\>\>\> IF013 - Fundamentos Teóricos de Informática

\>\>\> Licenciatura de Sistemas - UNPSJB - Sede Trelew

Name: Celia Cintas[†], Pablo Navarro[‡], Samuel Almonacid[§]

Date: September 4, 2017

[†] cintas@cenpat-conicet.gob.ar, cintas.celia@gmail.com, @RTFMCelia

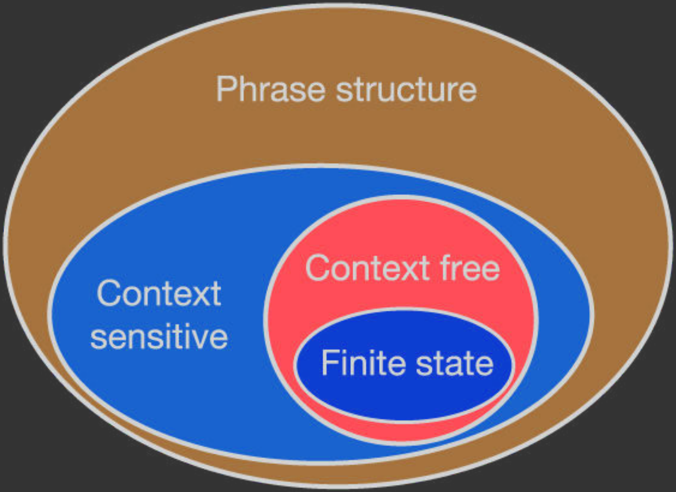
[‡] pnavarro@cenpat-conicet.gob.ar, pablo1n7@gmail.com

[§] almonacid@cenpat-conicet.gob.ar, almonacid.samuel.tw@gmail.com

>>> Unidad 3

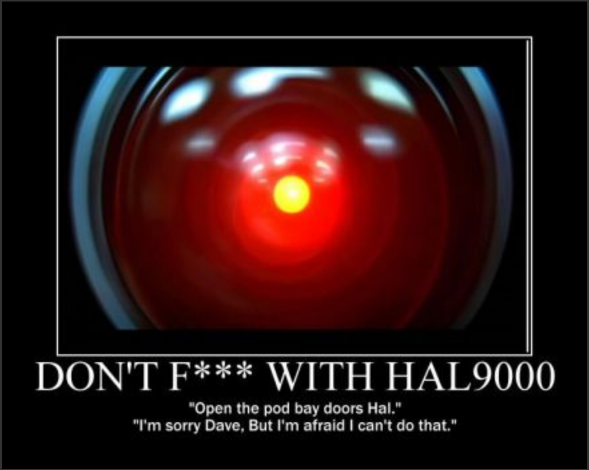
1. Gramáticas sensibles al contexto. Autómatas acotados linealmente.
2. Propiedades formales de los lenguajes sensibles al contexto.
3. Maquinas de Turing. Reconocedoras y Generadoras.
4. Tesis Turing-Church. Limitaciones de la Computabilidad Efectiva. Maquinas de Turing No Determinísticas.
5. Problema de detención de Máquina de Turing.
6. Reductibilidad.
7. Lenguajes y gramáticas sin restricciones.
8. Enumerabilidad de los lenguajes sensibles al contexto y argumento diagonal de Turing.

>>> Clasificación de Chomsky



Computational and volutionary aspects of language. Martin A. Nowak, Natalia L. Komarova and Partha Niyogi, 2002. Nature.

>>> Procesamiento de Lenguaje Natural



http://www.nltk.org/

Definición

Sea $G = (V_n, V_t, S, X_0, P)$ una gramática entonces G es una gramática sensible al contexto (GSC) si toda producción perteneciente a P de la forma $\alpha \rightarrow \beta$ es tal que $long(\alpha) \leq long(\beta)$, con $\alpha \in (V_n \cup V_t)^* V_n (V_n \cup V_t)^*$, $\beta \in (V_n \cup V_t)^*$.

>>> Gramáticas Sensibles al Contexto (Cont.)

$$G = (V_n, V_t, X_0, P)$$
$$V_n = \{X_0, X_1, X_2\}$$
$$V_t = \{a, b, c\}$$

$$P = \{X_0 \to abc | aX_1bc$$
$$X_1b \to bX_1$$
$$X_1c \to X_2bcc$$
$$bX_2 \to X_2b$$
$$aX_2 \to aaX_1$$
$$aX_2 \to aa\}$$

$$X_0 \to aX_1bc \to abX_1c$$
$$\to abX_2bcc \to aX_2bbcc \to aabbcc$$

>>> Gramáticas Sensibles al Contexto (Cont.)

$$G = (V_n, V_t, X_0, P)$$
$$V_n = \{X_0, X_1, X_2\}$$
$$V_t = \{a, b, c\}$$

$$P = \{X_0 \to abc | aX_1bc$$
$$X_1b \to bX_1$$
$$X_1c \to X_2bcc$$
$$bX_2 \to X_2b$$
$$aX_2 \to aaX_1$$
$$aX_2 \to aa\}$$

$$X_0 \to aX_1bc \to abX_1c$$
$$\to abX_2bcc \to aX_2bbcc \to aabbcc$$

L = $\{a^nb^nc^n | n \geq 1\}$

\>>> Propiedades de LSC

Teorema
La clase de los lenguajes sensible al contexto es cerrada bajo: Unión, Concatenación, Estrella de Kleene, Intersección, Sustitución libre de λ y Homomorfismo libre de λ.

	Lenguajes Regulares	Lenguajes Libres del Contexto	Lenguajes Sensibles al Contexto	Lenguajes Estructurados por Frases
	L_3	L_2	L_1	L_0
Unión	Sí	Sí	Sí	Sí
Concatenación	Sí	Sí	Sí	Sí
Estrella de Kleene	Sí	Sí	Sí	Sí
Intersección con un LenguajeRegular	Sí	Sí	Sí	Sí
Reversa	Sí	Sí	Sí	Sí
Sustitución Libre de λ	Sí	Sí	Sí	Sí
Sustitución	Sí	Sí	Sí	Sí
Homomorfismo	Sí	Sí	Sí	Sí
Intersección	Sí	No	Sí	Sí
Complemento	Sí	No	?	No

? = Teorema Immerman-Szelepcsényi

\>>> Automata Linealmente Acotado (ALA)

Definición

Es una máquina de Turing no-determinística que cumple con las siguientes condiciones:

* El alfabeto incluye dos elementos @ y \$, utilizados como topes.
* El ALA no se mueve ni a la izquierda de @ ni a la derecha de \$ ni los sobreescribe.

Definición

El ALA $M = (Q, \Sigma, \Gamma, \delta, q_0, @, \$, F)$ acepta el lenguaje L dado por:

$L = \{w | w \in (\Sigma - \{@, \$\})^* \; y \; q_0 @ w \$ \vdash^* \alpha q \beta, q \in F\}$

>>> Máquina de Turing

Definición

Un Procedimiento Efectivo es un conjunto de reglas escritas en un determinado lenguaje que son interpretables y ejecutables por una máquina.

* Está formado por una secuencia finita de instrucciones o sentencias.
* Existe un procesador capaz de interpretar las instrucciones y producir resultados predecibles y repetibles.
* El procesador tiene memoria para almacenar resultados intermedios.
* No existe límite finito ni a la entrada ni a la salida de los datos.
* No existe límite a la cantidad de almacenamiento requerido para realizar la computación.

http://www.dcc.fc.up.pt/~acm/turing-phd.pdf

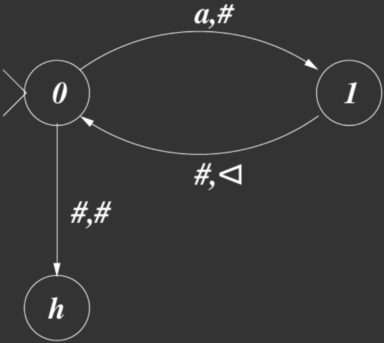
>>> Máquina de Turing (Cont.)

Definición

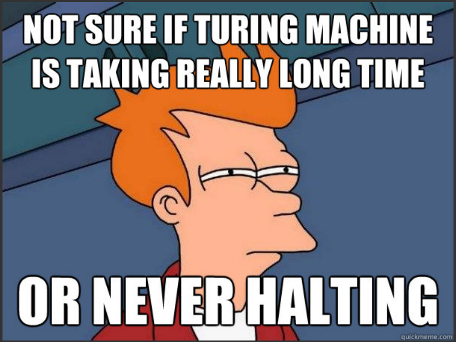
Una máquina de Turing es una quintupla $T = (S, \Sigma, \delta, s_0, F)$ donde:

* S es el conjunto de estados, $S \neq \emptyset$.
* Σ es el alfabeto de trabajo.
* δ es una función parcial, $\delta : S \times \Sigma \rightarrow S \times \Sigma \times \{I, D, N\}$ donde:
    * I denota: movimiento a izquierda
    * D denota: movimiento a derecha
    * N denota: sin movimiento
* s_0 es el estado inicial, $s_0 \in S$.
* F es el conjunto de estados finales, $F \subseteq S$.

>>> Máquina de Turing (Cont.)



δ	0	1
a	1,#	1,a
#	h,#	0,$\triangleleft$



MT Reconocedoras

Definición

Llamaremos configuración de una máquina de Turing $T = (S, \Sigma, \delta, s_0, F)$ a una terna (s, α, i), donde s es el estado corriente de T , $\alpha \in \Sigma^*$ e $i \in \mathbb{Z}^+$ que marca la posición donde está el cabezal.

Definición

Una transición de un máquina de Turing T será representada por la relación binaria $\vdash$ entre configuraciones:
$(s, \alpha, i) \vdash (s', \alpha', i')$ si existe en T una regla de transición $\delta(s, \alpha) = (s', \alpha', M)$ donde $s, s \in S; \alpha, \alpha' \in \Sigma^*; M \in \{I, N, D\}$, e

$$i' = \begin{cases} i - 1 \; si \; M = I \\ i \; si \; M = N \\ i + 1 \; si \; M = D \end{cases}$$

>>> Maquina de Turing (Cont.)

Definición

Se dice que una cadena w es aceptada o reconocida por una máquina de Turing $T = (S, \Sigma, \delta, s_0, F)$ si $(s_0, w, 1) \vdash^* (s_f, \alpha, i)$ para algún $s_f \in F$, $w, \alpha \in \Sigma^*, i \geq \mathbb{Z}^+$.

Definición

Dado un alfabeto Σ y un lenguaje $L \subseteq \Sigma^*$, L es aceptado por una máquina de Turing $T = (S, \Sigma, \delta, s_0, F)$ si:

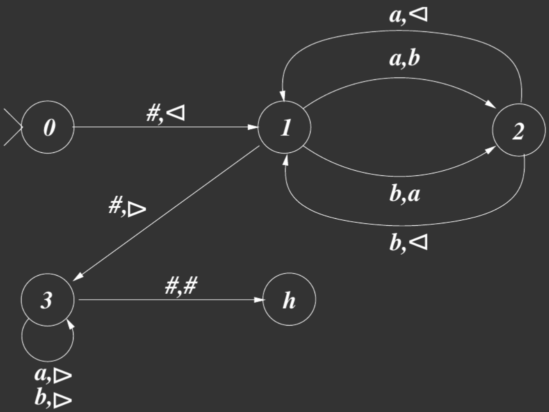
$L = L(T) = \{w | w \ \in \ \Sigma^* \ y \ w \ es \ aceptada \ por \ T\}$

En tales casos diremos que L es un lenguaje Turing Aceptable.

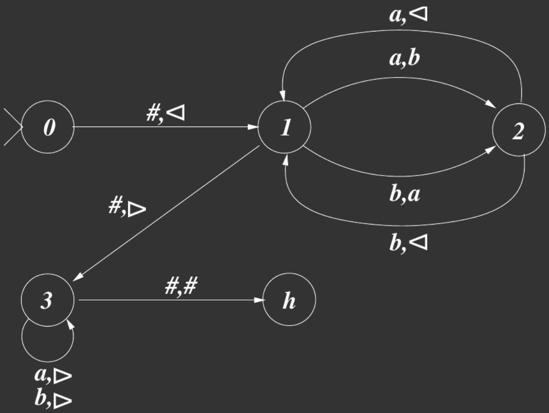
MT para Computar Funciones

Definición

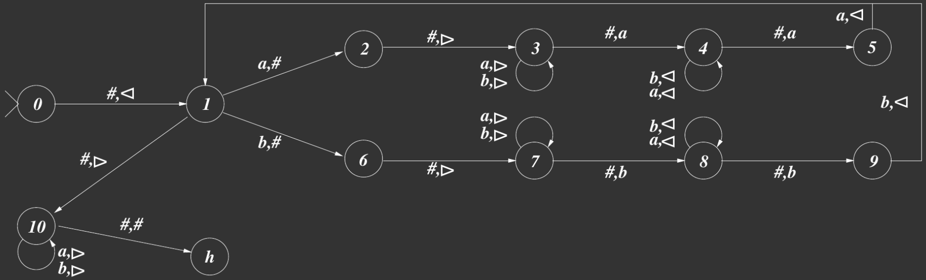
Se dice que una función $f_T : \Sigma^* \to \Sigma^*$ es computable por una máquina de Turing (Turing-computable) si existe una máquina de Turing $T = (S, \Sigma, \delta, s_0, F)$ tal que:
$(s_0, x, 1) \vdash^* (s_f, f_T(x), i)$ donde $w \in \Sigma, i \in \mathbb{Z}$ y $s_f \in F$.



>>> Máquina de Turing (Cont.)



$$f_T(w) = \bar{w}$$



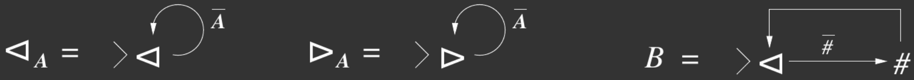
$$f_T(w) = ww^R$$

>>> Máquina de Turing (Cont.)

Definición

Las acciones básicas de la notación modular de MTs son:

* Moverse hacia la izquierda MT denominada $\lhd$.
* Moverse hacia la derecha MT denominada $\rhd$.
* Escribe $b, b \in \Sigma$.
* $\lhd_A$ y $\rhd_A$ se mueven hacia una dirección hasta encontrar el carácter A.
* La MT denominada B borra la cadena a su izquierda.



>>> Máquina de Turing (Cont.)

>>> Máquina de Turing (Cont.)

1. Busque hacia la izquierda hasta encontrar aa (dos a seguidas) y pare.
2. Decida el lenguaje $\{w \in \{a, b\}^*$, w contiene al menos una a$\}$.
3. Acepte el lenguaje a^*ba^*b.
4. Decida el lenguaje $\{w \in \{a, b\}^*, w$ contiene tantas a's como b's$\}$.

>>> Máquina de Turing y Computabilidad

Definición

Sea Σ un alfabeto y sean Y, N dos símbolos $\notin \Sigma$. Un lenguaje $L \subseteq \Sigma^*$ se dice decidible por una máquina de Turing (o Turing-decidible) si y sólo si la función $X_L : \Sigma^* \to \{Y, N\}$ es Turing-computable, y para cada $w \in \Sigma^*$, dá una respuesta acorde con:

$$X_L(w) =_{def} \begin{cases} Y si\ w \in L \\ N si\ w \notin L \end{cases}$$

Definición

Una máquina de Turing no determinista es una quintupla $T = (S, \Sigma, \delta, s_0, F)$ donde:

S es el conjunto de estados, $S \neq \emptyset$

Σ es el alfabeto de entrada.

$\delta : S \times \Sigma \to P(S \times \Sigma \times \{I, D, N\})$.

s_0 es el estado inicial, $s_0 \in S$.

F es el conjunto de estados finales, $F \subseteq S$.

>>> Máquina de Turing y Computabilidad (Cont.)

Teorema

Para cada máquina de Turing no determinista T_{nd} podemos construir una máquina de Turing determinista T_d equivalente.

>>> Tesis Turing-Church

Tesis 1 (de Turing) Un proceso naturalmente llamado procedimiento efectivo puede ser realizado por una máquina de Turing, es decir es Turing Computable.

Tesis 2 (de Church) Los procesos naturalmente llamados procedimientos efectivos o las funciones efectivamente computables son identificados con la clase de Funciones Recursivas Parciales.

>>> Tesis Turing-Church (Cont.)

1. Máquinas de Alan Turing (1936).
2. Teoría de Funciones Recursivas Parciales de K. Gödel y S. Kleene (1936).
3. λ-Cálculo de Alonzo Church (1941).
4. Sistemas Canónicos de Post (1943).
5. Algoritmo Generalizado de Markov (1951).
6. Gramáticas estructuradas por frases de Noam Chomsky (1959-1963).
7. Máquinas de registros de Shepardson y Sturgis (1963).
8. Redes de Petri (1962)

>>> Limitaciones de la Computabilidad Efectiva

Definición

Denominaremos Problema de Decisión a aquellos formulados a traves de una pregunta y que requieren una respuesta de tipo Si/No.

Definición

Un problema de decisión se dice Soluble si existe un algoritmo total para determinar si la propiedad es verdadera.

Definición

Un problema de decisión se dice Parcialmente Soluble si existe un procedimiento efectivo para determinar si la propiedad es verdadera.

Definición

Un problema de decisión es Insoluble si no existe un procedimiento efectivo para determinar si la propiedad es válida.

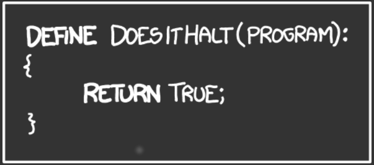
>>> Problema de la detención de la MT

Definición

Sean una máquina de Turing T y una cadena α, el problema de la detención de las máquinas de Turing se define como: ¿Existe un algoritmo para decidir si T se detendrá comenzando en el estado inicial con α en la cinta?

Teorema

El problema de la detención es Algorítmicamente Insoluble.



THE BIG PICTURE SOLUTION TO THE HALTING PROBLEM

Demostración

>>> Reductibilidad

Definición

Sean dos problemas de decisión, PD_1 y PD_2, diremos que el problema de decisión PD_1 se reduce al problema de decisión PD_2 y lo escribiremos $PD_1 \rightarrow_{reduct} PD_2$ si un algoritmo para solucionar PD_2 puede ser usado para construir la solución de PD_1.

Teorema

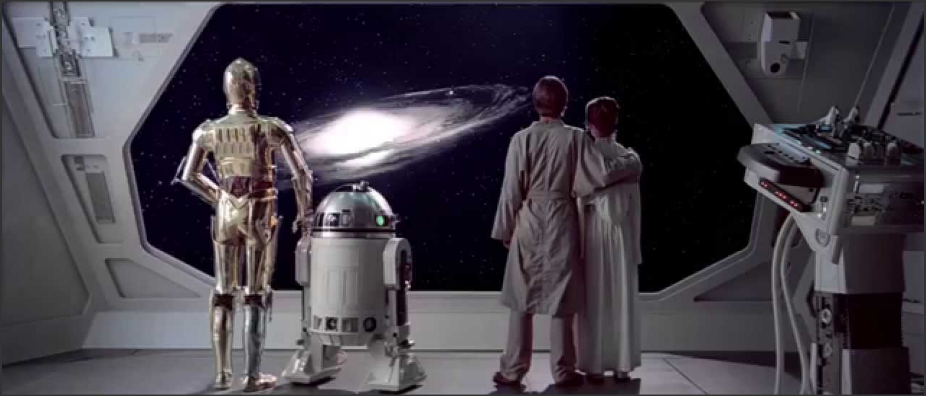
Sean PD_1 y PD_2 dos problemas de decisión, se cumple lo siguiente:

1. Si $PD_1 \rightarrow_{reduct} PD_2$ y PD_2 es soluble, entonces PD_1 también lo es.
2. Si $PD_1 \rightarrow_{reduct} PD_2$ es insoluble, entonces PD_2 también lo es.

>>> Lenguajes Recursivos y recursivos enumerables

something here

>>> Gracias!



Bibliografía

1. Introduction to Automata Theory, Languages, and Computation - Hopcroft et. al 2007 (3er ed.)
2. Teoría de la Computación - Gonzalo Navarro 2011.
3. Fundamentos de Cs. de la Computación - Juan Carlos Augusto 1995.