Guía 7: El paradigma imperativo de Neumann: El lenguaje $\mathcal{S}^\Sigma$

Sintaxis de $\mathcal{S}^\Sigma$

- *Definiciones*:
  - Definimos $Sig : Num^* \rightarrow Num^*$ como:

$$\begin{aligned} Sig(\varepsilon) &= 1 \\ Sig(\alpha 0) &= \alpha 1 \\ Sig(\alpha 1) &= \alpha 2 \\ Sig(\alpha 2) &= \alpha 3 \\ Sig(\alpha 3) &= \alpha 4 \\ Sig(\alpha 4) &= \alpha 5 \\ Sig(\alpha 5) &= \alpha 6 \\ Sig(\alpha 6) &= \alpha 7 \\ Sig(\alpha 7) &= \alpha 8 \\ Sig(\alpha 8) &= \alpha 9 \\ Sig(\alpha 9) &= Sig(\alpha)0 \end{aligned}$$

  - Definimos $Dec : \omega \rightarrow Num^*$ como:

$$\begin{aligned} Dec(0) &= \varepsilon \\ Dec(n+1) &= Sig(Dec(n)) \end{aligned}$$

    - Notar que para $n \in N$, $Dec(n)$ es la notación usual decimal de n
    - Para hacer más ágil la notación, escribiremos $\bar{n}$ en lugar de $Dec(n)$, por lo que $Dec = \lambda n[\bar{n}]$
  - *Sintaxis*: La sintaxis de $\mathcal{S}^\Sigma$ será dada utilizando el alfabeto $\Sigma \cup \Sigma_P$ donde

$$\Sigma_P = Num \cap \{\leftarrow, +, \dot{-}, ., \neq, ^\frown, \varepsilon, N, K, P, L, I, F, G, O, T, B, E, S\}$$

  - Las palabras de la forma:
    - $N\bar{k}$ con $k \in N$ son llamadas *variables numéricas de* $\mathcal{S}^\Sigma$
    - $P\bar{k}$ con $k \in N$ son llamadas *variables alfabéticas de* $\mathcal{S}^\Sigma$
    - $L\bar{k}$ con $k \in N$ son llamadas *labels de* $\mathcal{S}^\Sigma$
  - Una *instrucción básica de* $\mathcal{S}^\Sigma$ es una palabra $(\Sigma \cup \Sigma_P)^*$ la cual es de alguna de las siguientes formas (donde $a \in \Sigma;\ k, n \in N$):
    - $N\bar{k} \leftarrow N\bar{k} + 1$
    - $N\bar{k} \leftarrow N\bar{k} \dot{-} 1$
      - Si el contenido de $N\bar{k}$ es 0, dejarlo sin modificar. Caso contrario, disminuir en 1 el contenido de $N\bar{k}$
    - $N\bar{k} \leftarrow N\bar{n}$
      - Copiar el contenido de $N\bar{n}$ en $N\bar{k}$ sin modificar el contenido de $N\bar{n}$

        - $N\bar{k} \leftarrow 0$
        - $P\bar{k} \leftarrow P\bar{k}.a$
            - Modificar el contenido de $P\bar{k}$ agregando el símbolo a a la derecha
        - $P\bar{k} \leftarrow {}^{\curvearrowright}P\bar{k}$
            - Si el contenido de $P\bar{k}$ es ε, dejarlo sin modificar. Caso contrario, eliminar el primer símbolo del contenido de $P\bar{k}$
        - $P\bar{k} \leftarrow P\bar{n}$
        - $P\bar{k} \leftarrow \varepsilon$
        - IF $N\bar{k} \neq 0$ GOTO $L\bar{n}$
        - IF $P\bar{k}$ BEGINS a GOTO $L\bar{n}$
            - Si el contenido de $P\bar{k}$ comienza con el símbolo a, entonces ir a la instrucción con label $L\bar{n}$. Caso contrario, continuar con la siguiente instrucción
        - GOTO $L\bar{n}$
        - SKIP
    - Una *instrucción de* $\mathcal{S}^\Sigma$ es ya sea una instrucción básica de $\mathcal{S}^\Sigma$, o una palabra de la forma αI, donde $\alpha \in \{L\bar{n} : n \in N\}$ e I es una instrucción básica de $\mathcal{S}^\Sigma$.
        - Usaremos Ins^Σ para denotar el conjunto de todas las instrucciones de $\mathcal{S}^\Sigma$
        - Cuando I es de la forma $L\bar{n}J$ con J una instrucción básica, diremos que $L\bar{n}$ es la *label de* I
    - Un *programa* de $\mathcal{S}^\Sigma$ es una palabra de la forma $I_1 I_2 . . I_n$ donde $n \geq 1, I_1, . . , I_n \in Ins^\Sigma$ y además se cumple la *ley de los GOTO*: $\forall i \in \{1, . . , n\}$, si GOTO$L\bar{m}$ es un tramo final de I_i, entonces $\exists j \in \{1, . . , n\}$ tal que I_j tiene label $L\bar{m}$
        - Usaremos Pro^Σ para denotar el conjunto de todos los programas de $\mathcal{S}^\Sigma$
        - Definimos $n(\mathcal{P})$ como la cantidad de instrucciones de $\mathcal{P}$, e $I_i^\mathcal{P}$ como la i-ésima instrucción de $\mathcal{P}$ para $i \in \{1, . . , n(\mathcal{P})\}$. Además, $I_i^\mathcal{P} = \varepsilon$ cuando $i = 0$ o $i > n(\mathcal{P})$
            - Notamos con $\lambda\mathcal{P}[n(\mathcal{P})]$ y $\lambda i\mathcal{P}[I_i^\mathcal{P}]$
- *Lemas*:
    - *Parseo de programas*: Sea Σ un alfabeto finito, se tiene que:
        - Si $I_1 . . I_n = J_1 . . J_m$ con $I_1, . . , I_n, J_1, . . , J_m \in Ins^\Sigma$, entonces $n = m$ y $I_i = J_i \forall i \geq 1$
        - Si $\mathcal{P} \in Pro^\Sigma$, entonces existe una *única* sucesión de instrucciones $I_1, . . , I_n$ tal que $\mathcal{P} = I_1 . . I_n$
          Luego, esto significa que, dado un programa $\mathcal{P}$, tenemos unívocamente determinados $n(\mathcal{P})$ e $I_1, . . , I_{n(\mathcal{P})}$ tales que $\mathcal{P} = I_1 . . I_{n(\mathcal{P})}$

Semántica de $\mathcal{S}^\Sigma$

- *Definiciones*:

- *Bas*: Definimos $Bas : Ins^{\Sigma} \to (\Sigma \cup \Sigma_P)^*$ dada por

$$Bas(I) = \begin{cases} J & \text{si } I \text{ es de la forma } L\bar{k}J \text{ con } J \in Ins^{\Sigma} \\ I & \text{en otro caso} \end{cases}$$

- Asumiremos siempre que en una computación vía un programa de $\mathcal{S}^{\Sigma}$, todas excepto una cantidad finita de las variables numéricas tienen el valor 0 y todas excepto una cantidad finita de las variables alfabéticas tienen el valor ε
- *Estado*: Un estado es un par $(\vec{s}, \vec{\sigma}) = ((s_1, s_2, \dots), (\sigma_1, \sigma_2, \dots)) \in \omega^{[N]} \times \Sigma^{*[N]}$ y, si $i \geq 1$, entonces diremos que s_i es el contenido o valor de la variable $N\bar{i}$ en el estado $(\vec{s}, \vec{\sigma})$ y σ_i es el contenido o valor de la variable $P\bar{i}$ en el estado $(\vec{s}, \vec{\sigma})$
- *Descripción instantánea*: Una descripción instantánea es una terna $(i, \vec{s}, \vec{\sigma})$ tal que $(\vec{s}, \vec{\sigma})$ es un estado e $i \in \omega$.
    - Es decir, $\omega \times \omega^{[N]} \times \Sigma^{*[N]}$ es el conjunto de todas las descripciones instantáneas
    - Intuitivamente, $(i, \vec{s}, \vec{\sigma})$ nos dice que las variables están en el estado $(\vec{s}, \vec{\sigma})$ y que la instrucción que debemos realizar es $I_i^{\mathcal{P}}$
- *Sucesora*: Dado un programa $\mathcal{P}$, definimos $S_{\mathcal{P}} : \omega \times \omega^{[N]} \times \Sigma^{*[N]} \to \omega \times \omega^{[N]} \times \Sigma^{*[N]}$ como la función que asignará a una descripción instantánea $(i, \vec{s}, \vec{\sigma})$ la descripción instantánea sucesora de $(i, \vec{s}, \vec{\sigma})$ con respecto a $\mathcal{P}$. Es decir, hay varios casos posibles:
    - Si $i \notin \{1, .., n(\mathcal{P})\}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i, \vec{s}, \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{k} \dot{-} 1$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, (s_1, .., s_{k-1}, s_k \dot{-} 1, s_{k+1}, ..), \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{k} + 1$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, (s_1, .., s_{k-1}, s_k + 1, s_{k+1}, ..), \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow N\bar{n}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, (s_1, .., s_{k-1}, s_n, s_{k+1}, ..), \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = N\bar{k} \leftarrow 0$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, (s_1, .., s_{k-1}, 0, s_{k+1}, ..), \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = \text{IF } N\bar{k} \neq 0 \text{ GOTO } L\bar{m}$, entonces hay dos posibilidades:
        - Si el valor de $N\bar{k}$ es 0, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, \vec{s}, \vec{\sigma})$
        - Si el valor de $N\bar{k}$ es no nulo, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
    - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow {}^{\curvearrowright}P\bar{k}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, \vec{s}, (\sigma_1, .., \sigma_{k-1}, {}^{\curvearrowright}\sigma_k, \sigma_{k+1}, ..))$
    - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow P\bar{k}.a$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, \vec{s}, (\sigma_1, .., \sigma_{k-1}, \sigma_k a, \sigma_{k+1}, ..))$
    - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow P\bar{n}$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, \vec{s}, (\sigma_1, .., \sigma_{k-1}, \sigma_n, \sigma_{k+1}, ..))$
    - Si $Bas(I_i^{\mathcal{P}}) = P\bar{k} \leftarrow \varepsilon$, entonces $S_{\mathcal{P}}(i, \vec{s}, \vec{\sigma}) = (i+1, \vec{s}, (\sigma_1, .., \sigma_{k-1}, \varepsilon, \sigma_{k+1}, ..))$

- Si $Bas(I_i^{\mathcal{P}}) = \text{IF } P\bar{k} \text{ BEGINS } a \text{ GOTO } L\bar{m}$, entonces hay dos posibilidades:
  - Si el valor de $P\bar{k}$ comienza con a, entonces $S_{\mathcal{P}}(i,\vec{s},\vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
  - Si el valor de $P\bar{k}$ no comienza con a, entonces $S_{\mathcal{P}}(i,\vec{s},\vec{\sigma}) = (i+1,\vec{s},\vec{\sigma})$
- Si $Bas(I_i^{\mathcal{P}}) = \text{GOTO } L\bar{m}$, entonces $S_{\mathcal{P}}(i,\vec{s},\vec{\sigma}) = (\min\{l : I_l^{\mathcal{P}} \text{ tiene label } L\bar{m}\}, \vec{s}, \vec{\sigma})$
- Si $Bas(I_i^{\mathcal{P}}) = \text{SKIP}$, entonces $S_{\mathcal{P}}(i,\vec{s},\vec{\sigma}) = (i+1,\vec{s},\vec{\sigma})$

- *Computación partiendo de un estado*: Dado un programa $\mathcal{P}$ y un estado $(\vec{s},\vec{\sigma})$, a la infinitupla

$$((1,\vec{s},\vec{\sigma}), S_{\mathcal{P}}(1,\vec{s},\vec{\sigma}), S_{\mathcal{P}}(S_{\mathcal{P}}(1,\vec{s},\vec{\sigma})), ..)$$

  la llamaremos la *computación de $\mathcal{P}$ partiendo del estado* $(\vec{s},\vec{\sigma})$.
  - Diremos que $S_{\mathcal{P}}(S_{\mathcal{P}}(\ldots(S_{\mathcal{P}}(1,\vec{s},\vec{\sigma}))\ldots)) = (j,\vec{u},\vec{\eta})$ con $S_{\mathcal{P}}$ aplicado t veces, es la *descripción instantánea obtenida luego de t pasos partiendo del estado* $(\vec{s},\vec{\sigma})$, y $(\vec{u},\vec{\eta})$ es el estado obtenido luego de t pasos partiendo del estado $(\vec{s},\vec{\sigma})$
- *Detención*:
  - Cuando la primer coordenada de $S_{\mathcal{P}}(S_{\mathcal{P}}(\ldots(S_{\mathcal{P}}(1,\vec{s},\vec{\sigma}))\ldots))$ (con $S_{\mathcal{P}}$ aplicado t veces) es $n(\mathcal{P})+1$, diremos que $\mathcal{P}$ se detiene (luego de t pasos), partiendo desde el estado $(\vec{s},\vec{\sigma})$
  - Caso contrario, si ninguna de las primeras coordenadas en la infinitupla de la computación es igual a $n(\mathcal{P})+1$, diremos que $\mathcal{P}$ no se detiene partiendo desde el estado $(\vec{s},\vec{\sigma})$

Funciones Σ-computables

- *Consideraciones*:
  - Dados $x_1,\ldots,x_n \in \omega$ y $\alpha_1,\ldots,\alpha_m \in \Sigma^*$ con $n,m \in \omega$, usaremos $||x_1,\ldots,x_n,\alpha_1,\ldots,\alpha_m||$ para denotar al estado $((x_1,\ldots,x_n,0,\ldots),(\alpha_1,\ldots,\alpha_m,\varepsilon,\ldots))$
  - Dado $\mathcal{P} \in Pro^{\Sigma}$, definamos las funciones $\Psi_{\mathcal{P}}^{n,m,\#}$ y $\Psi_{\mathcal{P}}^{n,m,*}$ como:
    - $D_{\Psi_{\mathcal{P}}^{n,m,\#}} = \{(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m} : \mathcal{P} \text{ termina partiendo de } ||\vec{x},\vec{\alpha}||\}$
    - $D_{\Psi_{\mathcal{P}}^{n,m,*}} = \{(\vec{x},\vec{\alpha}) \in \omega^n \times \Sigma^{*m} : \mathcal{P} \text{ termina partiendo de } ||\vec{x},\vec{\alpha}||\}$
    - $\Psi_{\mathcal{P}}^{n,m,\#}(\vec{x},\vec{\alpha}) = \text{valor de N1 cuando } \mathcal{P} \text{ termina partiendo de } ||\vec{x},\vec{\alpha}||$
    - $\Psi_{\mathcal{P}}^{n,m,*}(\vec{x},\vec{\alpha}) = \text{valor de P1 cuando } \mathcal{P} \text{ termina partiendo de } ||\vec{x},\vec{\alpha}||$
- *Definición de función Σ-computable*: Una función Σ-mixta $f : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ es Σ-computable si existe un programa $\mathcal{P} \in \mathcal{S}^{\Sigma}$ tal que $f = \Psi_{\mathcal{P}}^{n,m,\#}$
  - Se define de forma análoga para funciones Σ-mixtas $f : S \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ con $f = \Psi_{\mathcal{P}}^{n,m,*}$
  - En tal caso decimos que f es *computada* por $\mathcal{P}$
- *Propiedades*:

- Si f es Σ-computable, entonces f es Σ-efectivamente computable

Macros

- Hay dos tipos de macros:
  - Los de asignación: cuando son expandidos nos dan un programa que simula la asignación a una variable dada del resultado de aplicar una función a los contenidos de ciertas otras variables
  - Los de tipo IF: cuando son expandidos nos dan un programa salvo por la ley de los GOTO, el cual direcciona al label cuando se cumple cierta propiedad (predicado) relativa a los contenidos de las variables
- Idea: consideraremos que una *macro* es un "molde" (que llamaremos M), el cual puede ser de la siguiente forma (ejemplo)

$$\begin{array}{ll} & V4 \leftarrow V2 \\ & V5 \leftarrow V3 \\ & V1 \leftarrow V1 \\ A1 & IFV5 \neq 0 \text{ GOTO } A2 \\ & \text{GOTO } A3 \\ A2 & V5 \leftarrow V5 \dot{-} 1 \\ & V1 \leftarrow V1 + 1 \\ & \text{GOTO } A1 \\ A3 & \text{SKIP} \end{array}$$

  Luego, la forma de usar esta *macro* será "reemplazando" cada ocurrencia de $V1, V2, V3$ por las variables numéricas que correspondan, y cada ocurrencia de $V4, V5$ por variables que no nos importen en el programa (*auxiliares*). De igual modo, cada ocurrencia de $A1, A2, A3$ serán reemplazadas por labels auxiliares.
  - A las palabras de la forma $V\bar{n}$, con $n \in N$, las llamaremos *variables numéricas de macro*
  - A las palabras de la forma $W\bar{n}$, con $n \in N$, las llamaremos *variables alfabéticas de macro*
  - A las palabras de la forma $A\bar{n}$, con $n \in N$, las llamaremos *labels de macro*
  - *Nota:* siempre supondremos que la primera instrucción de las macros no es labeleada, porque muchas veces cuando expandamos un macro nos interesará labelear la primera instrucción de dicha expansión

Macros asociados a funciones Σ-computables

- Dada una función $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$, usaremos $[V\overline{n+1} \leftarrow f(V1, .., V\bar{n}, W1, .., W\bar{m})]$ para denotar a la macro M que cumpla las siguientes propiedades
  - Las variables oficiales de M son $V1, .., V\overline{n+1}, W1, .., W\bar{m}$
  - M no tiene labels oficiales
  - Si reemplazamos:

- las variables oficiales de M por variables concretas $N\overline{k_1},..,N\overline{k_{n+1}},P\overline{j_1},..,P\overline{j_m}$,
- las variables auxiliares de M por variables concretas distintas de a dos y NO pertececientes a $\{N\overline{k_1},..,N\overline{k_{n+1}},P\overline{j_1},..,P\overline{j_m}\}$,
- los labels auxiliares de M por labels concretos distintos de a dos,

entonces la palabra obtenida es un programa de $\mathcal{S}^\Sigma$ que denotaremos con $[N\overline{k_{n+1}} \leftarrow f(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})]$ y tiene la siguiente propiedad:
- Si corremos $[N\overline{k_{n+1}} \leftarrow f(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})]$ partiendo de un estado e que asigne a $N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m}$ los valores $x_1,..,x_n,\alpha_1,..,\alpha_m$ respectivamente, entonces independientemente de los valores que les asigne e a las demás variables, se dará que:
- Si $(\vec{x},\vec{\alpha}) \notin D_f$, entonces $[N\overline{k_{n+1}} \leftarrow f(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})]$ **no** se detiene partiendo de e
- Si $(\vec{x},\vec{\alpha}) \in D_f$, entonces $[N\overline{k_{n+1}} \leftarrow f(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})]$ se detiene partiendo de e y llega a un estado e' que cumple que:
- e' le asigna a $N\overline{k_{n+1}}$ el valor $f(x_1,..,x_n,\alpha_1,..,\alpha_m)$
- e' solo puede diferir de e en los valores que le asigna a $N\overline{k_{n+1}}$ o a las variables que fueron a reemplazar a las variables auxiliares de M

  - De forma análoga se define para $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ con $[W\overline{m+1} \leftarrow f(V1,..,V\bar{n},W1,..,W\bar{m})]$
- Propiedades:
  - Sea Σ un alfabeto finito, entonces:
    - Sea $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$ una función Σ-computable, entonces en S^Σ hay un macro $[V\overline{n+1} \leftarrow f(V1,..,V\bar{n},W1,..,W\bar{m})]$
    - Sea $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ una función Σ-computable, entonces en S^Σ hay un macro $[W\overline{m+1} \leftarrow f(V1,..,V\bar{n},W1,..,W\bar{m})]$
  - Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$ es tal que en S^Σ hay un macro $[V\overline{n+1} \leftarrow f(V1,..,V\bar{n},W1,..,W\bar{m})]$, entonces f es Σ-computable

Macros asociados a predicados Σ-computables

- Dado un predicado $P : D_P \subseteq \omega^n \times \Sigma^{*m} \to \omega$, usaremos $[\text{IF } P(V1,..,V\bar{n},W1,..,W\bar{m}) \text{ GOTO } A1]$ para denotar un macro M el cual cumpla las siguientes propiedades:
  - Las variables oficiales de M son $V1,..,V\bar{n},W1,..,W\bar{m}$
  - $A1$ es el único label oficial de M
  - Si reemplazamos
    - las variables oficiales de M por variables concretas $N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m}$,
    - el label oficial $A1$ por el label concreto $L\bar{k}$
    - las variables auxiliares de M por variables concretas distintas de a dos y NO pertececientes a $\{N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m}\}$,

- los labels auxiliares de M por labels concretos distintos de a dos y ninguno de ellos igual a $L\bar{k}$,

  entonces la palabra obtenida es un programa de $\mathcal{S}^{\Sigma}$, salgo por la *ley de los GOTO* respecto de $L\bar{k}$, que denotaremos con $[\mathrm{IF}\ P(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})\ \mathrm{GOTO}\ L\bar{k}]$ y tiene la propiedad de que si lo hacemos correr partiendo de un estado e que asigne a $N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m}$ los valores $x_1,..,x_n,\alpha_1,..,\alpha_m$ respectivamente, entonces independientemente de los valores que les asigne e a las demás variables, se dará que:
  - Si $(\vec{x},\vec{\alpha}) \notin D_P$, entonces $[\mathrm{IF}\ P(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})\ \mathrm{GOTO}\ L\bar{k}]$ **no** se detiene partiendo de e
  - Si $(\vec{x},\vec{\alpha}) \in D_P$ y $P(x_1,..,x_n,\alpha_1,..,\alpha_m) = 1$, entonces, luego de una cantidad finita de pasos, $[\mathrm{IF}\ P(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})\ \mathrm{GOTO}\ L\bar{k}]$ direcciona al label $L\bar{k}$ quedando en un estado e', el cual solo puede diferir de e en los valores que le asigna a las variables que fueron a reemplazar a las variables auxiliares de M
  - Si $(\vec{x},\vec{\alpha}) \in D_P$ y $P(x_1,..,x_n,\alpha_1,..,\alpha_m) = 0$, entonces, luego de una cantidad finita de pasos, $[\mathrm{IF}\ P(N\overline{k_1},..,N\overline{k_n},P\overline{j_1},..,P\overline{j_m})\ \mathrm{GOTO}\ L\bar{k}]$ se detiene partiendo de e, y quedando en un estado e' que solo puede diferir de e en los valores que le asigna a las variables que fueron a reemplazar a las variables auxiliares de M

- Propiedades:
  - Sea $P : D_P \subseteq \omega^n \times \Sigma^{*m} \to \omega$ un predicado Σ-computable, entonces en S^{Σ} hay un macro $[\mathrm{IF}\ P(V1,..,V\bar{n},W1,..,W\bar{m})\ \mathrm{GOTO}\ A1]$
  - Si $P : D_P \subseteq \omega^n \times \Sigma^{*m} \to \omega$ es tal que en S^{Σ} hay un macro $[\mathrm{IF}\ P(V1,..,V\bar{n},W1,..,W\bar{m})\ \mathrm{GOTO}\ A1]$, entonces P es Σ-computable
  - Si $P : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ y $Q : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ son predicados Σ-computables, entonces $P \wedge Q, P \vee Q, \neg P$ son Σ-computables

Conjuntos Σ-enumerables

- Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-enumerable si es vacío o existe una función $F : \omega \to \omega^n \times \Sigma^{*m}$ tal que $I_F = S$ y $F_{(i)}$ sea una función Σ-computable para todo $i \in 1,..,n+m$
  - Es decir, un conjunto no vacío $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-enumerable si y solo si hay programas $\mathcal{P}_1,..,\mathcal{P}_{n+m}$ tales que:
    - $Dom(\Psi^{1,0,\#}_{\mathcal{P}_1}) =\ldots= Dom(\Psi^{1,0,\#}_{\mathcal{P}_n}) = \omega$
    - $Dom(\Psi^{0,1,*}_{\mathcal{P}_{n+1}}) =\ldots= Dom(\Psi^{0,1,*}_{\mathcal{P}_{n+m}}) = \omega$
    - $S = Im[\Psi^{1,0,\#}_{\mathcal{P}_1},..,\Psi^{1,0,\#}_{\mathcal{P}_n},\Psi^{0,1,*}_{\mathcal{P}_{n+1}},..,\Psi^{0,1,*}_{\mathcal{P}_{n+m}}]$

      es decir, $\mathcal{P}_1,..,\mathcal{P}_{n+m}$ enumeran a S.
- Propiedad: Sea $S \subseteq \omega^n \times \Sigma^{*m}$ un conjunto no vacío, entonces son equivalentes:
  - S es Σ-enumerable

- Hay un programa $\mathcal{P} \in Pro^{\Sigma}$ tal que
  - $\forall x \in \omega$, $\mathcal{P}$ se detiene partiendo de $||x||$ y llega a un estado de la forma $((x_1, .., x_n, y_1, \dots), (\alpha_1, .., \alpha_m, \beta_1, ..))$ con $(x_1, .., x_n, \alpha_1, .., \alpha_m) \in S$
  - $\forall (x_1, .., x_n, \alpha_1, .., \alpha_m) \in S$, $\exists x \in \omega$ tal que $\mathcal{P}$ se detiene partiendo de $||x||$ y llega a un estado de la forma $((x_1, .., x_n, y_1, \dots), (\alpha_1, .., \alpha_m, \beta_1, ..))$

  Decimos que $\mathcal{P}$ *enumera* a S

Conjuntos Σ-computables

- Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-computable si $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ-computable
  - Es decir, $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-computable si y solo si hay un programa $\mathcal{P} \in Pro^{\Sigma}$ que computa a $\chi_S^{\omega^n \times \Sigma^{*m}}$:
    - Si $(\vec{x}, \vec{\alpha}) \in S$, entonces $\mathcal{P}$ se detiene partiendo de $||x_1, .., x_n, \alpha_1, .., \alpha_m||$ y la variable $N1$ queda con contenido igual a 1
    - Si $(\vec{x}, \vec{\alpha}) \notin S$, entonces $\mathcal{P}$ se detiene partiendo de $||x_1, .., x_n, \alpha_1, .., \alpha_m||$ y la variable $N1$ queda con contenido igual a 0

      Decimos que $\mathcal{P}$ decide la pertenencia a S respecto al conjunto $\omega^n \times \Sigma^{*m}$
- *Macros*: Si $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-computable entonces, como $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ-computable, hay un macro $[\text{IF } \chi_S^{\omega^n \times \Sigma^{*m}}(V1, .., V\bar{n}, W1, .., W\bar{m}) \text{ GOTO } A1]$
  - Lo escribiremos mejor como $[\text{IF } (V1, .., V\bar{n}, W1, .., W\bar{m}) \in S \text{ GOTO } A1]$