Guía 4: El paradigma de Turing

Tres paradigmas de la computabilidad efectiva

- En la guía anterior se vieron los conceptos de:
  - *Procedimiento efectivo*
  - *Función Σ-efectivamente computable*
  - *Conjunto Σ-efectivamente computable*
  - *Conjunto Σ-efectivamente enumerable*
    Los cuales fueron definidos de forma intuitiva e imprecisa, sin formalismo matemático.
    Debido a que son fundamentales en el estudio teórico de la computabilidad, es muy importante poder dar un modelo o formalización matemática de estos conceptos.
- Como los conceptos de conjunto Σ-efectivamente computable y conjunto Σ-efectivamente enumerable se desprenden del concepto de función Σ-efectivamente computable, una formalización matemática de este último concepto es suficiente para poder modelizar los otros dos.
- Veremos en las siguientes guías las tres formalizaciones matemáticas más clásicas del concepto de función Σ-efectivamente computable, las cuales corresponden a tres paradigmas de la computabilidad efectiva:
  - **Paradigma de Turing**
  - **Paradigma de Godel**
  - **Paradigma imperativo de Von Neumann**

El paradigma de Turing

Descripción informal de las máquinas de Turing

- *Descripción*:
  - Modelo abstracto de máquina con una cantidad finita de estados (Q) que trabaja sobre una cinta de papel dividida en cuadros e interactúa o recibe acciones externas por medio de una cabeza lectora
  - La cabeza lectora lee de a un cuadro de la cinta a la vez, puede borrar el contenido del cuadro leído y escribir en él un símbolo, y también puede moverse un cuadro a la izquierda o a la derecha
  - La cinta tiene un primer cuadro hacia su izquierda pero hacia la derecha puede extenderse todo lo necesario
  - En un cuadro de la cinta podrá haber un símbolo o un cuadro puede simplemente estar en blanco

        - Es decir, habrá un alfabeto Γ el cual consiste de todos los símbolos que pueden figurar en la cinta
        - En Γ se incluye el símbolo que significa que el cuadro está en blanco (B)
    - La máquina, en función del estado en que se encuentre y d elo que vea su cabeza lectora en el cuadro escaneado, podrá modificar lo que encuentre en dicho cuadro (borrando y escribiendo algún nuevo símbolo), moverse a lo sumo un cuadro (izquierda, derecha o quedarse en el mismo lugar) y cambiar de estado (posiblemente al mismo estado)
    - Cada máquina tiene un estado especial, el cual será llamado su *estado inicial* (q_0) y será el estado en el que estará la máquina al comenzar a trabajar sobre la cinta
    - Las máquinas poseen un *alfabeto de entrada* que está contenido en Γ y en el cual están los símbolos que se usarán para formar la configuración inicial de la cinta (excepto B)
        - En general, se denotará con Σ al alfabeto de entrada y $\Gamma - \Sigma$ contiene los *símbolos auxiliares*
- *Detención*:
    - Puede pasar que para un determinado estado p y un $\sigma \in \Gamma$, la máquina no tenga contemplada ninguna acción posible, por lo que se detiene
    - Otro caso, es cuando está escaneando el primer cuadro de la cinta y su única acción posible implica moverse un cuadro a la izquierda
    - Todas las palabras para las cuales pase esto, formarán al conjunto $H(M)$, el cual será el conjunto de palabras aceptadas por detención por la máquina M
- *Alcance de estado final*:
    - Habrá un conjunto $F \subseteq Q$ cuyos elementos serán llamados *estados finales*.
    - Diremos que una palabra α es aceptada por la máquina si, al comenzar a trabajar sobre la cinta con la configuración inicial $B\alpha_1, .., \alpha_n, B$, la máquina llega a un estado de F
    - Todas estas palabras formarán al conjunto $L(M)$, el cual será el conjunto de palabras aceptadas por alcance final por la máquina M

Definición matemática de máquina de Turing

- Una máquina de Turing es una 7-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ donde:
    - Q es un conjunto finito cuyos elementos son llamados *estados*
    - Γ es un alfabeto que contiene a Σ
    - Σ es un alfabeto llamado el *alfabeto de entrada*
    - $B \in \Gamma - \Sigma$ es un símbolo de Γ llamado el *blank symbol*
    - $\delta : D_\delta \subseteq Q \times \Gamma \to Q \times \Gamma \times \{L, R, K\}$
    - q_0 es un estado llamado el *estado inicial* de M
    - $F \subseteq Q$ es un conjunto de estados llamados *finales*

- Notar que δ da la *personalidad* de la máquina ya que:
  - $\delta(p,\sigma) = (q,\gamma,L)$ significa que la máquina estando en estado p y leyendo σ, lo borrará, escribirá un γ en su lugar, se moverá un cuadro a la izquierda y cambiará al estado q (en caso que no esté en el primer cuadro)
  - $\delta(p,\sigma) = (q,\gamma,R)$ significa que la máquina estando en estado p y leyendo σ, lo borrará, escribirá un γ en su lugar, se moverá un cuadro a la derecha y cambiará al estado q
  - $\delta(p,\sigma) = (q,\gamma,K)$ significa que la máquina estando en estado p y leyendo σ, lo borrará, escribirá un γ en su lugar, no se moverá y cambiará al estado q

Descripciones instantáneas

- Una *descripción instantánea* será una palabra de la forma $\alpha q\beta$, donde $\alpha,\beta \in \Gamma^*, [\beta]_{|\beta|} \neq B$ y $q \in Q$
- La descripción instantánea $\alpha_1 .. \alpha_n q \beta_1 .. \beta_m$, con $\alpha_1, .., \alpha_n, \beta_1, .., \beta_m \in \Gamma^*, n, m \geq 0$ representará la siguiente situación:

$$\alpha_1, \alpha_2, .., \alpha_n, \beta_1(q), \beta_2, .., \beta_m, B, B$$

- Sea Des el conjunto de todas las descripciones instantáneas, definimos $St : Des \to Q$ tal que $St(d) =$único símbolo de Q que ocurre en d

Relación $\vdash$

- *Conceptos necesarios*
  - Dado $\alpha \in (Q \cup \Gamma)*$, definamos $\lfloor \alpha \rfloor$ como el resultado de remover de α el tramo final más grande de la forma B^n. Es decir:

$$\lfloor \varepsilon \rfloor = \varepsilon$$

$$\lfloor \alpha\sigma \rfloor = \alpha\sigma \text{ si } \sigma \neq B$$

$$\lfloor \alpha B \rfloor = \lfloor \alpha \rfloor$$

  - Dada cualquier palabra $alpha$ definimos las funciones:

$${}^{\curvearrowright}\alpha = \begin{cases} [\alpha]_2 .. [\alpha]_{|\alpha|} & \text{si } |\alpha| \geq 2 \\ \varepsilon & \text{si } |\alpha| \leq 1 \end{cases}$$

$$\alpha^{\curvearrowleft} = \begin{cases} [\alpha]_1 .. [\alpha]_{|\alpha|-1} & \text{si } |\alpha| \geq 2 \\ \varepsilon & \text{si } |\alpha| \leq 1 \end{cases}$$

- *Definición*: Dadas $d_1, d_2 \in Des$, diremos que $d_1 \vdash d_2$ si $\exists \sigma \in \Gamma, \alpha, \beta \in \Gamma^*, p, q \in Q$ tales que se cumple alguno de los siguientes:
  - $d_1 = \alpha p \beta, \delta(p, [\beta B]_1) = (q, \sigma, R), d_2 = \alpha\sigma q {}^{\curvearrowright}\beta$
  - $d_1 = \alpha p \beta, \delta(p, [\beta B]_1) = (q, \sigma, L), \alpha \neq \varepsilon, d_2 = \lfloor \alpha^{\curvearrowleft} q [\alpha]_{|\alpha|} \sigma {}^{\curvearrowright}\beta \rfloor$
  - $d_1 = \alpha p \beta, \delta(p, [\beta B]_1) = (q, \sigma, K), d_2 = \lfloor \alpha q \sigma {}^{\curvearrowright}\beta \rfloor$
- Escribiremos $d \nvdash d'$ si no se cumple que $d \vdash d'$

- Para $d, d' \in Des, n \geq 0$, escribiremos $d \overset{n}{\vdash} d'$ si $\exists d_1, .., d_{n+1}$ tales que $d = d_1, d' = d_{n+1}$ y $d_i \vdash d_{i+1}$ para $i = 1, .., n$
  - *Notar que* $d \overset{0}{\vdash} d' \Leftrightarrow d = d'$
- Definimos $d \overset{*}{\vdash} d' \Leftrightarrow \exists n \in \omega : d \overset{n}{\vdash} d'$

Detención

- Para $d \in Des$, diremos que M se detiene partiendo de d si existe $d' \in Des$ tal que:
  - $d \overset{*}{\vdash} d'$
  - $d' \nvdash d'' \forall d'' \in Des$
- **El lenguaje** $L(M)$
  - Diremos que una palabra $\alpha \in \Sigma^*$ es aceptada por M por *alcance de estado final* cuando $\lfloor q_0 B\alpha \rfloor \overset{*}{\vdash} d$, con $d : St(d) \in F$.
  - Luego, el *lenguaje aceptado por M por alcance de estado final* se define como $L(M) = \{\alpha \in \Sigma^* : \alpha \text{ es aceptada por } M \text{ por alcance de estado final}\}$
- **El lenguaje** $H(M)$
  - Diremos que una palabra $\alpha \in \Sigma^*$ es aceptada por M por *detención* cuando M se detiene partiendo de $\lfloor q_0 B\alpha \rfloor$.
  - Luego, el *lenguaje aceptado por M por detención* se define como $H(M) = \{\alpha \in \Sigma^* : \alpha \text{ es aceptada por } M \text{ por detención}\}$
- *Lemma*: Sea $L \subseteq \Sigma^*$, entonces son equivalentes:
  - Existe una máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ tal que $L = L(M)$
  - Existe una máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ tal que $L = H(M)$

Funciones Σ-Turing computables

- *Concepto previo*: Para poder computar funciones mixtas con una máquina de Turing, necesitaremos un símbolo para poder representar números sobre la cinta. A este símbolo lo llamaremos *unit* y lo denotaremos con $\shortmid$.
  - Más formalmente, una máquina de Turing es una 8-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una máquina de Turing y $\shortmid \in \Gamma - (\{B\} \cup \Sigma)$
- *Definición*:
  - Diremos que $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ es Σ-Turing computable si existe una máquina de Turing con *unit* $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que:
    - Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces $\exists p \in Q : \lfloor q_0 B \shortmid^{x_1} B..B \shortmid^{x_n} B\alpha_1 B..B\alpha_m \rfloor \overset{*}{\vdash} \lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor$ y $\lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor \nvdash d \forall d \in Des$
    - Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M **no** se detiene partiendo de $\lfloor q_0 B \shortmid^{x_1} B..B \shortmid^{x_n} B\alpha_1 B..B\alpha_m \rfloor$
  - En forma similar, una función $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$ es Σ-Turing computable si existe una máquina de Turing con *unit* $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que:

- Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces $\exists p \in Q$: $\lfloor q_0 B \mid^{x_1} B..B \mid^{x_n} B\alpha_1 B..B\alpha_m \rfloor \overset{*}{\vdash} \lfloor pB \mid^{f(\vec{x},\vec{\alpha})} \rfloor$ y $\lfloor pB \mid^{f(\vec{x},\vec{\alpha})} \rfloor \not\vdash d \forall d \in Des$
    - Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M **no** se detiene partiendo de $\lfloor q_0 B \mid^{x_1} B..B \mid^{x_n} B\alpha_1 B..B\alpha_m \rfloor$
  - En estos casos, decimos que f es computada por M.
- *Propiedades*:
  - Si f : $D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ con $O \in \{\omega, \Sigma^*\}$ es computada por una máquina de Turing con unit $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \mid, F)$, entonces f es Σ-efectivamente computable
  - Para toda función Σ-efectivamente computable f : $D_f \subseteq \omega^n \times \Sigma^{*m} \rightarrow O$ con $O \in \{\omega, \Sigma^*\}$, existe una máquina de Turing con unit $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \mid, F)$ que computa a f