

Teoría de Lenguajes
Teoría de la Programación I
Soluciones

Ejercicio 1

a)

El lenguaje es recursivamente enumerable como vemos en la parte b) al construir una G.I. que lo genera o en c) al construir una M.T. que lo reconoce. Demostraremos que NO es libre de contexto, utilizando el CR del PL (para Libres de Contexto). Para eso elegimos la tira $z = a^{N+1}b^Nc^{N+1}d^{N+2}$, siendo N la constante del Pumping Lemma.

Consideramos todas las descomposiciones posibles de la tira z en uvwxy que cumplen $|vx| > 0$ y

$|vwx| \leq N$, y probamos que existe un i tal que uv^iwx^iy no pertenece a L_1

Familia	a^{N+1}	b^N	c^{N+1}	d^{N+2}
1	v x			
2		v x		
3			v x	
4				v x
5	v	x		
6		v	x	
7			v	x
8	v xx			
9	vv x			
10		v xx		
11		vv x		
12			v xx	
13			vv x	

Familia 1

$$u = a^j$$

$$v = a^k$$

$$w = a^l$$

$$x = a^m$$

$$y = a^{N+1-j-k-l-m}b^Nc^{N+1}d^{N+2} \text{ con } k+m>0$$

Eligiendo $i=3$, $z_3 = a^{N+1+2(k+m)}b^Nc^{N+1}d^{N+2}$ no pertenece al lenguaje. Porque la cantidad de a_s y b_s queda mayor a la cantidad de c_s y d_s , y eso no sucede para ninguna tira de L_1 .

Familias 2, 5, 8 y 9

En forma análoga a la familia 1, se puede demostrar que eligiendo $i=3$, z_3 no pertenece al lenguaje por la misma razón.

Familia 3

$$u = a^{N+1}b^Nc^j$$

$$v = c^k$$

$$w = c^l$$

$$x = c^m$$

$$y = c^{N+1-j-k-l-m}b^Nc^{N+1-(k+m)}d^{N+2} \text{ con } k+m>0$$

Eligiendo $i=0$, $z_0 = a^{N+1}b^Nc^{N+1-(k+m)}d^{N+2}$ no pertenece al lenguaje. Porque la cantidad de c_s y d_s no supera en al menos 2, a la cantidad de a_s y b_s .

Observación: todas la tiras del lenguaje cumplen que la cantidad de a_s y b_s (sumadas) es $2q+p$ y a cantidad de c_s y d_s (sumadas) es $2r+p$, por lo tanto la diferencia es $2r+p-(2q+p) = 2(r-q)$ y como $r-q>0$ entonces la diferencia entre la

cantidad de c_s y d_s (sumadas) respecto a la cantidad de a_s y b_s (sumadas) tiene que ser al menos 2 o más.

Por eso es importante distinguir el caso 3 de los 1 y 2, puesto que los argumentos son otros.

Familias 4, 7, 12 y 13

En forma análoga a la familia 3, se puede demostrar que eligiendo $i=0$, z_0 no pertenece al lenguaje, porque la cantidad de c_s y d_s no supera en al menos 2, a la cantidad de a_s y b_s .

Familias 6

$$u = a^{N+1}b^j$$

$$v = b^k$$

$$w = b^{N-j-k}c^l$$

$$x = c^m$$

$$y = c^{N+1-l-m}d^{N+2} \text{ con } k+m>0$$

suponemos además que $k>0$ y $m>0$ sino caigo en algunos de los casos anteriores.

Eligiendo $i=2$, $z_2 = a^{N+1}b^{N+k}c^{N+1+m}d^{N+2}$ no pertenece al lenguaje. Porque la cantidad de b_s es mayor o igual a la cantidad de a_s , y eso no sucede para ninguna tira de L_1 .

Familias 10

Asumo que la x contiene b_s y c_s , sino caigo en algunos de los casos anteriores. En ese caso, eligiendo $i=2$ genero mezcla de b_s y c_s , lo cual no sucede para ninguna tira del L_1 , por lo tanto z_2 no pertenece al lenguaje.

Familia 11

En forma análoga a la familia 10, pero asumiendo que y contiene b_s y c_s , eligiendo $i=2$, se concluye que z_2 no pertenece al lenguaje porque se generan mezclas de b_s y c_s .

Por lo tanto:

Como estas son todas las descomposiciones posibles de z en $uvwxy$ que cumplen $|vx|>0$ y

$|vwx| \leq N$, por el CR del PL el lenguaje **no** es libre de contexto.

b)

$S \rightarrow aSd \mid aITFd$

$T \rightarrow QTR \mid TR \mid R$ genera: $a^p I Q^q R^r F d^p$

$Q \rightarrow Ab$

$R \rightarrow cD$ genera: $d^p I (Ab)^q (cD)^r F d^p$

$bA \rightarrow Ab$ mueva las A a la izquierda

$Dc \rightarrow cD$ mueva las D a la derecha

$IA \rightarrow aI$ transforma todas las A en a

$Ib \rightarrow bI$ y controla que estén antes de la primera b

$DF \rightarrow Fd$ transforma todas las D en d

$cF \rightarrow c$ y controla que estén después la última c

c)

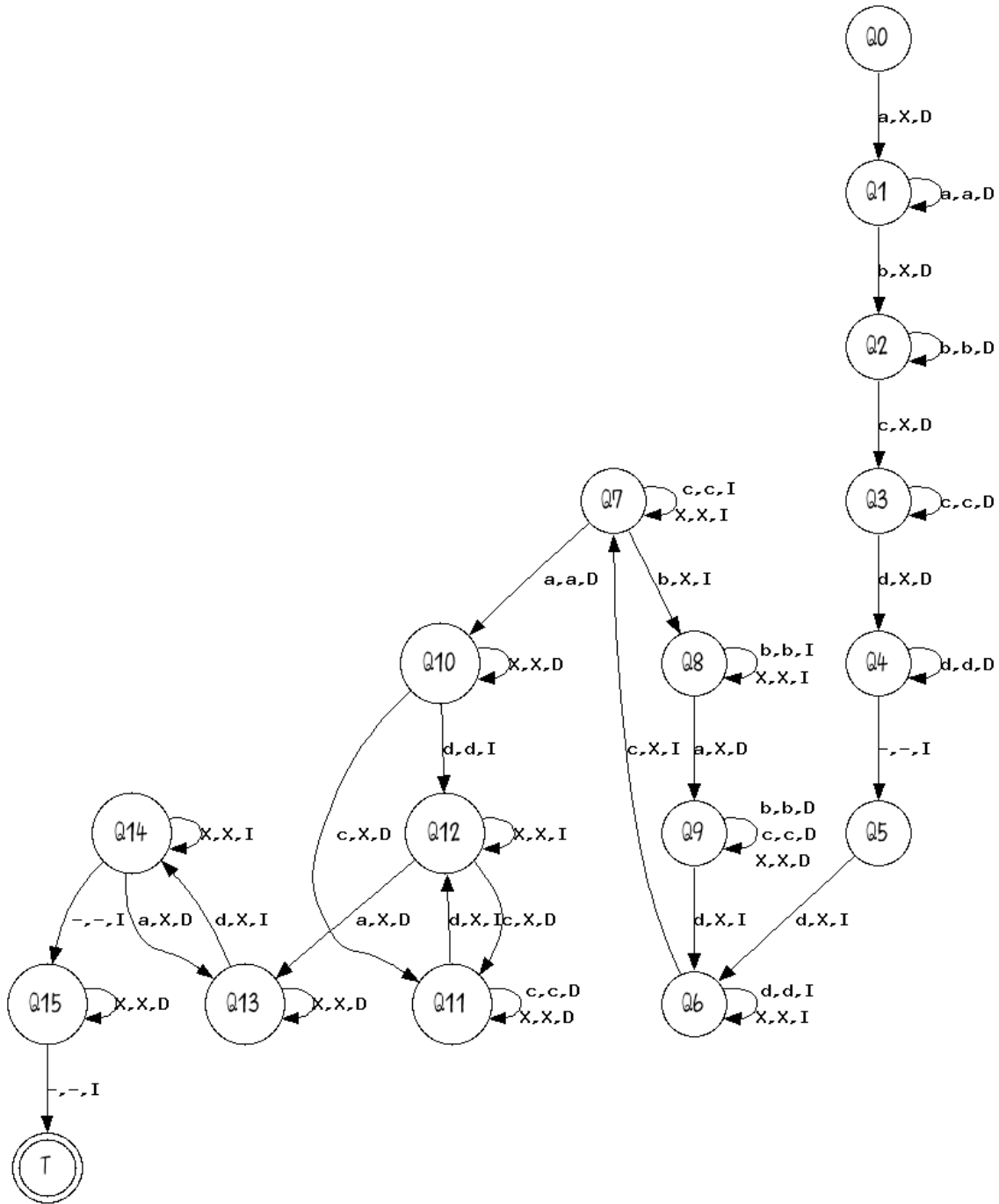
La idea básica del funcionamiento de la MT es:

1) ver que estén las a_s , b_s , c_s y d_s en ese orden, y marcar un símbolo de cada letra, dado que $0 < q < r$;

2) marcar por cada d una a, b y c (esto se debería repetir $q - 1 = \text{cantidad de } b_s - 1$, pues al inicio ya marcamos una vez;

3) para las c_s y d_s que quedan, marcar una c y luego un d (esto se debería repetir $r - q$ veces, para garantizar $r > q$ se sale de Q7 luego de haber marcado un d y c);

- 4) para las as y ds que quedan, marcar una a y luego un d (esto se debería repetir p veces, al menos una vez);
5) sino encuentro ninguna a más, me muvo a la derecha para verificar que solo quedan marcas



Ejercicio 2

Sea $L_2 = \{ \#w\#w' \mid w \in \{0,1\}^* \text{ donde: } w = a_1a_2a_3a_4\dots a_{2n} \text{ y } w' = a_{2n-1}a_{2n}\dots a_3a_4a_1a_2 \}$

- Clasifique a L_2 según la Jerarquía de Chomsky. Justifique
- Construya un autómata $M_2 / L_2 = L(M_2)$
- Construya una gramática $G_2 / L_2 = L(G_2)$

a)

El lenguaje es libre de contexto, no regular.

Que es L.C. se demuestra con la construcción del A.P.D. del parte b) o con la construcción de la G.L.C. de la parte c)

Que no es Regular se demuestra usando el contra-recíproco del P.L. para Leng. Reg.
Sea $n \in \mathbb{N}$

Elegimos $z = \#0^{2n}\#0^{2n} \in L$ y $|z| = 4n+2 \geq n$

Estudiamos todas las descomposiciones de $z = uvw$ talque $|uv| \leq n$ y $|v| \geq 1$ y trataremos de encontrar un $i \geq 0$ para c/descomposición tal que $uv^i w \notin L$

Familias	#	0^{2n}	#	0^{2n}
1	v	v		
2		v		

Familia 1

$u = \epsilon$

$v = \#0^q$ con $1+q \leq n$ y $q \geq 0$

$w = 0^{2n-q}\#0^{2n}$

$z_i = uv^i w = (\#0^q)^i 0^{2n-q}\#0^{2n}$

Eligiendo $i=0$ se ve que $uv^0 w \notin L$, porque la tira z_0 no comienza con #

Familia 2

$u = \#0^q$

$v = 0^p$ con $1+q+p \leq n$ y $p \geq 1$

$w = 0^{2n-q-p}\#0^{2n}$

$z_i = uv^i w = \#0^q(0^p)^i 0^{2n-q-p}\#0^{2n} = \#0^q 0^{pi} 0^{2n-q-p}\#0^{2n} = \#0^{2n+(i-1)p}\#0^{2n}$

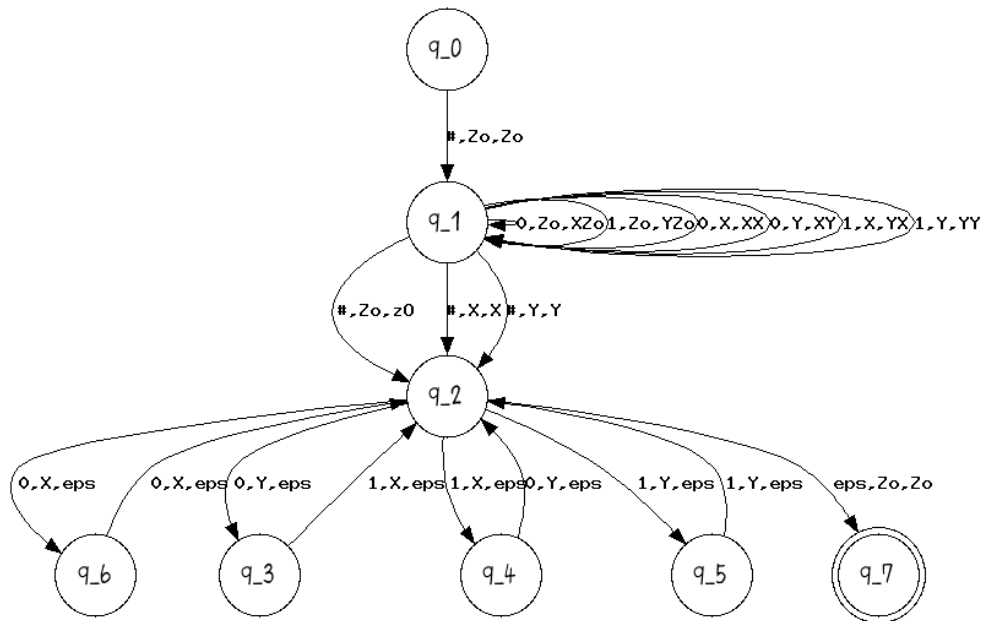
Eligiendo $i=0$ se ve que $uv^0 w \notin L$, porque como $p \geq 1$, la tira z_0 contiene menos 0's del lado izquierdo del 2do # que del lado derecho del 2do #

Por lo tanto:

Como estas son todas las descomposiciones posibles de z en uvw que cumplen $|v| > 0$ y

$|uv| \leq n$, por el CR del PL el lenguaje **no** es regular.

b)



c)

$S \rightarrow \#X$
 $X \rightarrow 01X01$
 $\quad | 00X00$
 $\quad | 10X10$
 $\quad | 11X11$
 $\quad | \#$

Ejercicio 3

- Si L_a es libre de contexto y L_b es R.E., entonces $L_a.L_b$ es R.E.
- Si $L_a \cap L_b$ es regular y no vacío y L_b es libre de contexto no regular, entonces L_a es regular.
- Si $L_a \cap L_b$ es regular, $L_a \cap L_c$ es libre de contexto no regular y $L_a \neq L_b$, entonces $L_a \cap L_b \cap L_c$ es regular.

Respuestas:

- VERDADERO.** Como L_a es libre de contexto, entonces (por Jerarquía de Chomsky) es recursivamente enumerable. Sea $G_a = \langle T_a, V_a, P_a, S_a \rangle$ la gramática que genera L_a y $G_b = \langle T_b, V_b, P_b, S_b \rangle$ la gramática que genera L_b y supongamos (sin pérdida de generalidad) que V_a y V_b son disjuntos (si no lo son, pueden renombrarse las variables en las gramáticas). Suponemos también que T_a y T_b son disjuntos (en caso de no serlo, deberían generarse reglas del tipo $V_i \rightarrow t$ por cada terminal t y sustituir en las reglas de las gramáticas). La gramática $G = \langle T_a \cup T_b, V_a \cup V_b, P_a \cup P_b \cup \{S \rightarrow S_a S_b\}, S \rangle$ genera $L_a.L_b$ y por lo tanto es recursivamente enumerable.
- FALSO.** Contraejemplo: $L_b = \{a^n b^n \mid n \text{ natural}\}$ que es libre de contexto no regular, y $L_a = \{b^n c^n \mid n \text{ natural}\}$.
 $L_a \cap L_b = \{\epsilon\}$
 - $\}$ que es regular, pero L_a no lo es.

Nota: Las gramáticas y los autómatas deben corresponderse con el tipo del lenguaje considerado en cada caso, según la Jerarquía de Chomsky. Se valora positivamente la simplicidad de las soluciones propuestas así como una breve explicación de éstas. Todas las respuestas deben estar debidamente justificadas.

- iii. *FALSO*. Contraejemplo: $L_a = \Sigma^*$, $L_b = L(a^*b^*)$, $L_c = \{a^n b^n \mid n \text{ natural}\}$. La intersección es L_c que no es regular

Ejercicio 4 [Teoría de Lenguajes]

Sean los siguientes lenguajes:

$$L_5 = \{ a^p b^q \mid p \bmod 3 = 0 ; 0 < q \}$$

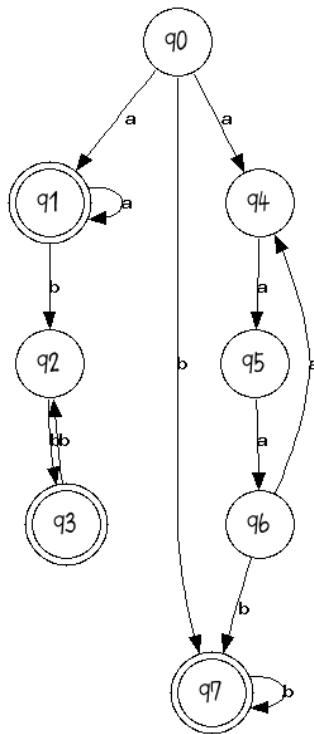
$$L_6 = \{ a^q b^r \mid r \bmod 2 = 0 ; 0 < q \}$$

$$L_4 = L_5 \cup L_6$$

- Construya un autómata mínimo $M_4 / L_4 = L(M_4)$
- ¿Cuántas clases de equivalencia tiene L_4 según R_L ? Justifique.
- Dé una expresión regular $r_4 / L_4 = L(r_4)$. Justifique.

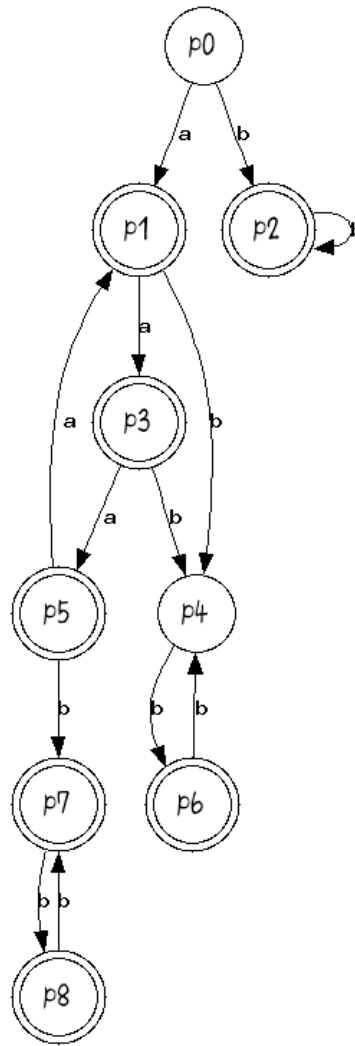
a)

AF no determinista que reconoce el lenguaje:



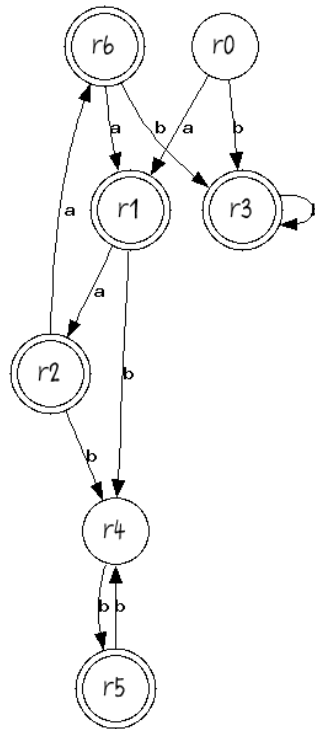
Obtengo autómata determinista:

		a	b
p0	q0	[q1 q4]	[q7]
p1	[q1 q4]	[q1 q5]	[q2]
p2	[q7]		[q7]
p3	[q1 q5]	[q1 q6]	[q2]
p4	[q2]		[q3]
p5	[q1 q6]	[q1 q4]	[q2 q7]
p6	[q3]		[q2]
p7	[q2 q7]		[q3 q7]
p8	[q3 q7]		[q2 q7]



Obtengo autómata mínimo:

[p4 p0] [p1 p2 p3 p5 p6 p7 p8]
 [p4] [p0] [p1 p3 p6] [p2 p5 p7 p8]
 [p4] [p0] [p1] [p3] [p6] [p5] [p2 p7 p8]



b)

Por el teorema de Myhill-Nerode, las clases de equivalencia del autómata mínimo (R_M) coinciden con las clases de equivalencia de R_L . Dado que el autómata mínimo posee 8 estados (considerando el estado pozo), L_4 posee 8 clases de equivalencia.

c)

Planteo sistema de ecuaciones para obtener las clases de equivalencia de R_M

$$X_0 = \text{eps}$$

$$X_1 = X_0a \mid X_6a$$

$$X_2 = X_1a$$

$$X_3 = X_0b \mid X_3b \mid X_6b$$

$$X_4 = X_2b \mid X_1b \mid X_5b$$

$$X_5 = X_4b$$

$$X_6 = X_2a$$

$$X_P = X_3a \mid X_4a \mid X_5a \mid X_P(a|b)$$

$$X_1 = a \mid X_6a = a \mid X_2aa = a \mid X_1aaa = a(aaa)^*$$

$$X_2 = a(aaa)^*a$$

$$X_6 = a(aaa)^*aa$$

$$X_3 = b \mid X_3b \mid a(aaa)^*aab = (b|a(aaa)^*aab)b^*$$

$$X_4 = a(aaa)^*ab \mid a(aaa)^*b \mid X_4bb = (a(aaa)^*ab \mid a(aaa)^*b)(bb)^*$$

$$X_5 = (a(aaa)^*ab \mid a(aaa)^*b)(bb)^*b$$

$$X_P = ((b|a(aaa)^*aab)b^*a \mid (a(aaa)^*ab \mid a(aaa)^*b)(bb)^*a \mid (a(aaa)^*ab \mid a(aaa)^*b)(bb)^*ba)(a|b)^*$$

La expresión regular del lenguaje se obtiene a partir de la unión de las expresiones regulares de los estados finales del autómata mínimo:

$$X_1 \mid X_2 \mid X_3 \mid X_5 \mid X_6$$

Ejercicio 4 [Teoría de la Programación 1]

- i. **f no es computable.** Supongamos que f es una función computable; luego, existe un programa en P que la computa. Sea MP una macro que lo hace.

Se construye una macro MQ que dado un índice de programa i y un natural n , calcula el índice del siguiente programa:

```
PROGRAM(X0)
  X1:= EVAL_PROG(i,n);
  X2:= SUC(0);
RESULT(X0)
```

(*)Observar que es condición necesaria y suficiente para que este programa asigna a X_2 el valor 1, que el programa de índice i con entrada j converja, sin importar la entrada dada al programa en sí.

Ahora, construimos el siguiente programa en P de índice p :

```
PROGRAM(<i,n>)
  X1:= MQ(<i,n>);
  X2:= MF(<X1,0,2,1>);
RESULT(X2)
```

Este programa siempre para, pues MF y MQ siempre lo hacen (f es total). Además se cumple que:

$$\begin{aligned} <I_x(p), <i,n>> \Rightarrow 1 \Leftrightarrow^1 MF(<MQ(<i,n>), 0, 2, 1>) = 1 \Leftrightarrow^2 \\ MQ(<i,n>) \text{ con entrada } 0 \text{ asigna a } X_2 \text{ el valor } 1 \Leftrightarrow^3 <I_x(i), n> \downarrow \Leftrightarrow^4 STOP(i,n)=1 \end{aligned}$$

1. Por construcción de p .
2. Definición de la función f .
3. Por lo expresado en (*).
4. Definición de la función $STOP$.

Luego, el programa de índice p computa a la función $STOP$, lo que es un absurdo que se desprende de suponer que f es computable $\Rightarrow f$ no es computable.

- ii. **g no es computable.** Supongamos que g es una función computable; luego, existe una macro MG en P que la computa.

Sea MQ la macro de la parte anterior. Construimos el siguiente programa en P de índice p :

```
PROGRAM(X0)
  X1:= MQ(<X0,X0>);
  X2:= MG(<X1,0,2,1>);
RESULT(X2)
```

Este programa cumple que:

$$\begin{aligned} <I_x(p), i> \downarrow \Leftrightarrow^1 MF(<MQ(<i,i>), 0, 2, 1>) \downarrow \Leftrightarrow^2 \\ MQ(<i,i>) \text{ con entrada } 0 \text{ no asigna a } X_2 \text{ el valor } 1 \Leftrightarrow^3 <I_x(i), i> \uparrow \Leftrightarrow^4 i \in C(K) \end{aligned}$$

1. Por construcción de p .

2. Definición de la función f .
3. Por lo expresado en (*) en la parte anterior.
4. Definición de $C(K)$.

Cuando el programa de índice p converge, devuelve 1; luego, este programa computa a la función característica parcial de $C(K)$, de donde $C(K)$ es un conjunto r.e. Esto es un absurdo que se desprende de suponer que g es computable $\Rightarrow g$ no es computable.