.

A. Turing describió a la máquina universal $\mathcal{M}$ como una máquina que tiene programa para resolver un problema que es codificado como una máquina específica o de propósito especial M_1 y un conjunto de datos a procesarse que será la entrada de M_1.

Así la máquina universal $\mathcal{M}$ simulará el comportamiento de M_1 que procesará los datos de entrada.

2. Una codificación de máquinas de Turing

Para llevar a cabo la simulación es necesario que las máquinas de propósito especial junto con sus datos de entrada sean codificados en cadenas para que $\mathcal{M}$ pueda procesarlos. Veamos una forma de codificación de una máquina específica para obtener una cadena.
Se tienen las siguientes convenciones:

- Se fija un alfabeto de entrada Σ.
- Se asume $Q = \{q_0, \dots, q_n\}$ siendo q_0 el estado inicial y q_1 el único estado final.
- El alfabeto de la cinta es de la forma $\Gamma = \{s_1, s_2, \dots, s_p\}$ siendo $s_1 = \sqcup$.

Codificación de una cinta Las siguientes características definen una codificación para la cinta de una máquina dada:

- El símbolo s_i se codifica como 1^i.
- Las cadenas $\Gamma^\star$ se codifican separando cada símbolo con un 0.
- En general, si $w = s_{n_1} s_{n_2} \dots s_{n_k}$ la codificación de w es: $01^{n_1}01^{n_2}0\dots01^{n_k}0$

Por ejemplo, si $\Gamma = \{\sqcup, a, b\}$ entonces codificamos $\sqcup := 1,\ a := 11,\ b := 111$. Así, la palabra $bab\sqcup aa$ se codifica como: 011101101110101101110

Codificación de los estados Considerando que los estados de una máquina están dados por $Q = \{q_0, q_1, \dots, q_n\}$ entonces se pueden codificar de la siguiente forma:

- El estado q_i (en la posición i del conjunto Q visto como lista) se codifica análogamente a los símbolos, es decir mediante cadenas de 1's pero con un símbolo más que la posición del estado:

  $$q_i \text{ se codifica como } 1^{i+1}$$

- Así el estado inicial q_0 se codifica con 1 y el estado final q_1 con 11.

Codificación de las direcciones de desplazamiento Consideramos que una máquina puede tener tres desplazamientos $\{\rightarrow, \leftarrow, -\}$ codificados respectivamente como 1, 11 y 111.

Codificación de las transiciones Recordemos que una transición en una máquina de Turing es $\delta(q_i, s_k) = (q_j, b_\ell, D)$. Entones, los estados, símbolos y dirección de desplazamiento se codifican de la manera indicada anteriormente. Y la transición se codifica escribiendo en orden los códigos respectivos separados por ceros:

$$01^{i+1}01^k01^{j+1}01^\ell01^n0 \quad \text{donde } n = 1, 2, 3$$

Por ejemplo $\delta(q_2, s_3) = (q_0, s_5, \leftarrow)$ se codifica como $01^301^30101^501^20$

Una máquina de Turing queda completamente determinada mediante su función de transición δ. Y se codifica mediante la sucesión de los códigos de sus transiciones sin separaciones. Es decir, si $C_1, \dots, C_k$ son los códigos de todas las transiciones de M. Entonces M se codifica mediante $C_1C_2 \dots C_k$

Obsérvese que no hay ambigüedad pues cada transición tiene exactamente seis ceros, además dos ceros consecutivos indican que inicia otra transición.

Ejemplo: Considere la máquina M dada por:

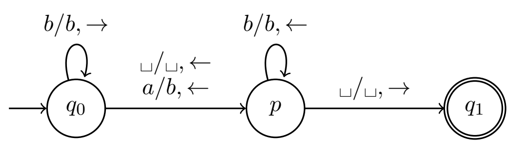
$$\begin{array}{ll} \delta(q_0, a) = (q_2, b, \rightarrow) & \delta(q_2, b) = (q_3, c, \rightarrow) \\ \delta(q_3, a) = (q_1, c, \rightarrow) & \delta(q_1, b) = (q_3, \sqcup, -) \end{array}$$

Se codifica mediante $a := 11,\ b := 111, c := 1111$ como sigue:

0101101110111010011101110111101111010

01111011011011110100110111011110101110

Ejemplo: Considere el siguiente macro que modifica la a de más a la izquierda de una cadena cambiándola por una b y dejando la cabeza lectora al inicio de la cadena:



La codificación es la siguiente:

01011101011101001011011101110110

0101011101011001110111011101110110

01110101101010

donde $\sqcup := 1$, $a := 11$, $b := 111$ y los estados se definen como $q_0 := 1$, $q_1 := 11$ y $p := 111$.

Observaciones:

- La codificación de una máquina de Turing no es única, puesto que el orden de las transiciones no importa y un orden distinto genera una codificación distinta.
- De hecho si una máquina M tiene n transiciones, existen $n!$ codificaciones distintas para M.
- El proceso de codificación puede revertirse, no es difícil definir un algoritmo que decida si una secuencia binaria representa un código válido para una máquina de Turing y en tal caso lo decodifique.

3. Uso de la Máquina Universal

Como mencionamos antes, la máquina universal como dispositivo de programación y ejecución de programas es principalmente la simulación de una máquina con un propósito específico.

Veamos una forma de cómo simular una máquina de Turing M que se ejecutará con cierta entrada w en la máquina universal $\mathcal{M}$:

- Codificar M y w, respectivamente las llamaremos $\mathcal{C}_M$ y $\mathcal{C}_w$
- La máquina universal $\mathcal{M}$ tiene 5 cintas:
  - T1 la cinta de entrada que sólo almacenará la cadena de entrada codificada $\mathcal{C}_w$.
  - T2 la cinta de descripción que contiene la codificación $\mathcal{C}_M$ y la cual no se reescribirá.
  - T3 la cinta de trabajo que contendrá exactamente la cinta de M.
  - T4 la cinta para saber el estado actual de M.
  - T5 una cinta especial para guardar la dirección en que se moverá la cabeza de M.
- La fase de inicialización tiene las siguentes acciones:
  1. copiar la cadena de entrada en la cinta T3
  2. inicializar T4 con el código del estado inicial de M
  3. almacenar los códigos de las direcciones en T5
  4. mover las cabezas de las 5 cintas a la parte más izquierda de las cadenas que almacenan.
- La máquina universal $\mathcal{M}$ está en un ciclo continuo para la simulación, es decir que simulará cada transición M en cada iteración.
- El ciclo de la máquina inicia siempre con las cabezas de las cintas lo más la izquierda de los datos almacenados excepto por la cinta T3 cuya cabeza está en el estado actual.
- El ciclo tiene varios pasos:
  1. La máquina universal $\mathcal{M}$ se mueve en la codificación $\mathcal{C}_M$ para escoger una transición adecuada al comparar las entradas y los datos en las otras cintas correspondientes.
  2. Cuando encuentra la transición deseada copia el nuevo estado en T4, el símbolo en T3 y mueve la cabeza de T3 según la dirección.
  3. Al final del ciclo mueve las cabezas de T2 y T4 a la izquierda.

Esta codificación es exagerada en el número de cintas en $\mathcal{M}$ ya que se pueden simplificar a menos e incluso se puede usar una sóla cinta. El hecho de tener los datos en diferentes cintas facilita la selección de la transición dado que todo está codificado.

La parte de simulación sólo se describirá dado que la máquina universal es un aparato muy complejo pero se puede pensar que se tienen macros para cada una de las acciones de la máquina.