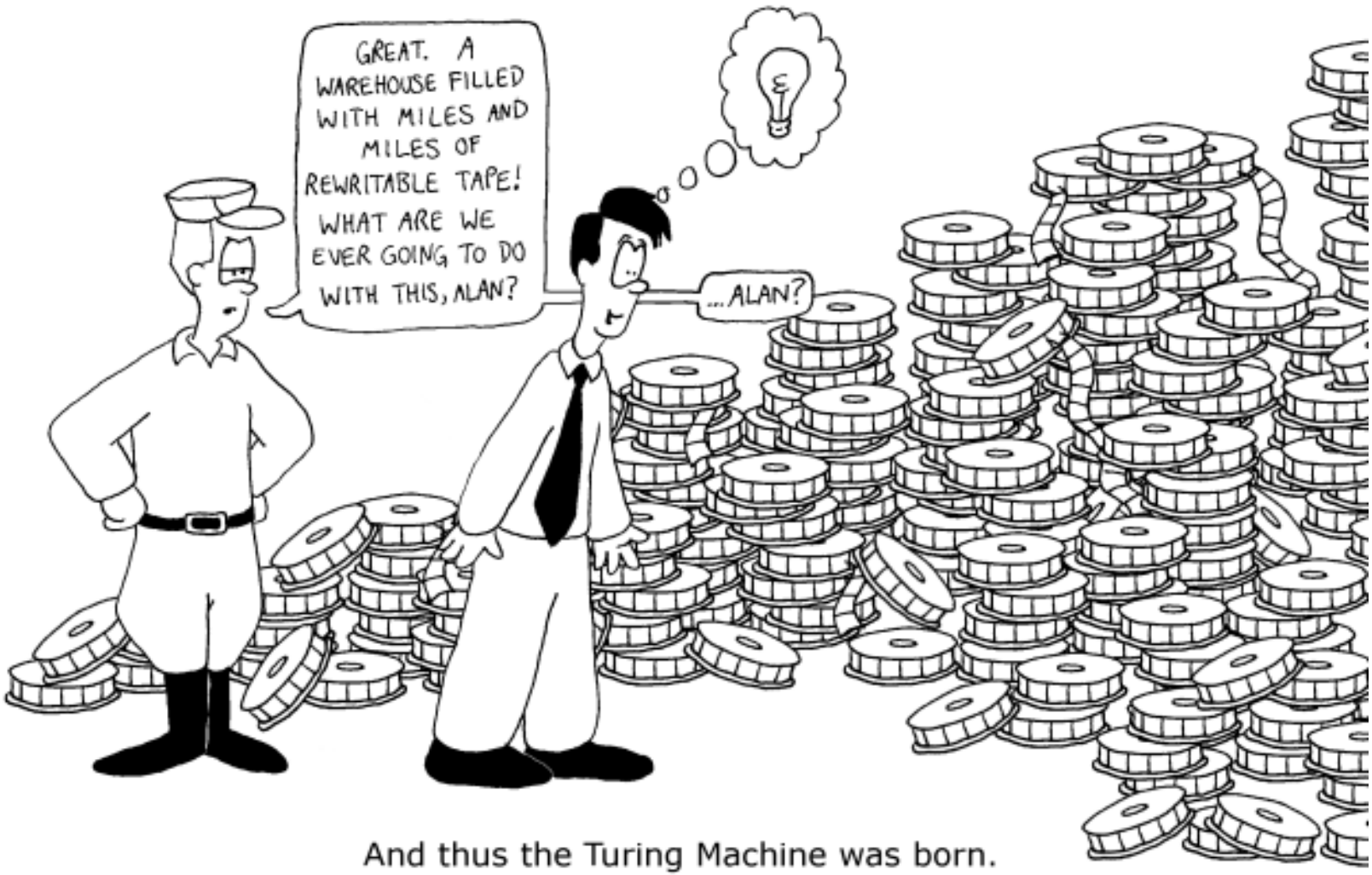
Fundamentos Teóricos de Informática

Pablo Toledo

UNPSJB

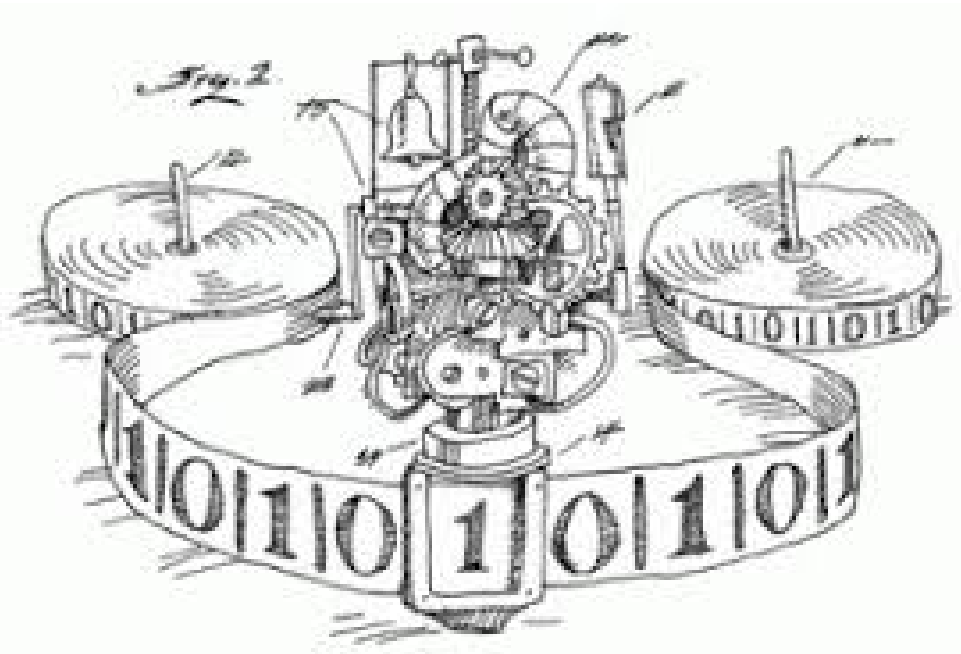
Feb 27, 2014

Máquinas de Turing



¿Qué es una TM?

- Un modelo que define una *Máquina abstracta* que manipula símbolos sobre una (o más cintas) de acuerdo a un conjunto de reglas.



Máquinas de Turing (cont...)

- Inventada en 1936 por Alan Turing
- Con esta se puede probar las limitaciones fundamentales de los algoritmos
- En particular, Turing la utilizó para resolver el **Halting Problem** o **Problema de la Detención...** y ganar WW2 (y probablemente haber salvado al mundo)

when the Brits are asked how they thanked Turing for saving the world

¿Pero cómo es una Máquina de Turing?

Una Máquina de Turing es una quintupla

$$T = (S, \sum, \delta, s_0, F)$$

Donde

- S es el conjunto de estados posibles
- $\sum$ es el alfabeto de trabajo
- δ es una funcion parcial
  $\delta : S \times \sum \to S \times \sum \times \{I, N, D\}$
  Donde
  - I denota movimiento a la izquierda
  - D denota movimiento a la derecha
  - N sin movimiento
- $F \subseteq S$ es el conjunto de estados finalizadores

Configuración

Llamaremos configuración de una $T = (S, \sum, \delta, s_0, F)$ a una terna (s, α, i) donde s es el estado acutal de T, $\alpha \in \sum^*$ (la secuencia de caracteres acutalmente sobre la cinta) e $i \in \mathbb{Z}^+$ siendo la posición de la cabeza lectora en la cinta.

Transición

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

$$i' = \begin{cases} i - 1 & \text{si M=I} \\ i & \text{si M=N} \\ i + 1 & \text{si M=D} \end{cases} \qquad (1)$$

Aceptación de una cadena y Lenguaje

Definición

Se dice que w es reconocida por $T = (S, \sum, \delta, s_0, F)$ si $(s_0, w, 1) \vdash^* (s_f, \alpha, i)$ para algún $s_f \in F, w, \alpha \in \sum^*, i \in \mathbb{Z}^+$

Definición

Dado un alfabeto $\sum$ y un lenguaje $L \subseteq \sum^*$, L es aceptado por $T = (S, \sum, \delta, s_0, F)$ si:
$L = L(T) = \{w | w \in \sum^* \text{ y } w \text{ es aceptada por } T\}$
En tal caso diremos que L es un lenguaje **Turing Aceptable**

MT para computar funciones

Definición

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

Computabilidad

Sea $\sum$ un alfabeto y sean Y, N dos símbolos fuera de éste ($\notin \sum$). Un lenguaje $L \subseteq \sum^*$ se dice *decibile por una Máquina de Turing* o *Turing-Decidible* si y sólo si la función $X_L : \sum \to \{Y, N\}$ es Turing-computable, y para cada $w \in \sum^*$, da una respuesta acorde

$$X_L(w) =_{def} \begin{cases} \text{Y si w} \in \text{L} \\ \text{N si w} \notin \text{L} \end{cases} \tag{2}$$

Máquina de Turing NO Determinista

Una máquina de Turing no determinista se puede definir de la siguiente manera:

$$T = (S, \sum, \delta, s_0, F)$$

Donde todos los elementos son los mismos de una Máquina de Turing determinista, salvo que δ se define de la siguiente manera

$$\delta : S \times \sum \rightarrow P(S \times \sum \times \{I, D, N\})$$

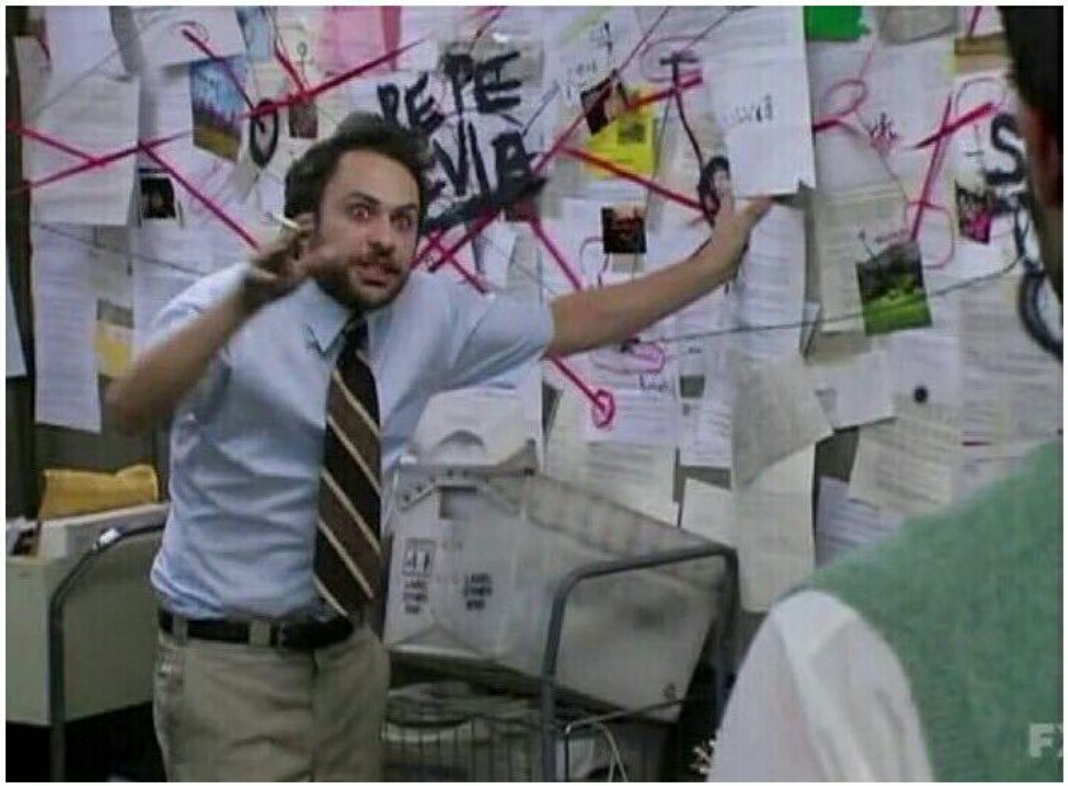
Teorema Para cada máquina de Turing no determinista T nd podemos construir una máquina de Turing determinista T d equivalente.

Tesis 1 (de Turing)
Un proceso naturalmente llamadopro cedimiento 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.

- Teoría de Funciones Recursivas Parciales de K. Godel y S. Kleene (1936)
- Cálculo Lambda de Alonzo Church (1941).
- Sistemas Canónicos de Post(1943).
- Redes de Petri (1962)

when you are a computer scientist and realize that many computational models are equivalent



when you learn to make an if statement using lambda calculus

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.

waiting for your turing machine to halt

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.

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

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 $PD_1 \rightarrow PD_2$ es insoluble, entonces PD_2 también lo es.