Curso:
Teoría de la Computación.
Unidad 1, Sesión 6: Otros formalismos

Instituto de Computación, Facultad de Ingeniería
Universidad de la República, Montevideo, Uruguay

dictado semestre 2 - 2009

Contenido:

1. Máquina de Turing
   (a) MT
   (b) MT de k-cintas
   (c) MT de acceso randómico
   (d) MT randómica
2. Caracterización de Kleene
3. Gramáticas Irrestrictas

Aquí presentamos mayor evidencia que avale la tesis de Church. Mencionaremos otros modelos de computabilidad que difieren radicalmente de los programa **while**, pero sin embargo definen la misma clase de funciones computables.

Comenzaremos dando una breve introducción a uno de los primeros modelos para la especificación de algoritmos, las máqunas de Turing.

Máquina de Turing

Una máquina de Turing (MT) es un autómata finito determinista, que consta de una cinta finita que puede ser extendida indefinidamente hacia la derecha. Esta cinta se encuentra dividida en *celdas*, las cuales pueden contener: un símbolo especial B que representa un blanco, el símbolo $\triangleright$ índica el límite izquierdo en la cinta, y cualquier otro símbolo de un alfabeto finito Σ. El autómata consta de estados y un cabezal de lectura/escritura que apunta a una celda, en cada instante el autómata puede estar en un único estado y el cabezal apuntando a una única celda. También existe una función de transición que determina para un cierto estado y el símbolo apuntado por el cabezal de lectura, cual será el próximo estado, el símbolo que sobrescribirá al actualmente leído y a su vez como mover el cabezal de lectura/escritura. El movimiento del cabezal de lectura estará limitado a quedarse en el lugar o moverse una única celda hacia la izquierda o derecha. Se asume que siempre que la cinta se extiende hacia la derecha se encuentra un símbolo B en esta.

MT

Formalizando, una MT es una quintupla $M = (Q, X, q_0, q_H, \delta)$, donde:

- Q es un conjunto de estados
- $X = \Sigma \cup \{B, \triangleright\}$ es el conjunto de símbolos de la cinta, donde $B, \triangleright \notin \Sigma$
- $q_0 \in Q$ es el estado inicial
- $q_H \in Q$ es el estado de parada
- $\delta : Q \times X \to Q \times X \times \{L, N, R\}$ es la función parcial de transición, tal que cumple las siguientes restricciones:
  1. $\delta(q_H, x)$ está indefinida para todo estado $x \in X$
  2. $\forall q \in (Q - \{q_H\}), \exists p \in Q, \delta(q, \triangleright) = (p, \triangleright, R)$
  3. $\forall q \in (Q - \{q_H\}), \forall a, b \in X, \exists p \in Q, \exists m \in \{L, N, R\}$, si $\delta(q, a) = (p, b, m)$ entonces $b \neq \triangleright$

Veamos que podemos caracterizar el estado de una MT en un cierto instante, lo que llamaremos *configuración*, por una cuaterna perteneciente a $Q \times X^* \times X \times X^*$ (* representa la Clausura de Kleene de un conjunto de símbolos Σ, la cual es el conjunto de secuencias de símbolos que se pueden formar sobre el conjunto de símbolos Σ, donde ε representa la secuencia vacía de símbolos), compuestas del estado, los símbolos a la derecha del cabezal de lectura/escritura, el símbolo leído por el cabezal y la cinta a la derecha del cabezal.

Las MT parecen ser máquinas procesadores de símbolos sobre un alfabeto Σ, pero veamos que pueden ser vistas como máquinas computadoras de funciones numéricas. Para esto suponemos la notación unaria de los naturales, y se define que una cierta MT M computa la función $f : N^k \to N$ si y sólo si partiendo del estado $(q_0, \triangleright, B, 1^{n_1}B1^{n_2}B \ldots B1^{n_k})$ la MT M para en el estado $(q_H, \triangleright, B, 1^{f(n)}B^m)$, con $m \in N$. Veremos a continuación la MT que computa la función sucesor.

Ejemplo: **MT computadora de la función sucesor.** Sea $\Sigma = \{1\}, Q = \{q_0, q_1, q_H\}$ y

$$\delta = \{(q_0, B, q_1, B, R), (q_1, 1, q_1, 1, R), (q_1, B, q_2, 1, L), (q_2, 1, q_2, 1, L), (q_2, B, q_H, B, N)\}$$

Se deja como ejercicio hallar la MT que computa la función suma entre naturales.

Finalmente se puede ir al libro de referencia del curso ("A programming approach to computability". A.J. Kfoury. Robert N. Moll. Michael A. Arbib. 1982) para ver una prueba inductiva que las MT pueden computar al menos las funciones computables por los programas **while**. Por otro lado, se deja como ejercicio ver que los programas **while** pueden a su vez computar cualquier función computable por una MT. Llegando así a la equivalencia de ambos modelos.

A su vez veremos a continuación algunas extensiones a la MT, se puede ir a "Elements of the Theory of Computation" de Harry R. Lewis y Christos H. Papadimitriou para un mayor detalle. Estas extensiones a la MT buscan reafirmar la tesis de Church, buscando modelos de computación de alguna forma más poderosos, justamente agregándole más poder a las MT, pero llegando indefectiblemente a que siguen siendo equivalentes a la MT básica.

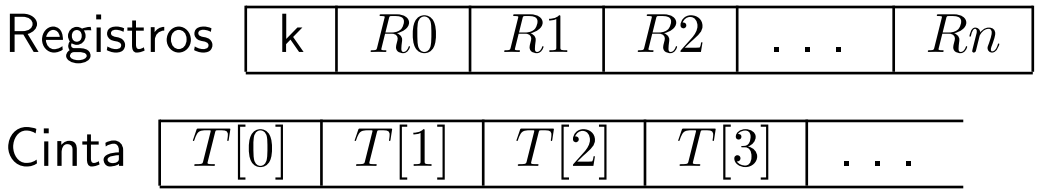
MT de k-cintas

Una primera extensión es la MT de múltiples cintas, cada una de ellas con su propio cabezal de lectura/escritura. Se demuestra que una MT de k-cintas es equivalente a una MT básica con un alfabeto extendido Σ' tal que $\Sigma' = \Sigma \cup (\Sigma \times \{0,1\})^k)$. Por cada cinta se tiene una cinta de ceros y unos, donde un único uno indica la posición del cabezal de lectura en cada una de las k cintas.

En el libro de referencia ("A programming approach to computability". A.J. Kfoury. Robert N. Moll. Michael A. Arbib. 1982) se presenta la MT con una cinta extensible en ambas direcciones, esta variante de la MT presentada en dicho texto, se la puede ver como un caso especial de una MT de múltiples cintas, específicamente una MT de dos cintas.

MT de acceso randómico

Una limitante que de las MTs vistas hasta el momento es el acceso secuencial a la memoria, en contraste las computadoras reales tienen memorias de *acceso randómico*. Para esto además de la cinta, dotamos de un conjunto registros finito $R0, R1, \ldots, Rn$, capaces de almacenar y manipular direcciones en la cinta. Este autómata actúa sobre la cinta y los registros de la forma dictada por un conjunto de instrucciones finito y fijo. A su vez, tenemos un registro especial *k*, el cual almacena el *program counter*, dicho registro almacena la dirección de la próxima instrucción a ser ejecutada. A la cinta las podemos ver como un arreglo T de naturales infinitamente extensible hacia la derecha.



A continuación enumeramos el conjunto de instrucciones posibles para una MT randómica:

Intrucción	Operando	Semántica
read	j	$R0 := T[Rj]$
write	j	$T[Rj] := R0$
store	j	$Rj := R0$
load	j	$R0 := Rj$
load	$= c$	$R0 := c$
add	j	$R0 := R0 + Rj$
add	$= c$	$R0 := R0 + c$
sub	j	$R0 := max(R0 - Rj, 0)$
sub	$= c$	$R0 := max(R0 - c, 0)$
half		$R0 := \lfloor \frac{R0}{2} \rfloor$
jump	s	$k := s$
jpos	s	if $R0 > 0$ then $k := s$
jzero	s	if $R0 = 0$ then $k := s$
halt		$k := 0$ provoca que el programa termine

Formalmente una MT de acceso randómico es una tupla $M = (k, \Pi,$ donde $k > 0$ es el número de registros, y $\Pi = \{\pi_1, \pi_2, \ldots, \pi_p\}$ es una secuencia finita de instrucciones que forman el programa. Asumimos que la última instrucción π_p es siempre halt. Una *configuración* es una $k+2$ tupla $(k, R0, R1, \ldots, Rn, T)$, donde:

- $k \in N, p \geq k \geq 0$ es el program counter, si $k = 0$ diremos que es una configuración de parada
- $\forall j, 0 \leq j < k, Rj \in N$ es el valor del registro j
- T representa el contenido de la cinta y es representado por un subconjunto de $(N - \{0\}) \times (N - \{0\})$ tal que $\forall i \geq 1$, existe a lo sumo un par de la forma $(i, m) \in T$, se asume que siempre que el arreglo se extiende hacia la derecha se encuentra un 0 en este.

Podemos ver a la MT de acceso randómico con k registros como una máquina computadora de funciones $f : N \to N$, utilizando la notación unaria de los naturales, si para todo $n \in N$ partiendo de la configuración

$$(1, 0, \ldots, 0, \{\underbrace{(1,1),(2,1),\ldots,(n,1)}_{n}\})$$

llegamos a una configuración de parada de la siguiente forma,

$$(0, n_1, \ldots n_k, \{\underbrace{(1,1),(2,1),\ldots,(f(n),1)}_{f(n)}\}) \text{ con } n_1, \ldots, n_k \in N$$

Ejemplo: **MT de acceso randómico computadora de la función sucesor.** Sea $\Sigma = \{1\}$, $M =$ $(2,$ load 1, add =1, store 1, read 1, jzero 7, jump 1, load =1, write 1,halt) ♣

MT randómica

La MT randómica es una MT clásica en la cual la función δ pasa a ser una relación, con lo cual deja de ser determinista la opción a seguir dado un estado y una lectura del cabezal de lectura/escritura.

Se demuestra que estas MT son equivalentes a una MT de tres cintas y por ende a la MT clásica. Para esto es utilizada la técnica de *dovetailing*, dicha técnica es utilizada para ir recorriendo todas las posibles configuraciones en que se puede encontrar en un instante dado la MT y avanzar cada una de ellas un paso, hasta encontrar una configuración de parada.

Caracterización de Kleene

La habilidad de usar definiciones recursivas para definir nuevas funciones basándose en un conjunto finito ya definido es la característica distintiva de la caracterización de Kleene de las funciones computables. Existen dos pasos en la formulación de Kleene. Primero define las funciones recursivas primitivas, las cuales forman la mayoría de las funciones matemáticas posibles. Sin embargo todas estas funciones son totales, por lo que no puede formar todas las funciones computables. La función de Ackermann es un ejemplo de función no primitiva recursiva. La segunda etapa, extiende las funciones recursivas primitivas, agregándole un operador adicional, el operador μ, el cual introduce búsquedas infinitas, introduciendo por tanto las funciones parciales.

Gramáticas Irrestrictas

Veremos a continuación otro modelo de computación equivalente a la MT, los *Semi-Thue Systems* o también conocido como gramáticas irrestrictas.

Una gramática irrestricta T sobre un alfabeto Σ es una terna ordenada $T = (\Sigma, P, V)$, donde V es un conjunto de símbolos extra y P es un conjunto de reglas o producciones de la forma $\alpha \to \beta$, tal que $\alpha \in (\Sigma \cup V)^+$ y $\beta \in (\Sigma \cup V)^*$. Escribimos $x \Rightarrow_T y$, con $x, y \in (\Sigma \cup V)^*$ para indicar que x puede ser escrita como $u\alpha w$, $\alpha \to \beta$ es una producción de P y $u\beta w = y$. Llamamos $\Rightarrow_T^*$ a la clausura transitiva de la relación $\Rightarrow_T$.

La forma natural de ver las gramáticas es como generadoras de lenguajes, pero también se pueden ver como computadoras de funciones.

Definición de **Función computada por una gramática irrestricta.**

Dados: Una función parcial $f : \Sigma^* \to \Sigma^*$.

Escribimos: Una gramática irrestricta G computa f

Condición: $SwS \Rightarrow_T^* v \Leftrightarrow v = f(w)$

♣

Ejemplo: **Gramática irrestricta computadora de la función sucesor.**

♣ $G = (\{1\}, \{1S \to S1, SS \to 1\}, \{S\})$, veamos un ejemplo de computo $S11S \Rightarrow_T S1S1 \Rightarrow_T SS11 \Rightarrow_T 111$