

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (www.tesisenxarxa.net) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (www.tesisenred.net) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (www.tesisenxarxa.net) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author

UNIVERSITAT POLITÈCNICA DE BARCELONA
FACULTAT D'INFORMÀTICA
2/6/83
N.º 309
ENTRADA

1400190423
còpia 1

FUNCIONES DE COMPLEJIDAD
Y SU RELACION CON
LAS FAMILIAS ABSTRACTAS DE LENGUAJES

J. GABARRO.



Rep: 1217

INDICE

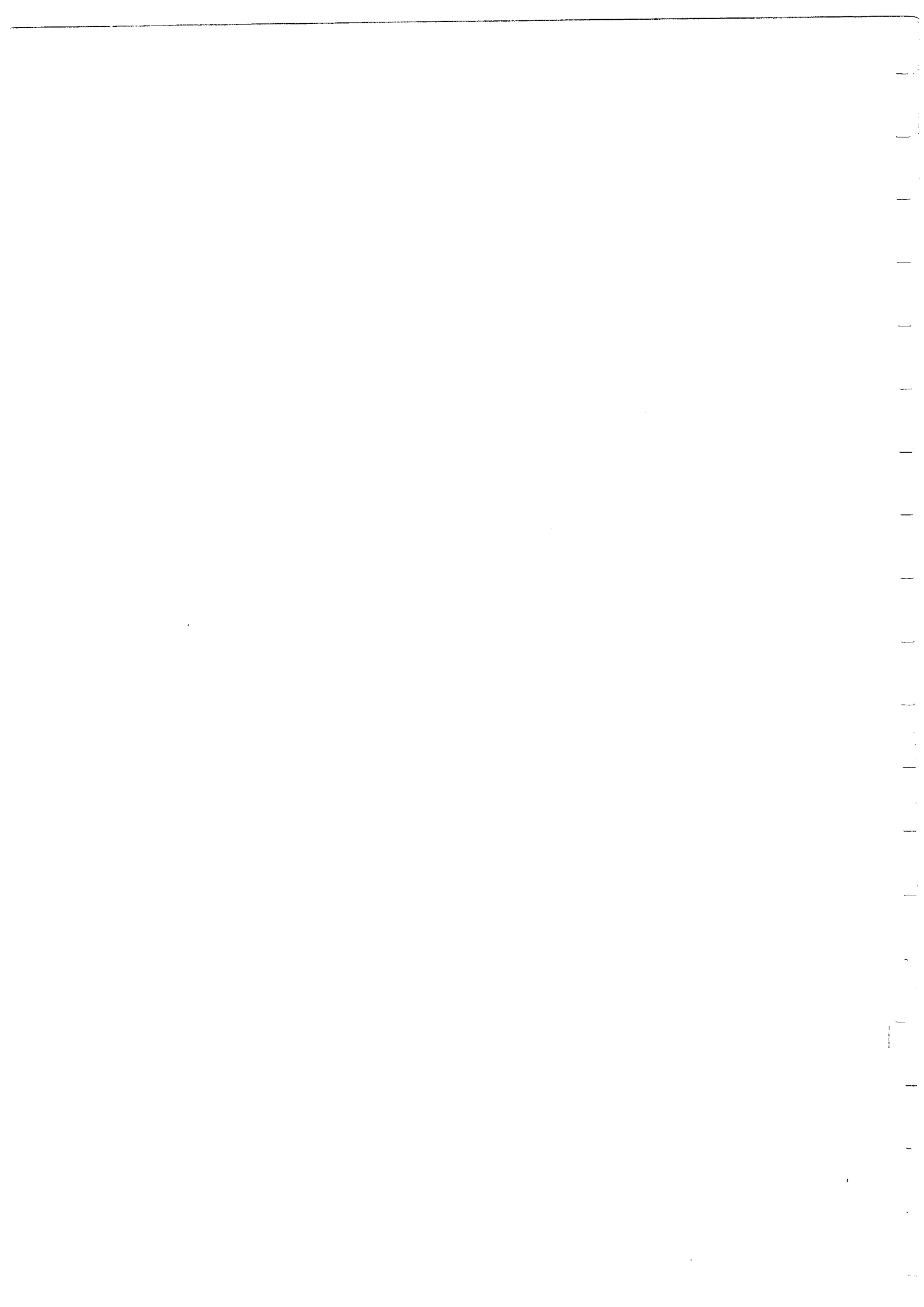
<u>Introducción</u>	1
---------------------	---

Capítulo I. Preliminares

I.1 Notaciones.	13
I.2 Lenguajes racionales y reconocibles.	15
I.3 Gramáticas, lenguajes algebraicos y autómatas a pila.	17
I.4 Transducciones racionales y conos racionales, conos racionales fieles.	22
I.5 Operaciones de substitución de full-A.F.L. y de A.F.L.	26
I.6 Centro y familias uniformes.	27

Capítulo II. Funciones de medida en lenguajes formales.

II.1 Función de crecimiento	34
II.2 α -medida	40
II.3 Índice racional.	57
II.4 Costo rígido.	87
II.5 Coste boleano.	97
II.6 h -medida.	102

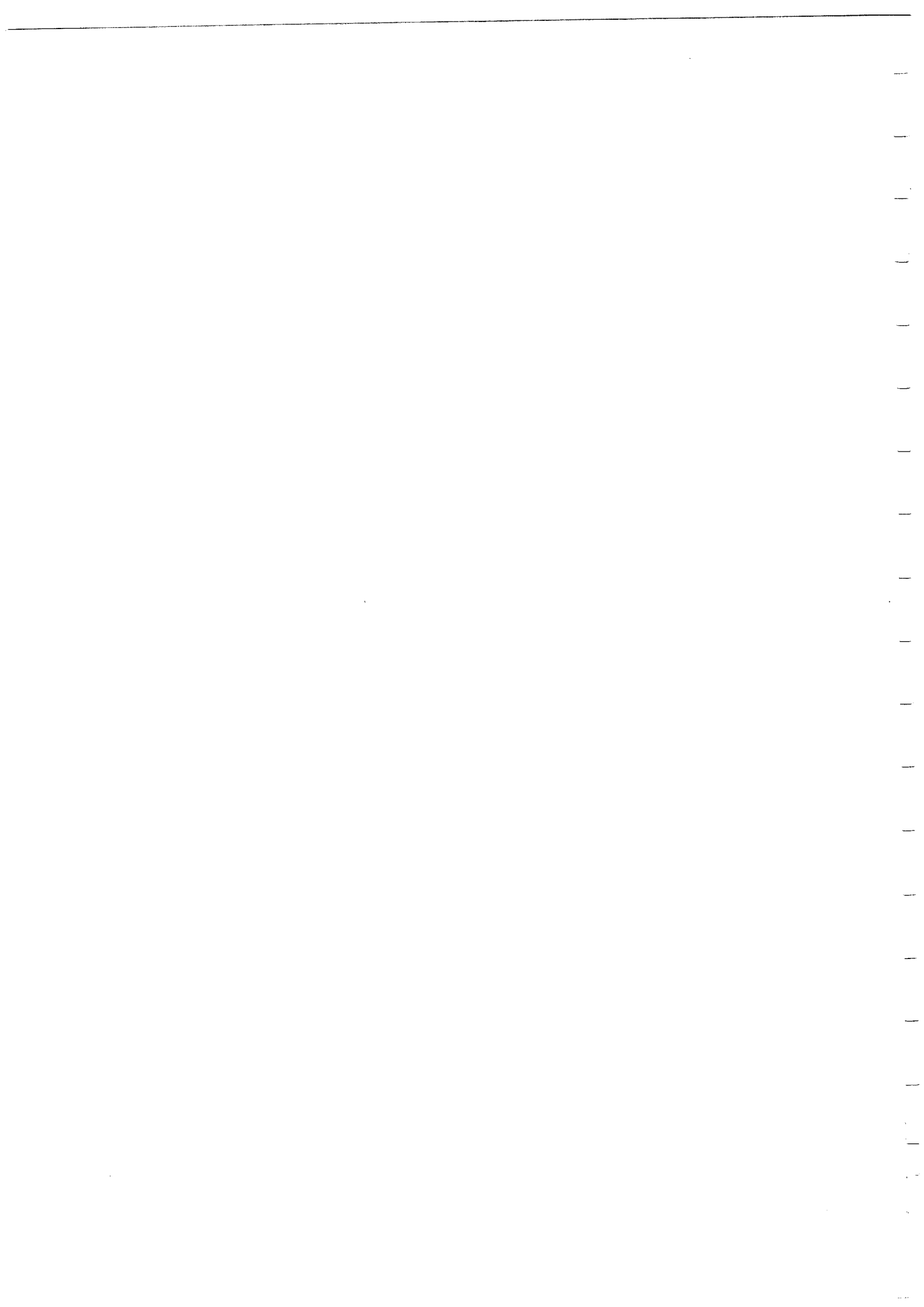


Capítulo III. Índice inicial, lenguajes polinomiales
y exponenciales.

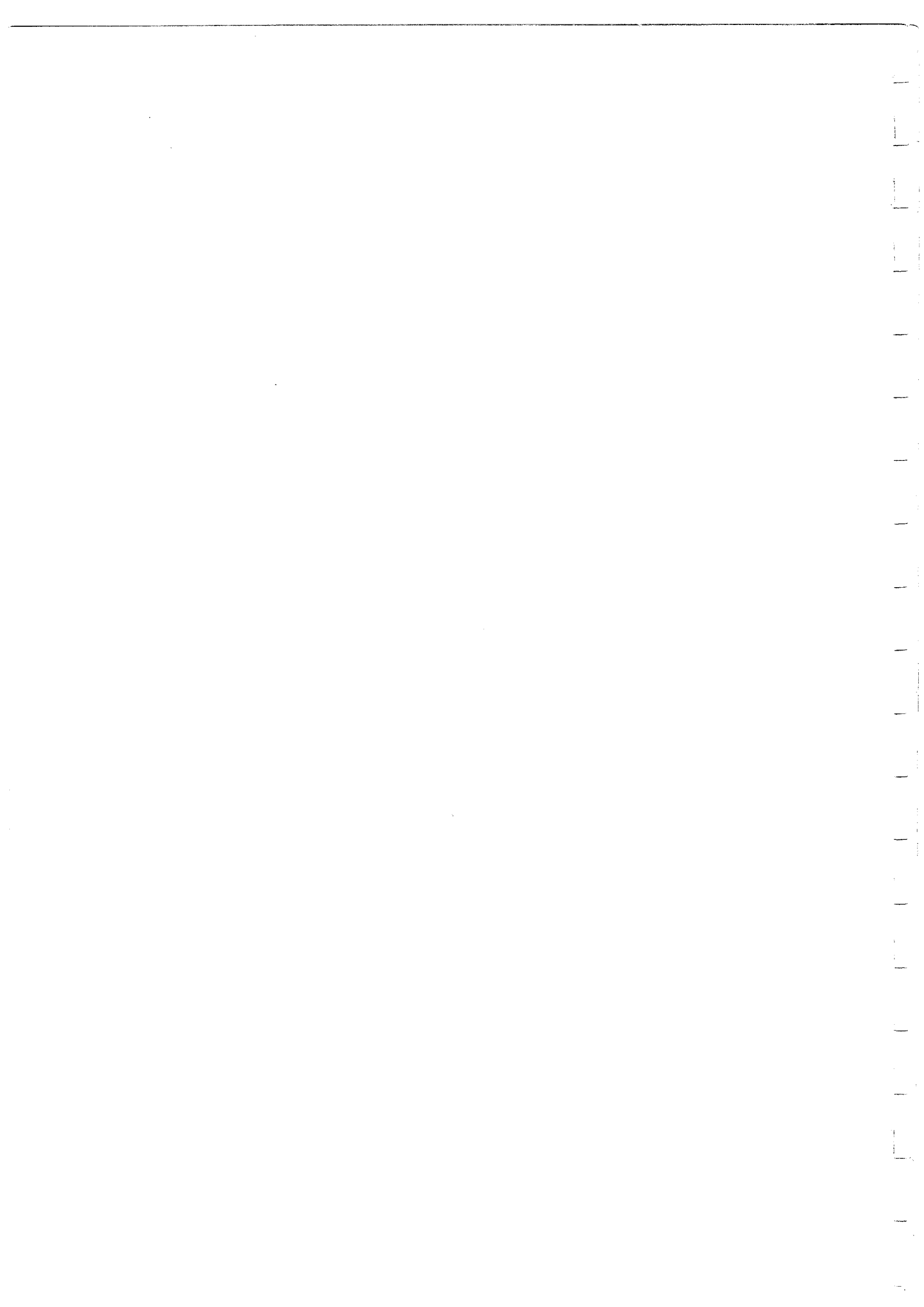
III.1	Índice inicial, definición y propiedades de crecimiento.	122
III.2	Ejemplos de cálculo de índices iniciales. Las familias $\text{Pol } \mu$ y $\overline{\text{Exp } \mu}$.	130
III.3	Índice inicial y su relación con las operaciones de lenguajes.	197

Capítulo IV. Índice inicial y familias clásicas
de lenguajes.

IV.1	Lenguajes polinomiales algebraicos y a multicontador.	225
IV.2	Lenguajes exponenciales y generadores fieles.	291



<u>Capítulo V.</u>	<u>Lenguajes de altura de pila sublineal.</u>	
V.1	Índice de pila, lenguajes p-sublineales.	296
V.1.1	Primeras definiciones y ejemplos.	297
V.1.2	Lenguajes $\overline{p\text{-Root}(q)}$ y $\overline{p\text{-Log}}$.	314
V.2	Operaciones entre lenguajes y cadenas de full-A.F.L.	332
V.2.1	Operaciones entre lenguajes.	333
V.2.2	Lemas de iteración y límites inferiores.	341
V.2.3	Cadena de full-A.F.L.	359
<u>Capítulo VI.</u>	<u>Comparación entre distintas medidas de complejidad.</u>	371
<u>Bibliografía</u>		407



INTRODUCCION



La teoría de los autómatas y lenguajes formales, nació en los años 50 de la reunión de trabajos de cibernéticos como Mc.Culloch y W. Pitts, de lógicos como S.Kleene y de lingüistas como N.Chomsky.

Fue N.Chomsky quien tuvo la idea de escribir un lenguaje, describiendo su proceso de generación, a partir de un número finito de reglas. Llamaremos gramática a dicho conjunto de reglas. Limitando, de modo gradual, el tipo de reglas de la gramática, N.Chomsky, pudo definir, en el año 56, los siguientes cuatro tipos de lenguajes [16] .

Dicha clasificación, ya clásica, contiene:

- La familia de los lenguajes engendrados por las gramáticas con estructura de frase. Es la familia de los lenguajes recursivamente enumerables.
- La familia de los lenguajes engendrados por las gramáticas de contexto ligado. Estos son los lenguajes a contexto ligado (context-sensitive) que son todos los lenguajes recursivos.
- La familia de lenguajes engendrada por las gramáticas a contexto libre (context-free). Estos son los lenguajes que llamamos algebraicos.
- La familia de lenguajes engendrada por las gramáticas lineales uniláteras. Esta es la familia de lenguajes racionales.

Podemos, también, definir un lenguaje por un proceso de reconocimiento. Podemos, entonces, caracterizar los lenguajes a partir de los autómatas que los aceptan. Podemos de manera dual caracterizar la jerarquía de Chomsky por:

- La familia de lenguajes con estructura de frase corresponde a los lenguajes reconocidos por las máquinas de Turing.
- Los lenguajes de contexto ligado pueden definirse con la ayuda de



los autómatas linealmente acotados.

- Los lenguajes algebraicos coinciden con los lenguajes reconocidos por los autómatas a pila.
- Los lenguajes racionales son los lenguajes reconocidos por los autómatas finitos.

Los estudios llevados a cabo en las diferentes familias de lenguajes, utilizaron, en principio, métodos directos. La comparación de los distintos métodos empleados permite constatar numerosas similitudes. Podemos, entonces, tratar de estudiar de modo axiomático, las familias de lenguajes estables bajo un conjunto fijado de operaciones. Este procedimiento permite demostrar resultados globales comunes a distintas familias. Es bajo este aspecto que se ha desarrollado la noción de cono racional. Un cono racional es una familia de lenguajes estable por transducción racional.

Esta noción ha sido introducida por C. Elgolt y J. Mezei [19] y transformada en manipulable por M. Nivat [46]. Paralelamente S. Ginsburg y S. Greibach la noción de full-A.F.L. (full Abstract Family of Languages) [29] que es un cono racional estable en las operaciones racionales de unión, producto y estrella.

La consecuencia lógica de esta manera de abordar el problema, consiste, dada una familia de lenguajes \mathcal{L} estable bajo un conjunto de operaciones E , en preguntarse si existe un lenguaje L en \mathcal{L} que permite obtener cualquier elemento de \mathcal{L} con la ayuda de las operaciones de E . En este caso, diremos que la familia \mathcal{L} es E -principal y que L es un generador de \mathcal{L} . Dicho lenguaje L aparece, entonces, como un prototipo de los lenguajes de \mathcal{L} y su estudio merece es-

pecial interés.

El estudio de la principalidad de las familias de lenguajes, se encuentra hoy día en el punto central de la teoría. La mayoría de estudios han sido realizados en el caso en que E corresponde a un cono racional o full-A.F.L. [7] ; sin embargo, hay otros conjuntos de operaciones interesantes, como se muestra en [2] .

Otro método de estudio de lenguajes formales, es a través de medidas de complejidad. Fijada una medida, podemos asociar a cada lenguaje su complejidad en relación a dicha medida. El caso más importante surge cuando la medida es una función sobre los naturales, la llamaremos en este caso "función de complejidad" .

Siguiendo a R.Karp y a J.Lipton [4 1] podemos clasificar las funciones de complejidad en dos grandes grupos, las uniformes y las no uniformes.

- Diremos que una medida es uniforme si se obtiene fijando mediante una función, una determinada limitación al funcionamiento del autómata. Sin duda, el ejemplo más conocido, consiste en acotar por un polinomio el tiempo de aceptación de las máquinas de Turing. Dicha idea ha permitido a S.Cook [1 3] definir las clases P, NP y NPC.

- Diremos que una medida es no uniforme si se basa en medir alguna propiedad de aproximaciones finitas de un lenguaje. Como ejemplo característico, citaremos el trabajo de J.Savage [5 0] , que consiste en medir un lenguaje por el número de puertas necesarias para sintetizar, con la ayuda de un circuito lógico, todas las palabras de longitud dada.

De modo general, fijada una función de complejidad, es posible clasi-

sificar los lenguajes respecto a su comportamiento asintótico en dicha medida. Podemos, de este modo, definir nuevas familias de lenguajes. Un problema fundamental consiste, entonces, en estudiar dichas familias en términos de la teoría de familias abstractas de lenguajes y tratar de demostrar si son o no principales. Este trabajo se sitúa en dicha óptica.

En los párrafos siguientes, se enuncia el contenido de cada capítulo.

- En el primer capítulo introducimos las notaciones y nociones básicas en el estudio de familias de lenguajes. Definimos rápidamente las distintas operaciones entre lenguajes, haciendo especial hincapié en la operación de transducción racional y en el teorema de M. Nivat. Seguidamente definimos distintos tipos de familias de lenguajes, como, por ejemplo, los full-A.F.L., los A.F.L, los conos racionales y los cilindros. Introducimos, luego, varias familias de lenguajes, los lenguajes lineales, quasiracionales, a contador restringido, a contador iterado y de Greibach; notadas Lin, Ort, Rocl, Fcl y Gre; dichos lenguajes son de especial importancia en el estudio de los lenguajes algebraicos.

Definimos, finalmente, la familia Gen de los lenguajes que son generadores de la familia Alg de los lenguajes algebraicos.

- En el capítulo segundo introducimos distintas funciones de complejidad que han aparecido en la literatura en el estudio de los lenguajes algebraicos.

Comencemos por citar las medidas uniformes:

. Tenemos una función de complejidad introducida por J. Paredaens y

R. Wyncke [47] y que llamaremos en lo sucesivo a-medida. Dicha función se basa en contar el número de estados accesibles a una determinada distancia del autómata minimal determinista que reconoce un lenguaje dado.

. Si nos restringimos a los lenguajes algebraicos, podemos considerar la h-medida. Esta medida ha sido introducida por K. Culik y H. Maurer [15] y se funda en medir la altura del árbol de derivación en relación a la longitud de la palabra engendrada.

Demos seguidamente las medidas no uniformes:

. Consideremos seguidamente la función de crecimiento. Dicha medida ha sido introducida por Milnor [45] en su aplicación al estudio de grupos finitamente presentados.

Nosotros consideraremos su aplicación a los lenguajes abstractos. La función de crecimiento se obtiene contando el número de palabras de una longitud dada que aparecen en un lenguaje. Para un lenguaje dado L , notaremos g_L su función de crecimiento.

. Otra medida que consideraremos es el índice racional, notado ρ_L . Dicha medida ha sido introducida por L. Boasson, B. Courcelle y M. Nivat [8] en relación al problema de la no principalidad en los lenguajes algebraicos. ρ_L mide la disparidad que presenta L con relación a los lenguajes racionales.

. Introduciremos seguidamente el coste boleano, notado b_L . Dicho coste ha sido introducido por J. Savage [50] en su estudio de las funciones booleanas. El coste boleano mide un lenguaje del alfabeto $\{0,1\}$ por el número de puertas necesarias para sintetizar la función característica de las palabras de longitud n .

. La última medida que trataremos en este capítulo es la uc-medida notada uc_L . Dicha función ha sido introducida por G. Goodrich, R. Ladner y M. Fischer [3 1] . Esta medida cuenta el número de uniones y concatenaciones que utiliza un programa rígido (straight-line) a fin de calcular aproximaciones finitas de un lenguaje.

Para cada una de ellas, mostramos su comportamiento en los lenguajes más usuales, como el lenguaje de Dyck, las palabras palíndromes o las palabras cuadradas.

En cada una de ellas se dan, cuando se conocen, sus cotas asintóticas y se enuncian los teoremas más importantes que de ellas se conocen. Hacemos especial hincapié en los que tienen relación con transducciones racionales y operaciones de A.F.L.

Cuando es posible, definimos las familias de lenguajes exponenciales y polinomiales asociados, a cada una de ellas. Remarquemos que en caso del índice racional, los lenguajes polinomiales algebraicos, son un full-A.F.L. cerrado por substitución [8] . Demostramos que dicha familia contiene estrictamente los lenguajes de Greibach [2 4] , y que dicha prueba utiliza la noción de centro no empleando ningún lema de iteración, de aplicación siempre tediosa.

- En el capítulo tercero introducimos una nueva medida de complejidad no uniforme, el índice inicial [2 5] .

Dicha medida es de naturaleza combinatoria y se basa en la función definida, contando el mínimo número de estados de un autómata no determinista que reconoce aproximaciones del lenguaje. Para un lenguaje dado L , notamos su índice inicial μ_L .

Calculamos el índice inicial de los lenguajes más usuales. Dicho cálculo se basa en dos etapas. En la primera etapa buscamos una cota superior y para ello es preciso dar una sucesión de autómatas que reconocen las aproximaciones del lenguaje. En una segunda etapa buscamos una cota inferior, mostrando que cualquier autómata debe tener al menos un cierto número de estados si se quiere reconocer una aproximación dada del lenguaje.

Definimos, seguidamente, la familia de los lenguajes de índice inicial acotado superiormente por un polinomio, notada $\text{Pol } \mu$, y los lenguajes acotados inferiormente por una exponencial, notada $\overline{\text{Exp}} \mu$.

Tenemos, entonces, un primer teorema que muestra que los lenguajes Dyck de dos letras, palíndromes, cuadrados, TWIN_2 y anti-Dyck de dos letras, pertenecen a $\overline{\text{Exp}} \mu$, mientras que los lenguajes Dyck de una letra, Lukasiewickz, $\{ a^n b^n \mid n \geq 1 \}$ pertenecen a $\text{Pol } \mu$.

Estudiamos, seguidamente, el comportamiento del índice inicial con respecto a las distintas operaciones entre lenguajes como unión, intersección, producto, estrella, morfismo, ϵ -libre, morfismo inverso y substitución propia.

En particular, damos un teorema que muestra el comportamiento del índice inicial con relación a las transducciones fieles.

El comportamiento agradable del índice inicial en estas operaciones, tiene como consecuencia el teorema que dice que la familia $\text{Pol } \mu$ es un A.F.L. cerrado por intersección y substitución propia.

- En el capítulo cuarto relacionamos las familias $\text{Pol } \mu$ y $\overline{\text{Exp}} \mu$ definidas en el capítulo tercero, con las familias clásicas de len-

guajes. Tratamos separadamente los lenguajes polinomiales y exponenciales.

Comenzamos mostrando que en el caso de lenguajes algebraicos, el lema de iteración de Ogden permite tratar el índice inicial con relación a las operaciones de morfismo y sustitución arbitrarias. Concluimos, entonces, que los lenguajes polinomiales algebraicos, son un full-A.F.L. cerrado por sustitución.

Utilizando la sucesión de lenguajes

$$P_k = \left\{ a_1^{n_1} a_2^{n_2} \dots a_k^{n_k} \# a_k^{n_k} \dots a_2^{n_2} a_1^{n_1} \mid n_i \geq 1 ; i = 1, \dots, k \right\}$$

que satisfacen $\mu_{P_k}(n) = \Theta(n^k)$ llegamos a que dicho full-A.F.L. no es principal.

Dado que el lenguaje de Dyck a una letra tiene índice inicial cuadrático, obtenemos un teorema que relaciona Pol μ con Alg a saber, los lenguajes algebraicos cuyo índice inicial está acotado superiormente por un polinomio son un full-A.F.L. cerrado por sustitución, que contiene la familia de los lenguajes a contador iterado. Conjeturamos que dicha inclusión es estricta.

Salimos seguidamente del marco de los lenguajes algebraicos para situarnos en el de los multicontadores. Entre los diversos lenguajes a multicontador, ocupan un lugar importante las secuencias de transición de las redes de Petri, introducidas por Peterson [48] y que notaremos \mathcal{ESS} . Demostramos que todo lenguaje \mathcal{ESS} tiene índice inicial polinomial. Dado que $\{a^n b^n \mid n \geq 0\}^*$ tiene índice inicial polinomial pero no pertenece a \mathcal{ESS} dicha inclusión es estricta. Mostramos, en fin, que las familias "on-line-quasi-realttime" y "on-line-polyno-

mial-time" son también polinomiales.

Estudiamos, seguidamente, los lenguajes de índice inicial exponencial. Mostramos que cualquier generador fiel de las familias Lin, Single-Reset, Alg y Post-languages tienen índice inicial exponencial.

Podemos concluir este capítulo, diciendo que, por una parte, los lenguajes polinomiales contienen subfamilias bien conocidas de los lenguajes a multicontador, y que los lenguajes exponenciales contienen los generadores de otras familias bien conocidas.

En el capítulo quinto introducimos una nueva medida de complejidad uniforme, el índice de pila [2 6]. Dicha medida es exclusiva de los lenguajes algebraicos y se basa en la altura de pila que es necesaria para reconocer palabras de longitud dada. Dado un lenguaje L notaremos p_L su índice de pila.

Tras dar las definiciones básicas, mostramos lenguajes que tienen un índice de pila sublineal, entre ellos encontramos el lenguaje de Goldstine y el de Patterson. Estos hechos conducen a definir la familia $p\text{-Sub}$ de lenguajes de índice de pila sublineal.

Estudiamos, seguidamente, las familias $p\text{-Root}(q)$ y $p\text{-Log}$ cuyo índice de pila está acotado por la raíz q -ésima y el logaritmo respectivamente.

En una segunda parte, analizamos las distintas familias hasta ahora definidas en términos de A.F.L. Para ello, estudiamos el comportamiento del índice de pila respecto a las operaciones de unión, producto, estrella, intersección con un lenguaje racional, morfismo, morfismo inverso y sustitución sintáctica. Podemos, entonces, dar el comportamien-

to del índice de pila en las transducciones racionales y concluir que las familias $\overline{p\text{-Sub}}$, $\overline{p\text{-Root}(q)}$ y $\overline{p\text{-Log}}$ son full-A.F.L. cerrados por substitución.

A fin de demostrar la inclusión estricta de los distintos A.F.L. que han sido encontrados, analizamos las propiedades de iteración de los autómatas a pila; a partir de ello se demuestra que unas versiones modificadas de los lenguajes de Goldstine y Patterson necesitan estrictamente de la raíz cuadrada para poder ser aceptados.

La idea de imbricar el lenguaje de Patterson en el lenguaje de Patterson, permite construir el lenguaje de "Patterson-menor, de orden q ", que precisa estrictamente de la raíz q -ésima para ser aceptado.

Se concluye el capítulo demostrando:

$$\overline{p\text{-Log}} \subsetneq \dots \subsetneq \overline{p\text{-Root}(q)} \subsetneq \overline{p\text{-Root}(q-1)} \subsetneq \dots \subsetneq \overline{p\text{-Root}(2)} \subsetneq \overline{p\text{-Sub}}$$

Podemos, entonces, concluir que el full-A.F.L. de los lenguajes sublineales contiene una jerarquía infinita de full-A.F.L. estrictamente incluidos los unos en los otros.

El sexto y último capítulo, está destinado a comparar las distintas medidas de complejidad que hemos tratado. Recordemos que dada una medida α podemos definir la familia de lenguajes polinomiales según α , $\text{Pol } \alpha$ y los lenguajes exponenciales según α , $\overline{\text{Exp } \alpha}$.

Diremos que dos medidas α y β son comparables si una de las familias que se definen con α está contenida en otra de las que se definen con β . Como ejemplo de medidas comparables tenemos:

$$(g_L, \mu_L) \quad , \quad (g_L, uc_L) \quad , \quad (g_L, b_L) \quad , \quad (\mu_L, a_L)$$

(μ_L, uc_L) , (μ_L, b_L) , (a_L, uc_L)
 (a_L, b_L)

Diremos que dos medidas α y β no son comparables si Pol α in-
 tersecta Exp β o viceversa. Como ejemplos de medidas no comparables
 tenemos:

(g_L, ρ_L) , (g_L, a_L) , (μ_L, ρ_L) , (ρ_L, a_L)

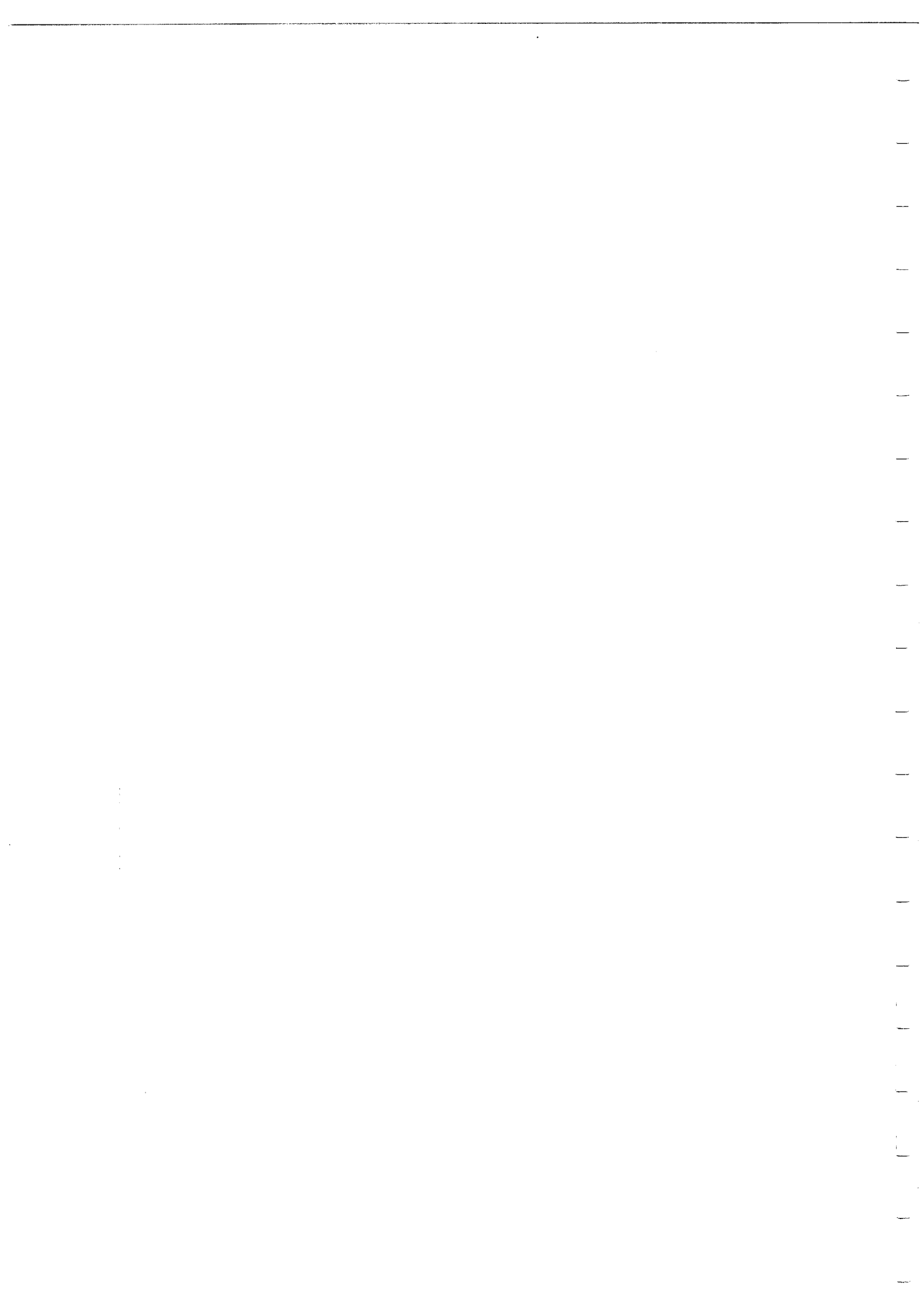
Respecto a las medidas h_L y p_L mostramos que módulo una constante p_L
 es siempre menor que h_L .

Queda abierto el problema de la equivalencia de dichas medidas.



CAPITULO I. PRELIMINARES.

- I.1 Notaciones.
- I.2 Lenguajes racionales y reconocibles.
- I.3 Gramáticas, lenguajes algebraicos y autómatas a pila.
- I.4 Transducciones racionales y conos racionales, conos racionales fieles.
- I.5 Operaciones de substitución de full-A.F.L. y de A.F.L.
- I.6 Centro y familias uniformes.



En este capítulo damos las notaciones que emplearemos, y citamos las distintas operaciones y familias de lenguajes que se utilizarán seguidamente.

I.1 Notaciones.

Un alfabeto X es un conjunto finito no vacío, llamaremos letras a sus elementos. Notaremos X^* el monoide libre engendrado por X . Llamaremos palabras a los elementos de X^* , llamaremos ϵ a su elemento neutro. Notaremos $X^+ = X^* \setminus \epsilon$.

La longitud $|f|$ de una palabra f de X^* es el número de letras de f ; tenemos $|\epsilon| = 0$. Diremos que g es un factor de f si existen u, v que cumplen $f = ugv$. Si $g \neq \epsilon$ y $uv \neq \epsilon$ diremos que g es un factor propio. Una palabra g es un factor izquierdo (respectivamente derecho) notado $g < f$ si existe v que satisface $f = gv$ (respectivamente $f = ug$).

Llamaremos imagen espejo de una palabra $f = x_1 x_2 \dots x_n$, ($n \geq 0$, $x_i \in X$) a la palabra $\tilde{f} = x_n \dots x_2 x_1$.

Con la palabra morfismo, designamos los morfismos $\psi : X^* \longrightarrow Y^*$ en que X e Y son alfabetos. Los tipos especiales de morfismo que nos interesan son:

- ψ es continuo si $\forall x \in X, |\psi(x)| \geq 1$
- ψ es alfabético si $\forall x \in X, |\psi(x)| \leq 1$
- ψ es estrictamente alfabético si $\forall x \in X, |\psi(x)| = 1$

Un lenguaje L de X^* es un subconjunto de X^* . Demos como ejemplos

de lenguajes:

$$S_1 = \{a^n b^n \mid n \geq 1\}$$

$$PAL_2 = \{f\bar{f} \mid f \in \{0,1\}^*\}$$

$$COPY_2 = \{ff \mid f \in \{0,1\}^*\}$$

Diremos que un lenguaje es acotado si existen un conjunto f_1, f_2, \dots, f_k de palabras tales que

$$L \subset f_1^* f_2^* \dots f_k^*$$

Los lenguajes acotados han sido largamente estudiados en [28,30]

Entre lenguajes podemos definir las operaciones siguientes:

- La unión, notada \cup que es la unión de conjuntos
- El producto, notado \cdot , definido por:

$$L_1 \cdot L_2 = \{f_1 \cdot f_2 \mid f_1 \in L_1 \text{ y } f_2 \in L_2\}$$

- La estrella, notada $*$, definida por:

$$L^0 = \varepsilon, L^{n+1} = L^n \cdot L, L^* = \bigcup_{n \geq 0} L^n$$

- El más, notado $+$, definido por:

$$L^+ = L^* \setminus \varepsilon$$

I.2 Lenguajes racionales y reconocibles.

Definición 2.1 La familia de los lenguajes racionales, notada Rat, es la familia más pequeña que contiene los conjuntos finitos y que es estable en las operaciones de unión, producto y estrella.

Vamos seguidamente a introducir el primer tipo de autómeta.

Definición 2.2 Un autómeta finito no determinista

$\mathcal{A} = \langle X, Q, Q_-, Q_+, T \rangle$ está definido por

- . X un alfabeto finito, llamado alfabeto de entrada.
- . Q es un conjunto finito no vacío, llamado conjunto de estados.
- . $Q_- \subset Q$ es el conjunto de estados iniciales.
- . $Q_+ \subset Q$ es el conjunto de estados finales.
- . $T \subset Q \times X \times Q$ es el conjunto de transiciones, notaremos (q, x, q') los elementos de T .

Un cálculo válido es una sucesión:

$(q_0, x_1, q_1) (q_1, x_2, q_2) \dots (q_{n-2}, x_{n-2}, q_{n-1}) (q_{n-1}, x_n, q_n)$

en que

$q_0 \in Q_-$ y $q_n \in Q_+$. La etiqueta de dicho camino es la palabra $m =$

$= x_1 x_2 \dots x_{n-2} x_{n-1}$. Si no nos interesamos en los estados interme-

dios podemos notar dicho camino por $q_0 \xrightarrow{m} q_n$, o notando la

función de transición por un \cdot , es decir, $q_n \in q_0 \cdot m$. El lenguaje re-

conocido por \mathcal{A} es $L(\mathcal{A}) = \{ m \mid q \xrightarrow{m} q', q \in Q_-, q' \in Q_+ \}$

Diremos que un lenguaje es reconocible si existe un autómeta finito

que lo reconoce. Llamaremos Rec a la familia de los lenguajes reconocibles.

La relación entre las familias Rat y Rec viene dada por:

Teorema de Kleene [3 8] 2.1 Rat = Rec

Se cumple:

Teorema 2.2 Los lenguajes racionales (o reconocibles) son estables por unión, producto, estrella, morfismo inverso, imagen espejo y intersección.

I.3 Gramáticas, lenguajes algebraicos y autómatas a pila.

Vamos a centrarnos en los lenguajes context-free que en dicho trabajo llamaremos algebraicos.

Definición 3.1 Una gramática algebraica $G = \langle X, V, P \rangle$ viene dada por:

- . X es un alfabeto finito llamado alfabeto terminal.
- . V es un alfabeto finito, disjunto de X, llamado alfabeto terminal
- . $P \subset V \times (X \cup V)^*$ es el conjunto de reglas, si $(v, m) \in P$ notaremos $v \longrightarrow m$.

Diremos que $f \in (X \cup V)^*$ se deriva en $g \in (X \cup V)^*$ si $f = m_1 v m_2$, $g = m_1 m m_2$ y $(v, m) \in P$, notaremos entonces $f \longrightarrow g$. Notaremos $\xrightarrow{*}$ el cierre reflexivo y transitivo de la relación de derivación.

Llamaremos lenguaje engendrado por la gramática G a partir del axioma $v \in V$ al conjunto:

$$L(G, v) = \{ m \in X^* \mid v \xrightarrow{*} m \}$$

Diremos que un lenguaje es algebraico si existe una gramática y un axioma que lo engendra, notaremos Alg la familia de los lenguajes algebraicos.

Damos seguidamente varios ejemplos de lenguajes algebraicos.

Ejemplo 3.1 Cuando no hay ambigüedad respecto al axioma, notamos la gramática entre paréntesis. Si en P aparecen las reglas $v \longrightarrow m_1$ y

$v \rightarrow m_2$ notaremos $v \rightarrow m_1 + m_2$.

Lenguaje de Dyck a n letras $D_n^* = \langle S \rightarrow \sum_{k=1}^n a_k S \bar{a}_k S + \epsilon \rangle$

Lenguaje de Lukasiewicz $t = \langle S \rightarrow aSS + b \rangle$

Lenguaje de Palíndrome $PAL_n = \langle S \rightarrow \sum_{k=1}^n a_k S a_k + \epsilon \rangle$

Lenguaje E_n $E_n = \langle S \rightarrow \sum_{k=1}^n a_k S b S c_k + d \rangle$

Recordemos que:

Teorema 3.1 Una gramática tiene forma normal de Greibach si

$P \subset VxXV^*$. Todo lenguaje algebraico propio ($L \subset X^+$) admite una gramática bajo la forma normal de Greibach que lo engendra.

Damos seguidamente las propiedades de cierre de Alg.

Teorema 3.2 Los lenguajes algebraicos son cerrados por unión, producto, estrella, imagen espejo, morfismo inverso e intersección con los lenguajes racionales.

A fin de estudiar las propiedades de iteración locales, introducimos el concepto de par iterante.

Definición [6] 3.2 Sea X un alfabeto finito y $L \subset X^*$, diremos que

$\pi = (\alpha, u, \beta, v, \gamma)$ es un par iterante de L si

$$\cdot \alpha u^n \beta v^n \gamma \in L \text{ para todo } n \geq 0$$

$$\cdot |uv| > 0$$

Llamaremos exponentes del par al conjunto

$$\text{Exp } \pi = \{ (n, n') \mid \alpha u^n \beta v^{n'} \gamma \in L \}$$

Diremos que un par es estricto si

$$\text{Exp } \pi = \text{Diag } \pi = \{ (n, n) \mid n \geq 0 \}$$

Cuando L es un lenguaje algebraico, diremos que π es gramatical respecto a la gramática $G = \langle X, V, P \rangle$ y al axioma S si

$$S \xrightarrow{*} \alpha T \beta, \quad T \xrightarrow{*} uTv, \quad T \xrightarrow{*} \beta$$

tenemos entonces

Teorema de Ogden 3.3 Para toda gramática algebraica $G = \langle X, V, P \rangle$

existe un entero k tal que para toda variable S y todo $m \in L(G, S)$

que contiene al menos k posiciones distinguidas, existe un par iterante

gramatical $\pi = (\alpha, u, \beta, v, \gamma)$ que cumple:

- α y u ó v y γ contienen posiciones distinguidas
- β contiene posiciones distinguidas
- $u \beta v$ contiene como máximo k posiciones distinguidas

Demos seguidamente una definición alternativa de lenguajes algebraicos en términos de autómatas.

Definición 3.3 Un autómata a pila viene dado por:

$$A = \langle X, Z, Q, T, q_-, z_0, Q_+ \rangle$$

en que:

- . X es el alfabeto finito de entrada
- . Z es el alfabeto finito de pila, $z_0 \in Z$ es el fondo de pila
- . Q es el conjunto finito de estados, $q_- \in Q$ es el estado inicial
- . y $Q_+ \subset Q$ es el conjunto de estados finales
- . El conjunto de transiciones es $T \subset Q \times \{ X + \varepsilon \} \times Z \times Q \times Z^*$ si $(q, a, z, q', \beta) \in T$ notaremos $(q, a, z) \vdash (q', \beta)$

Definimos movimientos en el autómata por:

$$(q, au, \alpha z) \vdash (q', u, \alpha \beta)$$

en que $(q, a, z) \vdash (q', \beta)$ es una regla de T.

El lenguaje reconocido por dicho autómata viene dado por:

$$TN(A) = \{ w \in X^* \mid (q_-, w, z_0) \vdash^* (q, \varepsilon, \varepsilon) q \in Q_+ \}$$

tenemos el teorema

Teorema 3.4

Un lenguaje es algebraico si y sólo si existe un autómata a pila que

lo reconoce.

Vamos seguidamente a introducir dos subfamilias de lenguajes algebraicos.

Definición 3.4

(a) Diremos que una gramática $G = \langle X, V, P \rangle$ es lineal si $P \subseteq VxX^*VX^*$. Notaremos Lin la familia de los lenguajes lineales.

(b) Diremos que un autómata a pila $A = \langle X, Z, Q, T, q_-, z_0 \rangle$ es un contador restringido si $Z = \{z_0\}$, notaremos Rocl dicha familia.

Ejemplo 3.2 $PAL_n = \langle S \longrightarrow \sum a_k S a_k + \epsilon \rangle$ se cumple

$PAL_n \in \text{Lin}$

Además D_1^* es reconocido por el autómata cuyas transiciones son:

$(q_-, a, z) \longmapsto (q_-, z^2)$, $(q_-, a, z) \longmapsto (q_-, \epsilon)$, $(q_-, \epsilon, z) \longmapsto (q_+, \epsilon)$

con lo que $D_1^* \in \text{Rocl}$

(3) Existe un alfabeto Z , dos morfismos $\varphi : Z^* \longrightarrow X^*$, $\psi : Z^* \longrightarrow Y^*$ y un racional $K = AZ^* \cup Z^*B \setminus Z^*VZ^*$ en que $A, B \subset Z$ y $V \subset Z^2$ tales que

$$\forall m \in X^* , \quad \tau(m) = \psi(\varphi^{-1}(m) \cap K)$$

(4) Si $X \cap Y \neq \emptyset$ existe un racional $K \subset (X \cup Y)^*$ que cumple

$$\forall m \in X^* , \quad \tau(m) = \pi_Y(\pi_X^{-1}(m) \cap K)$$

en que π_X y π_Y son las proyecciones de $(X \cup Y)^*$ en X^* e Y^* respectivamente.

Cuando τ es una transducción fiel que satisface $\varepsilon \in \tau(m) \implies m = \varepsilon$, existe un morfismo $\varphi : Z^* \longrightarrow X^*$, un morfismo $\psi : Z^* \longrightarrow Y^*$ estrictamente alfabético y un racional $K \subset Z^*$ que satisfacen

$$\forall m \in X^* , \quad \tau(m) = \psi(\varphi^{-1}(m) \cap K)$$

Tenemos que las transducciones racionales satisfacen

Teorema de Elgolt y Mezei [19] 4.2 La composición de dos transducciones racionales (transducciones racionales fieles) es una transducción racional (transducción racional fiel).

Podemos ahora definir las familias de lenguajes que tienen un comportamiento agradable en términos de familias de lenguajes.

Definición [4] 4.2 Diremos que una familia de lenguajes \mathcal{L} es un cono racional si es estable por transducción racional. Notaremos $\mathcal{L}(\mathcal{L})$ la clausura de \mathcal{L} por transducción racional. Diremos que \mathcal{L} es principal si existe un lenguaje $L \in \mathcal{L}$ que satisface $\mathcal{L} = \mathcal{L}(L)$. En este caso diremos que L es un generador de \mathcal{L} . Llamaremos $\text{Gen}(\mathcal{L})$ la familia de los lenguajes generadores de \mathcal{L} . Llamaremos $\text{Nge}(\mathcal{L}) = \mathcal{L} \setminus \text{Gen}(\mathcal{L})$ a la familia de los lenguajes no generadores de \mathcal{L} . Cuando tratamos con transducciones racionales fieles, obtenemos los conceptos de cono racional fiel, en este caso notaremos

$\mathcal{L}^f(\mathcal{L})$; llamaremos generador fiel a un lenguaje $L \in \mathcal{L}$ tal que $\mathcal{L} = \mathcal{L}^f(L)$, obtenemos entonces $\text{Gen}^f(\mathcal{L})$ y $\text{Nge}^f(\mathcal{L})$.

El teorema de Elgolt y Mezei nos permite concluir

Lema 4.1

- (a) Una familia de lenguajes es un cono racional ssi es estable en las operaciones de morfismo, morfismo inverso e intersección con los lenguajes racionales.
- (b) Una familia de lenguajes es un cono racional fiel ssi es estable en las operaciones de morfismo \mathcal{E} -limitado, morfismo inverso e intersección con los lenguajes racionales.

El teorema siguiente muestra la principalidad de las familias Alg, Lin, Rcl.

Teorema [4,33] 4.3

$$\underline{\text{Alg}} = \mathcal{L}(D_2^! *) = \mathcal{L}^f(D_2^! *)$$

$$\underline{\text{Lin}} = \mathcal{L}(S_2) = \mathcal{L}^f(S_2) \quad \text{con } S_2 = \langle s \rightarrow \sum_{k=1}^2 a_k s \bar{a}_k + \epsilon \rangle$$

$$\underline{\text{Rcl}} = \mathcal{L}(D_1^! *) = \mathcal{L}^f(D_1^! *)$$

Recordemos que uno de los grandes problemas abiertos en la teoría de lenguajes algebraicos es demostrar

Conjetura 4.1 Nge (Alg) es un cono no principal.

Podemos adoptar un subconjunto de las operaciones de cono racional, por ejemplo:

Definición [2] 4.3 Llamaremos cilindro una familia de lenguajes estable por morfismo inverso e intersección con los lenguajes racionales.

I.5 Operaciones de substitución y de full-A.F.L.

Comencemos por definir la substitución

Definición [7] 5.1 Una substitución σ es un morfismo de X^* en

$\mathcal{P}(Y^*)$. Diremos que σ es una \mathcal{L} -substitución si $\sigma(x) \in \mathcal{L}$ para cualquier $x \in X$. Sea \mathcal{M} una familia de lenguajes notaremos

$\mathcal{M} \square \mathcal{L}$ la familia:

$\mathcal{M} \square \mathcal{L} = \{ \sigma(L) \mid L \in \mathcal{M}, \sigma : X_L^* \longrightarrow Y^* \text{ es una } \mathcal{L} \text{ substitución} \}$

Notaremos la clausura por substitución de \mathcal{L} por $\mathcal{L} \bar{\mathcal{L}}^*$ en que:

$$\mathcal{L}^1 = \mathcal{L}, \quad \mathcal{L}^k = \mathcal{L} \square \mathcal{L}^{k-1}, \quad \mathcal{L} \bar{\mathcal{L}}^* = \bigcup_{k \geq 1} \mathcal{L}^k$$

Definimos, seguidamente, los full-A.F.L. y los A.F.L.

Definición [4] 5.2

. Llamaremos full-A.F.L. a todo cono racional estable en las operaciones unión, producto y estrella. Notaremos $\mathcal{F}(\mathcal{L})$ la clausura de \mathcal{L} respecto a las operaciones de A.F.L.

. Llamaremos A.F.L. a todo cono racional fiel estable en las operaciones unión, producto, más. Notaremos, en este caso, $\mathcal{F}^f(\mathcal{L})$ la clausura en tanto que A.F.L. de \mathcal{L} .

Se cumple:

Teorema [4] 5.1

$$\mathcal{F}(\mathcal{L}) = \underline{\text{Rat}} \square \mathcal{L}(\mathcal{L}) ; \mathcal{F}^{\text{f}}(\mathcal{L}) = \underline{\text{Rat}} \square \mathcal{L}^{\text{f}}(\mathcal{L})$$

$$\mathcal{F}(L) = \mathcal{L}(\{L \# \}^*) ; \mathcal{F}^{\text{f}}(L) = \mathcal{L}^{\text{f}}(\{L \# \}^*)$$

Podemos, ahora construir nuevas familias de lenguajes.

Teorema [6] 5.2 Las familias siguientes

$$\underline{\text{Qrt}} = \underline{\text{Lin}} \overline{\underline{\text{Lin}}}^*$$

$$\underline{\text{Fcl}} = \underline{\text{Rocl}} \overline{\underline{\text{Rocl}}}^*$$

$$\underline{\text{Gre}} = (\underline{\text{Rocl}} \cup \underline{\text{Lin}}) \overline{(\underline{\text{Rocl}} \cup \underline{\text{Lin}})}^*$$

son full-A.F.L. no principales estables por substitución.

I.6 Centro y familias uniformes

Vamos seguidamente a estudiar el centro de un lenguaje. Dicha noción ha sido introducida por L. Boasson y M. Nivat [9] en su estudio sobre las adherencias de lenguajes.

Definición [9] 6.1 Sea L un lenguaje de X^* definimos su centro por:

$$C(L) = \left\{ m \in X^* \mid \text{Card} (mX^* \cap L) = \infty \right\}$$

Damos seguidamente algunos ejemplos.

Ejemplo [23] 6.1

. Sea $L = \{ a^n b^n \mid n \geq 0 \}$ entonces $C(L) = a^*$

. Sea $D_2'^* = \langle S \rightarrow \sum_{k=1}^2 a_k S \bar{a}_k S + \epsilon \rangle$ entonces

$$C(D_2'^*) = \left\{ \sum_{k=1}^2 a_k D_2'^* \bar{a}_k + \sum_{k=1}^2 a_k \right\}^*$$

. Sea $E = \langle S \rightarrow aSbS + d \rangle$ entonces

$$C(E) = (a + aE)^* \left\{ \epsilon + a(aE)^* aEE \right\}$$

Se cumple que

Teorema [23] 6.1 Sea $L = L(G, v)$ un lenguaje algebraico, existe un lenguaje racional K que cumple

$$C(L) = FG \quad K \left[v_i \longleftarrow L(G, v_i) \right]$$

en que FG significa el conjunto de los factores izquierdos.

A fin de relacionar el centro con las familias de lenguajes, vamos

a definir las familias uniformes

Definición [10] 6.2

. Diremos que una gramática $G = \langle X, V, P \rangle$ es uniforme en \mathcal{L} si:

$$\forall v \in V \text{ se cumple } L(G, v) \in \mathcal{L}.$$

. Un lenguaje algebraico es uniforme en \mathcal{L} si existe una gramática uniforme que lo engendra

. Una familia \mathcal{L} es uniforme si cualquier $L \in \mathcal{L}$ es uniforme en \mathcal{L}

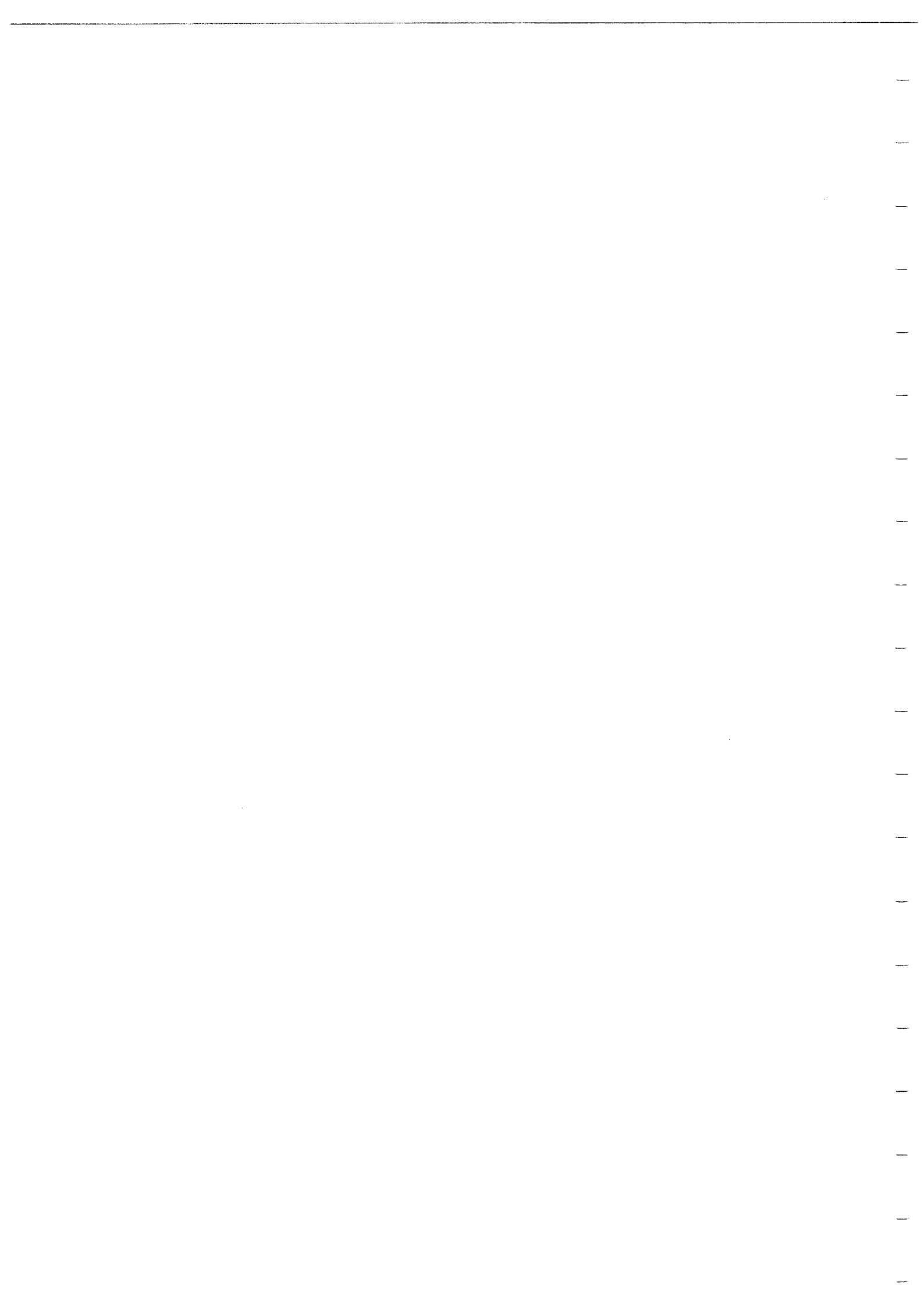
Se cumple:

Teorema 6.2 Los full-A.F.L. Qrt, Fcl y Gre son uniformes.

La relación entre las familias uniformes y el centro viene dada por

Teorema 6.3 [10, 23] Sea $\mathcal{L} \subset \text{Alg}$ un full-A.F.L. uniforme si $L \in \mathcal{L}$ entonces $C(L) \in \mathcal{L}$

tenemos un corolario que Qrt, Fcl y Gre son uniformes.



CAPITULO II. FUNCIONES DE MEDIDA EN LEGUAJES FORMALES.

II.1 Función de crecimiento.

II.2 a-medida.

II.3 Índice racional.

II.4 Costo rígido.

II.5 Costo boleano.

II.6 h-medida.



En este capítulo se introducen algunas de las distintas medidas de complejidad para lenguajes formales. En la medida de lo posible, se tratan para cada una de ellas distintos ejemplos significativos y se enuncian los principales resultados que de ellas se conocen. Comenzamos por estudiar la función de crecimiento introducida por Milnor. Dicha medida se obtiene contando el número de palabras que existen en un lenguaje, que tienen una longitud dada. Vemos seguidamente que un teorema, debido a Ginsburg y Spanier, afirma que en el caso de lenguajes algebraicos acotados, dicha función de crecimiento es un polinomio, siendo en los otros casos exponencial. Concluimos, entonces, que en el caso de lenguajes algebraicos es decidible saber si la función de crecimiento está acotada superiormente por un polinomio, o inferiormente por una exponencial. Nos centramos posteriormente a los lenguajes racionales. Citamos, entonces, los trabajos de A. Maurer y M. Nivat sobre las transducciones racionales entre lenguajes racionales, que son una biyección, y que se llaman biyecciones racionales. Se da la condición necesaria y suficiente para que una tal biyección exista entre dos racionales. Estudiamos seguidamente una segunda función de complejidad, la a-medida. Dicha función ha sido introducida por J. Paredaens y R. Vyncke. La a-medida se obtiene contando el número de estados del autómata minimal que reconoce un lenguaje dado. Mostramos el comportamiento de dicha medida en lenguajes como el de palíndromos o el de paréntesis. Estudiamos, luego, las propiedades de clausura de la a-medida, concluyendo que la familia de lenguajes de a-medida acotada por un polinomio, es un cilindro.

Seguidamente, se estudia el índice racional. Dicha función de complejidad ha sido introducida por L. Boasson, B. Courcelle y M. Nivat en relación al problema de los no generadores de los lenguajes algebraicos. Tras mostrar las técnicas empleadas en el cálculo del índice racional en numerosos lenguajes, estudiamos su relación con las familias abstractas de lenguajes. Se muestra que los lenguajes de índice racional polinomial, son un full-A.F.L. no principal, cerrado por substitución y abierto por complementación. Centrándonos en los lenguajes algebraicos, se demuestra la inclusión estricta de la familia de lenguajes de Greibach, en los lenguajes polinomiales.

La siguiente función de complejidad, que estudiamos, es el costo rígido. Dicha medida ha sido introducida por G. B. Goodrich, R. E. Ladner y M. J. Fischer. El costo rígido se basa en contar el número de uniones y concatenaciones que utiliza un programa rígido para calcular una aproximación de un lenguaje.

Vemos que existe una notable diferencia respecto al coste rígido entre el lenguaje de palíndromos y el lenguaje de cuadrados. Se muestra que todo lenguaje algebraico tiene un coste rígido acotado por una función cuadrática.

Otra medida que consideramos, es el coste boleano. Dicha función ha sido introducida por E. Savage. Se funda en el número de puertas lógicas necesarias para reconocer distintas aproximaciones de un lenguaje dado. Vemos que en este caso, no existe notable diferencia entre los palíndromos y los cuadrados. Citamos, finalmente, un teorema debido a M. J. Fischer y N. Pippenger, que da una cota al costo boleano en el caso de máquinas de Turing deterministas.

Analizamos, finalmente, la h-medida. Esta medida introducida por K. Culik y H.A.Maurer, es utilizable sólo en el caso de lenguajes algebraicos, y se funda en estudiar la altura del árbol de derivación en función de la longitud. Se muestra que todo algebraico no racional de a a b tiene una altura lineal; podemos, entonces, concluir que todo lenguaje estrictamente no racional tiene una altura lineal; en particular los deterministas.

II.1 Función de crecimiento.

La función de crecimiento ha sido introducida por Milnor [4 5] en el estudio de grupos finitamente generados en problemas de geometría diferencial. Nosotros estamos interesados en dicha función de crecimiento, en su aplicación a la teoría de lenguajes formales. Comencemos por dar su definición:

Definición [4 5] 1.1 Sea $L \subset X^*$, definimos la función de crecimiento de L por:

$$g_L(n) = \{ \text{Card } w \in L \mid |w| \leq n \}$$

Debido a su constante utilización en los ejemplos, introducimos la noción de segmento inicial de longitud n de L por

$$nL = \{ w \in L \mid |w| \leq n \}$$

Demos seguidamente algunos ejemplos de función de crecimiento.

Ejemplo 1.1 Sea $L = \{ a^n b^n \mid n \geq 0 \}$ evidentemente

$$g_L(2n) = n + 1$$

Ejemplo 1.2 Vamos a tratar seguidamente $X_k^* = \{ a_1, a_2, \dots, a_k \}^*$

Sabemos que $g_{X_k^*}(n) = \text{Card } n X_k^*$, además

$$n X_k^* = \bigcup_{t=0}^n \{ a_1, a_2, \dots, a_k \}^t. \text{ Dado que } \text{Card } \{ a_1, \dots, a_k \}^t = k^t$$

tenemos que $g_{X_k^*}(n) = \sum_{t=0}^n k^t = \frac{k^{n+1} - 1}{k - 1}$ resumimos diciendo

$$g_{X_k^*}(n) = \frac{k^{n+1} - 1}{k - 1} = \Theta(k^n) \text{ si } k > 1$$

Ejemplo 1.3 Demos seguidamente la función de crecimiento del lenguaje acotado a k letras

$$B_k = a_1^* a_2^* \dots a_k^*$$

Recordemos que el lenguaje a^* tiene por serie generatriz

$$\frac{1}{1-a} = \epsilon + a + a^2 + \dots$$

Sea z una nueva letra distinta de cualquier a_i , $1 \leq i \leq k$ se cumple:

$$\text{Card } n \ a_1^* a_2^* \dots a_k^* = \text{Card} \{ w \in a_1^* a_2^* \dots a_k^* z^* \mid |w| = n \}$$

El lenguaje $a_1 a_2 \dots a_k z$ tiene como serie generatriz:

$$\frac{1}{1-a_1} \cdot \frac{1}{1-a_2} \dots \frac{1}{1-a_k} \cdot \frac{1}{1-z}$$

A fin de contar los elementos de longitud n en dicha serie, basta hallar el coeficiente de x^n en la serie conmutativa que se obtiene confundiendo todas las letras con x ; dicha serie es

$$\left(\frac{1}{1-x} \right)^{k+1}$$

Es bien conocido que el coeficiente de x^n en dicha serie es

$$[n] \left(\frac{1}{1-x} \right)^{k+1} = \binom{n+k}{n}$$

tenemos entonces que

$$g_{B_k}(n) = g_{a_1^* a_2^* \dots a_k^*}(n) = \binom{n+k}{n}$$

Ejemplo 1.4 Vamos a estudiar seguidamente el lenguaje de Dyck a una letra D_1^* = $\langle S \rightarrow aS\bar{a}S + \epsilon \rangle$

Es bien conocido que el número de palabras de Dyck con n paréntesis abiertos, está dado por el número de Catalán, es decir:

$$\{w \in D_1^* \mid |w| = 2n\} = \frac{1}{n+1} \binom{2n}{n}$$

podemos entonces concluir

$$g_{D_1^*}(n) = \sum_{x=0}^n \frac{1}{x+1} \binom{2x}{x}$$

Ejemplo 1.5 Tratemos ahora D_p^* = $\langle S \rightarrow \sum_{i=1}^p a_i S \bar{a}_i S + \epsilon \rangle$

con $p > 1$. Evidentemente tenemos la igualdad:

$$\text{Card} \{w \in D_p^* \mid |w| = 2n\} = p^n \text{Card} \{w \in D_1^* \mid |w| = 2n\}$$

podemos concluir entonces

$$g_{D_p^*}(n) = \sum_{x=0}^n p^x \frac{1}{x+1} \binom{2x}{x}$$

Ejemplo 1.6 Vamos ahora a tratar el lenguaje FIFO_p definido en [4] como la clase \mathcal{E} en la congruencia definida en

$$\{a_1, \dots, a_p, \bar{a}_1, \dots, \bar{a}_p\} \text{ por}$$

$$w_1 \stackrel{\alpha}{\equiv} w_2 \iff w_1 = a_i x \bar{a}_i y, w_2 = xy, x \in \{a_1, \dots, a_p\}^*$$

Se cumple que

$$\text{Card} \{w \in D_p^* \mid |w| = 2n\} = \text{Card} \{w \in \text{FIFO}_p \mid |w| = 2n\}$$

podemos concluir entonces

$$g_{D_p^*}(n) = g_{\text{FIFO}_p}(n)$$

Estudiamos seguidamente cuál es el comportamiento de la función de crecimiento en el caso de los lenguajes algebraicos. Los resultados que siguen, son una reescritura de resultados debidos a Ginsburg y Spanier [3 0] en términos de función de crecimiento. Recordemos que un lenguaje L es acotado si existe un conjunto de palabras w_1, w_2, \dots, w_k que satisfacen

$$L \subset w_1^* w_2^* \dots w_k^*$$

Enunciemos seguidamente un lema respecto a la conmutatividad de dos palabras.

Lema [3 0] 1.1 Sean $u, v \in X^+$, las siguientes afirmaciones son equivalentes:

- . u y v conmutan ; $uv = vu$
- . existen $p, q \geq 1$ que satisfacen $u^p = v^q$
- . existe $w \in X^+, r, s \geq 1$ tales que $u = w^r$ y $v = w^s$

dicho lema admite como consecuencia

Lema [3 0] 1.2 Sean $u, v \in X^+$, las afirmaciones siguientes son equivalentes:

- . u y v no conmutan.
- . $L = \{u, v\}^*$, no es un lenguaje acotado.
- . L tiene una función de crecimiento exponencial.

Vamos a relacionar seguidamente la función de crecimiento con la aparición de bucles en la gramática, cuando L es algebraico.

Lema [30] 1.3 Sea $L \neq \emptyset$ engendrado por $G = \langle X, V, P \rangle$ para cada $v \in V$ notemos:

$$Y_v(G) = \{ m \mid m \in X^*, \exists n \text{ con } v \xrightarrow{*} m v n \}$$

$$Z_v(G) = \{ n \mid n \in X^*, \exists m \text{ con } v \xrightarrow{*} m v n \}$$

las tres condiciones siguientes son equivalentes.

. Existe $v \in V$ tal que al menos uno de los dos conjuntos

$Y_v(G)$ ó $Z_v(G)$ no es conmutativo

. L no es un lenguaje acotado

. g_L crece como una exponencial

Podemos resumir dichos lemas en:

Teorema [30] 1.1 Un lenguaje algebraico tiene una función de crecimiento polinomial ssi es acotado, si no, su función de crecimiento es una exponencial.

Dado que es decidible si un lenguaje algebraico es acotado, es también decidible si su función de crecimiento es un polinomio o una exponencial.

Vamos a dar seguidamente los resultados debidos a A.Maurer y M. Nivat que relacionan la función de crecimiento y las transducciones racionales en el caso de lenguajes racionales, tenemos:

Teorema [44] 1.2 Sean R_1 y R_2 dos lenguajes racionales, existe entre ellos una transducción racional que es una biyección ssi existe p_1 y p_2 y un entero n_0 , tales que:

$$g_{R_1}(p_1 n) \leq g_{R_2}(n) \leq g_{R_1}(p_2 n) , n \geq n_0$$

Si existe una biyección racional entre R_1 y R_2 notaremos $R_1 \longleftrightarrow R_2$. De modo informal el teorema precedente dice que existe una biyección racional entre dos racionales, si, y solamente si, tienen aproximadamente la misma función de crecimiento. El lema siguiente da varios ejemplos de biyecciones racionales.

Lema [44] 1.4

. Si $p \geq q$

$$a_1^* a_2^* \dots a_p^* + b_1^* b_2^* \dots b_q^* \longleftrightarrow a_1^* a_2^* \dots a_k^*$$

. Para todo $p \geq 1$

$$(a_1 + a_2)^* \longleftrightarrow (a_1 + a_2)^* + a_1^* a_2^* \dots a_p^*$$

. Para todo $p, q \geq 2$

$$(a_1 + a_2 + \dots + a_p)^* \longleftrightarrow (a_1 + a_2 + \dots + a_q)^*$$

. Para todo $p, q \geq 1$

$$a_1^* \dots a_p^* b_1^* \dots b_q^* \longleftrightarrow a_1^* a_2^* \dots a_{p+q}^*$$

. Para todo $p, q \geq 1$

$$a_1^* \dots a_p^* (b_1 + b_2)^* \longleftrightarrow (b_1 + b_2)^*$$

. Para todo $p \geq 1$

$$(ca_1^* \dots a_p^*)^* \longleftrightarrow (a_1 + a_2)^*$$

II.2 a-medida.

La a-medida ha sido introducida por J. Paredaens y R. Vyncke en [47]. Dicha medida es de naturaleza combinatoria, y consiste en contar en el autómata minimal que reconoce un lenguaje dado, el número de estados accesibles a partir de palabras de longitud acotada.

Vamos seguidamente a mostrar el comportamiento de dicha medida en los lenguajes más usuales.

A fin de dar la definición formal de a-medida, comencemos por recordar que un autómata determinista $\mathcal{A} = \langle X, Q, q_0, Q_+, \dots \rangle$ será dicho completo, cuando la función de transición es una aplicación total de $Q \times X$ en Q . En un autómata completo $q_0 \cdot w$ está definido para cualquier palabra $w \in X^*$.

Damos seguidamente la definición de T-medida.

Definición [47] 2.1 Sea L un lenguaje sobre X^* y sea $\mathcal{A} = \langle X, Q, q_0, Q_+, \dots \rangle$ su autómata determinista minimal completo, llamaremos a-medida de L a la función:

$$a_L(n) = \text{Card } q_0 \cdot nX^*$$

Remarca 2.1 La medida \hat{a}_L definida por Paredaens y Vyncke, contiene pequeñas variaciones con respecto a la aquí definida. Tenemos

$a_L(n) = 1 + \text{Card } q_0 \cdot X^n$. Nosotros hemos cambiado la definición a fin de que $a_L(n)$ sea una función creciente; tenemos:

Lema 2.1 La a-medida de $L \subset X^*$ satisface:

- $a_L(n) \leq \text{Card } n X^*$
- $a_L(n) \leq a_L(n + 1)$

Vamos seguidamente a calcular la a-medida de los lenguajes más usuales.

Ejemplo 2.1 Sea $S_1 = \{a^n b^n \mid n \geq 0\}$ tenemos que

$$a_{S_1}(n) = 2n - 1$$

(a) Comenzamos por dar el autómata minimal que describe S_1 . Es bien conocido que dicho autómata $A = \langle \{a, b\}, Q, q_-, q_+, \dots \rangle$ está definido por:

$$Q = \{(n, s) \mid n \in \mathbb{N}, s \in \{+, -\}\}$$

$$q_- = (0, +), \quad q_+ = (0, -)$$

La función de transición está definida por:

$$(n, +).a = (n + 1, +), \quad n \geq 0$$

$$(n, +).b = (n - 1, -), \quad n > 0$$

$$(n, -).b = (n - 1, -), \quad n > 0$$

Dicho autómata puede esquematizarse en la figura II.1

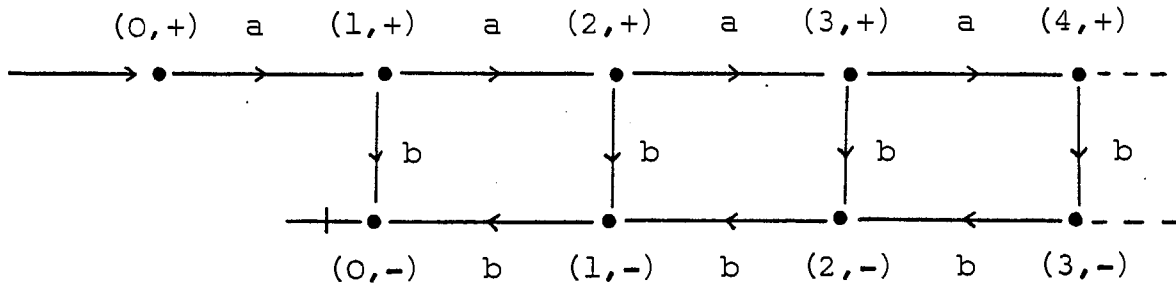


Figura II.11.

(b) A partir del autómata minimal podemos calcular la a-medida:

$$q_-.n \{a,b\}^* = \{ (p,+) \mid 0 \leq p \leq n \} \cup \{ (p,-) \mid 0 \leq p \leq n-2 \}$$

Vemos que el autómata minimal que hemos dado no es completo. A fin de completarlo debemos, añadir un estado. Llegamos a

$${}^a_{S_1}(n) = 2n - 1$$



Ejemplo 2.2 Vamos a tratar seguidamente el lenguaje de Dyck $D_1^* =$

$$\langle S \rightarrow aS\bar{a}S + \epsilon \rangle$$

(a) El autómata minimal para D_1^* es $\mathcal{A} = \langle \{a,b\}, Q, q_-, q_+, \cdot \rangle$

con:

$$Q = \{ (n) \mid n \in \mathbb{N} \}, \quad q_- = q_+ = (0)$$

La función de transición está definida por:

$$(n).a = (n+1), \quad n \geq 0$$

$$(n).\bar{a} = (n-1), \quad n > 0$$

Podemos esquematizar dicho autómata en la figura II.2.

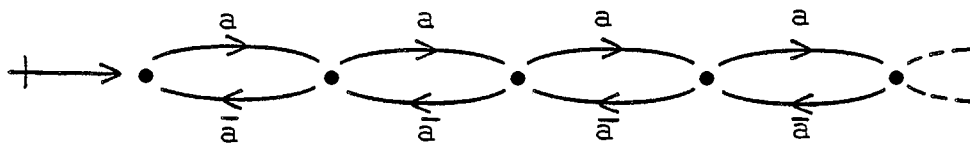


Figura II.2.

(b) La a -medida viene dada por:

$$q_{-}.n \{a, \bar{a}\}^* = \{ (p) \mid 0 \leq p \leq n \}$$

A fin de completar dicho autómata añadimos otro estado, con lo que:

$$a_{D_1}^*(n) = n + 1$$

Ejemplo 2.3 Vamos a tratar $L = \{ w \# \bar{w} \mid w \in \{a, b\}^* \}$

(a) Comenzamos por estudiar su autómata minimal

$$A = \langle \{a, b, \# \}, Q, q_-, q_+, \cdot \rangle$$

definido por

$$Q = \{ (w, s) \mid w \in \{a, b\}^*, s \in \{+, -\} \}$$

$$q_- = (\epsilon, +), \quad q_+ = (\epsilon, -)$$

$$(w, +).u = (wu, +), \quad u \in \{a, b\}$$

$$(w, +).\# = (w, -)$$

$$(wu, -).\bar{u} = (w, -), \quad u \in \{a, b\}$$

Veamos que dicho autómata es minimal. Dado que es determinista veamos que no podemos confundir dos estados cualquiera.

No podemos confundir $(w,+)$ con $(w',-)$ dado que $(w,+). \# \tilde{w} = q_-$ y $(w',-). \# \tilde{w}$ no está definido.

Si $w \neq w'$ no podemos, tampoco, confundir $(w,+)$ con $(w',+)$ dado que $q_-.w \# \tilde{w} = (w,+). \# \tilde{w} = q_+$ y $(w',+). \# \tilde{w}$ no está definido. De modo análogo tratamos $(w,-), (w',-)$ con $w \neq w'$. Dicho autómata se halla esquematizado en la figura siguiente, figura II.3.

(b) Vamos a calcular la a-medida de L. Tenemos que a partir de q_0 , podemos procesar dos tipos de palabras de $n \{a,b, \# \}^*$; procesamos las palabras $w \in n \{a,b \}^*$, dichas palabras conducen a estados $(w,+)$; procesamos también las palabras de la forma $wf \# \tilde{f}$ con

$|wf \# \tilde{f}| \leq n$ que nos conducen a estados $(w,-)$. Tenemos:

$$q_0.n \{a,b, \# \}^* = \{ (w,+) \mid w \in n \{a,b \}^* \} \cup \{ (w,-) \mid w \in (n-1) \{a,b \}^* \}$$

A fin de completar dicho autómata añadimos otro estado, con lo que

$$a_L(n) = (2^{n+1} - 1) + (2^n - 1) + 1 = 3 \cdot 2^n - 1$$



Ejemplo 2.4 Vamos a tratar seguidamente el lenguaje de Dyck a dos letras $D_2^* = \langle S \rightarrow a\bar{a}S + b\bar{b}S + \epsilon \rangle$

(a) Recordemos el autómata minimal $\mathcal{A} = \langle \{a, \bar{a}, b, \bar{b}\}, q_0, q_-, q_+, \cdot \rangle$ para D_2^* , definido por:

$$Q = \{ (w) \mid w \in \{a,b \}^* \}, \quad q_- = q_+ = (\epsilon)$$

La función de transición está definida por:

$$(w).u = (wu), \quad u \in \{a,b\}$$

$$(wu).\bar{u} = (w), \quad u \in \{a,b\}$$

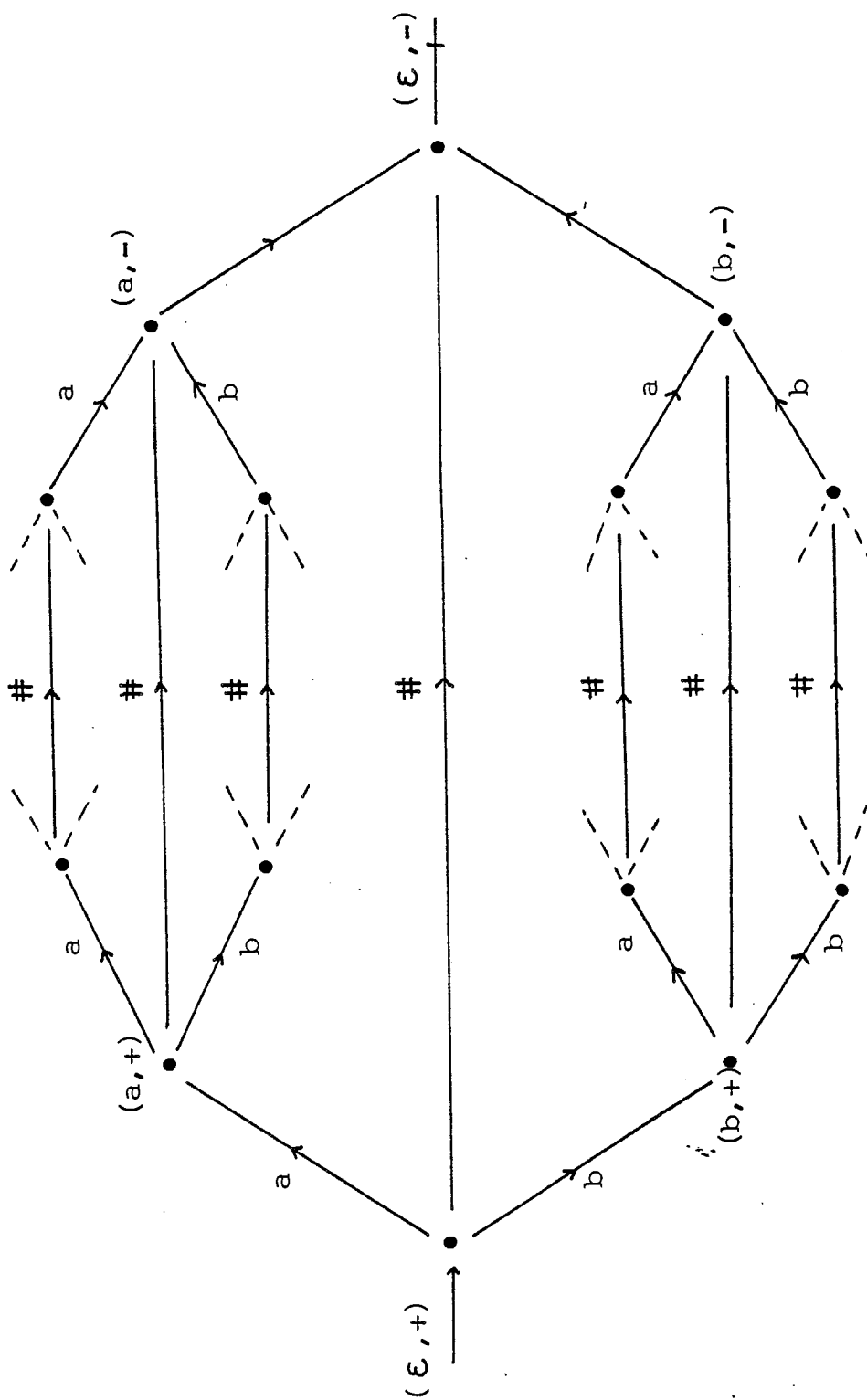


Figura II.3.

Dicho autómata puede esquematizarse en la figura siguiente, figura II.4.

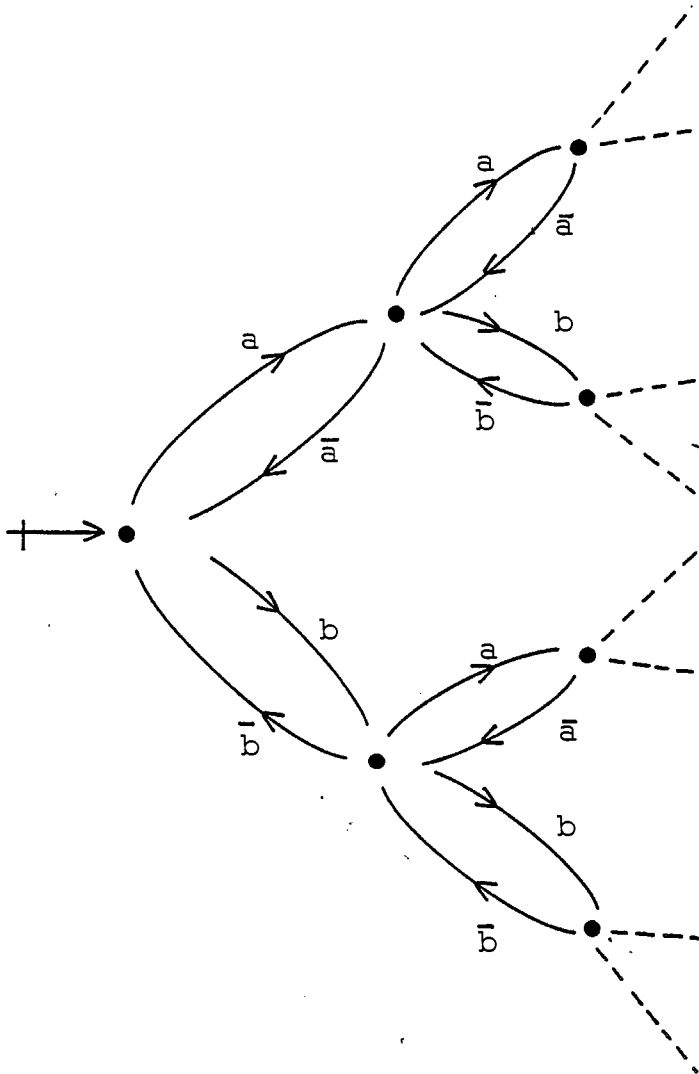


Figura II.4.

(b) A fin de calcular $a_{D_2^1}^*(n)$ calculamos

$$q_{-n} \{ a, \bar{a}, b, \bar{b} \}^* = \{ (w) \mid w \in n \{ a, b \}^* \}$$

$$\text{Dado que } \{ a, b \}^n = 2^n, \text{ tenemos } a_{D_2^1}^*(n) = \sum_{j=0}^n 2^j = 2^{n+1} - 1$$

Completando el autómata precedente llegamos a:

$$a_{D_2^1}^*(n) = 2^{n+1}$$

Ejemplo 2.5 Tratemos seguidamente el lenguaje de Dyck bilátero.

$$D_2^* = \langle s \longrightarrow aS\bar{a}S + bS\bar{b}S = \bar{a}SaS = \bar{b}SbS + \epsilon \rangle$$

Dicho lenguaje puede definirse como la clase de en la congruen-

cia en $X^* = \{ a, \bar{a}, b, \bar{b} \}^*$ engendrada por $a\bar{a} \equiv \bar{a}a \equiv b\bar{b} \equiv \bar{b}b \equiv \epsilon$

Consideremos la relación ρ definida en $\{ a, \bar{a}, b, \bar{b} \}$ por

$$\rho = \{ (a, \bar{a}), (\bar{a}, a), (b, \bar{b}), (\bar{b}, b) \}$$

A partir de ρ podemos inductivamente definir el autómata minimal

que reconoce D_2^* por

$$(\epsilon).x = (x) ; x \in \{ a, \bar{a}, b, \bar{b} \}$$

Sea (wx) un estado con x una letra, entonces

$$(wx).y = \begin{cases} \text{Si } (x,y) \in \rho & \text{entonces } (w) \\ \text{sino} & (wxy) \end{cases}$$

Siendo (ϵ) el estado inicial y final.

En la figura II.5. mostramos esquemáticamente los primeros estados de dicho autómata.

Dado que los factores izquierdos de D_2^* es todo X^* , dicho autómata es además completo.

Vemos que a partir del estado (a) se genera un árbol terciario infinito. Lo mismo ocurre a partir de $(\bar{a}), (b)$ y (\bar{b}) . Para ver cuantos estados accesibles a partir de $(\epsilon).X^n$, basta notar que $(\epsilon).X^n =$

$$= \bigcup_{x \in X} (x).X^{n-1} + (\epsilon) \text{ con lo que } a_{D_2^*}(n) = 4 \cdot \sum_{x=0}^{n-1} 3^x + 1 =$$

$$= 2 \cdot 3^n - 1.$$

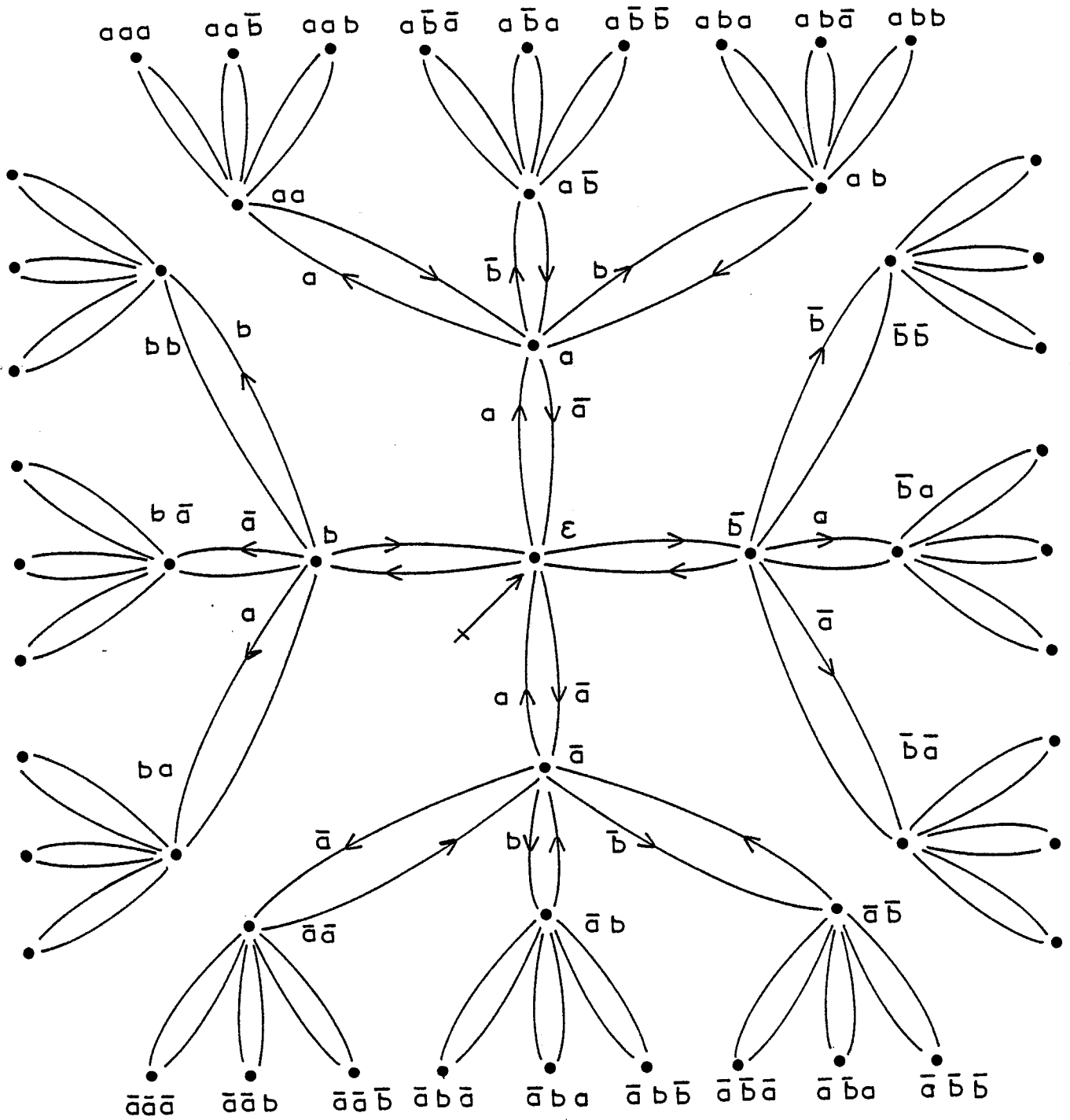


Figura II.5.

Ejemplo 2.6 Estudiamos seguidamente la a -medida de $FIFO_2$. Dicho lenguaje se define en [4] como la clase de ε en la congruencia

definida para palabras en $\{a, \bar{a}, b, \bar{b}\}^*$ por

$$w_1 \stackrel{\alpha}{\equiv} w_2 \iff w_1 = ax\bar{a}y \text{ ó } w_1 = bx\bar{b}y, w_2 = xy, x \in \{a, b\}^*$$

Si w es un factor izquierdo de $FIFO_2$, entonces $\alpha(w) \in \{a, b\}^*$

(a) Comencemos seguidamente por definir el autómata minimal

$\mathcal{A} = \langle \{a, \bar{a}, b, \bar{b}\}, Q, q_-, q_+, \cdot \rangle$ que reconoce $FIFO_2$.

$$Q = \{ (w) \mid w \in \{a, b\}^* \}, \quad q_- = q_+ = (\varepsilon)$$

$$(w) \cdot u = (wu), \quad u \in \{a, b\}$$

$$(uw) \cdot \bar{u} = (w), \quad u = \{a, b\}$$

La figura II.6 esquematiza dicho autómata.

(b) De modo análogo al lenguaje de Dyck, tenemos

$$q_- \cdot n \cdot \{a, \bar{a}, b, \bar{b}\}^* = \{ (w) \mid w \in \{a, b\}^* \}$$

con lo que $a_{FIFO_2}(n) = 2^{n+1}$

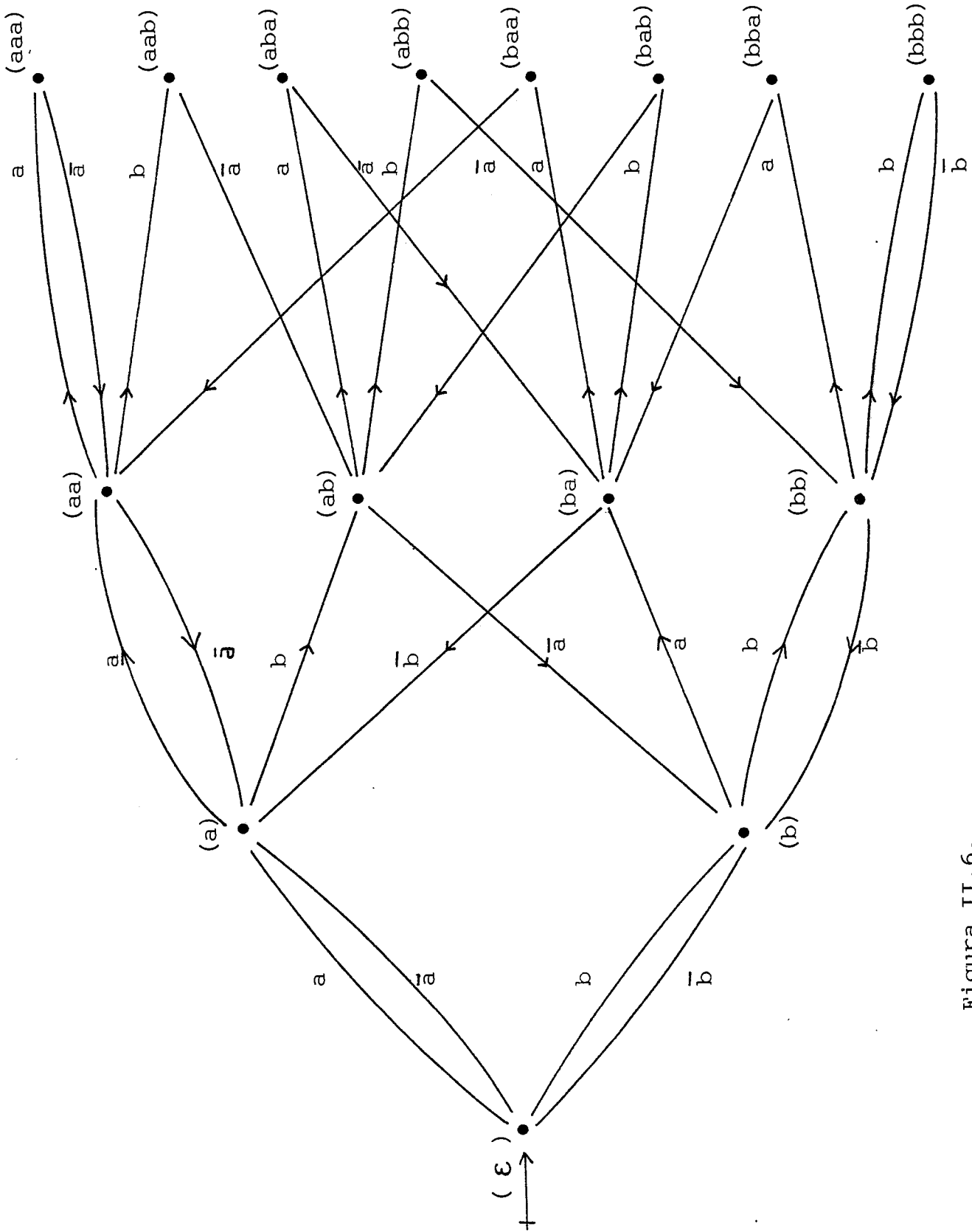


Figura II.6.

Ejemplo 2.7 Vamos a tratar seguidamente un lenguaje debido a Goldstine [4] definido por:

$$\text{Gold} = \{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_k} b \mid k \geq 1, \exists 1 \leq j \leq n \\ \text{con } i_j \neq j \}$$

Vamos a calcular su a -medida. A este fin, vamos a dar el autómata minimal completo que lo reconoce. Figura II.7. En dicho autómata, existe una sucesión infinita de estados que se utilizan para reconocer la palabra infinita.

$$\underline{m} = aba^2ba^3ba^4b \dots a^nba^{n+1}b \dots$$

Llamemos $(aba^2b \dots a^i b)$, el estado al que se llega tras la lectura de $aba^2b \dots a^i b$, si $i \neq j$, entonces $(aba^2b \dots a^i b) \neq (aba^2b \dots a^j b)$ si no, dado que $(aba^2b \dots a^i b) \cdot a^{j+1} b \in \text{Gold}$, $(aba^2b \dots a^j b) \cdot a^{j+1} b$ debe también ser aceptada. Llegamos, por tanto, a una contradicción.

Tenemos

$$a_{\text{Gold}}(n) = n + 3$$

Vamos a estudiar seguidamente algunas propiedades de cierre de la a -medida. Comenzamos por dar una caracterización alternativa. A este fin, recordemos la definición de residual. Sea $L \subset X^*$ y $w \in X^*$, definimos el residual de L con relación a w como

$$L \setminus w = \{ x \in X^* \mid wx \in L \}$$

tenemos entonces que:

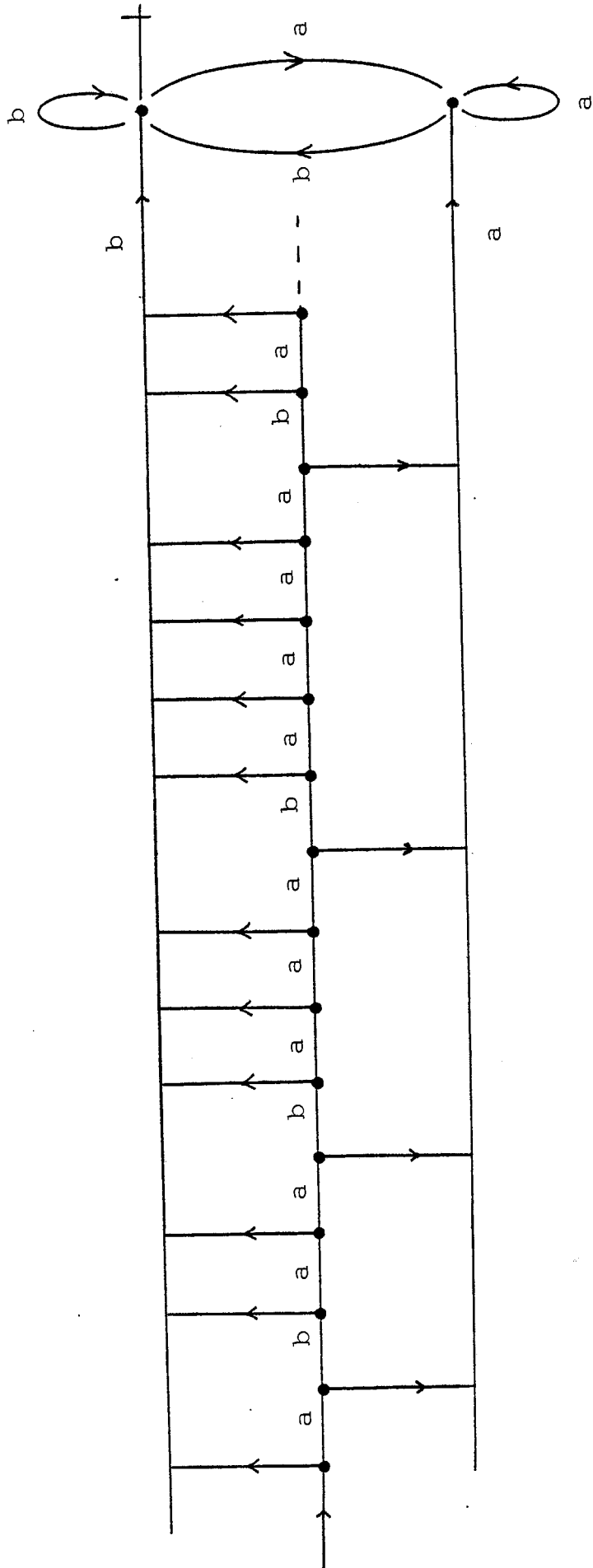


Figura II.7.

Lema [47] 2.2 Sea $L \subset X^*$, consideremos la relación de equivalencia

$$w \sim w' \text{ ssi } w, w' \in nX^* \text{ y } L \setminus w = L \setminus w'$$

si satisface $nX^*/\sim = \{ L \setminus w \mid w \in nX^* \}$ y

$$a_L(n) = \text{Card} \{ L \setminus w \mid w \in nX^* \}$$

Prueba Basta construir el autómata minimal completo que reconoce L a partir de los residuales; sea $\mathcal{A} = \langle X, Q, q_-, q_+, \dots \rangle$ con $Q = \{ L \setminus w \mid w \in X^* \}$, $q_+ = \mathcal{P}(L)$ y la función de transición está definida por

$$(L \setminus w) \cdot x = L \setminus wx$$

El comportamiento de la a -medida en la complementación, viene dado por [47] .

Lema 2.3 Sea $L \subset X^*$ se cumple

$$a_{X^* \setminus L}(n) = a_L(n)$$

Prueba Para obtener un autómata minimal completo para $X^* \setminus L$, basta intercambiar los estados finales y no finales en el autómata minimal completo para L .

Estudiemos seguidamente la unión y la intersección [47] .

Lema 2.4 La a -medida satisface

$$a_{L_1+L_2}(n) \leq a_{L_1}(n) \cdot a_{L_2}(n)$$

$$a_{L_1} a_{L_2}(n) \leq a_{L_1}(n) \cdot a_{L_2}(n)$$

Prueba

(a) Notemos que $(L_1 + L_2) \setminus w = L_1 \setminus w + L_2 \setminus w$.

Si $w \in nX^*$ hay como máximo $a_{L_1}(n)$ residuales $L_1 \setminus w$ distintos y $a_{L_2}(n)$ residuales $L_2 \setminus w$ distintos, con lo que hay $a_{L_1}(n) \cdot a_{L_2}(n)$ residuales $L_1 \setminus w + L_2 \setminus w$ distintos como máximo.

(b) Sabemos que $(L_1 \cap L_2) \setminus w = L_1 \setminus w \cap L_2 \setminus w$. El mismo razonamiento que (a) nos conduce a $a_{L_1 \cap L_2}(n) \leq a_{L_1}(n) \cdot a_{L_2}(n)$.

Vamos a tratar seguidamente el producto y la estrella marcados.

Lema [47] 2.5 Sean L_1, L_2 contenidos en X y $\$$ no contenido en X , tenemos

$$a_{L_1 \$ L_2}(n) \leq a_{L_1}(n) + a_{L_2}(n) + 1$$

$$a_{(L_1 \$)^*}(n) \leq n a_{L_1}(n) + 1$$

Prueba

(a) Comencemos por tratar la unión marcada. Vamos a calcular

$$\text{Card} \{ L_1 \$ L_2 \setminus w \mid w \in n(X + \$)^* \}$$

Hay tres casos. Supongamos que w no contiene $\$$, en este caso

$$\begin{aligned} L_1 \$ L_2 \setminus w &= \{ h \mid wh \in L_1 \$ L_2 \} = \{ h_1 \$ h_2 \mid wh_1 \$ h_2 \in L_1 \$ L_2 \} = \\ &= (L_1 \setminus w) \$ L_2. \end{aligned}$$

Supongamos que w contiene un solo símbolo $\$$, en este caso $w = w_1 \$ w_2$ y $L_1 \$ L_2 \setminus w_1 \$ w_2 = \begin{cases} L_2 \setminus w_2 & \text{si } w_1 \in L_1 \\ \emptyset & \text{sino} \end{cases}$

Si w contiene mas de un símbolo $\$$ entonces $L_1 \$ L_2 \setminus w = \emptyset$.

Tenemos entonces que $a_{L_1 \$ L_2}(n) \leq a_{L_1}(n) + a_{L_2}(n) + 1$

(b) Tratemos ahora la estrella marcada. En este caso hay que calcular:

$$\text{Card} \{ (L_1 \$)^* \setminus w \mid w \in n(X + \$)^* \}$$

Si $w \in n(X + \$)^*$, w contendrá como máximo n marcas $\$$. Tenemos

$$(L_1 \$)^* \setminus w_1 \$ w_2 \$ \dots \$ w_k = \begin{cases} (L_1 \setminus w_k) \$ (L_1 \$)^* & \text{con } w_i \in S, 1 \leq i \leq k-1 \\ \emptyset & \text{sino} \end{cases}$$

entonces da $a_{(L_1 \$)^*}(n) \leq n \cdot a_{L_1}(n) + 1$



Vamos a tratar finalmente el morfismo inverso:

Lema [47] 2.6 Sea $\varphi : X^* \longrightarrow Y^*$, sea $k = \max \{ |\varphi(x)| \mid x \in X \}$

se cumple que

$$a_{\varphi^{-1}(L)}(n) \leq k \cdot a_L(n)$$

Prueba Expresamos el comportamiento de los residuales en el morfismo inverso

$$L \setminus \varphi(w) = \{ h \mid \varphi(w)h \in L \}$$

consideremos

$$\varphi^{-1}(L \setminus \varphi(w)) = \{ \varphi^{-1}(h) \mid w \varphi^{-1}(h) \in \varphi^{-1}(L) \} = \varphi^{-1}(L) \setminus w$$

Si queremos tratar los w con $w \in nX^*$ basta partir con

$\varphi(w) \in knY^*$ con $k = \max\{|\varphi(x)| \mid x \in X\}$, tenemos entonces que:

$$\text{Card} \{ \varphi^{-1}(L) \setminus w \mid w \in nX^* \} \leq \text{Card} \{ L \setminus \varphi(w) \mid \varphi(w) \in knY^* \}$$

obtenemos que $a_{\varphi^{-1}(L)}(n) \leq k \cdot a_L(n)$

A partir de la a -medida, podemos definir las familias de lenguajes de a -medida polinomial y exponencial respectivamente, es decir

$$\text{Pol}_a = \{ L \mid \exists k \in \mathbb{N}_+ \text{ que satisface } a_L(n) = O(n^k) \}$$

$$\text{Exp}_a = \{ L \mid \exists \alpha \in \mathbb{R}_+ \text{ que satisface } a_L(n) = \Omega(2^{\alpha n}) \}$$

Si consideramos el lenguaje

$$L = \{ c^{2^{|w|}} w \# \tilde{w} \mid w \in \{0,1\}^* \}$$

Vemos que la familia Pol_a no está cerrada ni por paso al lenguaje espejo (Si $w \in L, \tilde{w} \in \tilde{L}$) ni por morfismo.

Dado que Pol_a está cerrada por morfismo inverso e intersección con los racionales, podemos decir

Lema 2.7 La familia Pol_a es un cilindro.

II.3 Índice racional.

El índice racional es una función de complejidad introducida en [8] por L.Boasson, B.Courcelle y M.Nivat, a fin de estudiar la no principalidad de los lenguajes algebraicos.

El índice racional mide la disparidad que presenta un lenguaje dado con la familia de los lenguajes racionales. Demos, seguidamente, su definición formal:

Definición [8] 3.1 Sea $L \subset X^*$, llamaremos $\text{Rat}_n X^*$ a la familia de lenguajes racionales aceptados por un autómata determinista, que tiene como máximo n estados. Llamaremos índice racional de L a la función:

$$\rho_L(n) = \left\{ \max \left\{ \min \{ |w| \mid w \in L \cap K \} \mid L \cap K \neq \emptyset \text{ y } K \in \text{Rat}_n X^* \right\} \right\}$$

Introducimos seguidamente dos notaciones que emplearemos a lo largo de este apartado.

$$\delta_{L \cap K} = \min \{ |w| \mid w \in L \cap K \}$$

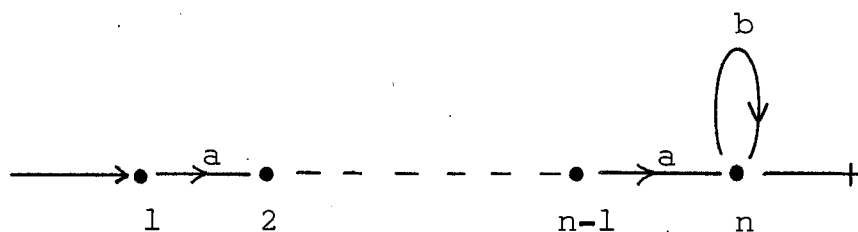
$$\min L \cap K = \{ w \mid w \in L \cap K \text{ y } |w| = \delta_{L \cap K} \}$$

Veamos en un ejemplo simple que el índice racional puede tomar como cota inferior cualquier valor.

Lema [8] 3.1 Sea $f(x)$ una función a valores enteros mayor que la identidad, el lenguaje $L = \{ a^n b^{f(n+1)} \mid n \geq 0 \}$ satisface

$$\rho_L(n) = \Omega(f(n))$$

Prueba A fin de dar un límite inferior, consideremos la sucesión de autómatas K_n siguiente:



tenemos que $\min L \cap K_n = a^{n-1}b^{f(n)}$ con lo que

$$\delta_{L \cap K_n} = n - 1 + f(n) = \Theta(f(n)) \text{ con lo que}$$

$$\rho_L(n) = \Omega(f(n))$$

Damos seguidamente una sucesión de ejemplos en los que se muestra la técnica empleada en el cálculo del índice racional.

Ejemplo 3.1 Sea $L = \{0,1\}^* 010$ su índice racional satisface

$$\rho_L(n) = n + 2$$

Vamos a proceder en dos etapas distintas

(a) Busquemos una cota inferior. Se trata, en este caso, de encontrar un autómata con n estados que nos dé una cota lo mayor posible.

Tomemos, en este caso, el racional $R \in \text{Rat}_n \{0,1\}^*$ reconocido por el autómata siguiente, figura II.8.

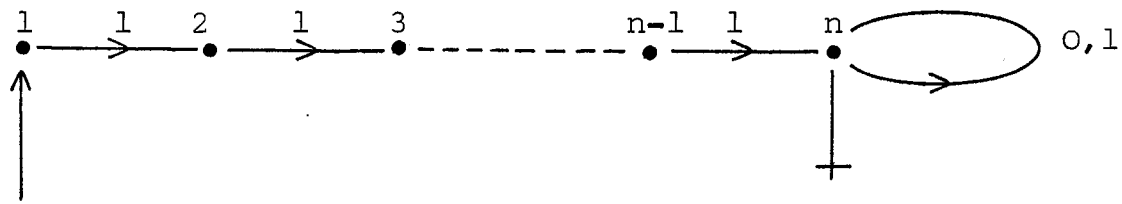


Figura II.8.

Vemos que $L \cap R = 1^{n-1}010$ con lo que $\min(L \cap R) = 1^{n-1}010$ y

$$\delta_{L \cap R} = n + 2. \text{ Tendremos, por lo tanto, que } \rho_L(n) \geq n + 2$$

(b) Busquemos una cota superior. Se trata de demostrar que la longitud de las palabras pertenecientes a la intersección con cualquier autómata a n estados, satisfacen una cierta cota.

En este caso vamos a demostrar que $\delta_{L \cap R} \leq n + 2$ cuando $R \in \text{Rat}_n \{0,1\}^*$

Razonemos por reducción al absurdo; supongamos que exista un $R \in \text{Rat}_n \{0,1\}^*$ con $\delta_{L \cap R} > n + 2$. Sea $w \in \min(L \cap R)$ dicha palabra se factoriza de la forma $w = m 010$ con $m \in \{0,1\}^*$. Consideremos un cálculo en R para w con longitud mínima, dicho cálculo se factoriza en:

$$q_- \xrightarrow{m} q \xrightarrow{010} q_+$$

con $q_- \in Q_-$ y $q_+ \in Q_+$. Si $w > n + 2$ tenemos que $m > n - 1$, con lo que en el camino $q_- \xrightarrow{m} q$ existe al menos un estado que se repite, sea q' dicho estado, tenemos

$$q_- \xrightarrow{u} q' \xrightarrow{\alpha} q' \xrightarrow{v} q \xrightarrow{010} q_+$$

con $m = u v$, $|\alpha| > 0$. Tenemos, entonces, que $uv 010$ pertenece a

$L \cap R$ y tiene una longitud estrictamente menor; hemos llegado, por lo tanto, a una contradicción. Esquematizamos dicho proceso de reducción en la figura II.9.

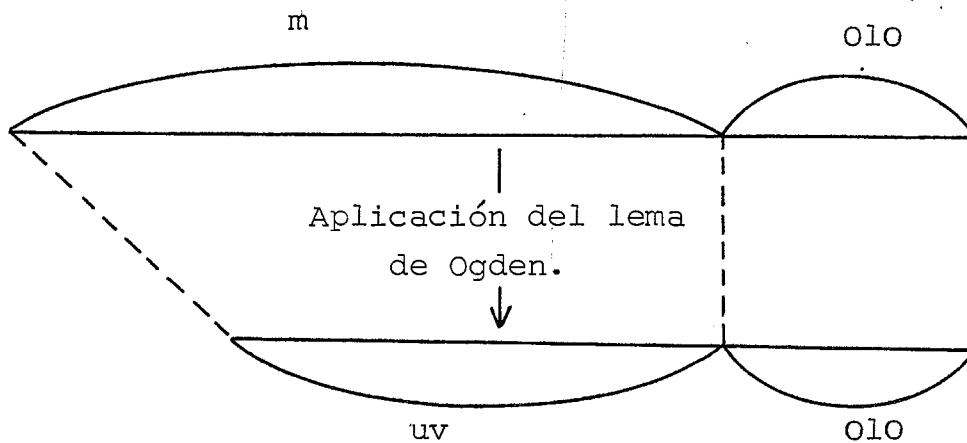


Figura II.9.

Ejemplo 3.2 Sea $L = \{0,1\}^* 010 \{0,1\}^*$ su índice racional satisface:

$$\rho_S(n) = 2(n - 1)$$

(a) Busquemos una cota inferior. Sea R el racional reconocido por el autómata dado en la figura II.10

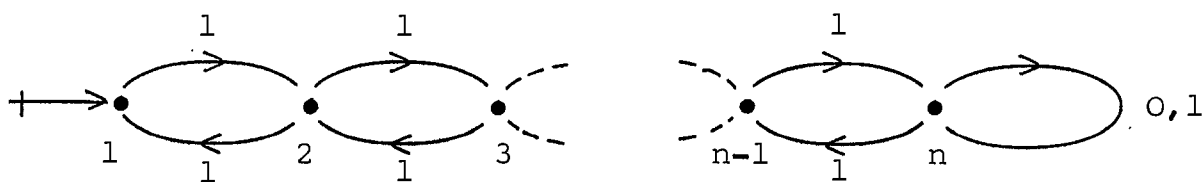


Figura II.10.

Vemos que $L \cap R = 1^{n-1}010 1^{n-1}$, con lo que $\delta_{L \cap R} = 2n + 1$ y en consecuencia $\rho_L(n) \geq 2n + 1$

(b) Busquemos una cota superior. Sea $R \in \text{Rat}_n \{0,1\}^*$ en este caso

$\delta_{L \cap R} \leq 2n + 1$, sino podemos aplicar el lema de Ogden del modo esquematizado en la figura II.11.

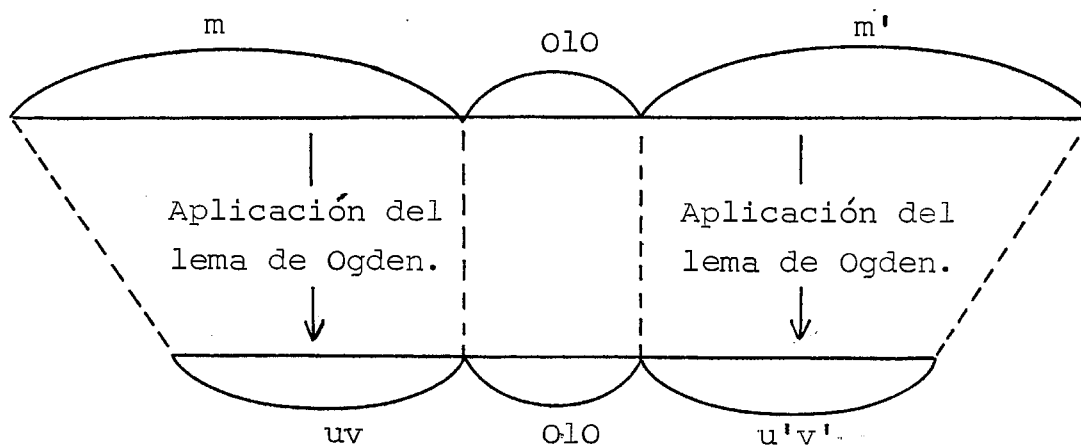


Figura II.11.

Vamos a tratar seguidamente un ejemplo no racional.

Ejemplo [23] 3.3 Sea $S_{\geq} = \{a^m b^n \mid m \geq n\}$ su índice racional satisface:

$$\rho_{S_{\geq}}(n) = 2(n - 1)$$

(a) Busquemos una cota inferior. Sea R el racional reconocido por el autómata dado en la figura II.12

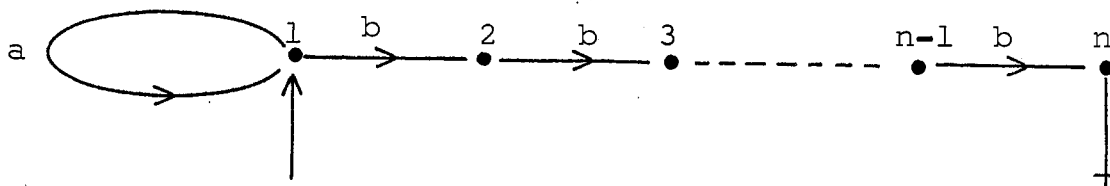


Figura II.12.

Tenemos $L \cap R = a^{n-1} b^{n-1}$ con lo que $\rho_{S_{\geq}}(n) \geq \delta_{L \cap R} = 2(n - 1)$

(b) Busquemos una cota superior. Vamos a demostrar que para cualquier $R \in \text{Rat}_n \{a, b\}^*$ se cumple $\delta_{S_{\geq} \cap R} \leq 2(n - 1)$.

Sea $w \in \min(L \cap R)$ con $|w| > 2(n - 1)$ vamos a llegar a una contradicción. La palabra w se factoriza en $w = a^x b^y$ con $x + y > 2(n - 1)$ como $x \geq y$, tenemos $x > n - 1$.

Consideremos un cálculo válido en R reconociendo w con longitud

mínima. Dicho cálculo se factoriza en:

$$q_- \xrightarrow{a^{n-1}} q \xrightarrow{a^{x_1} b^y} q_+$$

Dado que $x_1 + y > n - 1$ podemos aplicar el lema de Ogden a la segunda parte del camino, con lo que:

$$q \xrightarrow{u} q' \xrightarrow{\alpha} q' \xrightarrow{v} q_+$$

$u = a^{x_1} b^y$, con $\alpha \neq \epsilon$ la palabra $uv = a^{x_2} b^{y_2}$ es estrictamente más corta que $a^{x_1} b^y$, y además $a^{n-1} uv \in L \cap R$, hemos llegado, por lo tanto, a una contradicción. Dicho proceso de simplificación, se halla esquematizado en la figura II.13.

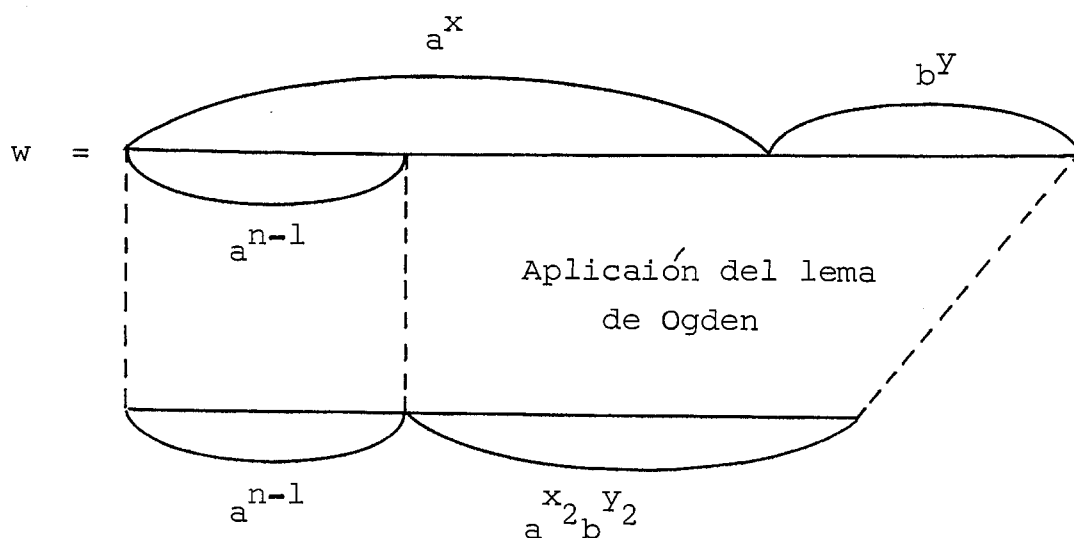


Figura II.13.

Veamos que la presencia de un marcador central, cambia ligeramente la situación.

Ejemplo 3.4 Sea $L = \{a^m \# b^n \mid m \geq n\}$ su índice racional satisface

$$\rho_L(n) = 3n - 2$$

(a) Busquemos una cota inferior. Sea R el racional dado en la figura II.14;

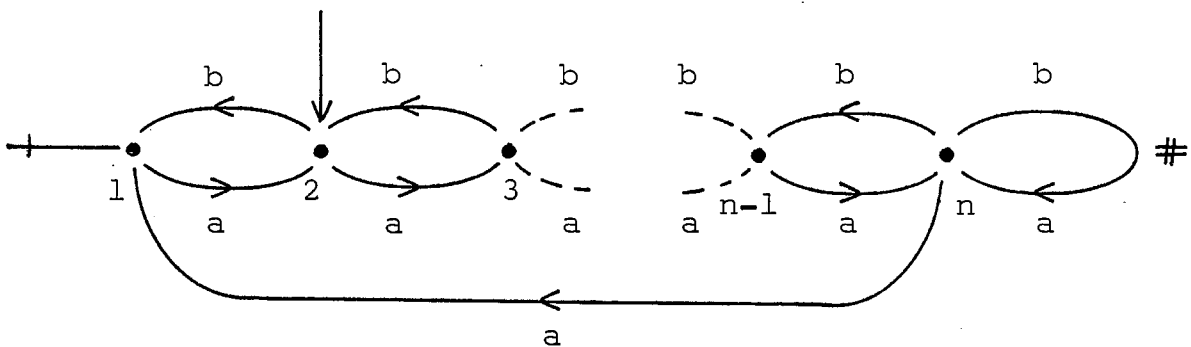


Figura II.14.

Vemos que $L \cap R = a^{2(n-1)} \# b^{n-1}$, con lo que $\rho_L(n) \geq \delta_{L \cap R} = 3n - 2$

(b) Busquemos una cota superior. Sea $R \in \text{Rat}_n \{a, b\}^*$ si $w = a^x \# b^y \in \min(L \cap R)$ con $|w| > 3n - 2$ realizemos la reducción esquematizada en la figura siguiente:

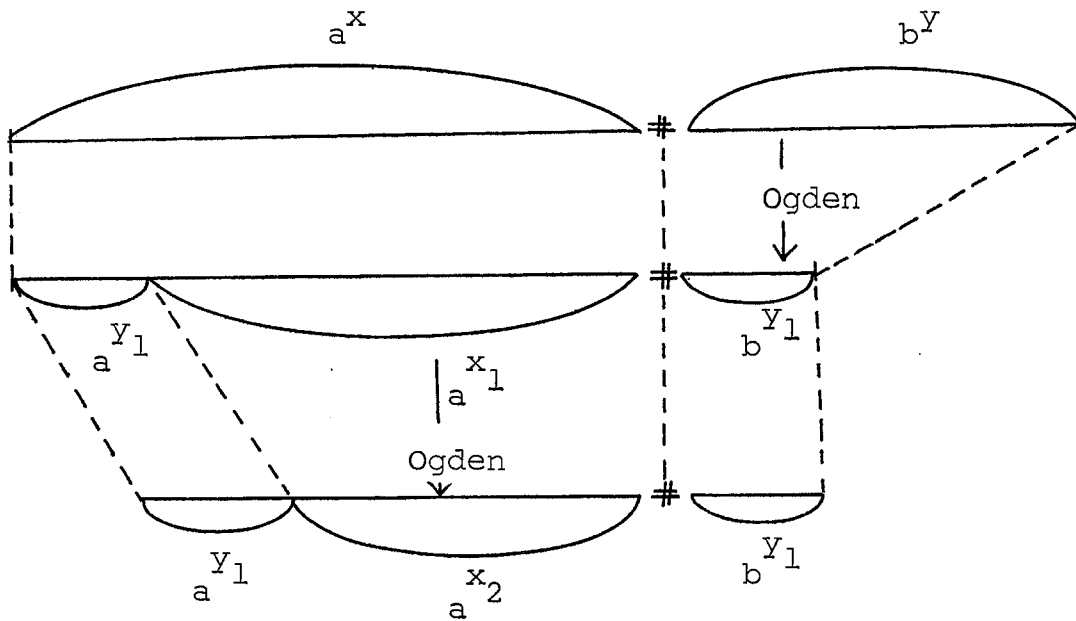


Figura II.15.

Ejemplo 3.5 [8] Vamos a tratar seguidamente el lenguaje $S_1 = \{ a^n b^n \mid n \geq 0 \}$, veremos que su índice racional verifica

$$\rho_{S_1}(n) = \Theta(n^2)$$

(a) Busquemos una cota inferior. Sea $K = (a^n)^* (b^{n+1})^+$ que pertenece a $\text{Rat}_{2n+1} \{a, b\}^*$ como vemos en la figura II.16.

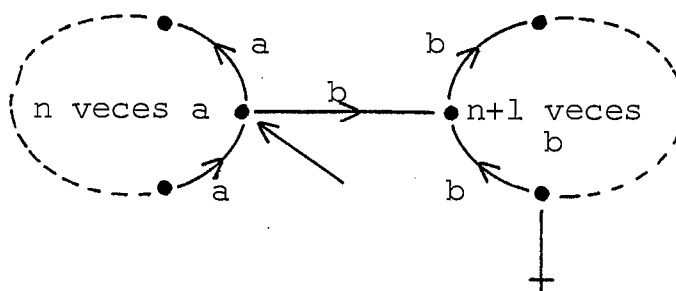


Figura II.16.

Dado que n y $n + 1$ son siempre primos entre sí, tenemos que:

$$S_1 \cap K = a^{n(n+1)} b^{n(n+1)} = \min(S_1 \cap K),$$

$$\delta_{S_1 \cap K} = \left| a^{n(n+1)} b^{n(n+1)} \right| \text{ con lo que } \rho_{S_1}(n) \geq \delta_{S_1 \cap K} = 2n(n+1)$$

(b) Busquemos una cota superior. Asociemos a toda palabra de S_1 una representación en el plano. Representaremos a por ↗ y b por ↘. A fin de encontrar una cota, vamos a especificar en qué estado se encuentra un autómata $K \in \text{Rat}_n \{a, b\}^*$ tras la lectura de cada letra. Esquematizamos dicho proceso en la figura II.17.

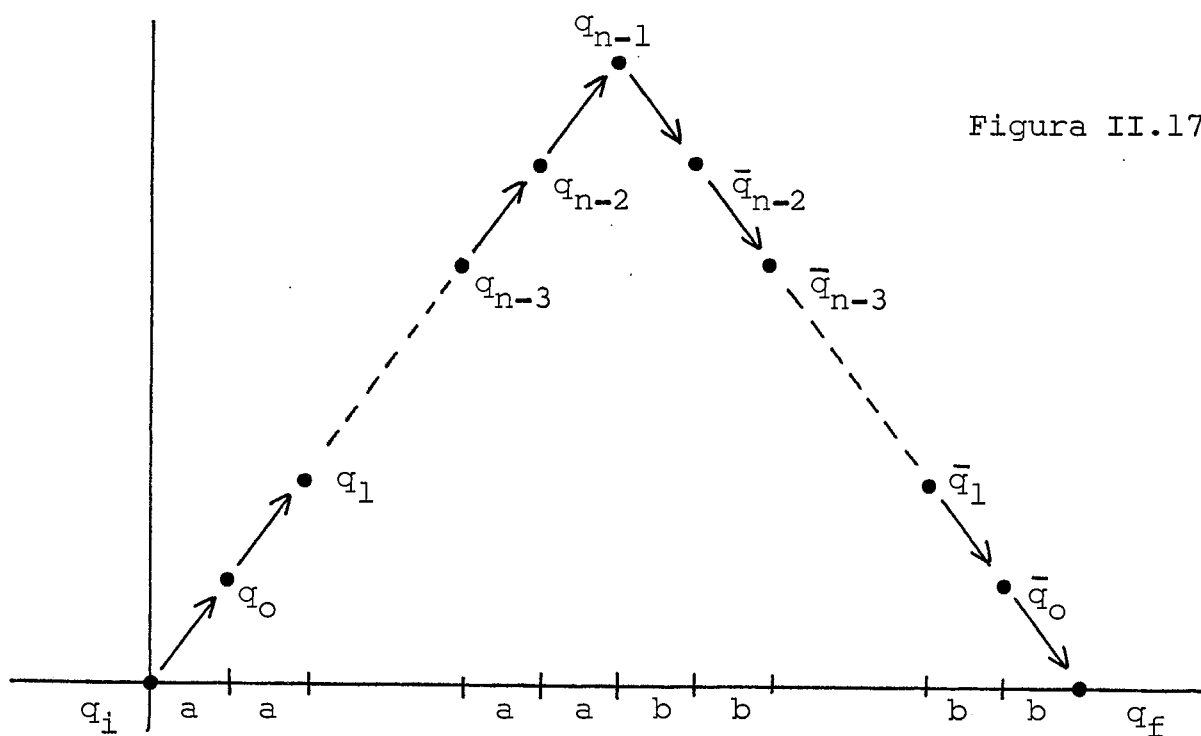


Figura II.17.

Si consideramos la sucesión: $(q_0, \bar{q}_0), (q_1, \bar{q}_1), \dots, (q_{n-3}, \bar{q}_{n-3}), (q_{n-2}, \bar{q}_{n-2})$, vemos que puede contener como máximo n^2 elementos distintos. Si dicha sucesión contiene más de n^2 elementos, utilizando el principio de Dirichlet, existen al menos dos elementos idénticos. Como consecuencia de esto, tenemos que $\delta_{S_1 \cap K} \leq 2(n^2 + 2)$. Si no, tenemos la longitud de la palabra de la manera esquematizada en la

figura II.18.

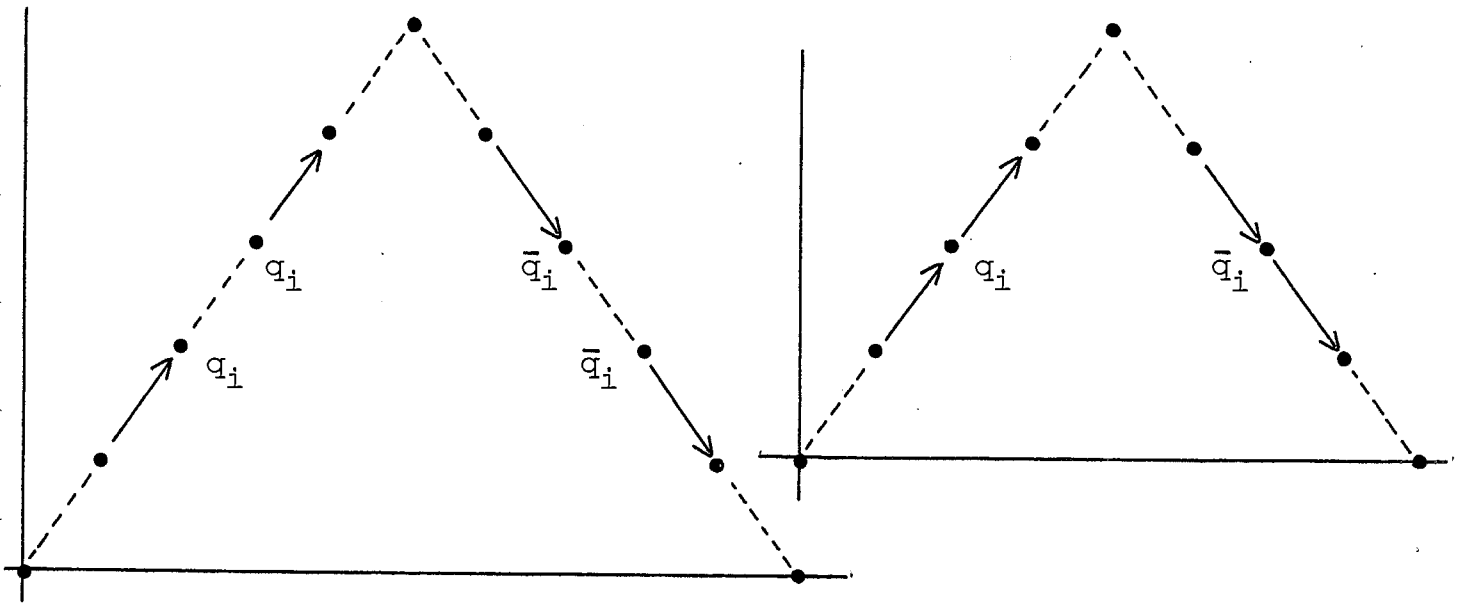


Figura II.18.

Vemos en este caso, que si bien en orden de magnitud el índice racional de S_1 es en n^2 , no hemos podido hallar su valor exacto.

Ejemplo 3.6 Con la ayuda del ejemplo precedente, vamos a tratar el caso de los palíndromes $PAL_2 = \{w\tilde{w} \mid w \in \{a,b\}^*\}$

$$\rho_{PAL_2}(n) = \Theta(n^2)$$

(a) Busquemos el límite inferior. En este caso basta modificar ligeramente el autómata dado precedentemente, en el que hemos desdoblado cada arco. La situación se halla esquematizada en la figura II.19.

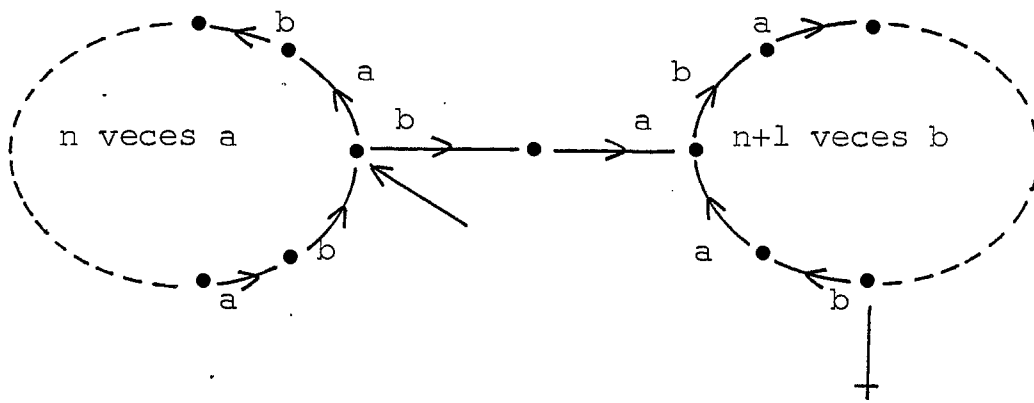


Figura II.19.

(b) Busquemos el límite superior. Basta considerar el mismo argumento que el dado en S_1 , figura II.20.

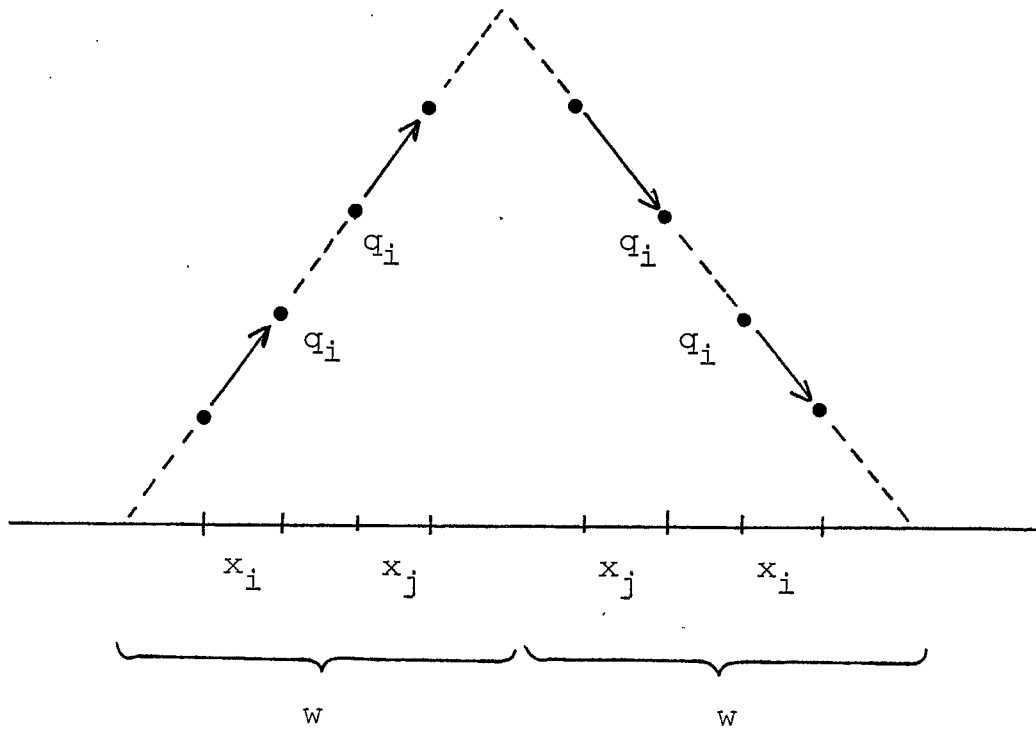


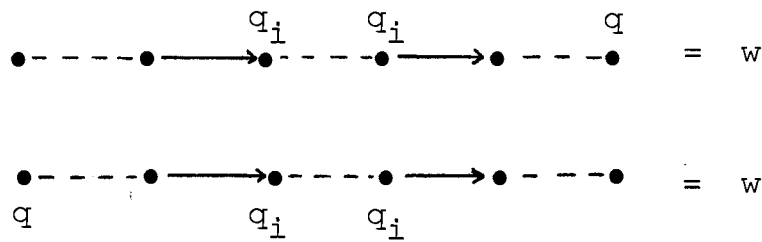
Figura II.20.

Vamos a dar una cota superior al índice racional de COPY.

Ejemplo 3.7 Sea $COPY_2 = \{ww \mid w \in \{a,b\}^*\}$, veamos que

$$\rho_{COPY_2}(n) = O(n^2)$$

Basta emplear la misma técnica de enumeración de estados, empleando el esquema:



Ejemplo [8] 3.8 Vamos a tratar seguidamente el lenguaje de Dyck a una letra D_1^* , engendrado por $S \rightarrow aS\bar{a}S + \epsilon$.

Sea $w \in \{a,b\}^*$, notamos $\beta(w) = |w|_{\bar{a}} - |w|_a$. En función de β podemos definir D_1^* como el conjunto de palabras m de $\{a,b\}^*$ que satisfacen

- $\beta(m) = 0$
- $\beta(m') \geq 0$ para todo factor izquierdo m' de m .

a toda palabra de Dyck podemos, entonces, asociar una representación gráfica, como vemos en la figura siguiente II.21.

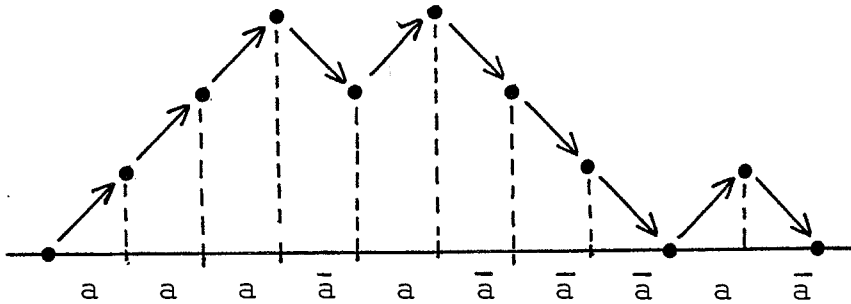


Figura II.21.

La longitud de dicha palabra está acotada por el "volumen" de su representación gráfica.

Sea $K \in \text{Rat}_n \{ a, \bar{a} \}^*$, tomemos $m \in \min D_1^* \cap K$, de modo aproximado podemos decir que la "anchura" de la representación gráfica de m está acotada por n , figura II.22.

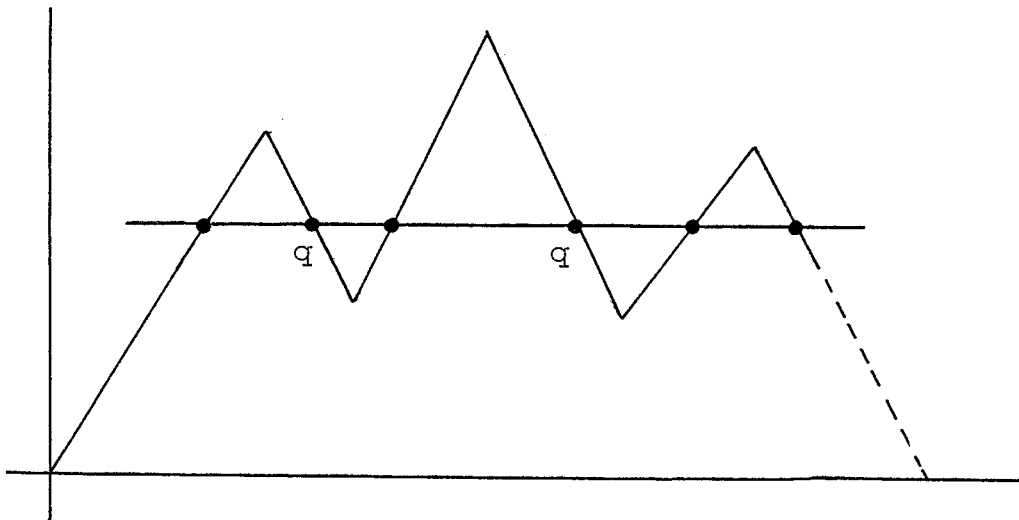


Figura II.22.

En efecto, si una horizontal corta dicha representación en más de n

puntos, existe un estado q repetido y podemos acortar m .

Por otra parte, considerando la factorización esquematizada en la figura II.23. y aplicando el mismo argumento que en S_1 , tenemos que la "altura" está acotada por n^2 .

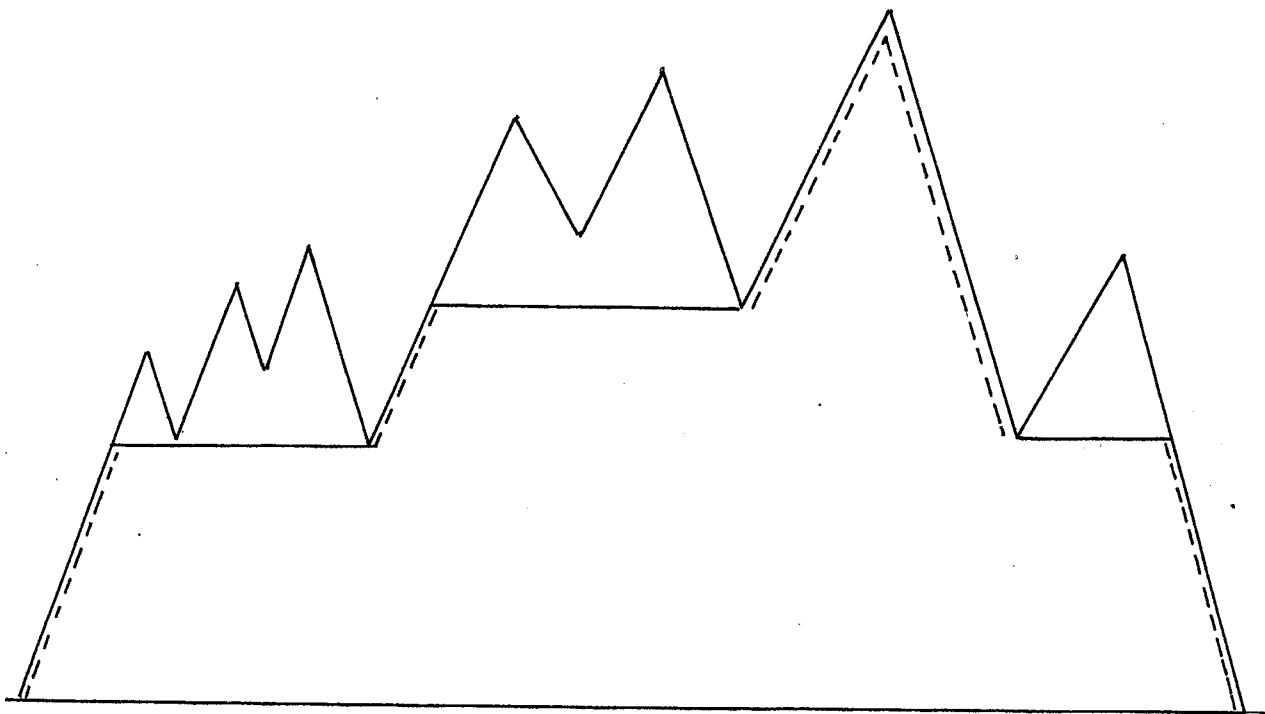


Figura II.23.

con lo que llegamos que $|m| = \Theta(n^3)$, con lo que $\rho_{D_1^*}(n) = O(n^3)$.

Tomando el autómata dado para S_1 obtenemos $\rho_{D_1^*}(n) = \Omega(n^2)$.

Hemos llegado por lo tanto a

$$\Omega(n^2) = \rho_{D_1^*}(n) = O(n^3)$$

Proponemos que:

Conjetura 3.1 $\rho_{D_1^*}(n) = \Theta(n^2)$

Ejemplo [8] 3.9 Vamos a tratar seguidamente el lenguaje E, engendrado por la gramática G, dada por $S \rightarrow aSc + d$.

$$\Omega(2^n) = \rho_{E^*}(n) = O(2^{kn^2})$$

(a) Busquemos una cota inferior. Para ellos consideremos el autómata K a n estados, dado en la figura II.24.

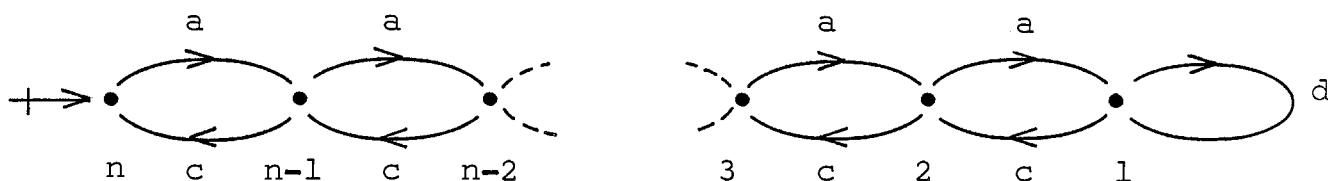


Figura II.24.

Razonando por inducción sobre el número de estados podemos ver que

$$\delta_{E^* \cap K} = \Omega(2^n) [23]$$

(b) Busquemos una cota superior. Vamos a utilizar el lema de Ogden de los lenguajes algebraicos. Sea $K \in \text{Rat}_n \{a, c, d\}^*$, si construimos la gramática que genera $L(G) \cap K = E^* \cap K$ tendrá como variables los tripletes $q^s q'$ siendo q y q' dos estados del autómata que reconoce K. El lema de Ogden nos permite concluir que existe una constante k que satisface

$$\delta_{E^* \cap K} \leq 2^{kn^2}$$

Conjetura [8] 3.2 $\rho_{E'}(n) = \Theta(2^{pn})$

Vamos a dar dos variaciones de lenguajes exponenciales.

Ejemplo 3.10 Trataremos seguidamente el lenguaje E generado por
 $S \rightarrow aSbSc + d$

$$\rho_E(n) = \Omega(2^n)$$

basta dar una ligera modificación del autómata indicado en la figura II.24., tenemos entonces: figura II.25.:

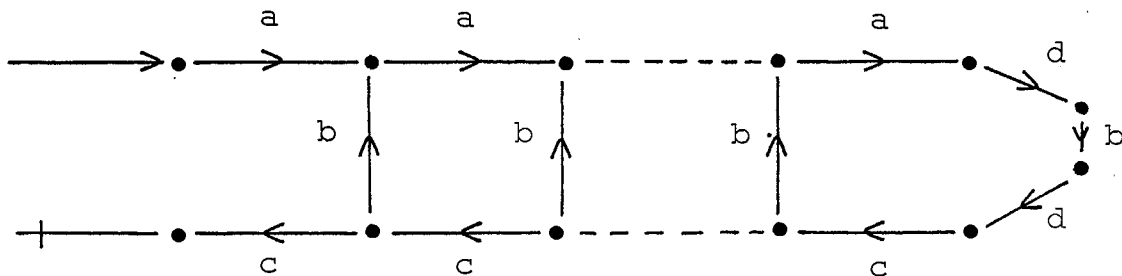


Figura II.25.

A partir de este ejemplo podemos tratar el lenguaje de Dyck de dos letras, en efecto

Ejemplo 3.11 Tratemos el lenguaje de paréntesis $D_2^{!*}$ engendrado por $S \rightarrow a\bar{a}S + b\bar{b}S + c\bar{c}S + \epsilon$

$$\rho_{D_2^{!*}}(n) = \Omega(2^n)$$

Notemos que a partir de la gramática dada para $D_2^{!*}$ podemos obtener una versión codificada de E, haciendo:

$$S \rightarrow a\bar{a}S \rightarrow ab\bar{b}\bar{a}S \rightarrow ab\bar{b}c\bar{c}S\bar{a}S \xrightarrow{*} ab\bar{b}c\bar{c}\bar{a}$$

en que hemos codificado a por ab, b por $\bar{b}c$, c por $\bar{c}\bar{a}$, obtenemos, entonces, el autómata esquematizado en la figura II.26.

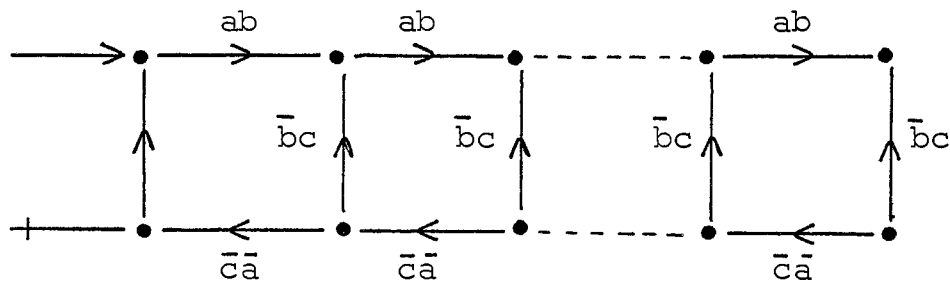


Figura II.26.

Vamos ahora a definir las familias de lenguajes de índice inicial

polinomial y exponencial, respectivamente, por:

$$\underline{\text{Pol}} \rho = \{ L \mid \exists k > 0 \text{ que satisface } \rho_L(n) = O(n^k) \}$$

$$\overline{\text{Exp}} \rho = \{ L \mid \exists k > 0 \text{ que satisface } \rho_L(n) = \Omega(2^{kn}) \}$$

Reunimos los ejemplos precedentes en el siguiente lema:

Lema 3.2

. Pertenecen a $\underline{\text{Pol}} \rho$ los lenguajes siguientes:

$$S_{\geq} = \{ a^m b^n \mid m \geq n \} ; S_1 = \{ a^n b^n \mid n \geq 0 \} ;$$

$$D_1^* = \langle S \rightarrow aS\bar{a}S + \varepsilon \rangle ; \text{PAL}_2 = \{ w\bar{w} \mid w \in \{a,b\}^* \}$$

$$\text{COPY}_2 = \{ ww \mid w \in \{a,b\}^* \}$$

. Pertenecen a $\overline{\text{Exp}} \rho$ los lenguajes siguientes:

$$E' = \langle S \rightarrow aS\bar{S}c + d \rangle ; E = \langle S \rightarrow aSbSc + d \rangle$$

$$D_p^* = \langle S \rightarrow \sum_{n=1}^p a_n S \bar{a}_n S + \varepsilon \rangle , p > 1$$

A fin de estudiar las familias $\underline{\text{Pol}} \rho$ y $\overline{\text{Exp}} \rho$ en términos de familias de lenguajes, damos el comportamiento del índice en relación a las operaciones racionales y de full-A.F.L.

Teorema [8] 3.1 En relación a las operaciones racionales, el índice racional satisface:

$$\cdot \rho_{L_1+L_2}(n) \leq \max \{ \rho_{L_1}(n) + \rho_{L_2}(n) \}$$

$$\cdot \rho_{L_1 \cdot L_2}(n) \leq \rho_{L_1}(n) \cdot \rho_{L_2}(n)$$

$$\cdot \rho_{L_1^*}(n) \leq n \rho_{L_1}(n)$$

Estudiar el comportamiento de ρ en relación a las operaciones de morfismo, morfismo inverso e intersección con los lenguajes racionales, es equivalente a estudiar su comportamiento en relación a las transducciones racionales; tenemos:

Teorema [8] 3.2 Sea L un lenguaje y τ una transducción racional existe un entero p que satisface

$$\rho_{\tau(L)}(n) \leq p \cdot n \cdot (\rho_L(pn + 1)) + \rho_L(n)$$

Vamos en el último lugar a estudiar en relación a la operación de substitución.

Teorema [23] 3.3 Sea $\sigma : X^* \rightarrow \mathcal{P}(Y^*)$ una substitución, $\sigma(x) = M_x$, existe un entero p que satisface

$$\rho_{\sigma(L)}(n) \leq \rho_L(pn) (1 + \max \rho_{M_x}(pn))$$

Vamos seguidamente a caracterizar Pol ρ en términos de familias de lenguajes, tenemos

Teorema [23] 3.4 La familia de lenguajes de índice racional polinomial es un full-A.F.L., no principal, cerrado por substitución.

Prueba Que Pol ρ sea un full-A.F.L. cerrado por substitución se sigue de la definición de Pol ρ y del comportamiento del índice racional en las distintas operaciones.

Para ver la no principalidad, basta considerar la sucesión de lenguajes:

$$L_0 = \{ a_0^n b_0^n \mid n \geq 0 \}$$

$$L_{p+1} = \{ a_{p+1}^n L_p a_{p+1}^n \mid n \geq 0 \}, \quad p \geq 0$$

que satisfacen $\rho_{L_{p+1}}(n) = \Theta(n \rho_{L_p}(n))$ con lo que $\rho_{L_p}(n) = \Theta(n^{p+2})$

■

Vamos a estudiar el comportamiento de Pol ρ y $\overline{\text{Exp}} \rho$ en relación a la complementación; para ello, estudiamos el índice racional de los complementarios de los lenguajes de Dyck. Comencemos por el lenguaje de Dyck a una sola letra. Vamos a demostrar ante todo, un lema de carácter combinatorico.

Lema 3.3 Sean f, m, g, n, h , palabras de $\{a, \bar{a}\}^*$ que cumplan.

$$fgh \in D_1^{!*}, \quad fmg h \in D_1^{!*}, \quad fgnh \in D_1^{!*}$$

se cumple, entonces, que $fmg nh \in D_1^{!*}$

Prueba Dado que $D_1^{!*}$ es la clase de \mathcal{E} en la congruencia

$\Theta = \langle a\bar{a} \equiv \varepsilon \rangle$ vamos a tomar de cada palabra su irreductible

$$\alpha = \Theta(f) = \bar{a}^p a^p$$

$$\beta = \Theta(g) = \bar{a}^r a^r$$

$$u = \Theta(m) = \bar{a}^q a^q$$

$$v = \Theta(n) = \bar{a}^s a^s$$

$$\gamma = \Theta(h) = a^{t'} a^t$$

Dado que $fg h \in D_1^*$ tenemos que $\alpha \beta \gamma \in D_1^*$ con lo que

$$\alpha \beta \gamma = \bar{a}^p a^p \bar{a}^{r'} a^{r'} \bar{a}^{t'} a^t$$

con lo que $p' = 0$, $t = 0$, $p \geq r'$ y $p - r' + r = t$

consideremos ahora la palabra $fmgh \in D_1^*$ se cumple que

$$\alpha u \beta \gamma \in D_1^* \text{ es decir}$$

$$u = a^p a^{q'} a^{q'} a^{r'} a^{r'} a^{t'}$$

con lo que $p \geq q'$ y $p - q' + q \geq r'$ y $p - q' + q - r' + r = t$

dado que $p - r' + r = t$, obtenemos $q = q'$

Si consideramos la palabra $fgnh$ tenemos

$$\alpha \beta v \gamma = a^p \bar{a}^{r'} a^{r'} \bar{a}^{s'} a^{s'} \bar{a}^{t'}$$

con lo que $p - r' + r \geq s'$ y $s' = s$ con lo que

$$\alpha = a^p$$

$$u = \bar{a}^q a^q \text{ con } p \geq q$$

$$\beta = \bar{a}^{r'} a^{r'} \text{ con } p \geq r'$$

$$v = \bar{a}^s a^s \text{ con } p - r' + r \geq s$$

$$\gamma = \bar{a}^{t'} \text{ con } p - r' + r = t$$

tenemos entonces que $\alpha u \beta v \gamma$ pertenece a D_1^* , dado que $\alpha u \beta v \gamma =$

$$= a^p \bar{a}^q a^q \bar{a}^{r'} a^{r'} \bar{a}^s a^s \bar{a}^{t'} \text{ dado que } p \geq q, a^p \bar{a}^q a^q \equiv a^p \text{ con lo que}$$

$$\alpha u \beta v \gamma \equiv \alpha \beta v \gamma \equiv \epsilon .$$

Dado que la palabra $fmgnh$ se obtiene a partir de $\alpha u \beta v \gamma$ añadiendo palabras de D_1^* entre dos letras consecutivas, llegamos al resultado.

Vamos a calcular seguidamente el índice racional de complementario de $D_1^!^*$, tenemos:

Lema 3.4 Sea $L = \{a, \bar{a}\} \setminus D_1^!^*$ se cumple

$$\rho_L(n) = \Theta(n)$$

Prueba

(a) Busquemos el límite inferior. Tratemos el caso impar. Damos la construcción de la sucesión de autómatas K_{2n+1} .

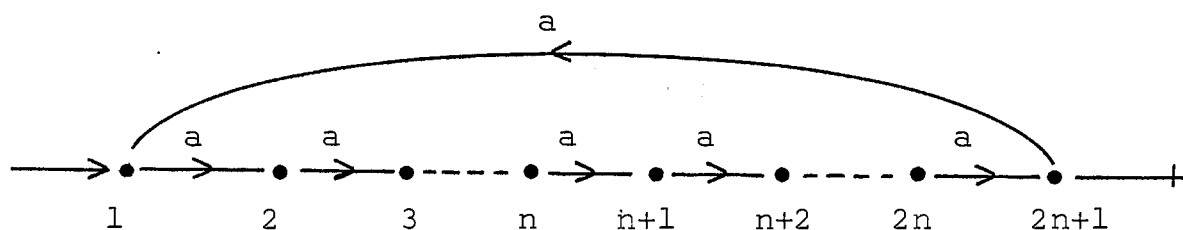
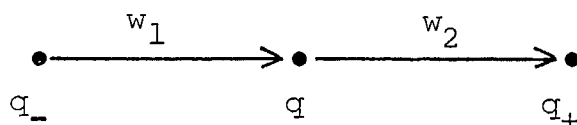


Figura II.27.

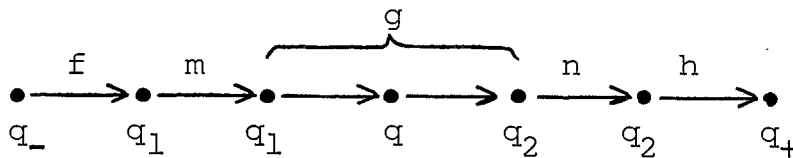
Tenemos que $\min(L \cap K_{2n+1}) = a^n \bar{a}^n a a^n \bar{a}^n$ con lo que

$$\rho_L(2n+1) \geq \delta_{L \cap K_{2n+1}} = 2n+1$$

(b) Tratemos el límite superior. Sea $K \in \text{Rat}_n \{a, \bar{a}\}^*$ que realiza el índice racional, tomemos $w \in \min(L \cap K)$. Supongamos que $|w| \geq 2n+2$, existe, entonces, una factorización



con $w_1 w_2 = w$ y $|w_1| > n$, $|w_2| > n$ existen, entonces, en los caminos q_- a q_+ y q_+ a q_- dos bucles, con lo que w se factoriza:



Dado que w es de longitud mínima, tenemos que tanto fgh como $fmgh$ o $fgnh$ son palabras de Dyck (no pertenecen a L), pero, entonces, el lema precedente nos dice que $fmgnh$ también es una palabra de Dyck; hemos llegado, por lo tanto, a una contradicción. ■

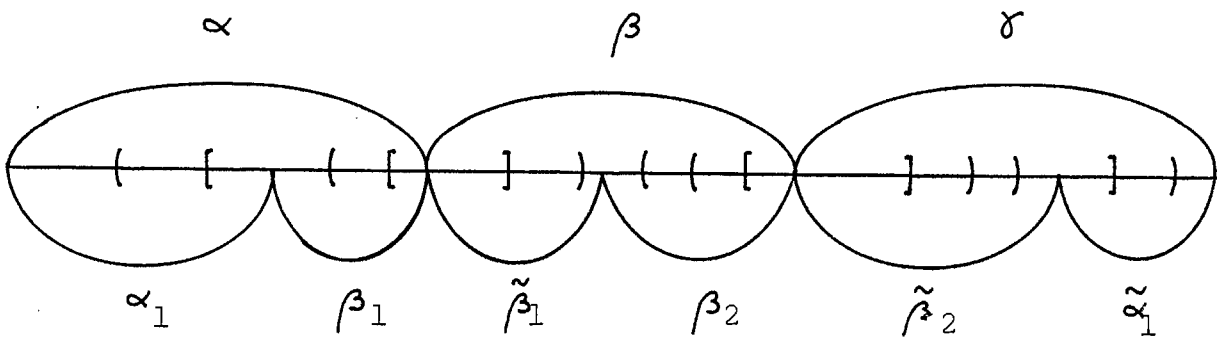
Vamos ahora a generalizar el lema dado precedentemente a $D_2'^*$.

Lema 3.5 Sean f, m, g, n, h , palabras de $\{a, \bar{a}, b, \bar{b}\}^*$ que satisfacen:

$$fgh \in D_2'^* \quad , \quad fmgh \in D_2'^* \quad , \quad fgnh \in D_2'^*$$

entonces se cumple que $fmgnh \in D_2'^*$

Prueba Dado que $D_2'^*$ es la clase de ε en la congruencia $\Theta = \langle aa \equiv bb \equiv \varepsilon \rangle$. Sean $\alpha = \Theta(f)$, $u = \Theta(m)$, $\beta = \Theta(g)$, $v = \Theta(n)$, $\gamma = \Theta(h)$. Demos un ejemplo de cómo pueden ser tomando a igual al paréntesis (y b al paréntesis [.



podemos generalizar una factorización de este estilo al caso general

$$\alpha = \alpha_1 \beta_1 \quad \text{con} \quad \alpha \in \{a, b\}^*$$

$$\beta = \tilde{\beta}_1 \beta_2 \quad \text{con} \quad \tilde{\beta}_1 \in \{\bar{a}, \bar{b}\}^*, \quad \beta_2 \in \{a, b\}^*$$

$$\gamma = \tilde{\beta}_2 \tilde{\alpha}_1 \quad \text{con} \quad \tilde{\beta}_2, \tilde{\alpha}_1 \in \{\bar{a}, \bar{b}\}^*$$

$$\text{además} \quad \beta_1 \tilde{\beta}_1 \equiv \varepsilon, \quad \beta_2 \tilde{\beta}_2 \equiv \varepsilon \quad \text{y} \quad \alpha_1 \tilde{\alpha}_1 \equiv \varepsilon$$

Vamos ahora a estudiar el caso $\alpha \vee \beta \vee \gamma \in D_2^*$

tenemos

$$\alpha = \alpha_1 \beta_1$$

$$\beta = \tilde{\beta}_1 \beta_2$$

$$\gamma = \tilde{\beta}_2 \tilde{\alpha}_1$$

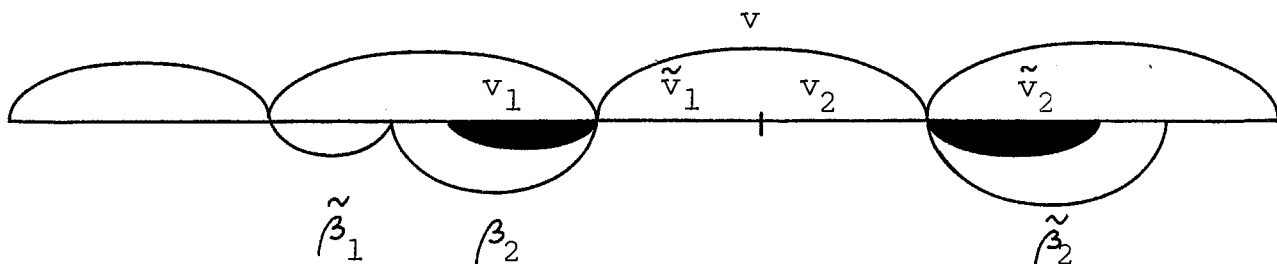
$$\delta = \tilde{\beta}_2 \tilde{\alpha}_1$$

Dado que $|\alpha_1 \beta_1| - |\tilde{\beta}_1| + |\beta_2| - |\tilde{\beta}_2| + |\tilde{\alpha}_1| =$

$= 0$; podemos concluir que $|\tilde{\beta}_1| = |\tilde{\alpha}_1|$. Queremos mostrar que

$v_2 = v_1$ en que v_1 es la palabra que se obtiene a partir de v_1 haciendo la palabra espejo y cambiando los paréntesis cerrados por abiertos.

Tenemos la factorización:



por inspección vemos que $v_1 = v_2$ con lo que podemos escribir

$$v = \tilde{v}_1 v_1$$

Por simetría, considerando la palabra $\alpha u \beta \gamma$, podemos concluir

$$u = \tilde{u}_1 u_1$$

Dado que $\Theta(\alpha u) = \alpha$ y $\Theta(\alpha u \beta v) = \Theta(\alpha \beta)$ podemos concluir que $\alpha u \beta v \gamma \in D_2'^*$ con lo que $\text{fmgnh } D_2'^*$

■

Utilizando los mismos argumentos que en $D_2'^*$ y utilizando el lema precedente, no presenta ninguna dificultad mostrar

Lema 3.6 Sea $L = \{a, \bar{a}, b, \bar{b}\}^* \setminus D_2'^*$ se cumple que

$$\rho_L(n) = \Theta(n)$$

Dicho lema admite como corolario:

Corolario 3.1 Las familias $\text{Pol } \rho$ y $\overline{\text{Exp}} \rho$ no están cerradas por complementación.

Prueba Basta tomar el lenguaje de Dyck a dos letras y su complementario.

Estudiamos la relación que existe entre las familias Pol_ρ y $\overline{\text{Exp}}_\rho$ y las distintas subfamilias de lenguajes algebraicos; tenemos:

Lema 3.7 Todos los lenguajes de la familia de Greibach tienen índice racional polinomial, formalmente $\text{Gre} \subset \text{Pol}_\rho$.

Prueba Dado que la familia Gre se construye haciendo la clausura por substitución de las familias de los lenguajes lineales, Lin , y de los lenguajes a contador, Fcl , y dado que S_2 y D_1^* son de índice racional polinomial, obtenemos el resultado.

Vamos a demostrar que la inclusión de Gre en Pol es estricta, para ello comenzamos por introducir el operador $\#$.

Definición [24] 3.2 Sea $L \subset X^*$ y $\#$ un marcador que no está contenido en X , definimos $\#(L) = L\# + K$ en que

$$K = \{ h\#^r \mid |h| \geq r \geq 0 \}$$

Notemos que si L es algebraico $\#(L)$ también. Damos seguidamente

un lema que muestra que la operación $\#$ degenera cualquier par iterante de L ; como consecuencia $\#(L)$ jamás será generador de los lenguajes algebraicos.

Lema [24] 3.8 Sea π un par iterante de $\#(L)$ se verifica:

- . Existe k_π que satisface $\{ (n, n') \mid n \geq n' \geq k \} \subset \text{Exp } \pi$
- . El lenguaje $\#(L)$ no tiene ningún par iterante estricto.

podemos entonces anunciar

Teorema [24] 3.5 Para todo $L \subset X^*$, $\#(L) \in \underline{\text{Nge}}$

Vemos también que $\#(L)$ puede "generar" ambigüedad, es decir

Lema [23] 3.9 El lenguaje $\#(S_{\underline{}})$ es inherentemente ambiguo.

Veamos seguidamente cuales son las relaciones entre la operación $\#$ con el centro y el índice racional.

Lema [24] 3.10

- . Para todo $L \subset X^*$ tenemos $C(\#(L)) = X^* + L\#^*$
- . Se cumplen las relaciones:
 - . $\rho_{(L)}(n) \leq 2(n - 1)$

$$\rho_L(n-1) + 1 \leq \rho_{C(\#(L))}(n) \leq \rho_L(n) + n$$

tenemos finalmente el teorema

Teorema [24] 3.6 Si L es un lenguaje algebraico exponencial (por ejemplo E), entonces $\#(L)$ es un lenguaje polinomial que no pertenece a la familia \underline{Gre} ($\#(E) \in \underline{Pol} \rho \setminus \underline{Gre}$)

Prueba Sabemos que $\#(L)$ tiene índice racional lineal cualquiera que sea L . La demostración que $\#(L)$ no está en \underline{Gre} utiliza la noción de centro e índice racional.

Dado que \underline{Gre} es uniforme, si $L' \in \underline{Gre}$, $C(L')$ también, con lo que si L' es polinomial $C(L')$ también debe serlo.

Esto no ocurre cuando tomamos $L' = \#(L)$, con L un lenguaje algebraico de índice racional exponencial.



Estudiemos seguidamente la familia $\overline{\text{Exp} \rho}$, tenemos el lema preliminar siguiente:

Lema 3.11 Si $L' = \tau(L)$ y L' tiene índice racional exponencial, entonces L tiene índice racional exponencial.

Prueba Evidente, considerando el comportamiento del índice racional en las transducciones.

Dicho lema tiene como consecuencia que cualquier generador de los recursivamente enumerables en tanto que cono racional, tiene índice racional exponencial, es decir:

Corolario 3.2 El lenguaje $FIFO_2$ definido como la clase de \mathcal{E} en la congruencia α definida en $\{a, \bar{a}, b, \bar{b}\}^*$ por

$$w_1 \equiv w_2 \iff w_1 = ax\bar{a}y \text{ o } w_1 = bx\bar{b}y, w_2 = xy, x \in \{a, b\}^*$$

tiene índice racional exponencial.

Prueba $FIFO_2$ genera R.E., en tanto que cono racional, por lo tanto domina E por transducción racional.

Lo mismo podemos decir respecto a cualquier generador de los lenguajes algebraicos.

Teorema [8] 3.7 Cualquier generador de la familia de los lenguajes algebraicos tiene índice racional exponencial, formalmente

$$\text{Gen} \subset \overline{\text{Exp}\rho}$$

Prueba Dado que los lenguajes algebraicos son un full-A.F.L. principal cerrado por substitución, los generadores en tanto que cono racional y full-A.F.L. coinciden [6 , pag.198] .

No tiene sentido, entonces, distinguir entre generadores en tanto que cono o full-A.F.L.

Dado que cualquier generador domina E por transducción racional, obtenemos el resultado.

II.4. Costo rígido.

Una manera de medir un lenguaje es contando el número de operaciones necesarias para engendrar sus aproximaciones a partir de programas rígidos (straight-line). Dicha medida ha sido introducida por G.B. Goodrich, R.E. Ladner y M.J. Fischer [31].

Los resultados que a continuación citamos se hallan en dicho artículo.

Comencemos por dar la definición de programa rígido que genera un lenguaje finito de X^* . Notaremos $\mathcal{P}_F(X^*)$ los subconjuntos finitos de X^* .

Definición 4.1 Consideremos un conjunto finito

$$\Omega = \{ h \mid h: \mathcal{P}_F(X^*) \times \mathcal{P}_F(X^*) \rightarrow \mathcal{P}_F(X^*) \}$$

de aplicaciones, llamado operaciones de base y el conjunto

$$\Gamma = \{ \varepsilon \} + \{ \{x\} \mid x \in X \}$$
 llamado conjunto de constantes.

Llamaremos programa rígido β_Ω a k pasos a

$$\beta_\Omega = (\beta_1, \beta_2, \dots, \beta_k)$$

en que para $1 \leq j \leq k$ tenemos que $\beta_j \in \Gamma$ ó $\beta_j = (h; \beta_p, \beta_q)$ con $h \in \Omega$ y $1 \leq p, q \leq j$.

Llamaremos costo de β_Ω , $c(\beta_\Omega)$ al número de operaciones de

β_Ω , formalmente:

$$c(\beta_\Omega) = \text{Card} \{ \beta_j \mid \beta_j \notin \Gamma \}$$

Si $\beta_j \in \Gamma$ diremos que el lenguaje calculado por β_j , L_{β_j} es $L_{\beta_j} = \beta_j$ si no dicho lenguaje se halla definido recursivamente por $L_{\beta_j} = h(L_{\beta_p}, L_{\beta_q})$.

Diremos que β_Ω calcula $L \in \mathcal{P}_F(X^*)$ si existe $1 \leq j \leq k$ que satisface $L = L_{\beta_j}$, notaremos $L = L(\beta_\Omega)$.

Vamos seguidamente a definir el coste r gido de $L \subset X^*$, tomando Ω como operaciones de base.

Definici n 4.2 Sea $L \subset X^*$, llamaremos coste r gido $\Omega_L(n)$ a la funci n

$$\Omega_L(n) = \min \{ c(\beta_\Omega) \mid L(\beta_\Omega) = L \cap X^n \}$$

Vamos en principio a tomar como operaciones de base la uni n y concatenaci n de conjuntos. $\Omega = (+, \cdot) = \text{uc}$.

En este caso, hablaremos de uc-medida y notaremos $\text{uc}_L(n)$.

Podemos encontrar un teorema an logo al dado por Shannon y Lupanov en el caso de funciones booleanas que nos mide el comportamiento asint tico de la uc-medida.

Teorema [31] 4.1 Sea $L \subset X^*$ con $\text{Card } X = s \geq 2$. Para todo $\epsilon > 0$ y para todo n suficientemente grande se cumple:

$$uc_L(n) \leq (1 + \epsilon) \frac{s^n}{n \log_2 s}$$

Ejemplo 4.1 Comencemos por tratar $L = a^*$. Vamos a mostrar que

$$uc_{a^*}(n) \leq r(n) = \Theta(\log n)$$

Con $r(n) = \lambda(n) + \nu(n) - 2$; $\lambda(n)$ es el número de bits en el desarrollo binario de n , y $\nu(n)$ es el número de cifras 1 en dicho desarrollo.

Comencemos por calcular a^n por el método dicotómico [7] que indicamos seguidamente

$$\begin{aligned} \text{exp}(a, n) = & \text{Si } n = 1 \text{ entonces } a \\ & \text{sino si } n = \text{par} \text{ entonces } \text{exp}(a, a, \frac{n}{2}) \\ & \text{sino } a \cdot \text{exp}(a, n - 1) \end{aligned}$$

El número de concatenaciones en este algoritmo es $r(n)$.

Veamos como calculamos a^{15} por el método dicotómico:

$$a^{15}, a \cdot a^4, a \cdot (a^7)^2, a \cdot (a \cdot a^6)^2, a \cdot (a \cdot (a^3)^2)^2, a \cdot (a \cdot (a \cdot a^2)^2)^2$$

Si consideramos este cálculo en orden ascendente tenemos:

$$\begin{aligned} a, a^2 = a \cdot a, a^3 = a \cdot a^2, a^6 = a^3 \cdot a^3, a^7 = a \cdot a^6, a^{14} = a^7 \cdot a^7, \\ a^{15} = a \cdot a^{14} \end{aligned}$$

$$\text{Vemos que } r(15) = 4 + 4 - 2 = 6$$

Podemos transformar dicho cálculo en un programa rígido. $\beta_{uc} =$

$$\begin{aligned} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7) \text{ con } \beta_1 = a, L_{\beta_1} = a, \\ \beta_2 = (\cdot; \beta_1, \beta_1), L_{\beta_2} = a^2, \beta_3 = (\cdot; \beta_1, \beta_2), L_{\beta_3} = a^3, \end{aligned}$$

$$\beta_4 = (. ; \beta_3, \beta_3) , L_{\beta_4} = a^6 , \beta_5 = (. ; \beta_1, \beta_4) , L_{\beta_5} = a^7$$

$$\beta_6 = (. ; \beta_5, \beta_5) , L_{\beta_6} = a^{14} , \beta_7 = (. ; \beta_1, \beta_6) , L_{\beta_7} = a^{15}$$

Tenemos que para todo $n \exp(a, n)$ nos permite hallar un programa rígido para el cálculo de a^n . Dado que el número de concatenaciones en dicho programa es $r(n)$, tenemos:

$$uc_{a^*}(n) \leq r(n)$$

Ejemplo 4.2 Sea $L_k = \{ a_1^n a_2^n \dots a_k^n \mid n \geq 1 \}$ veamos que:

$$uc_{L_k}(kn) \leq kr(n) + k = \Theta(\log n)$$

Sea $\beta_{uc}^i = (\beta_1^i, \beta_2^i, \dots, \beta_{r+1}^i)$ un programa rígido para a_i^m , $1 \leq i \leq k$, con $L_{\beta_{r+1}^i} = a_i^n$. Consideremos el programa rígido siguiente:

$$\beta_{uc} = (\beta_{uc}^1, \beta_{uc}^2, \dots, \beta_{uc}^k, \beta_{k(r+1)+1}, \dots, \beta_{k(r+2)-1})$$

con : $\beta_{k(r+1)+1} = \beta_{r+1}^1 \cdot \beta_{r+1}^2$

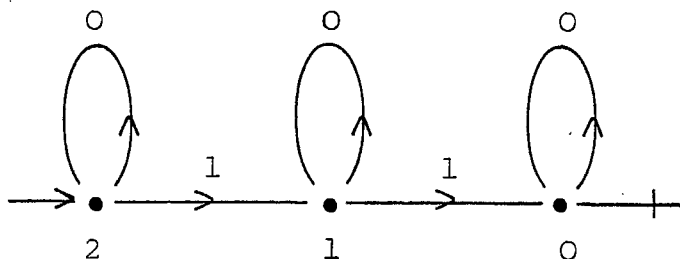
$$\beta_{k(r+1)+2} = \beta_{k(r+1)+1} \cdot \beta_{r+1}^3$$

$$\beta_{k(r+2)-1} = \beta_{k(r+2)} \cdot \beta_{r+1}^k$$

Tenemos que $L_{\beta_{k(r+2)-1}} = a_1^n a_2^n \dots a_k^n$

Ejemplo 4.3 Vamos a calcular la uc-medida de $L = 0^* 1 0^* 1 0^*$.

Consideremos el autómata finito \mathcal{A} que reconoce L



$$A = \langle q_2 \rightarrow 0q_2 + 1q_1 ; q_1 \rightarrow 0q_1 + 1q_0 ; q_0 \rightarrow 0q_0 ; q_- = q_2 ; q_+ = q_0 \rangle$$

Vamos a introducir los pasos $q[i]$ en que $q[i]$ genera las palabras de longitud i que se pueden obtener tomando q como axioma.

Utilizando una notación tipo Algol, que puede fácilmente trasladarse a los programas β_Ω definidos precedentemente, llegamos a:

principio

$$q_0[0] \leftarrow \varepsilon ;$$

$$q_1[0] \leftarrow \emptyset ;$$

$$q_2[0] \leftarrow \emptyset ;$$

para $i = 1$ a n hacer

principio

$$q_2[i] \leftarrow \{1\} \cdot q_1[i-1] + \{0\} \cdot q_2[i-1] ;$$

$$q_1[i] \leftarrow \{1\} \cdot q_0[i-1] + \{0\} \cdot q_1[i-1] ;$$

$$q_0[i] \leftarrow \{0\} \cdot q[i-1]$$

fin

fin

Vemos, entonces, que los índices i en $q[i]$ no son necesarios, esto nos conduce a [31]:

principio

$$q_0 \leftarrow \varepsilon ;$$

$$q_1 \leftarrow \emptyset ;$$

$$q_2 \leftarrow \emptyset ;$$

para $i = 1$ a n hacer

principio

$$\begin{aligned}
 q_2 &\leftarrow \{1\} \cdot q_1 + \{0\} \cdot q_2 ; \\
 q_1 &\leftarrow \{1\} \cdot q_0 + \{0\} \cdot q_1 ; \\
 q_0 &\leftarrow \{0\} \cdot q_0
 \end{aligned}$$

fin

fin

Vemos que $uc_L(n) = O(n)$. Notemos que dicha construcción es válida para cualquier lenguaje racional, con lo que todos los lenguajes racionales tienen una uc-medida lineal, como máximo.

Ejemplo 4.4 Vamos finalmente a tratar

$$PAL_2 = \{ w\tilde{w} \mid w \in \{a,b\}^* \}$$

El lenguaje PAL_2 está engendrado por la gramática:

$$S \rightarrow aSa + bSb + aa + bb$$

Dicha gramática nos muestra como dar un programa rígido para las palabras de longitud $2n$. Tenemos:

principio

$$S \leftarrow a.a + b.b$$

$$\text{para } i = 1 \text{ a } n \text{ hacer } S \leftarrow a.S.a + b.S.b$$

fin

Llegamos, por lo tanto, a que $uc_{PAL_2}(n) = O(n)$. Notemos que todos los lenguajes lineales tienen una uc-medida lineal.

Veamos que la situación es muy distinta cuando tratamos $COPY_2$.

Ejemplo [31] 4.5 Vamos a tratar seguidamente:

$$\text{COPY}_2 = \{ ww \mid w \in \{0,1\}^* \}$$

Se puede demostrar que para todo n :

$$uc_{\text{COPY}_2}(2n) \geq \frac{2^{n+1}}{n} - 1$$

y que para todo $\epsilon > 0$ y n suficientemente grande

$$uc_{\text{COPY}_2}(2n) \leq (2 + \epsilon) \frac{2^{n+1}}{n}$$

es decir; $uc_{\text{COPY}_2}(2n) = \Theta\left(\frac{2^{n+1}}{n}\right)$

Indiquemos rápidamente cómo construir el programa rígido que nos permite hallar el límite superior.

Sean $w_1, w_2 \in \{0,1\}^n$ diremos que w_1 y w_2 son conjugados si $w_1 = uv$ y $w_2 = vu$. La relación de conjugación es una relación de equivalencia que notaremos conj . Se cumple que

$$\text{Card}\left(\frac{\{0,1\}^n}{\text{conj}}\right) \leq \left(1 + \frac{\epsilon}{4}\right) \frac{2^n}{n}$$

Tomemos en cada clase un representante, y sea S el conjunto así construido. Tenemos que S contiene el mínimo número de elementos de $\{0,1\}^n$ que generan todo $\{0,1\}^n$ por permutación circular.

Vamos a construir las palabras de la forma ww con $|w| = 3m$, a partir del programa siguiente:

$$\sum_{\substack{u \in \{0,1\}^{2m} \\ u = u_1 \dots u_{2m}}} \left(\sum_{i=0}^{2m} u_{i+1} \dots u_{2m} \left(\sum_{\substack{t \in \{0,1\}^m \\ tu \text{ ó } ut \in S}} t \right) u_1 \dots u_i \right)$$

El siguiente teorema muestra que los lenguajes algebraicos tienen un costo cuadrático.

Teorema 4.2 [31] Sea L un lenguaje algebraico, se cumple:

$$uc_L(n) = O(n^2)$$

Prueba Sea $G = \langle X, V, P, S \rangle$ una gramática en forma normal de Chomsky que engendra L . Sea $p = \text{Card } P$. Podemos partir de G , construir un programa rígido β para $L \cap X_n$ de la forma siguiente. A la variable $v \in V$ vamos a asociar los pasos $v[i]$ con $1 \leq i \leq n$. El paso $v[i]$ construirá las palabras de longitud i derivables a partir de v . El programa rígido viene esquematizado por:

$$v[1] = \sum \{ x \mid v \rightarrow x \in P \}$$

$$v[i] = \sum \{ v'[j] \cdot v''[k] \mid j + k = i \text{ y } v \rightarrow v'v'' \in P \}$$

Tenemos que $S[n]$ calcula $L \cap X^n$

Para j, k fijos hay, como máximo, p productos; el número de posibles pares j, k está acotado por i , como $i \leq n$. El número de productos es como máximo pn , lo mismo pasa con el número de uniones. Dado que hay como máximo pn pasos $v[i]$ llegamos a $C(\beta) \leq 2p^2n^2$.

Tenemos que el límite precedente es óptimo, en efecto:

Lema 4.1 [31]: El lenguaje:

$$L = \{ a^i 0^j 1^k a^j b^k a^i \mid i, j, k > 1 \}$$

satisface $uc_L(n) = \Theta(n^2)$

Considerando el teorema que acabamos de dar, vemos que cualquier lenguaje algebraico tiene una uc-medida, como máximo, cuadrática en particular $PAL_2 = \{w\tilde{w} \mid w \in \{0,1\}^*\}$

Tenemos, por lo tanto, que mientras PAL_2 y $COPY_2$ no difieren en la función de crecimiento, en la \mathfrak{a} -medida o en el índice racional, difieren profundamente en la uc-medida.

Veamos que los resultados dados precedentemente, varían de modo notable cuando añadimos al conjunto de operaciones de base, la intersección, es decir:

$$\Omega = \{+, \cdot, \cap\} = uci$$

Veamos que en este caso, el resultado dado para $COPY$ ya no es válido; en efecto:

Ejemplo 4.6 [31] Veamos que cuando consideramos la intersección obtenemos:

$$uci_{COPY_2}(2n) \leq 11n$$

Para ello, consideremos el uci-programa siguiente

principio

$$A_0 \leftarrow \{\varepsilon\};$$

$$0 \leftarrow \{0\};$$

$$1 \leftarrow \{1\};$$

$$A_1 \leftarrow \{0 + 1\};$$

para $i = 2$ a n hacer $A_i \leftarrow A_1 \cdot A_{i-1}$;
para $i = 1$ a n hacer $B_i^0 \leftarrow A_{i-1} \cdot 0 \cdot A_{n-1} \cdot 0 \cdot A_{n-i}$;
para $i = 1$ a n hacer $B_i^1 \leftarrow A_{i-1} \cdot 1 \cdot A_{n-1} \cdot 1 \cdot A_{n-i}$;
para $i = 1$ a n hacer $B_i \leftarrow B_i^0 + B_i^1$;
 $C \leftarrow B_1$;
para $i = 2$ a n hacer $C \leftarrow C \cap B_i$
fin

Evidentemente la uci-medida admite la misma cota superior que la uc-medida, es decir:

$$uci_L(n) \leq (1 + \varepsilon) \frac{S^n}{n \log_2 S}$$

si $S = \text{Card } X \geq 2$, $L \subset X^*$

El teorema siguiente muestra que la "mayoría" de lenguajes son en

orden $\frac{S^n}{n \log_2 S}$

Teorema [31] 4.3 Sea $\text{Card } X = S \geq 2$. Para todo $\varepsilon > 0$ la fracción de lenguajes $L \subset X^n$ tales que

$$C_{uci}(L) \geq (1 - \varepsilon) \frac{S^n}{n \log S}$$

tiende a 1 cuando n tiende a ∞ .

Veremos en el último capítulo que los programas rígidos tomando uci como operaciones de base, están en ciertos aspectos muy próximos a los circuitos booleanos que analizamos seguidamente.

II.5. Coste booleano.

Dado un lenguaje $L \subset \{0,1\}^n$ su función característica $f_L(w) =$
Si $w \in L$ entonces 1 sino 0 es una función booleana de $\mathbb{B}^n \rightarrow \mathbb{B}$
 $\mathbb{B} = \{0,1\}$. Podemos, entonces, contar el número de puertas lógicas necesarias para sintetizarla. El primer artículo en esta dirección ha sido realizado por Savage [50]. Trataremos los circuitos lógicos en términos de programa rígido.

Definición 5.1 Consideremos el conjunto de operaciones booleanas

$$\Omega = \{ \wedge, \vee, \neg \}$$

A fin de tratar funciones booleanas de n variables, introducimos el conjunto de datos $\Gamma = \{0,1,x_1,x_2,\dots,x_n\}$ formado por el conjunto de constantes $\{0,1\}$ y el conjunto de variables $\{x_1,x_2,\dots,x_n\}$. Llamaremos circuito booleano a $\beta = (\beta_1, \beta_2, \dots, \beta_k)$ en el que para $1 \leq j \leq k$ tenemos que $\beta_j \in \Gamma$ ó $\beta_j = (h; \beta_p, \beta_q)$ con $h \in \Omega$ y $1 \leq p, q \leq j$

Llamaremos costo de β al número de puertas booleanas empleadas, formalmente:

$$c(\beta) = \text{Card} \{ \beta_j \mid \beta_j \notin \Gamma \}$$

Si $\beta_j \in \Gamma$ la función booleana calculada por β_j es $f_{\beta_j} = \beta_j$

Si no se halla definida por $f_{\beta_j} = h(\beta_p, \beta_q)$. Diremos que β sin-

tetiza f si existe j con $f_{\beta_j} = f$, en este caso notaremos $f_{\beta} = f$.

Vamos seguidamente a definir el coste boleano asociado a un lenguaje en $\{0,1\}^*$

Definición 5.2 Sea $L \subset \{0,1\}^*$ definimos el coste boleano de L por:

$$b_L(n) = \min \left\{ C(\beta) \mid f = f_{L_n} \text{ con } L_n = L \cap \{0,1\}^n \right\}$$

Vamos a dar seguidamente el comportamiento asintótico de $b_L(n)$

Teorema de Shannon Lupanov [37] 5.1 Sea $L \subset \{0,1\}^*$ se cumple

$$b_L(n) \leq \frac{2^n}{n} \left(1 + o\left(\frac{1}{\sqrt{n}}\right) \right)$$

Tenemos, en principio, que la "mayor parte" de funciones booleanas necesitan $\frac{2^n}{n}$ puertas para ser sintetizadas formalmente.

Teorema [37] 5.2 Para n suficientemente grande y para cualquier

$\varepsilon > 0$, todas las funciones de n variables necesitan al menos $(1 - \varepsilon) \frac{2^n}{n}$ puertas excepto una fracción arbitrariamente pequeña estrictamente positiva.

El teorema siguiente nos muestra que a pesar de todo los lenguajes que más frecuentemente encontramos como $\{a^n, b^n \mid n \geq 0\}$, D_n^* , D_n^* , $TWIN_n$, $\{a^n b^n c^n \mid n \geq 0\}$, SYM_n , $COPY_n$ admiten todos un coste booleano acotado por un polinomio.

Una primera versión de dicho teorema, se debe a Savage [50], nosotros damos aquí la versión debida a Fischer y Pippenger [22].

Teorema 5.3 Si existe una máquina de Turing determinista que reconoce L con tiempo $T(n) \geq n$, existe $d > 0$ tal que

$$b_L(n) \leq d \cdot T(n) \cdot \log T(n)$$

Notemos que en términos de b -medida no parece existir diferencia entre PAL_2 y $COPY_2$; en efecto:

Ejemplo 5.1 Comencemos por dar un circuito que sintetiza la función característica que corresponde a los palíndromes.

$$PAL_2 = \{ w\tilde{w} \mid w \in \{0,1\}^* \}$$

Tratemos el caso $2n$ para $n = 3$. Figura II.28.

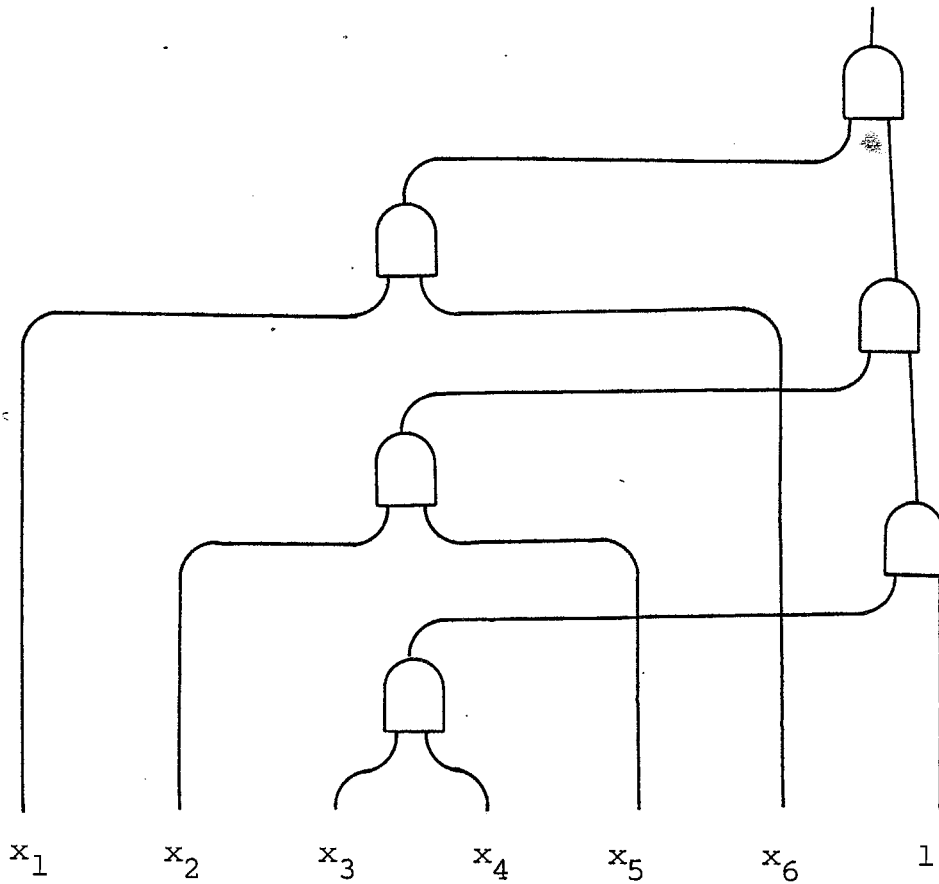


Figura II.28.

Tenemos que $b_{PAL_2}(2n) \leq 2n$

Ejemplo 5.2 Tratemos seguidamente el lenguaje de los cuadrados

$$COPY_2 = \{ww \mid w \in \{0,1\}^*\}$$

Damos un circuito booleano para $2n, n = 3$. Figura II.29.

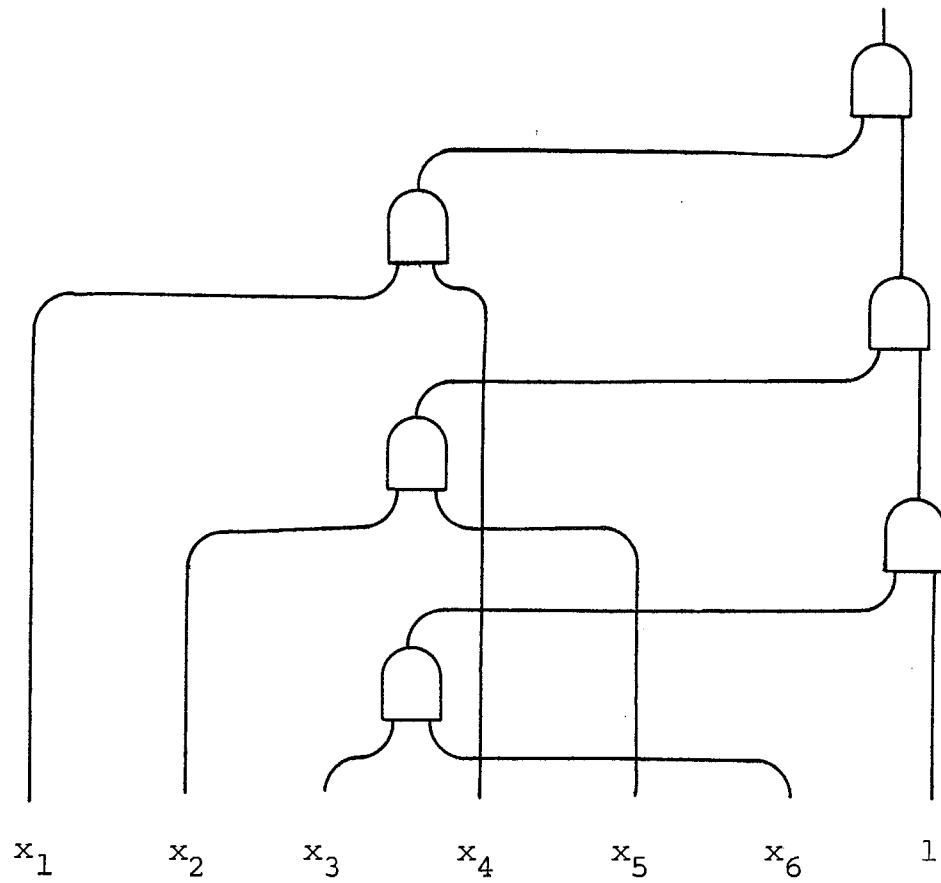


Figura II.29.

Tenemos que $b_{\text{COPY}_2}(2n) \leq 2n$



II.6. h-medida.

Otra manera de clasificar distintos lenguajes consiste en imponer algún tipo de limitación a los autómatas que los reconocen.

En el caso de los lenguajes algebraicos, un instrumento básico de análisis son los árboles de derivación asociados a una palabra dada. K. Culik II y H.A. Maurer [15] han estudiado la altura del árbol de derivación con relación a la longitud de la palabra que engendran.

Es esta la idea base en la definición de h-medida (h por height) que damos seguidamente.

Definición [15] 6.1 Diremos que un lenguaje L tiene una h-medida $h_L(n)$ si existe una gramática que lo engendra, tal que toda palabra de nL admite en G un árbol de derivación de altura acotada superiormente por $h_L(n)$.

Recordemos que una función f es sublineal cuando asintóticamente crece más suavemente que la identidad, formalmente f es sublineal

$$\text{ssi } \lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0.$$

Como ejemplo de función sublineal, podemos citar la función log.

Comencemos por estudiar el comportamiento asintótico de $h_L(n)$;

tenemos:

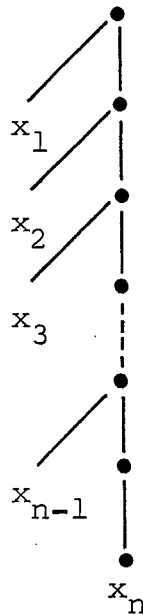
Lema 6.1 Sea L un lenguaje algebraico, su h-medida cumple:

$$h_L(n) = O(n) \text{ y } h_L(n) = \Omega(\log n)$$

Prueba

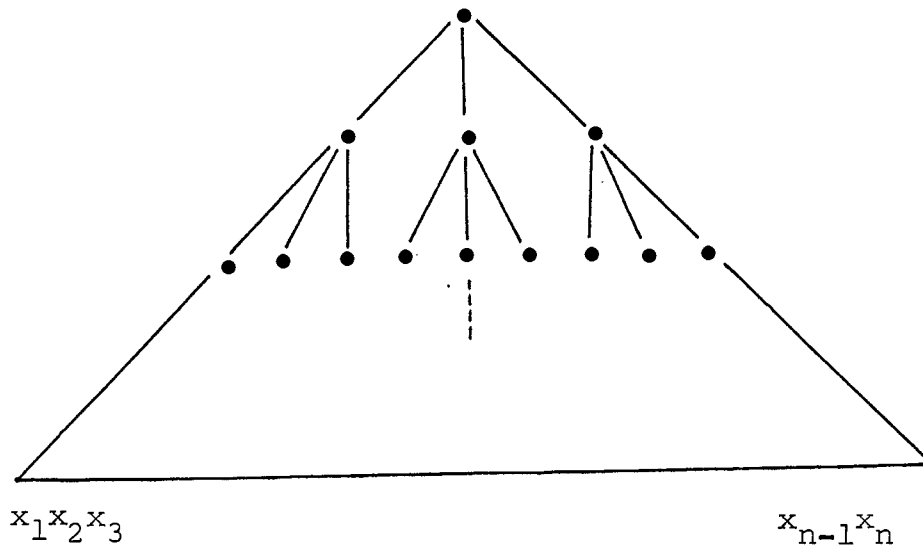
(a) Tomemos una gramática reducida que engendre L.

Sea $w = x_1 x_2 x_3 \dots x_{n-1} x_n \in L$; el caso más desfavorable consiste en un árbol del tipo:



En este caso la altura es una función lineal de la longitud.

(b) El caso más favorable se dará cuando aparezcan árboles n -arios completos, por ejemplo



en este caso la altura es una función logarítmica de la longitud. Notemos que respecto a la altura del árbol de derivación, la función logaritmo da un límite inferior; no podemos, entonces, esperar encontrar lenguajes algebraicos de altura $\log \log n$.

Veamos seguidamente una familia de lenguajes de altura sublineal.

Lema [15] 6.2 Si R es un lenguaje racional, entonces $h_R(n) = O(\log n)$

Prueba Sea $A = \langle X, Q, q_-, q_+, \cdot \rangle$ un autómata que acepta R , consideremos la gramática cuyas reglas vienen dadas por:

$$P = \{ S \rightarrow [q_-, q] \mid q \in Q_+ \} + \\ \{ [p, q] \rightarrow [p, r] [r, q] \mid p, q, r \in Q \} + \\ \{ [p, q] \rightarrow x \mid q \in p \cdot x \}$$

si R contiene ϵ añadimos $S \rightarrow \epsilon$. Dicha gramática genera R y su altura es $\log_2 n$.

Vamos a estudiar seguidamente el comportamiento de la h -medida con relación a distintas operaciones entre lenguajes. A este fin recordemos la definición de máquina secuencial generalizada (gsm) $g: \langle X, Y, q_0, Q_+, T \rangle$ en que X es el alfabeto de entrada, Y el de salida y la función de transición es un subconjunto de $Q \times X \times Y^* \times Q$.

Su funcionamiento es análogo al de un autómata no determinista al que se ha añadido una función de salida; diremos que $f \in g(w)$ si existe en g un camino que permite leer w y salir f . Si $L \subset X^*$ definimos $g(L) \subset Y^*$ como el conjunto de salidas, cuando se leen palabras de L . Diremos que g es ϵ -limitado sobre L si existe una constante $k > 0$, tal que el número de letras cuya imagen por g es ϵ en L , está limitado por k .

Demos ahora el comportamiento de h_L .

Lema [15] 6.3 Sean L_1, L_2 lenguajes algebraicos y R un lenguaje racional, tenemos:

$$h_{L_1+L_2}(n) \leq 1 + \max \{ h_{L_1}(n), h_{L_2}(n) \}$$

$$h_{L_1 \cdot L_2}(n) \leq 1 + \max \{ h_{L_1}(n), h_{L_2}(n) \}$$

$$h_{L_1^*}(n) \leq \log_2 n + h_{L_1}(n)$$

$$h_{L_1 \cap R}(n) \leq 1 + h_L(n)$$

Si $g: X^* \rightarrow Y^*$ es un gsm ϵ -limitado sobre $L_1 \subset X^*$, entonces existe k que cumple:

$$h_{g(L)}(n) \leq k \cdot h_L(n)$$

Prueba La suma y el producto son evidentes. A fin de tratar la estrella, consideremos la regla $S \rightarrow SS$ que sirve para fabricarnos S^n con una altura $\log_2 n$.

A fin de tratar la intersección con un racional, basta notar que la construcción triple $[q, v, q']$ para calcular dicha intersección, no aumenta la altura del árbol de derivación.

Basta finalmente tratar el caso de la máquina secuencial generalizada. Consideremos una gramática para L en forma normal de Chomsky que genera L y realicemos la triple construcción sobre dicha gramática.

$$P' = \{ S \rightarrow [q_0, S, q] \mid q \in Q_+ \} + \\ \{ [p, v, q] \rightarrow [p, v_1, r] [r, v_2, q] \mid v \rightarrow v_1 v_2 \in P, p, q, r \in Q \} + \\ \{ [p, v, q] \rightarrow w \mid v \rightarrow a \text{ y } (p, a, w, q) \in T \}$$

Dado que g es ε -limitado sobre L , la porción de palabra borrada será como máximo una función lineal de la longitud.

Vamos a ver, seguidamente, que en a^*b^* los únicos lenguajes sublineales que existen son los lenguajes racionales.

Comenzamos por demostrar varios lemas técnicos.

Lema [15] 6.4 Sea $L \subset a^*b^*$ un lenguaje algebraico y sean

$$R_1 = \bigcup_{\substack{u_i > 0 \\ 1 \leq i \leq k_1}} (a^{u_i})^*, \quad R_2 = \bigcup_{\substack{v_j > 0 \\ 1 \leq j \leq k_2}} (a^{v_j}) \quad \text{dos lenguajes racionales,}$$

el lenguaje $L' = R_1 L R_2$ es un lenguaje racional.

Prueba Sea $G = \langle \{a, b\}, V, P, S \rangle$ una gramática lineal reducida, de axioma S que engendra L . Sea G' una gramática que engendra L' definida por: $G' = \langle \{a, b\}, V', P', S' \rangle$ con

$$V' = V + \{ A_{u_i} \mid u_i \text{ en } R_1 \} + \{ B_{v_j} \mid v_j \text{ en } R_2 \} + \{ S' \}$$

$$P' = P + \{ \bar{S} \rightarrow A_{u_i} S B_{v_j} \mid u_i \text{ en } R_1, v_j \text{ en } R_2 \} +$$

$$+ \left\{ A_{u_i} \rightarrow A_{u_i} A_{u_i} + a^{u_i} + \varepsilon \mid u_i \text{ en } R_1 \right\} +$$

$$+ \left\{ B_{v_j} \rightarrow B_{v_j} B_{v_j} + b^{v_j} + \varepsilon \mid v_j \text{ en } R_2 \right\}$$

Evidentemente $L' = L(G')$.

Vamos a buscar los bucles elementales en G , llamados $B(V)$. Comenzamos por los bucles elementales de cada variable $B(v) = \{ (a^i, b^j) \mid$ existe en G una derivación.

$$v \rightarrow a^{i_1} v_1 b^{j_1} \rightarrow a^{i_2} v_2 b^{j_2} \rightarrow a^{i_3} v_3 b^{j_3} \rightarrow \dots \rightarrow a^i v b^j,$$

con v, v_1, v_2, v_3, \dots distintas }

Definimos, entonces, $B(V) = \bigcup_{v \in V} B(v)$. Llamamos $\tau = \text{Card } B(V)$,

el número de bucles elementales en G . Sea $\rho = \text{Card } V$, es decir, el número de variables en G . Sea finalmente

$$\mu = \text{mcm} \{ u_i v_j \mid \bar{S} \rightarrow A_{u_i} S B_{v_j} \text{ en } P' \}$$

Consideremos un árbol de derivación que comienza por:

$$S \rightarrow A_{u_i} S B_{v_j}$$

y que partiendo de S tiene un camino de longitud mayor que $\mu \cdot \tau \cdot \rho$, dicho camino contiene más de $\mu \cdot \tau$ bucles elementales de V , con lo que existe un bucle elemental $v \rightarrow a^x v b^y$ que ha sido empleado al menos μ veces.

Consideremos, entonces, μ de estos bucles elementales, han sido introducidos $x \cdot \mu$ letras a , y $y \cdot \mu$ letras b . Dado que en $S \rightarrow A_{u_i} S B_{v_j}$ u_i y v_j dividen a μ , podemos eliminar dichos bucles y fabricar $a^{x \cdot \mu}$ y $b^{y \cdot \mu}$ a partir de $A_{u_i} \rightarrow A_{u_i} A_{u_i}$ y $B_{v_j} \rightarrow B_{v_j} B_{v_j}$.

Podemos transformar todo árbol de manera que el camino que parta de S tenga como máximo $\mu \cdot \tau \cdot \rho$ producciones. Es decir, para fa-

bricar cualquier palabra de G' necesitamos como máximo $\mu.T.p$ producciones de G , es decir, las producciones de G sirven para fabricar un conjunto finito F de palabras. Podemos eliminar G y poner F en su lugar, obteniendo:

$$G'' = \langle \{a, b\}, V, P, S'' \rangle$$

$$\text{con } V = \{ A_{u_i} \mid u_i \text{ en } R_1 \} + \{ B_{v_j} \mid v_j \text{ en } R_2 \}$$

$$P = \{ S \rightarrow A_{u_i}^m B_{v_j} \mid A_{u_i} \text{ y } B_{v_j} \text{ en } V \text{ y } m \in F \} +$$

$$+ \{ A_{u_i} \rightarrow A_{u_i} A_{u_i} + a^{u_i} + \epsilon \mid u_i \text{ en } R_1 \} +$$

$$+ \{ B_{v_j} \rightarrow B_{v_j} B_{v_j} + b^{v_j} + \epsilon \mid v_j \text{ en } R_2 \}$$

Vamos a dar seguidamente un ejemplo.

Ejemplo 6.1 Sea G la gramática

$$G = \langle S \rightarrow aSb^2 + aTb ; T \rightarrow aTb + ab ; S \text{ axioma} \rangle$$

tomemos como racionales $R_1 = (a^2)^*$, $R_2 = (b^3)^*$

En este caso, G' es la gramática de axioma S' siguiente:

$$S' \rightarrow A_2 S B_3$$

$$S \rightarrow aSb^2 + T$$

$$T \rightarrow aTb + \epsilon$$

$$A_2 \rightarrow A_2 A_2 + a^2 + \epsilon$$

$$B_3 \rightarrow B_3 B_3 + b^3 + \epsilon$$

El lenguaje engendrado es:

$$L(G') = (a^2)^* \{ a^{p+n} b^{2p+n} \mid p \geq 0, n \geq 0 \} (b^3)^*$$

Vamos a mostrar que dicho lenguaje es racional. Consideremos una derivación a partir de S.

$$S \xrightarrow{p} a^p S b^{2p} \xrightarrow{n} a^p T b^{2p} \xrightarrow{n} a^{p+n} T b^{2p+n} \xrightarrow{} a^{p+n} b^{2p+n}$$

$$\text{En este caso } \mu = \text{mcm} \{ 2, 3 \mid S' \rightarrow A_2 S A_3 \} = 6$$

Si $p \geq 6$ podemos expresarlo $p = 6x + p_1$, podemos generar

$$S \xrightarrow{*} a^{6x} S b^{12x} \text{ a partir de } A_2 \text{ y } B_3; \text{ lo mismo ocurre con}$$

T, con lo que:

$$L'(G') = (a^2)^* \{ a^{p+n} b^{2p+n} \mid 0 \leq p \leq 6, 0 \leq n \leq 6 \} (b^3)^*$$

Vemos, entonces, que dicho lenguaje es racional.

Vamos a dar seguidamente un lema de normalización para las gramáticas de los lenguajes acotados.

Lema [15] 6.5 Sea $L \subset a^* b^*$ un lenguaje algebraico de altura $h_L(n)$, existe una gramática $G = \langle \{a, b\}, V, P, S \rangle$ de altura $h_L(n)$ que satisface:

(1) Las variables son una unión disjunta de:

$$V = V_a + V_b + V_{\text{medio}} + \{ \bar{S} \}$$

(2) Las producciones son una unión disjunta de:

$$P = P_a + P_b + P_{\text{término}} + P_{\text{regular}} + P_{\text{comienzo}} + P_{\text{bucle}}$$

En que:

• P_a es de la forma:

$$P_a = \{ A_p \rightarrow A_p A_p + a^p + \epsilon \mid A_p \in V_a \}$$

. P_b es de la forma:

$$P_b = \{ B_q \rightarrow B_q B_q + b^q + \epsilon \mid B_q \in V_b \}$$

. $P_{\text{término}}$ contiene solamente producciones de la forma

$$\alpha \rightarrow a^m b^n, \quad \alpha \in V_{\text{medio}}, \quad m, n \geq 0$$

. P_{regular} contiene solamente producciones de la forma

$$S \rightarrow \alpha a^m b^n \beta \quad \text{con} \quad \alpha \in V_a + \epsilon \\ \beta \in V_b + \epsilon, \quad m, n \geq 0$$

. P_{comienzo} contiene solamente reglas de la forma

$$\bar{S} \rightarrow \alpha a^m v b^n \beta \quad \text{con} \quad \alpha \in V_a + \epsilon \\ \beta \in V_b + \epsilon, \quad v \in V_{\text{medio}}, \quad m, n \geq 0$$

El caso $\alpha \neq \epsilon$ y $\beta \neq \epsilon$ está prohibido

. P_{bucle} contiene solamente reglas de la forma

$$v \rightarrow a^m v' b^n \quad \text{con} \quad v, v' \in V_{\text{medio}}$$

$m, n \geq 0$. Si $v \xrightarrow{*} a^x v b^y$, $v \in V_{\text{medio}}$, tenemos que

$$x > 0, \quad y > 0$$

A toda gramática de esta forma la llamaremos canónica.

Prueba Si L es algebraico, existe una gramática que lo engendra.

Sea $c \rightarrow c_1 c_2 \dots c_n$ una regla de dicha gramática. Hay dos casos:

Primer caso Existe una variable c_i tal que $c \rightarrow c_1 c_2 \dots c_i c_{i+1} \dots c_n$

con $L_{c_1 c_2 \dots c_i} \subset a^*$ y $L_{c_{i+1} \dots c_n} \subset b^*$. En este caso, dado que

todo lenguaje algebraico sobre una letra es racional, podemos substituir dicha regla por un conjunto de reglas $c \rightarrow \alpha a^m b^n \beta$ con

$$\alpha \in V_a + \varepsilon, \beta \in V_b + \varepsilon$$

Segundo caso Existe una variable c_i tal que $L_{c_i} \subset a^+b^+$, en este caso $c \rightarrow c_1c_2 \dots c_{i-1}c_i c_{i+1} \dots c_n$, $L_{c_1 \dots c_{i-1}} \subset a^*$ y

$L_{c_{i+1} \dots c_n} \subset b^*$. En este caso dicha regla se puede substituir

por un conjunto de reglas del tipo:

$$c \rightarrow \alpha a^m c_i b^n \beta$$

$$\text{con } \alpha \in V_a + \varepsilon, \beta \in V_b + \varepsilon$$

Vamos ahora a transformar la gramática de modo, a generar las distintas α y β a partir del axioma. Notemos que una vez α ha sido introducida, no necesitamos introducirla de nuevo, dado que puede ser generada a partir de A_p . Llamemos X_α al conjunto de las que hemos obtenido:

$$X_\alpha = \{ \alpha \mid \exists c \rightarrow \alpha a^m c b^n \beta \text{ ó } c \rightarrow \alpha a^m b^n \beta \}$$

definimos, también, X_β

$$X_\beta = \{ \beta \mid \exists c \rightarrow \alpha a^m c b^n \beta \text{ ó } c \rightarrow \alpha a^m b^n \beta \}$$

Sea $U \subset X_\alpha$, tenemos $U = \{ \alpha_1, \alpha_2, \dots, \alpha_k \}$. Consideremos el lenguaje $L_{\alpha_1 \alpha_2 \dots \alpha_k} \subset a^*$. Existe un conjunto finito $P(U)$ de pala-

bras del tipo a^m o $a^m A_p$ de modo que cualquier palabra de

$L_{\alpha_1 \alpha_2 \dots \alpha_k}$ o bien está en $P(U)$ o bien se puede obtener a par-

tir de él, aplicando $A_p \rightarrow A_p A_p + a^p + \varepsilon$. Sea $W \subset X_\beta$, de modo

análogo construimos $P(W)$. Si en este proceso hemos creado nuevas

variables A_p o B_q , las añadiremos a los conjuntos V_a y V_b definidos

precedentemente. Tomaremos como axioma \bar{S} , y las reglas P_{comienzo} serán:

$$\bar{S} \longrightarrow \alpha [U, S, W] \beta$$

con $U \subset V_a, W \subset V_b, \alpha \in P(U), \beta \in P(W)$

Las reglas P_{bucle} serán de la forma

$$[U, c, W] \longrightarrow a^m [U \setminus \alpha, c', W \setminus \beta] b^n$$

si $c \longrightarrow \alpha a^m c' b^n \beta$

Las reglas $P_{\text{término}}$ contendrán

$$[\alpha, c, \beta] \longrightarrow a^m b^n \text{ si } c \longrightarrow \alpha a^m b^n \beta$$

Notemos que en esta gramática podemos tener reglas del tipo

$$\bar{S} \longrightarrow \alpha a^m S b^n \beta$$

con $\alpha \neq \epsilon$ y $\beta \neq \epsilon$, el lema precedente permite eliminarlas obteniendo una gramática del tipo deseado. ■

Veamos seguidamente que en una gramática de altura sublineal, las reglas de la forma $\bar{S} \longrightarrow A_p a^i v b^j$ no son necesarias; formalmente:

Lema [15] 6.6 Sea $G = \langle \{a, b\}, V, P, \bar{S} \rangle$ una gramática canónica de altura sublineal, las reglas $\bar{S} \longrightarrow A_p a^i v b^j$ de G , engendran lenguajes racionales.

Prueba Hagamos el producto de los distintos p que aparecen en las variables A_p ; formalmente:

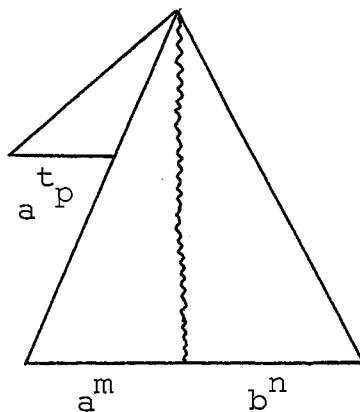
$$\pi = \prod \left\{ p \mid \exists \bar{S} \longrightarrow A_p a^i v b^j \text{ ó } \bar{S} \longrightarrow A_p a^i b^j B_q \right\}$$

Consideremos una regla $\bar{S} \longrightarrow A_p a^i v b^j$. Consideremos las palabras que podemos obtener a partir de S con "pocas" derivaciones racionales

y "muchas" derivaciones lineales; formalmente:

$$\bar{S} \rightarrow A_p a^i v b^j \xrightarrow{*} A_p a^m b^n \xrightarrow{*} a^{t \cdot p} a^m b^n, \quad 0 \leq t \leq \frac{\pi}{p}$$

Esquemáticamente podemos decir que corresponden a árboles del tipo



dado que el número de letras a debidas a la variable A_p , están acotadas; dichos árboles tienen altura lineal (excepto quizás en un número finito de casos). Dado que la gramática es sublineal, deben existir derivaciones sublineales que los engendran; es decir, para cada $0 \leq t \leq \frac{\pi}{p}$ debe existir una regla $\bar{S} \rightarrow A_{r_t} a^{i_t} b^{j_t} B_{r_t}$ que cumple

$$S \rightarrow A_{r_t} a^{i_t} b^{j_t} B_{r_t} \xrightarrow{*} a^{t \cdot p} a^m b^n, \quad 0 \leq t \leq \frac{\pi}{p}$$

Es decir, para los árboles con pocas reglas racionales y muchas lineales (altura lineal) debe existir un árbol sublineal, esto fuerza a que las reglas sean racionales.

Veamos ahora que las reglas que contienen mucho racional, pueden, también, ser engendradas de este modo. Supongamos que queremos engendrar $a^{x \cdot p} a^m b^n$ con x grande, entonces $x = u \cdot \frac{\pi}{p} + t$ con $0 \leq t \leq \frac{\pi}{p}$ entonces $a^{x \cdot p} a^m b^n = a^{u \cdot \pi + t \cdot p} a^m b^n$. Consideremos la derivación.

$$\bar{S} \rightarrow A_{r_t} a^{i_t} b^{i_t} B_{r_t} \xrightarrow{*} A_{r_t} a^{t \cdot p} a^m b^n \xrightarrow{*} a^u \cdot \pi a^{t \cdot p} a^m b^n$$

Dado que r_t divide a π sabemos que una tal derivación existe.

Hemos obtenido que cualquier palabra m que satisfaga

$$\bar{S} \rightarrow A_p a^i v b^j \xrightarrow{*} m$$

puede ser engendrada a partir de reglas racionales, excepto quizás en un número finito de casos. Es decir, la participación de las reglas lineales pueden servir como máximo a engendrar un lenguaje finito. Podemos, entonces, quitar dichas reglas y añadir dicho lenguaje finito. La regla $\bar{S} \rightarrow A_p a^i v b^j$ engendra, en consecuencia, un lenguaje racional.

Vamos seguidamente a demostrar el teorema que nos dice que todo lenguaje sublineal acotado es racional.

Teorema [15] 6.1 Sea $L \subset a^* b^*$ un lenguaje algebraico con $h_L(n)$ sublineal, entonces L es racional.

Prueba Consideremos una gramática en forma canónica que engendra L . En el lema precedente hemos visto que cuando L es sublineal, las reglas:

$$\bar{S} \rightarrow A_p a^i v b^j$$

sirven como máximo para engendrar un lenguaje finito; podemos quitar dichas reglas y añadir dicho lenguaje en forma de reglas terminales. El mismo razonamiento es utilizable para reglas del tipo:

$$\bar{S} \rightarrow a^i v b^j B_q$$

En consecuencia, en la gramática que engendra L sólo serán empleadas las reglas de la forma:

$$\bar{S} \longrightarrow \alpha a^i b^j \beta$$

con $\alpha \in V_a + \epsilon$ y $\beta \in V_b + \epsilon$ y

$$A_k \longrightarrow A_k A_k + a^k + \epsilon$$

$$B_q \longrightarrow B_q B_q + b^k + \epsilon$$

En consecuencia, el lenguaje engendrado es racional. ■

Dicho teorema admite como corolario inmediato:

Corolario 6.1 Si $L \subset a^* b^*$ es un lenguaje algebraico no racional, $h_L(n) = \Omega(n)$ para una infinidad de n .

Prueba Debe existir una infinidad de palabras para las que cualquier árbol que las engendra sea lineal, de lo contrario dicho lenguaje sería racional. ■

Dicho corolario nos induce a considerar los lenguajes de altura lineal, notados $\overline{h\text{-Lin}}$ definidos por:

Definición 6.2 Diremos que L es h -lineal, notado $\overline{h\text{-Lin}}$, si existe una infinidad de n para los que $h_L(n) = \Omega(n)$. Llamaremos $\overline{h\text{-Lin}}$ a la familia de los lenguajes $\overline{h\text{-Lin}}$.

Ejemplo 6.2 Entre los lenguajes $\overline{h\text{-Lin}}$ contamos:

$$S_1 = \{ a^n b^n \mid n > 1 \} , \quad S_{\geq} = \{ a^p b^q \mid p \geq q \}$$

$$S_{\leq} = \{ a^p b^q \mid p \leq q \} , \quad S_{\neq} = \{ a^p b^q \mid p \neq q \}$$

Dado que $D_1^* = \langle S \text{ — } aSbS + \epsilon \rangle$ satisface

$D_1^* \cap a^* b^* = S_1$, tenemos que D_1^* es lineal.

El mismo razonamiento es válido para D_n^* , $n \geq 1$



La remarca precedente sobre el lenguaje de Dyck, nos conduce a la siguiente definición:

Definición [1] 6.3 Diremos que un lenguaje algebraico L es estrictamente no regular si existe $(\alpha, u, \beta, v, \gamma)$ con $u \neq \epsilon$ y $v \neq \epsilon$ tal que $L \cap \alpha u^* \beta v^* \gamma$ es un lenguaje no regular.

Recordemos la definición de lenguajes IRS debida a S.Greibach:

Definición [3 2] 6.4 Diremos que L es IRS (infinite regular set) si L no contiene ningún lenguaje racional infinito.

Como ejemplos de lenguajes IRS tenemos $S_1 = \{a^n b^n \mid n \geq 1\}$ ó
 $PAL = \{w\tilde{w} \mid w \in \{0,1\}^*\}$. El interés de los lenguajes IRS es-
 tá en que:

Lema 6.7 Todo lenguaje algebraico, IRS, es estrictamente no regular.

Prueba Sea L un lenguaje algebraico IRS, sea $\pi = (\alpha, u, \beta, v, \gamma)$
 un par iterante de dicho lenguaje, entonces $u \neq \epsilon$, $v \neq \epsilon$ y
 $L \cap \alpha u^* \beta v^* \gamma$ no es racional, de lo contrario L no es IRS.



La conexión entre los lenguajes estrictamente no regulares y la al-
 tura del árbol de derivación, viene dada en:

Teorema [15] 6.2 Todo lenguaje estrictamente no racional es un
 lenguaje h-lineal.

Prueba Sea $L \subset X^*$ estrictamente no racional, existe $(\alpha, u, \beta, v, \gamma)$
 con $L \cap \alpha u^* \beta v^* \gamma$ no racional.

$$\text{Sea } \alpha = \alpha_1 \alpha_2 \dots \alpha_p, \quad \alpha_i \in X \quad 1 \leq i \leq p$$

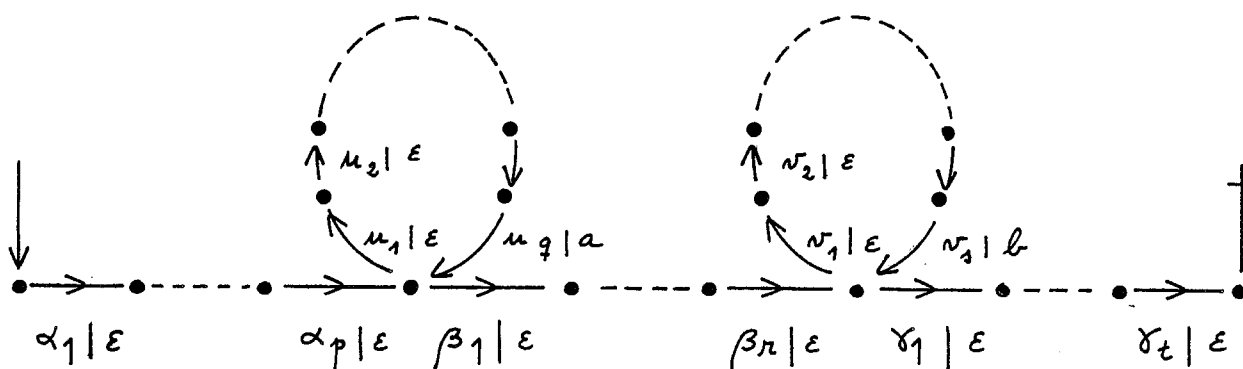
$$u = u_1 u_2 \dots u_q, \quad u_i \in X \quad 1 \leq i \leq q$$

$$\beta = \beta_1 \beta_2 \dots \beta_r, \quad \beta_i \in X \quad 1 \leq i \leq r$$

$$v = v_1 v_2 \dots v_s, \quad v_i \in X \quad 1 \leq i \leq s$$

$$\gamma = \gamma_1 \gamma_2 \dots \gamma_t, \quad \gamma_i \in X \quad 1 \leq i \leq t$$

Consideremos la máquina secuencial generalizada ϵ -limitada (el número de letras consecutivo que borra, está limitado por una constante) $g : X^* \longrightarrow \{a,b\}^*$ siguiente:



tenemos que $g(L) = \{a^i b^j \mid \alpha u^i \beta v^j \gamma \in L\}$

dado que $g(L) \subset a^* b^*$ no es racional es h-lineal y como g es ϵ -limitado L también es h-lineal.

Dicho teorema admite como corolario:

Corolario 6.2 Todo lenguaje IRS o determinista es h-lineal.

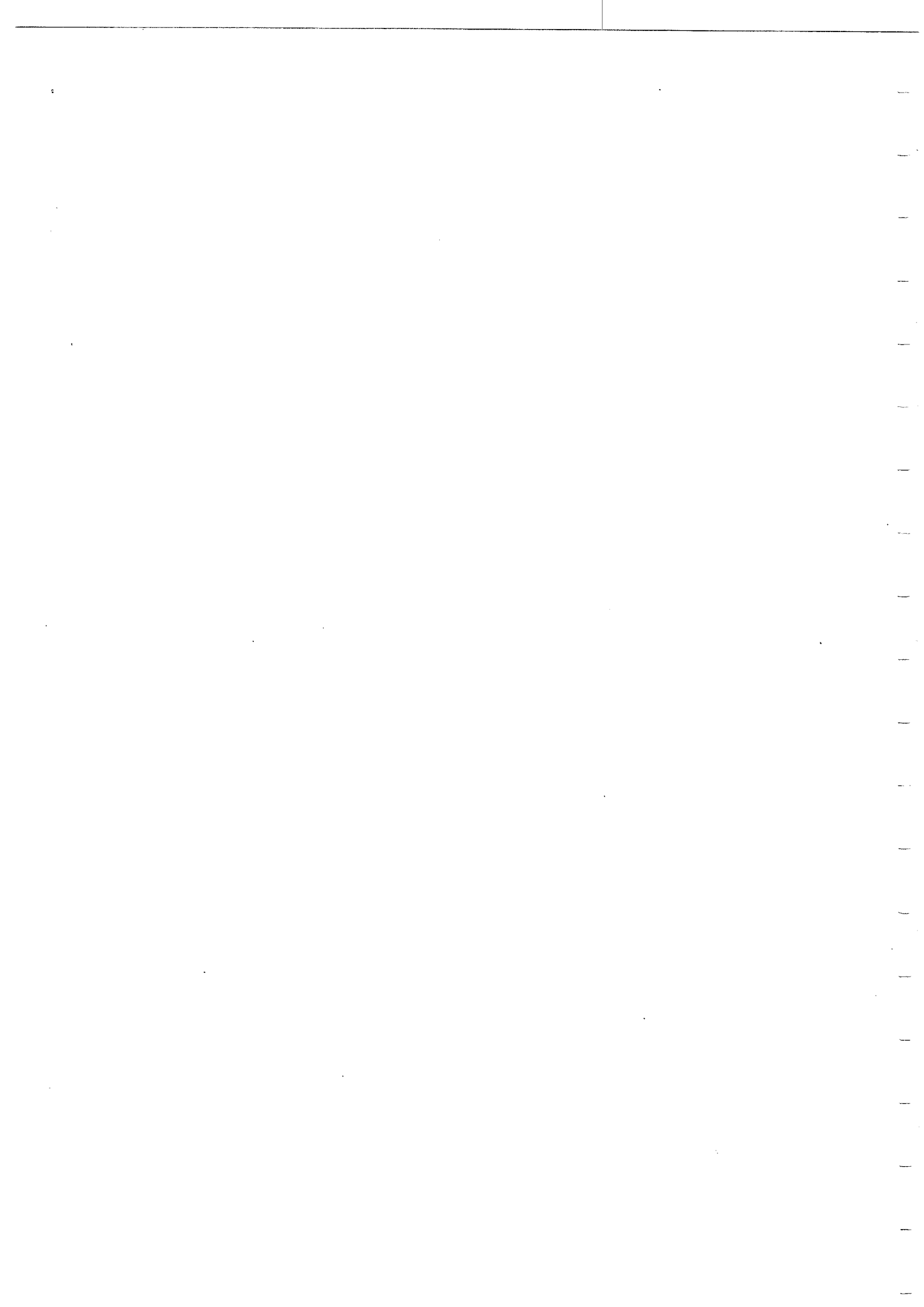
Ejemplo.6.3 Vamos a demostrar finalmente que

$PAL = \{w\tilde{w} \mid w \in \{a,b\}^*\}$ es h-lineal, en efecto, tenemos

(ϵ , ab, ϵ , ba, ϵ) se cumple que $PAL \cap (ab)^* (ba)^* = \{ (ab)^n (ba)^n \mid n \geq 1 \}$. Dado que PAL es estrictamente no racional es h-lineal.

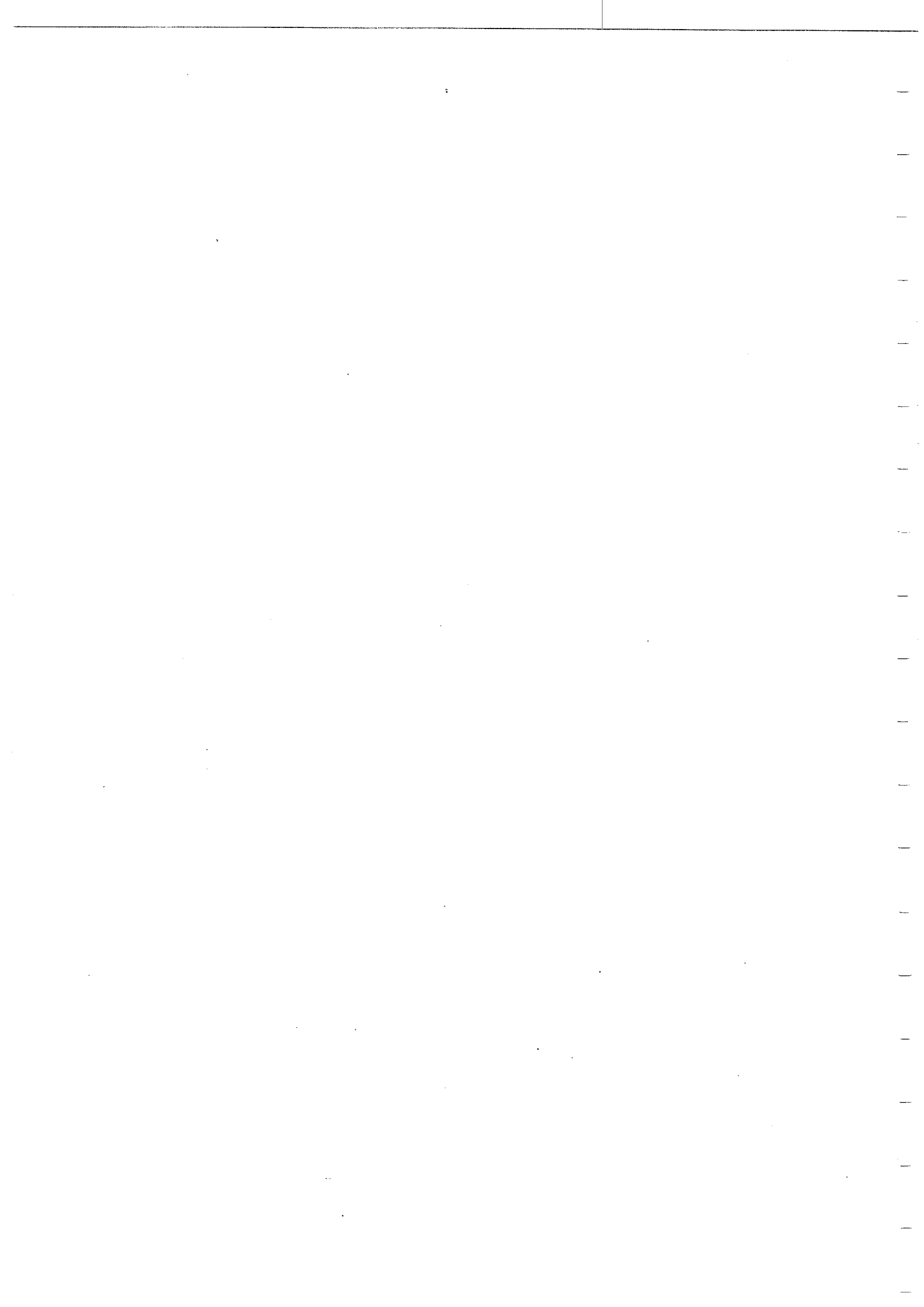
■

Queda abierto el problema de la existencia de lenguajes algebraicos sublineales no racionales; en los capítulos V y VI daremos algunas respuestas a estas cuestiones.



CAPITULO III. INDICE INICIAL, LENGUAJES POLINOMIALES
Y EXPONENCIALES.

- III.1 Índice inicial, definición y propiedades de crecimiento.
- III.2 Ejemplos de cálculo de índices iniciales. Las familias $\text{Pol } \mu$ y $\overline{\text{Exp } \mu}$.
- III.3 Índice inicial y su relación con las operaciones de lenguajes.



Vamos a estudiar en este capítulo una variante del problema de Milnor precedentemente tratado, definiendo una nueva función de complejidad: el índice inicial.

En un primer apartado daremos la definición formal de dicha función de complejidad. El índice inicial es una medida de carácter combinatorio, basada en contar el número mínimo de estados de un autómata finito que reconoce aproximaciones de un lenguaje. Demostraremos que está limitada superiormente por una exponencial, que en sentido largo, es una función creciente y que, si bien localmente puede decrecer, dicha oscilación está acotada.

En un segundo apartado mostraremos las diversas técnicas utilizadas en el cálculo de los índices iniciales de los lenguajes más usuales. Definiremos, tomando como medida el índice inicial, las familias de lenguajes polinomiales y exponenciales a ella asociada. En un primer teorema mostraremos a cual de dichas familias pertenecen los lenguajes más conocidos. En un último apartado estudiaremos el comportamiento del índice inicial con respecto a diversos grupos de operaciones, operaciones racionales, booleanas, de trios, de sustitución y de centro.

Un segundo teorema nos mostrará cómo pasa el índice inicial a través de las transducciones fieles.

Un último teorema nos permitirá situar la familia de los lenguajes polinomiales en el cuadro de las familias clásicas de lenguajes estudiadas por S. Greibach y S. Ginsburg en [29]. En dicho teorema mostraremos que los lenguajes polinomiales son un A.F.L. cerrado por intersección.

Es, sin duda, en este teorema que el índice inicial se muestra como un instrumento útil en el estudio de las familias de lenguajes.

III.1. Índice inicial, definición y propiedades de crecimiento.

En este párrafo vamos a definir una nueva medida de complejidad basada en asociar a todo lenguaje una función sobre los enteros, llamada su índice inicial.

Esta medida es de naturaleza combinatoria, consiste en contar el mínimo número de estados necesarios para reconocer aproximaciones del lenguaje dado.

Veremos, que en el deseo de obtener funciones económicas, debemos restringirnos a autómatas no deterministas. Demostraremos que en un caso concreto el paso del no-determinismo al determinismo puede transformarnos una función polinomial en una exponencial. Analizaremos finalmente las propiedades de crecimiento del índice inicial. Demostraremos que, si bien, en sentido largo es una función creciente, localmente puede presentar oscilaciones. La magnitud de dichas oscilaciones no puede, sin embargo, ser demasiado grande. Damos, para terminar, un lenguaje finito cuyo índice inicial presenta dichas oscilaciones.

Vamos a medir cada lenguaje por una función sobre los enteros. A fin de poderla definir, necesitamos primeramente poder precisar una aproximación al lenguaje que dependa de n , para ello vamos a definir los segmentos iniciales.

Definición 1.1 [12] Sea L un lenguaje, llamaremos n -segmento inicial al conjunto:

$$nL = \{ w \in L \mid |w| \leq n \}$$

Dado que nL es un lenguaje finito podemos hallar un autómata finito A que lo acepte. Notaremos por $\|A\|$ el número de estados de A . Podemos definir una nueva función de complejidad por:

Definición 1.2 Sea L un lenguaje llamamos índice inicial μ_L a la función entera definida por:

$$\mu_L(n) = \min \{ \|a\| \mid nL = L(a) \}$$

■

Notemos que en esta última definición intervienen autómatas no deterministas. Así $\mu_L(n)$ es el mínimo número de estados que precisa un autómata no determinista para reconocer nL . Diremos que un tal autómata es pequeño. Dicho autómata no es único en general. El problema del índice inicial se reduce entonces a encontrar autómatas económicos para un lenguaje dado y demostrar que son pequeños.

Dado que el proceso de determinación incluye un salto exponencial, el considerar autómatas deterministas, no sería interesante si queremos hallar funciones económicas para μ_L .

El siguiente ejemplo muestra uno de tales casos.

Ejemplo 1.1. Sea L el lenguaje sobre $\{0,1\}^* \# \{0,1\}^*$ definido por:

$$L = \{ f \# g \mid |f| = |g| \text{ y } f \neq \tilde{g} \}$$

en la que \tilde{g} es la palabra espejo de g .

. Existe un autómata no determinista para nL con $O(n^3)$ estados
 . El autómata determinista minimal para nL tiene $\Omega(2^{\frac{n}{2}})$ estados

Prueba Evidentemente el lenguaje L es algebraico.

(a) Vamos a demostrar que existe un autómata no determinista polinomial en el número de estados que acepta nL .

Una palabra f de L se puede factorizar en:

$$m = m_1 0 m_2 \# m'_2 1 m'_1$$

$$m = m_1 1 m_2 \# m'_2 0 m'_1$$

donde m_1, m_2, m'_1, m'_2 son palabras de $\{0,1\}^*$ y $|m_1| = |m'_1|$,

$|m_2| = |m'_2|$. Damos un autómata no determinista a_p en la figura III,1, $p = 7$ que mira si 0 ó 1 están en la posición correcta. Dicho autómata reconoce evidentemente

$$(\# + 1 + 0)^{2n+1} \cap (2n + 1)L \text{ y } \|a\| = \sum_{x=1}^n (2x + 1) = \Theta(n^2)$$

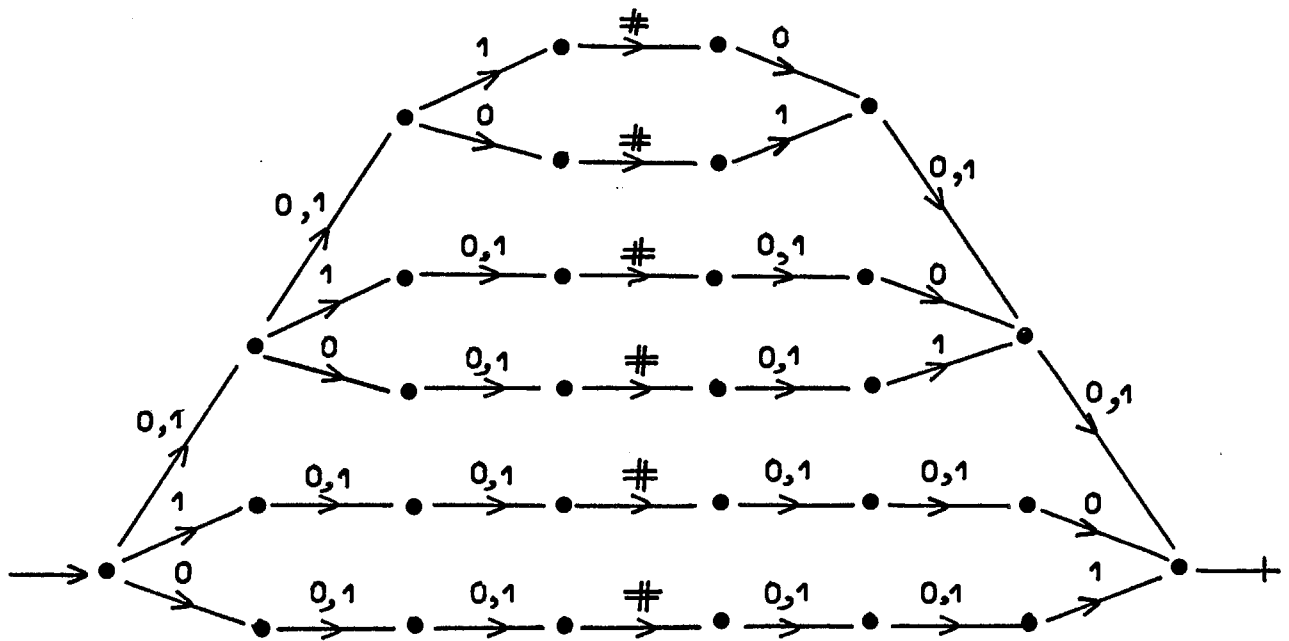


Figura III,1.

(b) Demostremos que cualquier autómata determinista que acepte $(2n + 1)L$ tiene 2^n estados. Sea q_0 el estado inicial de un tal autómata. Demostremos que el conjunto

$$E = \{ q_0 \cdot w\# \mid w \in \{0,1\}^n \}$$

tiene cardinal 2^n .

Supongamos $q_0 \cdot w_1\# = q_0 \cdot w_2\#$ con $w_1 \neq w_2$ entonces

$q_0 \cdot w_1\# \bar{w}_1 \in L$ y $q_0 \cdot w_2\# \bar{w}_1 \notin L$, tenemos pues una contradicción.

Tenemos entonces $\|E\| = \|\{0,1\}^n\| = 2^n$.

Dado que cualquier autómata determinista debe contener el conjunto de estados E llegamos al resultado deseado. ■

Vamos a estudiar seguidamente las propiedades de crecimiento del índice inicial. Tenemos en primer lugar que el índice inicial es como máximo una exponencial, como veremos en el lema siguiente.

Lema 1.1 Sea L un lenguaje en X^* tenemos:

$$\mu_L(n) = O(\|X\|^n)$$

Prueba Sabemos que $nL \subset nX^*$. Consideremos el autómata determinista minimal para nX^* . Dicho autómata $\mathcal{A}_n = \langle X, Q, q_0, Q_+, \delta \rangle$

está dado por:

$$Q = \{ (w) \mid w \in nX^* \}, \quad q_0 = (\varepsilon), \quad Q_+ = \{ (w) \mid w \in X^n \}$$

La función de transición δ está dada por:

$$(w) \cdot x = (wx), \quad w \in (n-1)X^*, \quad x \in X$$

Evidentemente $||Q|| = \sum_{i=0}^n ||X||^i = O(||X||^n)$

Podemos modificar los estados finales de A_n a fin que reconozca nL . Para ello basta tomar como nuevos estados finales el conjunto:

$$\{(w) \mid w \in nL\}$$

Evidentemente el nuevo autómata tiene el mismo número de estados que el precedente. ■

El lema siguiente nos dice que de manera aproximada el índice inicial crece al menos como una función lineal.

Lema 1.2 Sea L un lenguaje infinito en X^* , existe una sucesión $n_i \in \mathbb{N}$, $i = 1, 2, \dots, n, \dots$ que cumple:

$$\mu_L(n_i) \leq n_i + 1$$

Prueba Tomemos una sucesión n_i tal que $n_i L$ contenga al menos una palabra de longitud n_i . Dado que L es infinito y X finito, dicha sucesión debe existir forzosamente. Como cualquier autómata para $n_i L$ debe reconocer las palabras de longitud n_i , y como dicho autómata no tiene bucles, debe tener al menos $n_i + 1$ estados. ■

Aunque $\mu_L(n)$ es una función creciente en sentido largo, puede ser localmente decreciente, pero dicha oscilación no puede ser demasiado grande como vemos en el lema siguiente:

Lema 1.3 Sea L un lenguaje, su índice inicial verifica:

$$\mu_L(n) \leq 2\mu_L(n+1)$$

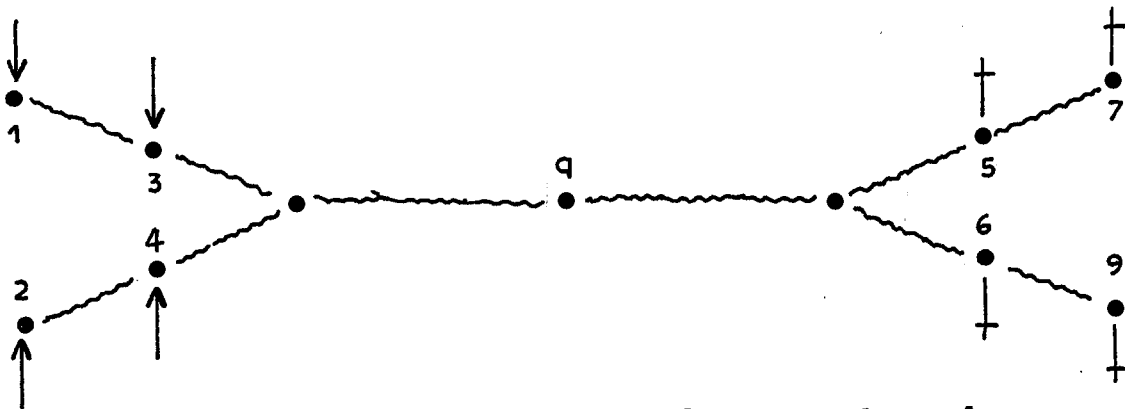
Prueba Consideremos un autómata pequeño para $(n + 1)L$.

Vamos a construir a partir de dicho autómata un autómata para nL .

En el autómata para $(n + 1)L$ hay dos tipos de caminos:

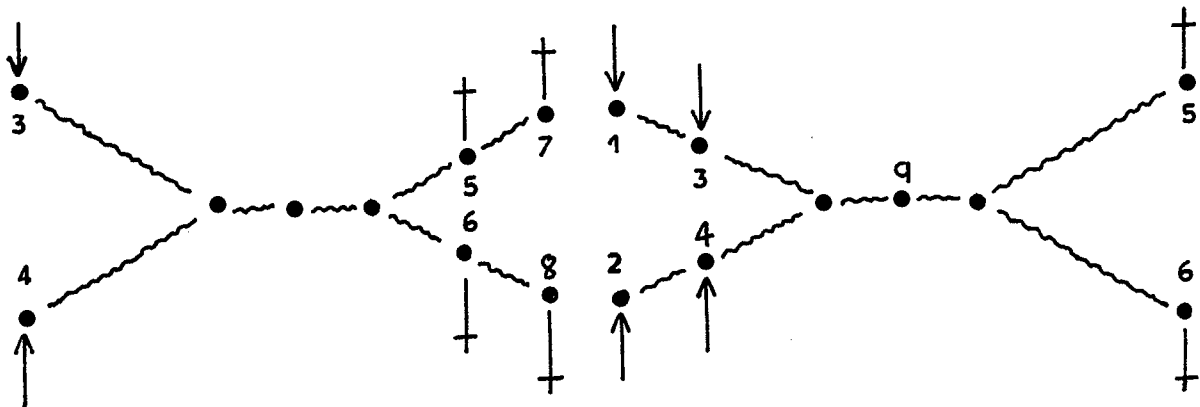
(1) Los caminos de longitud como máximo n , que no están contenidos en caminos de longitud $(n + 1)$. Dichos caminos pasarán sin modificación al autómata para nL .

(2) Los caminos contenidos en un camino de longitud $n + 1$. Supongamos dicha situación esquematizada en la figura siguiente:



Dado que los caminos $[1, q, 7]$, $[1, q, 8]$, $[2, q, 7]$ y $[2, q, 8]$ son de longitud $(n + 1)$, q está a la misma distancia de 1 y 2 y de 7 y 8.

Vamos a desdoblar el esquema siguiente en dos esquemas donde los caminos tienen como máximo longitud n :



Dado que dicho proceso dobla como máximo el número de estados, hemos construido a partir del autómata para $(n + 1)L$ un autómata para nL con un número de estados como máximo doble del precedente. Tenemos por lo tanto:

$$\mu_L(n) \leq 2\mu_L(n + 1)$$

Ejemplo 1.2 Vamos a dar un lenguaje finito cuyo índice inicial decrece. Sea $X = \{x_1, x_2, \dots, x_p\}$ consideremos:

$$L = \{w \in X^* \mid \exists q, r \text{ tales que } w = x_q x_{q+1} \dots x_{q+r}, q+r \leq p\}$$

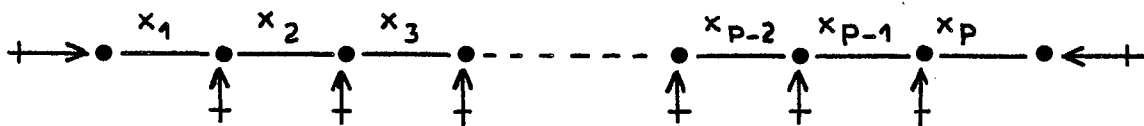
El índice inicial de L es:

$$\mu_{Lp}(1) = 2$$

$$\mu_{Lp}(n) = (p - n)(n - 2) + p + 1; \quad p \leq n \leq 2$$

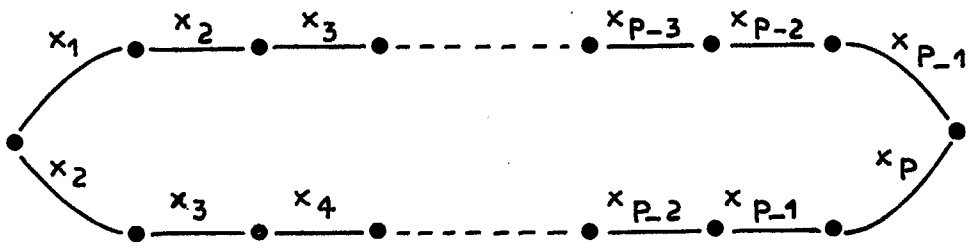
$$\mu_{Lp}(n) = p + 1 \quad ; \quad n > p$$

Prueba Un autómata pequeño para pL es evidentemente:



Evidentemente $\mu_{Lp}(p) = p + 1$

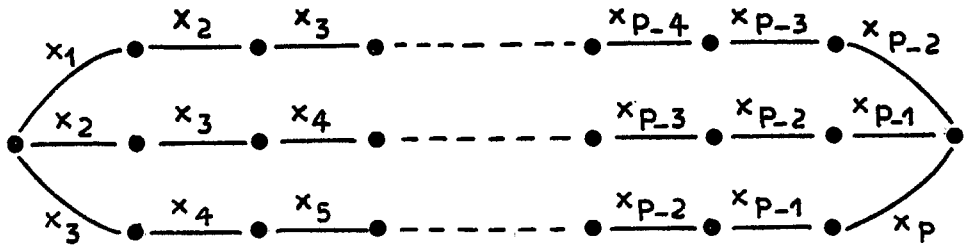
Un autómata pequeño para $(p - 1)L$ es:



Donde todos los estados son iniciales y finales. Evidentemente:

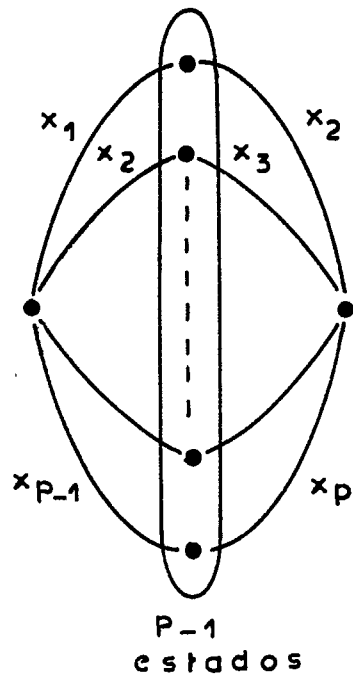
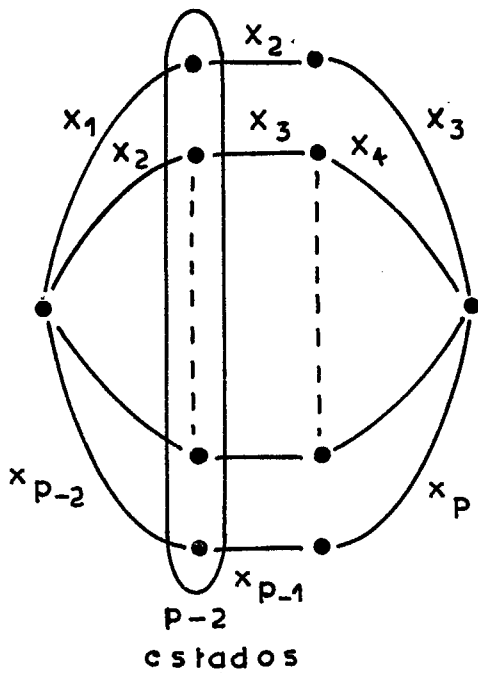
$$\mu_{Lp}(p - 1) = 2(p - 2) + 2$$

Para $(p - 2)Lp$ un autómata pequeño es:



Tenemos $\mu_{Lp}(p - 2) = 3(p - 3) + 2$

Para $3L$ y $2L$ tenemos respectivamente:



Podemos expresar el índice inicial como:

$$\mu_{Lp}(p - t) = (t + 1)(p - t - 1) + 2 = t(p - t - 2) + p + 1$$

Damos explícitamente el cálculo para $p = 6$ y $p = 9$

n	1	2	3	4	5	6
$\mu_{L_6}(n)$	2	7	10	11	10	7

n	1	2	3	4	5	6	7	8	9
$\mu_{L_9}(n)$	2	10	16	20	22	22	20	16	10

III.2. Ejemplos de cálculo de índices iniciales, las familias Pol μ y Exp μ .

En este párrafo damos numerosos ejemplos de cálculo del índice inicial para distintos lenguajes. Se tratan como ejemplos los lenguajes más usuales tanto algebraicos como no algebraicos. El cálculo del índice inicial consiste en dos partes. En una primera parte se darán autómatas económicos para los segmentos ini ciales. En esta parte la dificultad consiste en demostrar que di chos autómatas reconocen efectivamente dichos lenguajes. En una segunda parte analizaremos el número mínimo de estados que son

precisos para procesar los segmentos iniciales. Es de este modo que lograremos hallar autómatas pequeños y calcular el índice inicial. No existe método general para el cálculo del índice inicial, basándose su búsqueda en las propiedades combinatorias de cada lenguaje concreto. En los casos en que existen ideas intuitivas que conduzcan dicho cálculo se ha procurado mostrarlas con el máximo de claridad posible.

Tras esto podemos definir la familia de los lenguajes de índice inicial como máximo polinomial y la familia de los lenguajes de índice inicial como mínimo exponencial.

Para concluir esta sección clasificamos los lenguajes antes estudiados dentro de estas dos familias.

Vamos en el siguiente ejemplo a mostrar con todo detalle como calcular el índice inicial de un lenguaje dado.

Ejemplo 2.1 Consideremos el lenguaje $S_1 = \{ a^n b^n \mid n \geq 0 \}$

El cálculo de su índice inicial consta de dos etapas.

(a) En una primera etapa damos un autómata con pocos estados que acepte nS_1 , con esto obtenemos un límite superior al índice inicial.

Notemos que para este lenguaje tenemos $(2n + 1)S_1 = 2nS_1, n \geq 0$.

Consideremos el autómata A_n dado en la figura III.2.

$$\mu_{S_1}(n) = \text{Si } n = \text{impar entonces } n \text{ sino } n + 1$$

que podemos reescribir como $\mu_{S_1}(n) \leq 2 \lfloor \frac{n}{2} \rfloor + 1$

(b) En esta segunda etapa vamos a obtener un límite inferior al índice inicial contando el número mínimo de estados que todo autómata pequeño para $2nS_1$ debe tener.

En este caso concreto esto es de una extrema sencillez. Dado que $2nS_1$ es un lenguaje finito todo autómata pequeño que lo acepte no debe tener bucles. Como dicho autómata debe aceptar la palabra $a^n b^n$ necesita al menos $2n + 1$ estados. Tenemos entonces que:

$$\mu_{S_1}(n) \geq 2 \lfloor \frac{n}{2} \rfloor + 1$$

Considerando conjuntamente (a) y (b) tenemos:

$$\mu_{S_1}(n) = 2 \lfloor \frac{n}{2} \rfloor + 1, \quad n \geq 0$$

En general lo que nos interesará del índice inicial será su comportamiento asintótico, en este caso tenemos:

$$\mu_{S_1}(n) = \Theta(n)$$

(c) Notemos que pueden existir varios autómatas pequeños para un mismo lenguaje. En la figura III.3 damos un autómata pequeño no determinista para $2nS_1$.

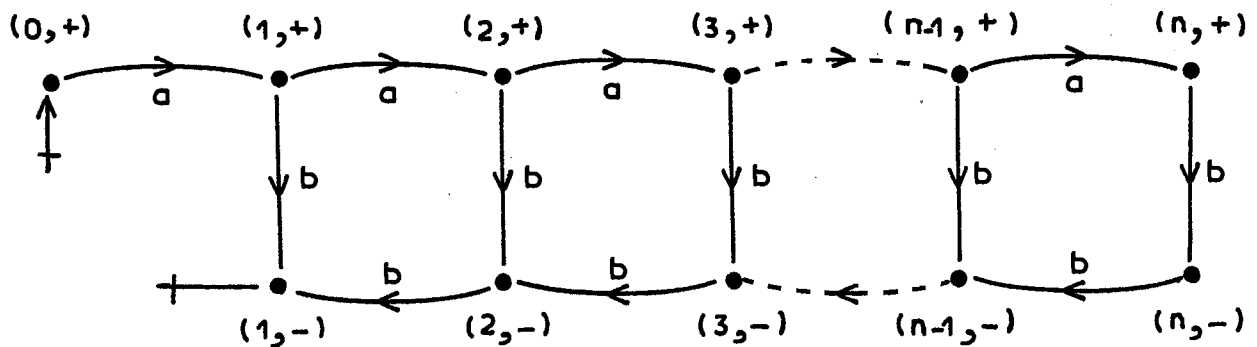


Figura III.2

Dicho autómata está descrito por $A_n = \langle \{a, b\}, Q, q_0, Q_+, \delta \rangle$

donde:

El conjunto de estados es $Q = \{(x, +) \mid 0 \leq x \leq n\} \cup \{(y, -) \mid 1 \leq y \leq n\}$

El estado inicial q_0 es $(0, +)$

Los estados finales son $Q_+ = \{(0, +), (1, -)\}$

La función de transición δ es:

$$(x, +). a = (x + 1, +), \quad 0 \leq x < n$$

$$(x, +). b = (x, -), \quad 0 < x \leq n$$

$$(x, -). b = (x - 1, -), \quad 0 < x \leq n$$

Por construcción las únicas palabras procesadas son de la forma: $w = a^p b^q$, $p \leq n$, $p \geq q$, en este caso tenemos:

$$(0, +). a^p b^q = (p - q, \text{Si } q = 0 \text{ entonces } + \text{ sino } -).$$

Evidentemente $L(A_n) = 2nS_1$ y $\|A_n\| = 2n + 1$

Tenemos entonces que todo autómata pequeño para $2nS_1$ tiene al menos $2n + 1$ estados, en consecuencia:

$$\mu_{S_1}(2n) = \mu_{S_1}(2n + 1) \leq 2n + 1, \quad n \geq 0$$

Dicha ecuación puede escribirse:

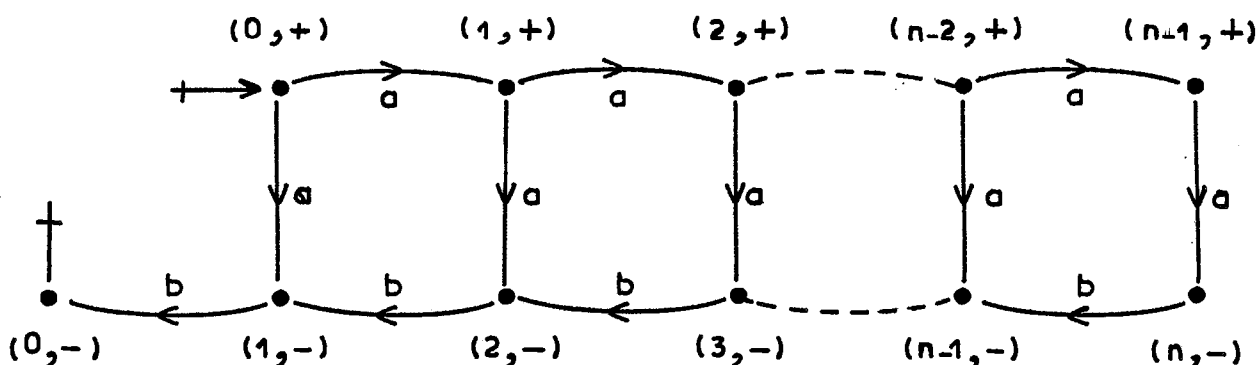


Figura III.3.

La definición formal de dicho autómata es:

$$\mathcal{B}_n = \langle \{a, b\}, Q, q_-, q_+, \delta \rangle$$

$$Q = \{ (x, s) \mid 0 \leq x \leq n-1, s \in \{+, -\} \} \cup \{ (n, -) \}$$

$$q_- = (0, +), \quad q_+ = \{ (0, +), (0, -) \}$$

La función de transición δ es:

$$(x, +). a = \{ (x+1, +), (x+1, -) \}, \quad 0 \leq x \leq n-2$$

$$(n-1, +). a = (n, -)$$

$$(x, -). b = (x-1, -), \quad 1 \leq x \leq n$$

Las palabras $a^p b^p$, $p < n$ son procesadas anticipando de manera no determinista la última a , i.e:

$$\begin{aligned} (0, +). a^p b^p &= (1, +). a^{p-1} b^p = (2, +). a^{p-2} b^p = (p-1, +). a b^p = \\ &= (p, -) b^p = (p-1, -) b^{p-1} = (0, -) \end{aligned}$$

(d) Queda la cuestión de saber si hay algún modo canónico de obtener los autómatas \mathcal{A}_n y \mathcal{B}_n . En este caso la respuesta es afirmativa, dichos autómatas pueden obtenerse como intersección de autómatas a pila con autómatas finitos.

Obtención de \mathcal{A}_n . Sea el p.d.a. siguiente: $M_1 = \langle X, Z, Q, q, z, T \rangle$

El alfabeto de lectura es $X = \{a, b\}$, el de pila es $Z = \{z, a\}$,

el fondo de pila es z , el conjunto de estados $Q = \{q, +, -\}$
 el estado inicial q , y la función de transición:

$$(q, z). a = (+, za)$$

$$(+, a). a = (+, aa)$$

$$(+, a). b = (-, \epsilon)$$

$$(-, a). b = (-, \epsilon)$$

Dicho autómata reconoce por fondo de pila S_1 .

Consideremos el autómata R_{2n} , la figura III.4 lo dá para $n = 3$

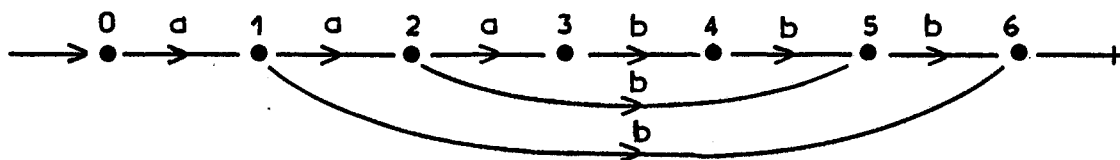


Figura III.4

Veamos que $M_1 \cap R_{2n} = A_n$, notaremos los estados de $M_1 \cap R_{2n}$

por $[(\text{estado de } M_1, \text{pila}), \text{estado de } R_6]$, la figura III.5 realiza tal intersección para $n = 3$.

Veamos que en este caso tanto el autómata a pila como el autómata finito que cuenta las distancias son deterministas.

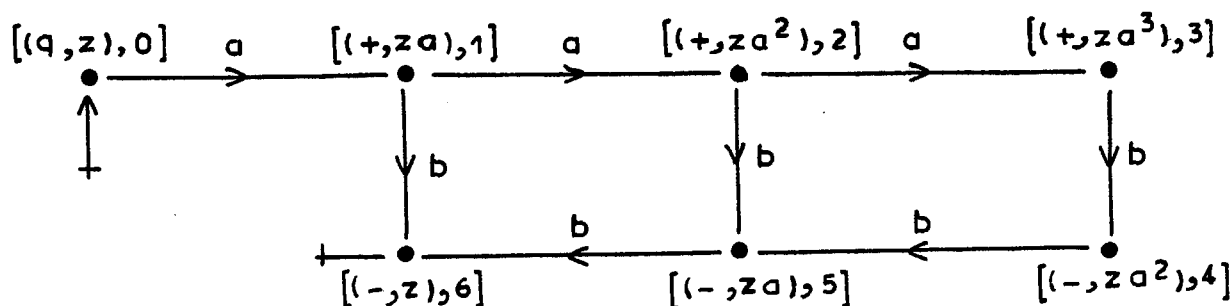


Figura III.5.

Obtención de \mathcal{B}_n . Para este caso necesitaremos encontrar un pda y un autómata no deterministas. Sea M_2 un autómata a pila cuya función de transición es:

$$\begin{aligned} (q, z). a &= \{ (+, za), (-, za) \} \\ (+, a). a &= \{ (+, aa), (-, aa) \} \\ (-, a). b &= (-, \epsilon) \end{aligned}$$

dicho autómata reconoce por fondo de pila S_1 .

Consideremos el autómata C_{2n} , que en el caso $n = 3$ está dado en la figura III.6

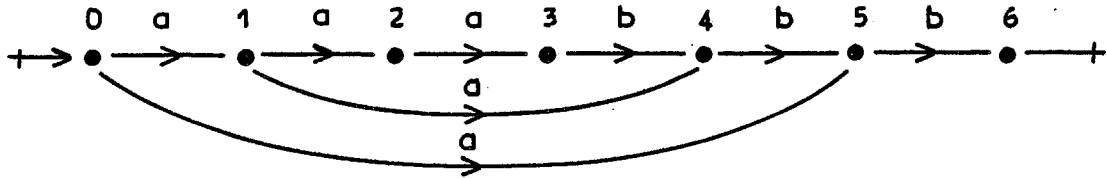


Figura III.6

En este caso tenemos $M_2 \cap C_{2n} = \mathcal{B}_n$, la figura III.7 lo dá para $n = 3$.

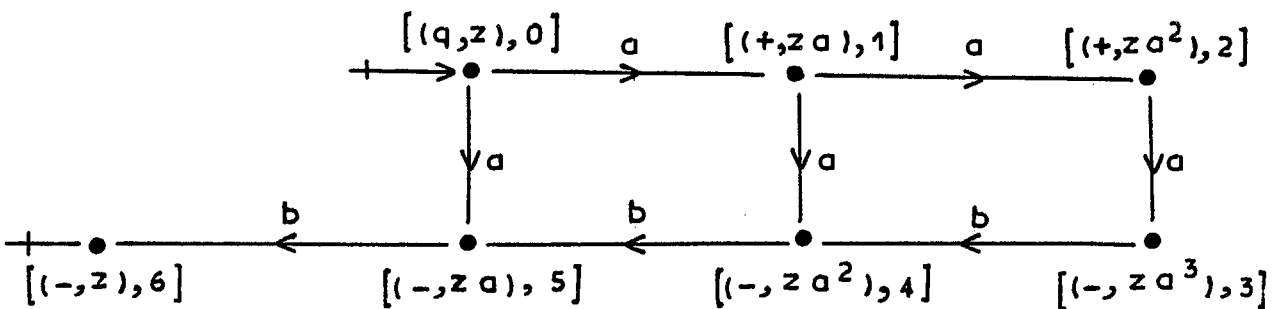


Figura III.7

(e) Considerando conjuntamente autómatas A_n y B_n podemos obtener otros tipos de autómatas, por ejemplo el autómata esquematizado en la figura III.8.

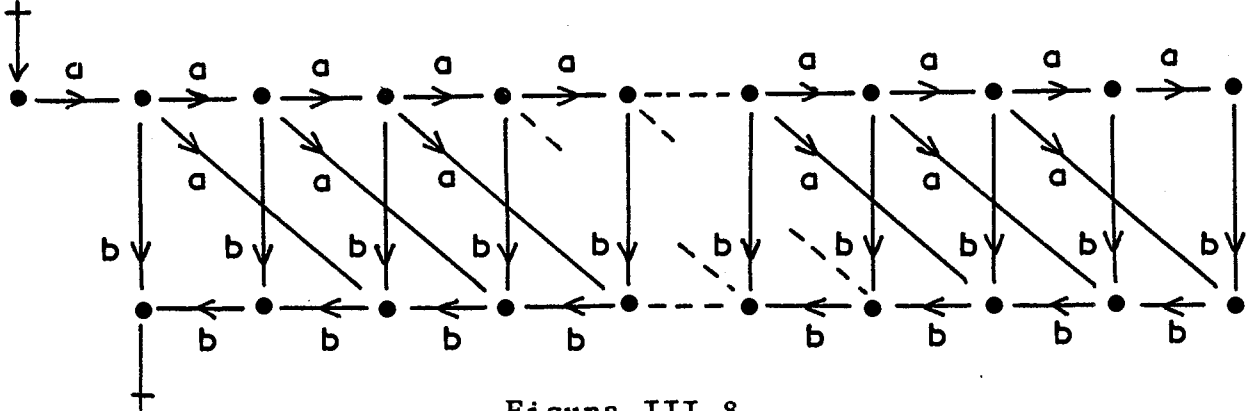


Figura III.8.

■

Ejemplo 2.2 Sea el lenguaje $S_{\geq} = \{ a^n b^p \mid n \geq p \}$, consideremos el siguiente autómata, figura III.9.

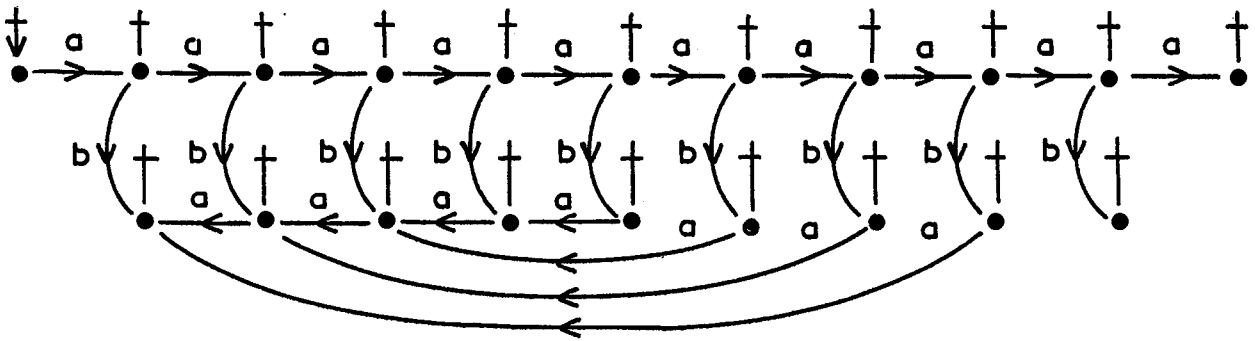


Figura III.9.

evidentemente $\mu_{S_{\geq}}(n) = \Theta(n)$

Ejemplo 2.3 Sea el lenguaje $S_{\leq} = \{ a^n b^p \mid n \leq p \}$, consideremos

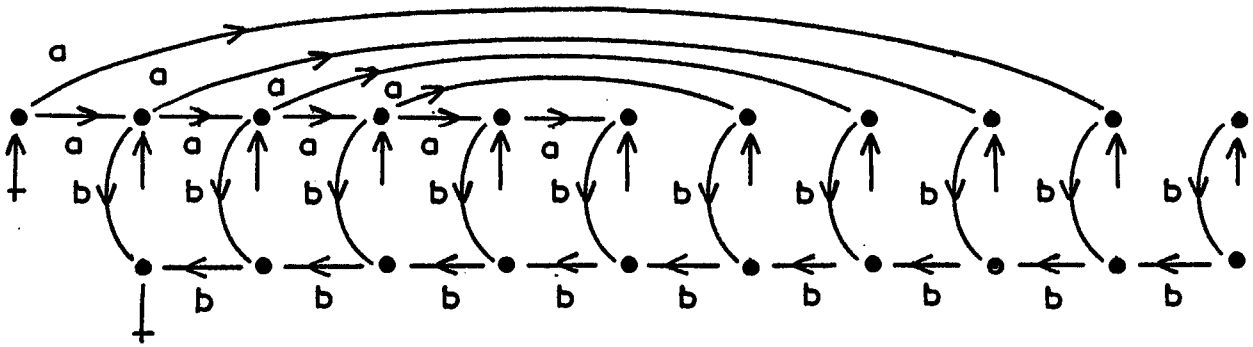


Figura III.10.

evidentemente $\mu_S(n) = \Theta(n)$

Ejemplo 2.4 Consideremos ahora el lenguaje $L = \{a^n b^n c^n \mid n \geq 0\}$

(a) Vamos a dar un límite inferior al índice inicial.

Notemos que $3nL = (3n + 1)L = (3n + 2)L$. Consideremos el autómata

A_n , para $n = 18$ dado en la figura siguiente.

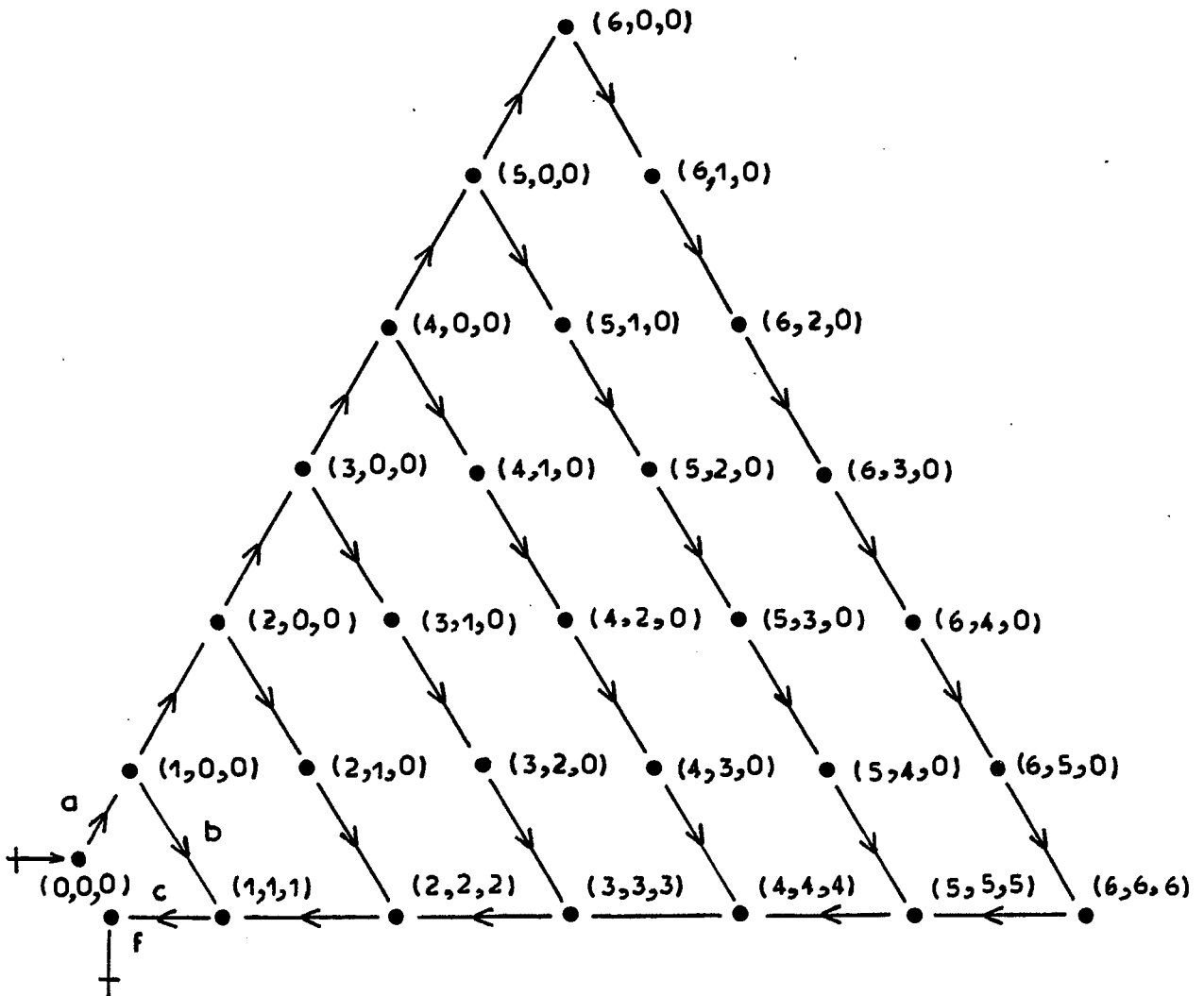


Figura III.11.

Dicho autómata puede ser descrito por $A_n = \langle \{a, b, c\}, Q, q_-, q_+, \delta \rangle$

$$Q = \{ (x, y, 0) \mid 0 \leq x \leq n; 0 \leq y \leq n-1 \} \cup \{ (x, x, x) \mid 1 \leq x \leq n \} \cup \{f\}$$

$$q_- = (0, 0, 0) \quad , \quad q_+ = f$$

La función de transición δ está dada por:

$$(x, 0, 0). a = (x + 1, 0, 0) \quad ; \quad 0 \leq x \leq n - 1$$

$$(x, 0, 0). b = (x, 1, 0) \quad ; \quad 1 \leq x \leq n$$

$$(x, y, 0). b = (x, y + 1, 0) \quad ; \quad 0 \leq y \leq x - 2$$

$$(x, x - 1, 0). c = (x, x, x) \quad ; \quad 1 \leq x \leq n$$

$$(x, x, x). c = (x - 1, x - 1, x - 1) \quad ; \quad 2 \leq x \leq n$$

$$(1, 1, 1). c = f$$

Evidentemente $L(A_n) = 3nL$

Notemos que $||A_n|| = (1 + 2 + \dots + n) + 1 = \Theta(n^2)$

Tenemos entonces que $\mu_L(n) = O(n^2)$

(b) Vamos a encontrar un límite inferior al índice inicial.

Para ello vamos a hallar un límite inferior al número de estados de cualquier autómata pequeño reconociendo $3nL$.

Notemos que dichos autómatas son coacesibles (a partir de cualquier estado podemos llegar a un estado final).

Sea $n\mathcal{A} = \langle \{a, b, c\}, Q, q_-, q_+, \delta \rangle$ un autómata pequeño para nL .

Sean $q_1, q_2 \in Q_-$, $1 \leq x \leq n$, $0 \leq y, z \leq x$ vamos a demostrar

que $\delta(q_1, a^x b^y) \neq \delta(q_2, a^x b^z)$ siempre que $y \neq z$. Razonando por el absurdo supondremos que $\delta(q_1, a^x b^y) = \delta(q_2, a^x b^z) = q$

Dado que q es coacesible la situación puede esquematizarse en la figura III.12.

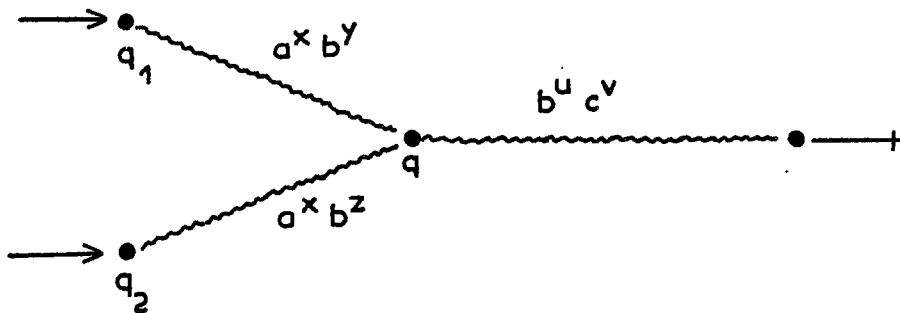


Figura III.12

tenemos que $x = y + u = z + u = v$ entonces $y = z$, tenemos pues una contradicción. Consideremos para cada $1 \leq x \leq n$ el conjunto $E_x = \{ \delta(q, a^x b^y) \mid q \in Q_-, 0 \leq y \leq x \}$, entonces $\|E_x\| = x + 1$. Además dichos conjuntos son disjuntos $E_x \cap E_{x'} = \emptyset$ si $x \neq x'$.

Prueba evidente. Entonces $\|n \mathcal{L}\| \leq \sum_{x=1}^n \|E_x\| = \Theta(n^2)$

Tenemos entonces que $\mu_L(n) = \Theta(n^2)$. ■

Ejemplo 2.5 Lenguaje de Goldstine [4], definido por

$$\text{GOLD} = \{ a^{i_1} b a^{i_2} b \dots a^{i_p} b \mid p \leq 1; \exists j: 1 \leq j \leq p, i_j \neq j \}$$

Podemos dar una definición alternativa de dicho lenguaje a partir de los factores izquierdos de una palabra infinita.

La palabra infinita de Goldstine $g: \mathbb{N}_+ \longrightarrow \{a, b\}$ está definida por:

$$g\left(\frac{1}{2}n(n+3)\right) = b, \quad n \geq 1$$

$$g\left(\frac{1}{2}n(n+3) + m\right) = a, \quad n \geq 1, 1 \leq m \leq n+1$$

Notaremos $g(n) = g_n$ y $g = g_1 g_2 \dots g_n \dots$. Observemos que:

$$g = ab a^2 b a^3 b \dots a^n b a^{n+1} b \dots$$

Tenemos que $\text{GOLD} = (a^* b)^+ \setminus \text{FG}(g)$

Definamos el factor izquierdo n por $f_n = ab a^2 b \dots a^n b$, tenemos

$$|f_n|_b = n, \quad |f_n| = \frac{1}{2} n(n+3)$$

(a) A partir de las consideraciones precedentes vamos a definir un autómata económico para GOLD. Remarquemos que:

$$\frac{1}{2} n(n+3) \text{GOLD} = \frac{1}{2} n(n+3) (a^* b)^+ \setminus \text{FG}(ab a^2 b \dots a^n b)$$

Sea $A_n = \langle \{a, b\}, Q, q_-, q_+, \delta \rangle$ definido por:

$$Q = \left\{ (x, +) \mid 0 \leq x \leq \frac{1}{2} n(n+3) \right\} \cup \left\{ (x, -) \mid 0 \leq x \leq \frac{1}{2} n(n+3) - 1 \right\}$$

$$q_- = (0, +), \quad q_+ = \left(\frac{1}{2} n(n+3), + \right)$$

La función de transición está definida por:

$$(x, +) \cdot u = \begin{cases} (x+1, +), u = g_{x+1}, & 0 \leq x < \frac{1}{2} n(n+3) \\ (x+1, -), u = a \neq g_{x+1}, & 0 \leq x < \frac{1}{2} n(n+3) - 2 \\ \left\{ (x+1, -), q_+ \right\}, u = b \neq g_{x+1}, & 0 \leq x < \frac{1}{2} n(n+3) - 1 \end{cases}$$

$$(x, -) \cdot a = (x+1, -), \quad 0 \leq x < \frac{1}{2} n(n+3) - 2$$

$$(x, -) \cdot b = \left\{ (x+1, -), q_+ \right\}, \quad 0 \leq x < \frac{1}{2} n(n+3) - 1$$

La figura III.13 da una muestra de A_n , $n = 5$.

Vamos a demostrar que A_n reconoce $\frac{1}{2} n(n+3)$ GOLD.

Para ello veamos que el lenguaje Goldstine puede escribirse

como:

$$\text{GOLD} = \left\{ ab a^2 b a^3 b \dots a^i b a^j b (a^* b)^* \mid j \neq i+1, i_j \geq 1 \right\}$$

Notemos que $(a^* b)^* = b + (a + b)^* b$. Al bloque $a^{i_j} b$ le llamaremos "mala transición". El autómata A_n fabrica a partir del factor izquierdo $f_n = ab a^2 b \dots a^n b$ todas las malas transiciones posibles manteniendo las condiciones de longitud. Dicho proceso puede esquematizarse en la figura III.14:

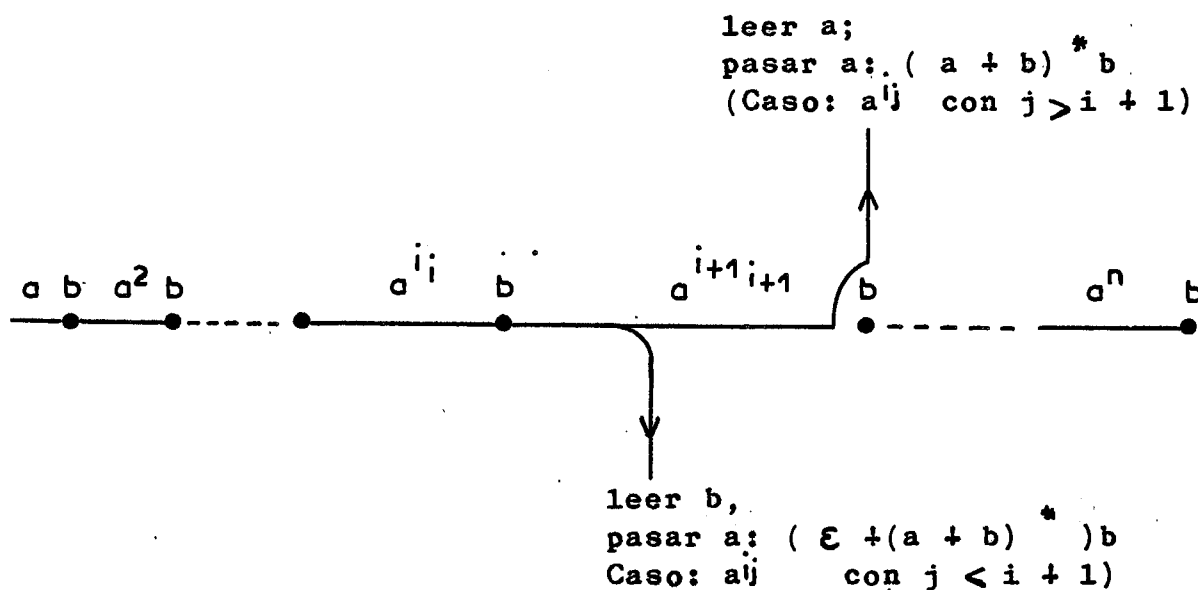


Figura III.14.

Evidentemente $\|a_n\| = \Theta(n^2)$ con lo que $\mu_{\text{GOLD}}(n) = O(n)$

(b) Dado que todo autómata pequeño para $\frac{1}{2} n(n+3)$ GOLD debe reconocer la palabra $ab a^2 b \dots a^{n-1} b a^{n-1} b^2$ de longitud $\frac{1}{2} n(n+3)$ dicho autómata contiene al menos n^2 estados con lo que $\mu_{\text{GOLD}}(n) = \Omega(n)$. Por lo que $\mu_{\text{GOLD}}(n) = \Theta(n)$.

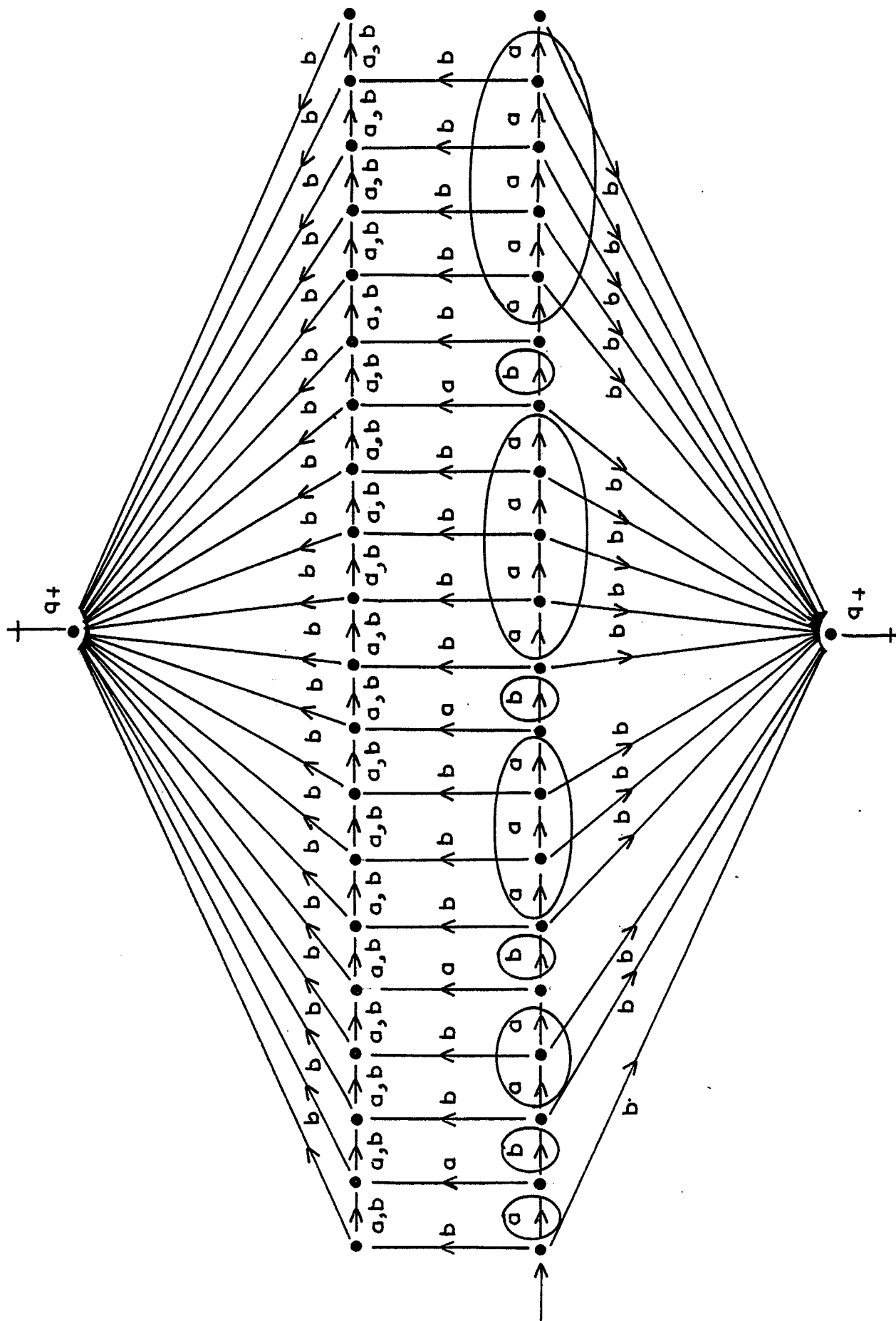


Figura III.13.

Ejemplo 2.6 Vamos a calcular el índice inicial del lenguaje simétrico de dos letras $S_2 = \text{SYM}_2 = \langle S \rightarrow aS\bar{a} + bS\bar{b} + \epsilon \rangle$

(a) Límite superior. Evidentemente $2n \text{SYM}_2 = (2n + 1)\text{SYM}_2$.

Sea $A_n = \langle \{a, b, \bar{a}, \bar{b}\}, Q, q_-, q_+, \delta \rangle$ el autómata definido

por:

$$Q = \{ (w, s) \mid w \in (n-1)\{a, b\}^*, s \in \{+, -\} \} \cup \{ (w, +) \mid w \in \{a, b\}^n \}$$

$$q_- = (\epsilon, +), \quad q_+ = \{ (\epsilon, +), (\epsilon, -) \}$$

La función de transición δ está definida por:

$$(w, +) \cdot u = (wu, +), \quad u \in \{a, b\}, \quad |w| \leq n-1$$

$$(wu, +)\bar{u} = (w, -), \quad u \in \{a, b\}, \quad \bar{u} \text{ copia barrada de } u$$

$$(wu, -)\bar{u} = (w, -)$$

La figura III.15 muestra A_n .

Veamos un cálculo válido en A_n . Sea $m = u_1 u_2 \dots u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1$,

$p \leq n$, tenemos:

$$(\epsilon, +) \cdot m = (\epsilon, +) \cdot u_1 u_2 \dots u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 = (u_1, +) \cdot u_2 u_3 \dots u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 =$$

$$= (u_1 u_2, +) \cdot u_3 \dots u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 = (u_1 u_2 \dots u_p, +) \cdot \bar{u}_p \dots \bar{u}_2 \bar{u}_1 =$$

$$= (u_1 u_2 \dots u_{p-1}, -) \cdot \bar{u}_{p-1} \dots \bar{u}_2 \bar{u}_1 = (u_2 u_1, -) \cdot \bar{u}_2 \bar{u}_1 = (u_1, -) \cdot \bar{u}_1 = (\epsilon, -)$$

Se verifica que A_n reconoce $2n \text{SYM}_2$.

Vamos a estimar el número de estados de A_n . Sabemos que en

$\{a, \bar{a}, b, \bar{b}\}$ hay 2^p palabras distintas de longitud p , es decir

$$\|\{a, \bar{a}, b, \bar{b}\}^n\| = 2^p.$$

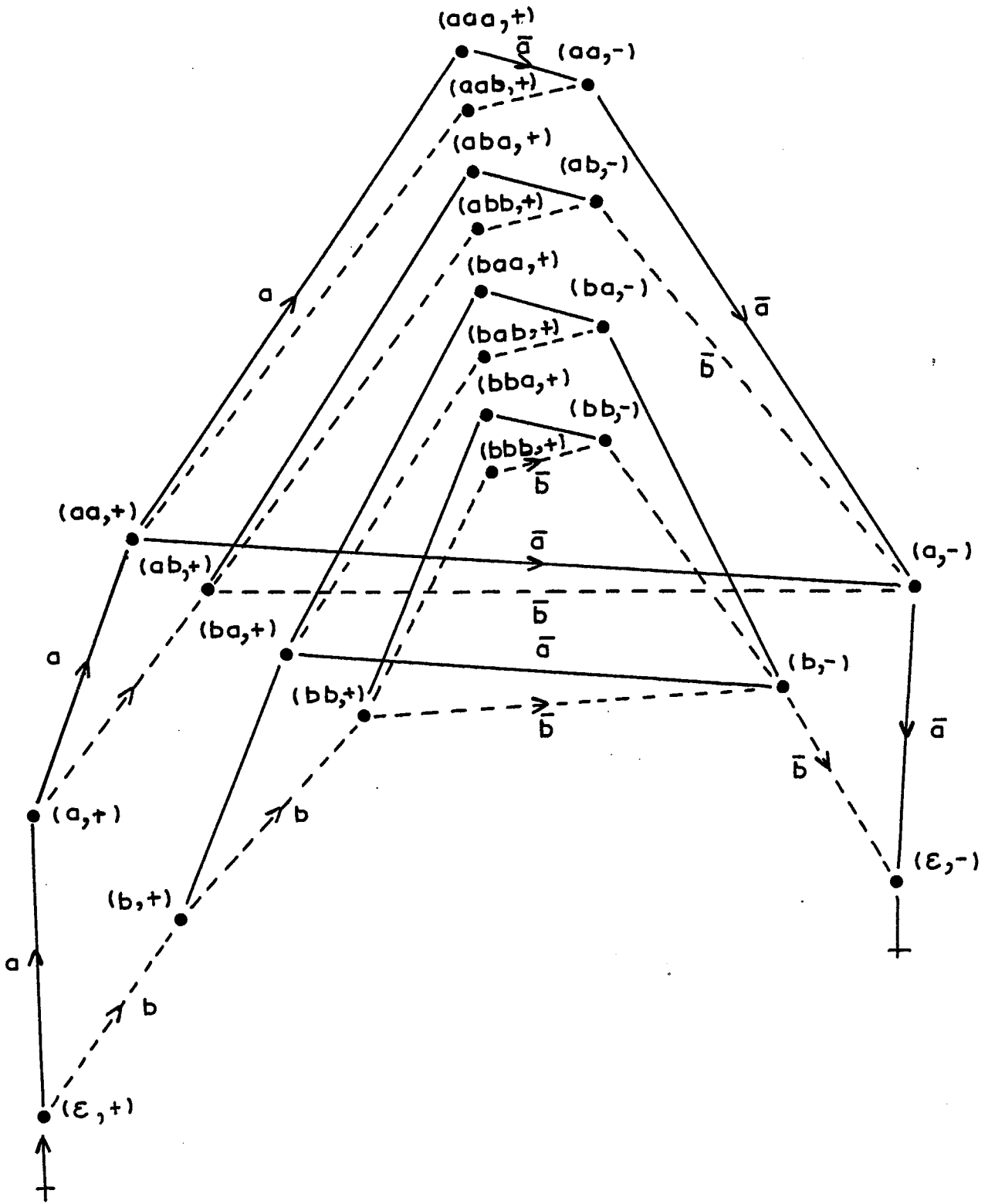


Figura III.15.

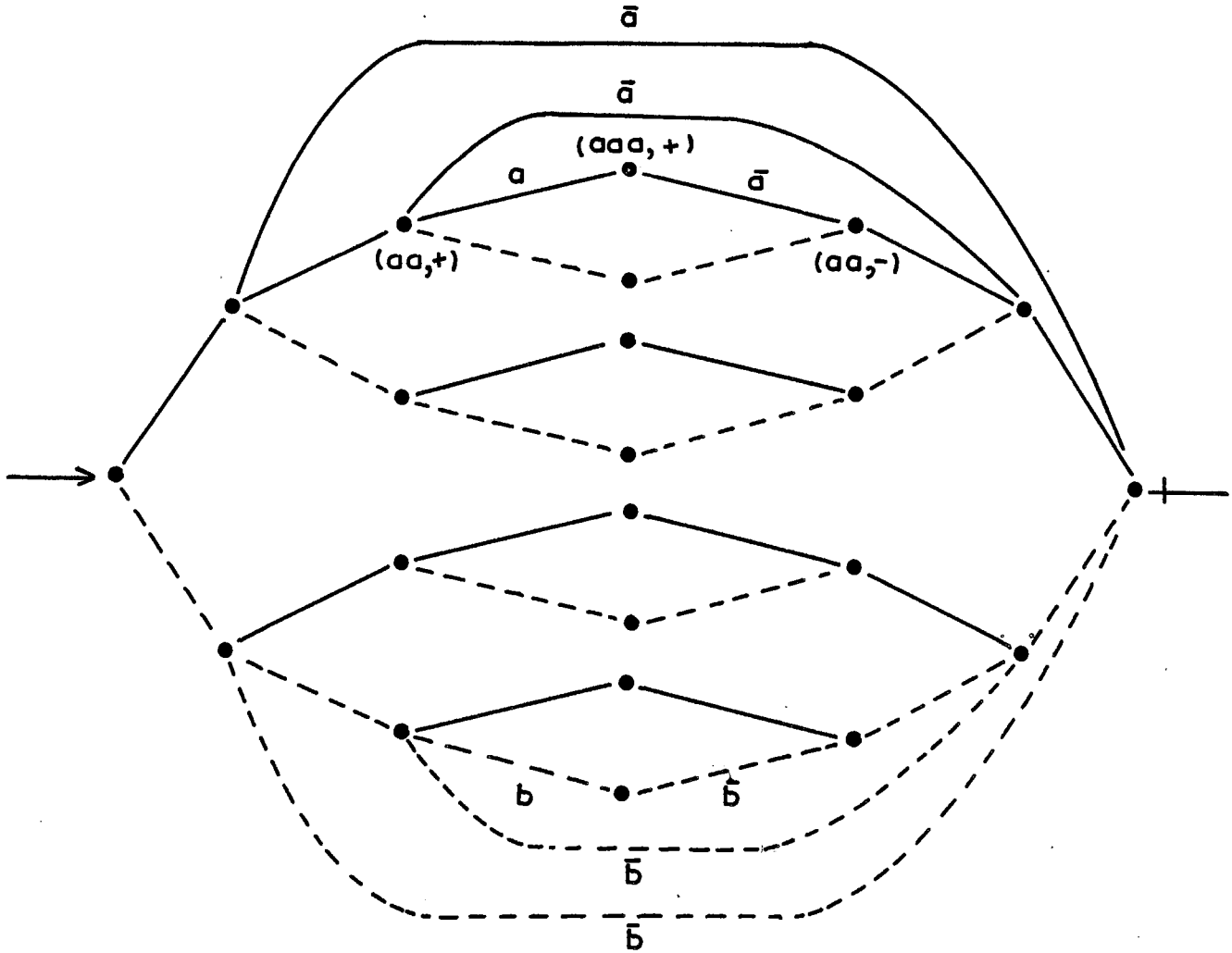


Figura III.15. bis

Entonces:

$$Q = 2 \cdot \sum_{p=0}^{n-1} \left(\left\| \{ a, \bar{a}, b, \bar{b} \}^p \right\| + \left\| \{ a, \bar{a}, b, \bar{b} \}^n \right\| \right)$$

$$= 2 \sum_{p=0}^{n-1} 2^p + 2^n = \Theta(2^n).$$

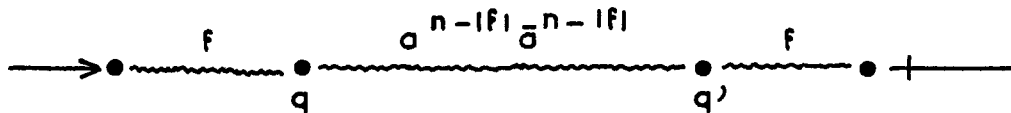
El índice inicial satisface $\mu_{SYM_2}(n) \leq O(2^{\frac{n}{2}})$

(b) Límite inferior. Sea h un factor izquierdo de SYM_2 , h se factoriza de modo único en $h = u_1 u_2 \dots u_p u_{p+1} \dots u_r \bar{u}_r \dots \bar{u}_{p+1} = fg\bar{g}$,
 $f = u_1 u_2 \dots u_p$, $g = u_{p+1} \dots u_r$, $\bar{g} = \bar{u}_r \dots \bar{u}_{p+1}$. Vamos a llamar a la

palabra f el peso de h y la notaremos por $p(h)$.

Si consideramos el conjunto $2n\text{SYM}_2$ vemos que su conjunto de pesos es $n \{ a, b \}^*$. Sea $2n \mathcal{S}ym_2$ un autómata pequeño para $2n\text{SYM}_2$, dada su coaccesibilidad tenemos que a todo estado corresponde un único peso, podemos entonces hablar del peso de un estado dado.

Consideremos los pares de palabras $f, fa^{n-|f|} \bar{a}^{n-|f|} \bar{f}$, esquematisado en:



los estados q y q' son distintos dada la falta de bucles y tienen el mismo peso. Tenemos entonces que a todo peso de longitud menor que n corresponden al menos dos estados.

En consecuencia:

$$2n \mathcal{S}ym_2 \geq 2 \left\| (n-1) \{ a, b \}^* \right\| + \left\| \{ a, b \}^n \right\| = 2 \sum_{p=0}^{n-1} 2^p + 2^n$$

y el autómata dado anteriormente es pequeño.

Considerando (a) y (b) conjuntamente tenemos $\mu_{\text{SYM}_2}^{(n)} = \Theta(2^{\frac{n}{2}})$

(c) Tampoco en este caso existe un único autómata pequeño que acepta $2n \text{SYM}_2$. Consideremos el autómata no determinista:

$$\mathcal{B}_n = \langle \{ a, b, \bar{a}, \bar{b} \}, Q, q_-, q_+, \delta \rangle$$

$$Q = \{ (w, s) \mid w \in (n-1) \{ a, b \}^*, s \in \{ +, - \} \} \cup \{ (w, -) \mid w \in \{ a, b \}^n \}$$

$$q_- = (\epsilon, +), \quad q_+ = \{ (\epsilon, +), (\epsilon, -) \}$$

La función de transición está dada por:

$$(w, +).u = \{ (wu, +), (wu, -) \}, \quad |w| \leq n-2$$

$$(w, +).u = (wu, -), \quad |w| = n-1$$

$$(wu, -).u = (w, -)$$

La figura III.16 muestra 6 \mathcal{B}_n .

En este autómata hay que anticipar de manera no determinista la última letra no barrada y realizar la transición de + a -.

Consideremos, a modo de ejemplo, $f = u_1 u_2 \dots u_{p-1} u_p \bar{u}_p \bar{u}_{p-1} \dots \bar{u}_2 \bar{u}_1$.

$$\begin{aligned} (\varepsilon, +) \cdot u_1 u_2 \dots u_{p-1} u_p \bar{u}_p \bar{u}_{p-1} \dots \bar{u}_2 \bar{u}_1 &= (u_1, +) \cdot u_2 \dots u_{p-1} u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 = \\ &= (u_1 u_2 \dots u_{p-1}, +) \cdot u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 = (u_1 u_2 \dots u_{p-1} u_p, -) \bar{u}_p \dots \bar{u}_2 \bar{u}_1 = \\ &= (u_1 u_2 \dots u_{p-1}, -) \bar{u}_{p-1} \dots \bar{u}_2 \bar{u}_1 = (u_2 u_1, -) \cdot \bar{u}_2 \bar{u}_1 = (\varepsilon, -) \end{aligned}$$

Evidentemente $\|\mathcal{A}_n\| = \|\mathcal{B}_n\|$

Comentario. Las construcciones precedentes admiten una generalización evidente para SYM_p , $p \in \mathbb{N}$. Si $p > 1$ obtenemos sin dificultad que:

$$\mu_{\text{SYM}_p}(n) = \Theta\left(p^{\frac{n}{2}}\right)$$

En el caso $p = 1$ tenemos que $\text{SYM}_1 = S_1$ y los autómatas de las figuras III.15, 16 coinciden con aquellos de las figuras III.2, 3.

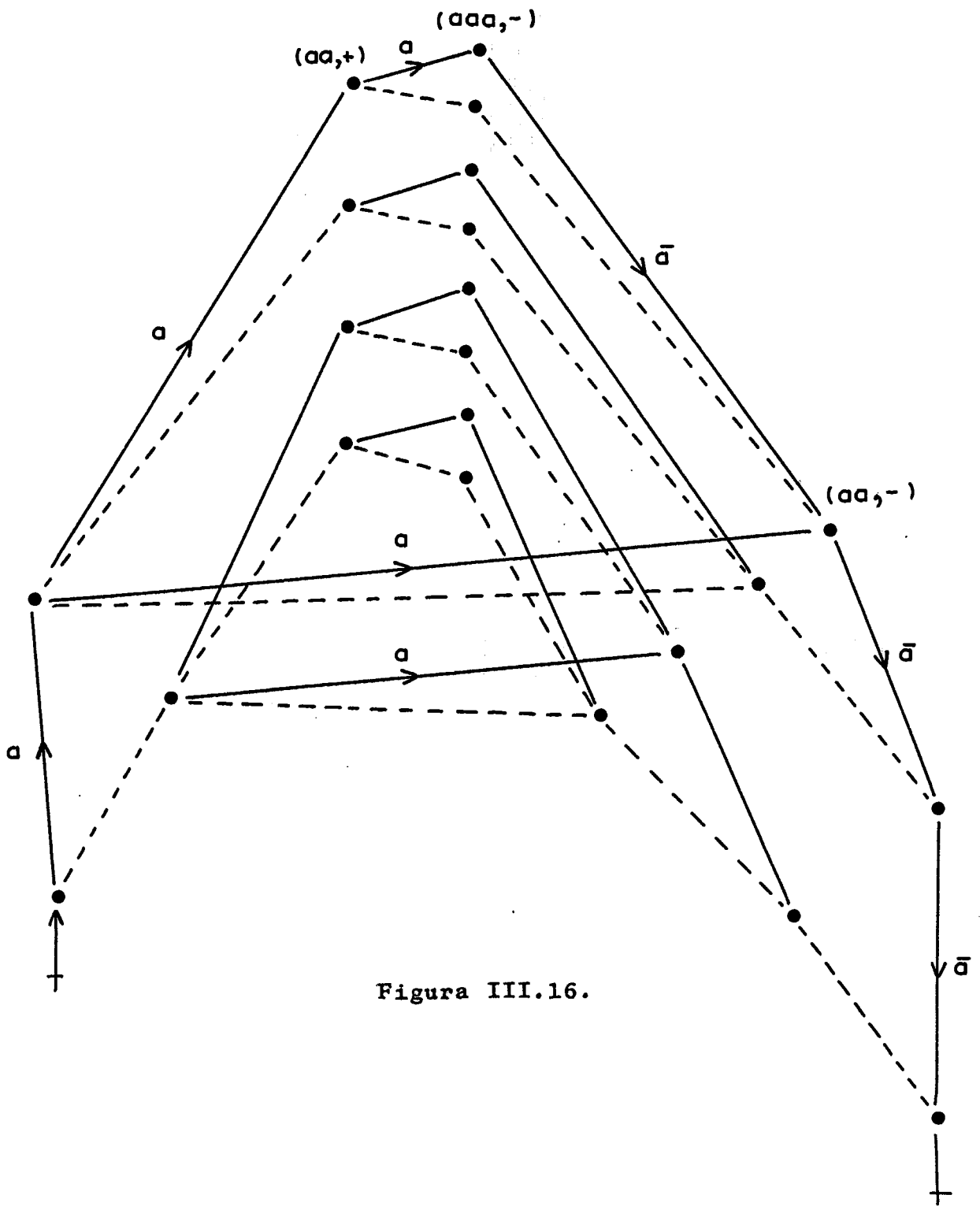


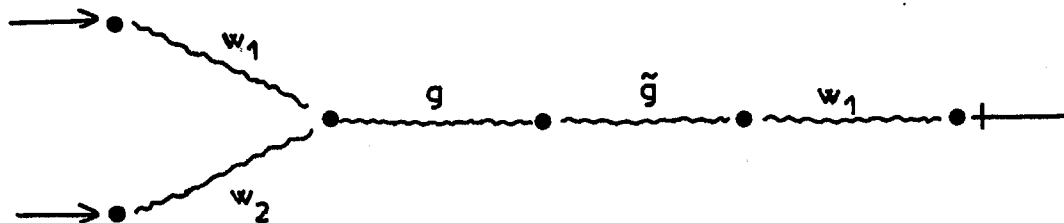
Figura III.16.

la palabra w_1 a $|w_2| - n$ a $|w_2| - n$ w_2 es demasiado larga para pertenecer a $2n \text{ PAL}_2$.

.Tenemos también que para todo i, j tenemos $Q_i \cap Q_{n+j} = \emptyset$

Si no son disjuntos tomando $|w_1| = i$ y $|w_2| = n + j$ con $w_2 = u_1 u_2 \dots u_j \dots u_n u_n \dots u_j$ en la construcción precedente llegamos a una contradicción.

.Demostremos que $\|Q_i\| \geq 2^i$. Supongamos $\|Q_i\| < 2^i$, dado que existen 2^i palabras de longitud i debemos tener un estado al que forzosamente llegan dos palabras de longitud i distintas. La situación es esquematizada en la figura siguiente.



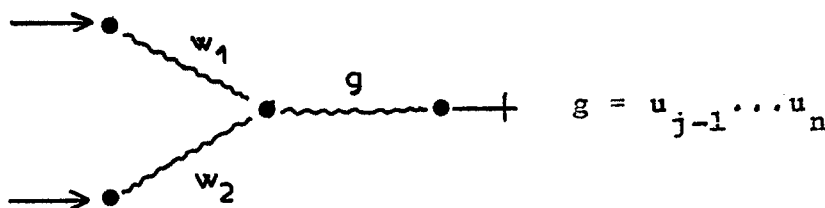
Tenemos que $f_2 g \tilde{g} f_1$ no pertenece al lenguaje.

.Demostremos que $\|Q_{n+j}\| \geq 2^{n-j}$. Supongamos $\|Q_{n+j}\| < 2^{n-j}$,

en este caso existen $w_1 = u_n \dots u_{j-1} u_j \dots u_n u_n \dots u_j$,

$w_2 = v_n \dots v_{j-1} v_j \dots v_n v_n \dots v_j$ con $u_n \dots u_{j-1} \neq v_n \dots v_{j-1}$,

entonces:



la palabra $w_2 g$ no pertenece al lenguaje.

.En consecuencia $\|Q\| = \sum_{0 \leq i \leq n} \|Q_i\| + \sum_{1 \leq j \leq n} \|Q_{n+j}\| =$
 $= 2 \sum_{0 \leq i \leq n-1} 2^p + 2^n$ y el autómata dado precedentemente es pe

queño. ■

(c) Si consideramos los autómatas $\psi(A_n)$ o $\psi(B_n)$ vemos que son no deterministas. Dado el carácter no determinista del autómata a pila que reconoce PAL_2 , de manera intuitiva, el autómata determinista que reconoce $2nPAL_2$ debe ser "complicado". Las figuras III.17,18 muestran dichos autómatas para 6 PAL_2 y 8 PAL_2 . ■

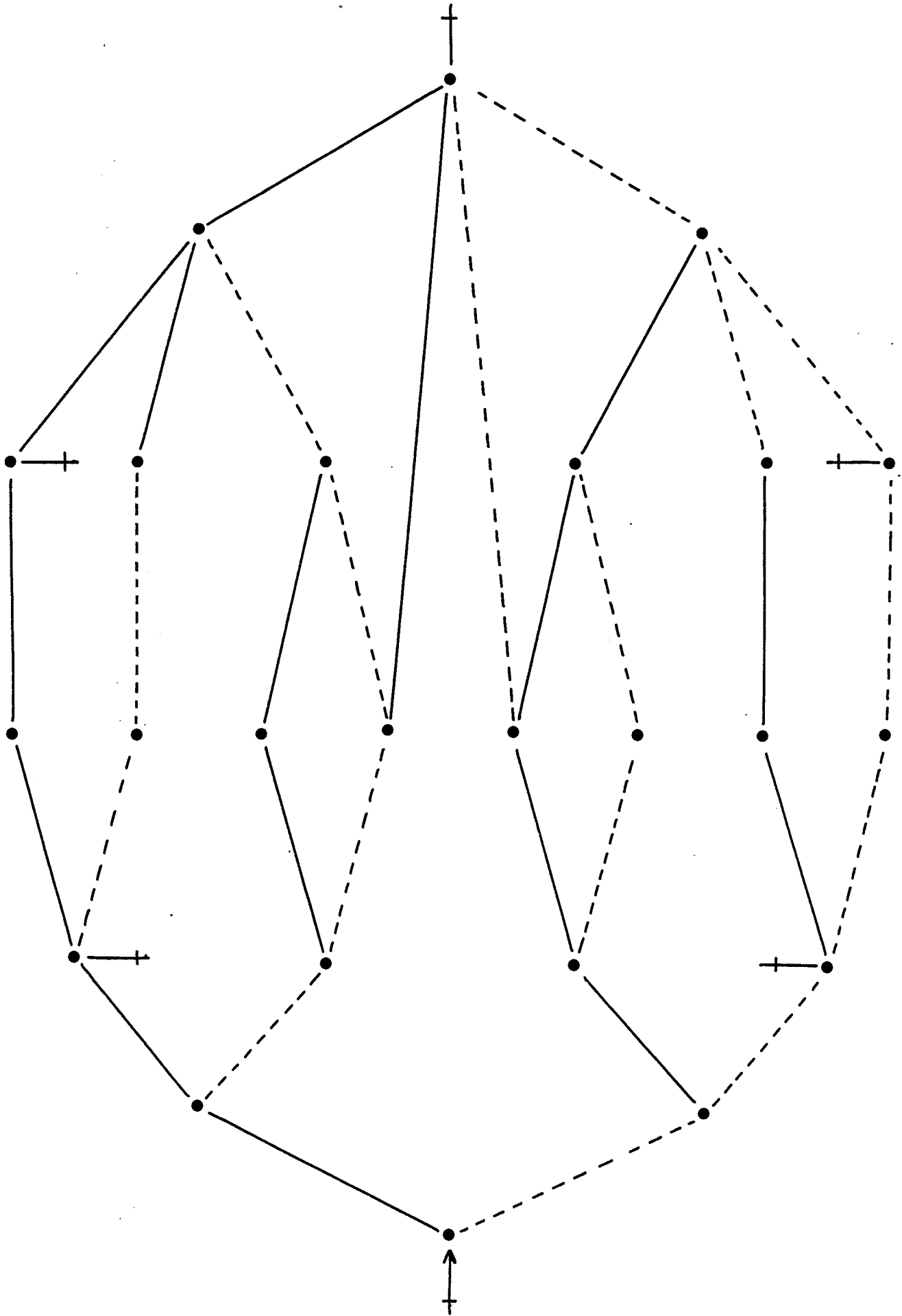


Figura III.17.

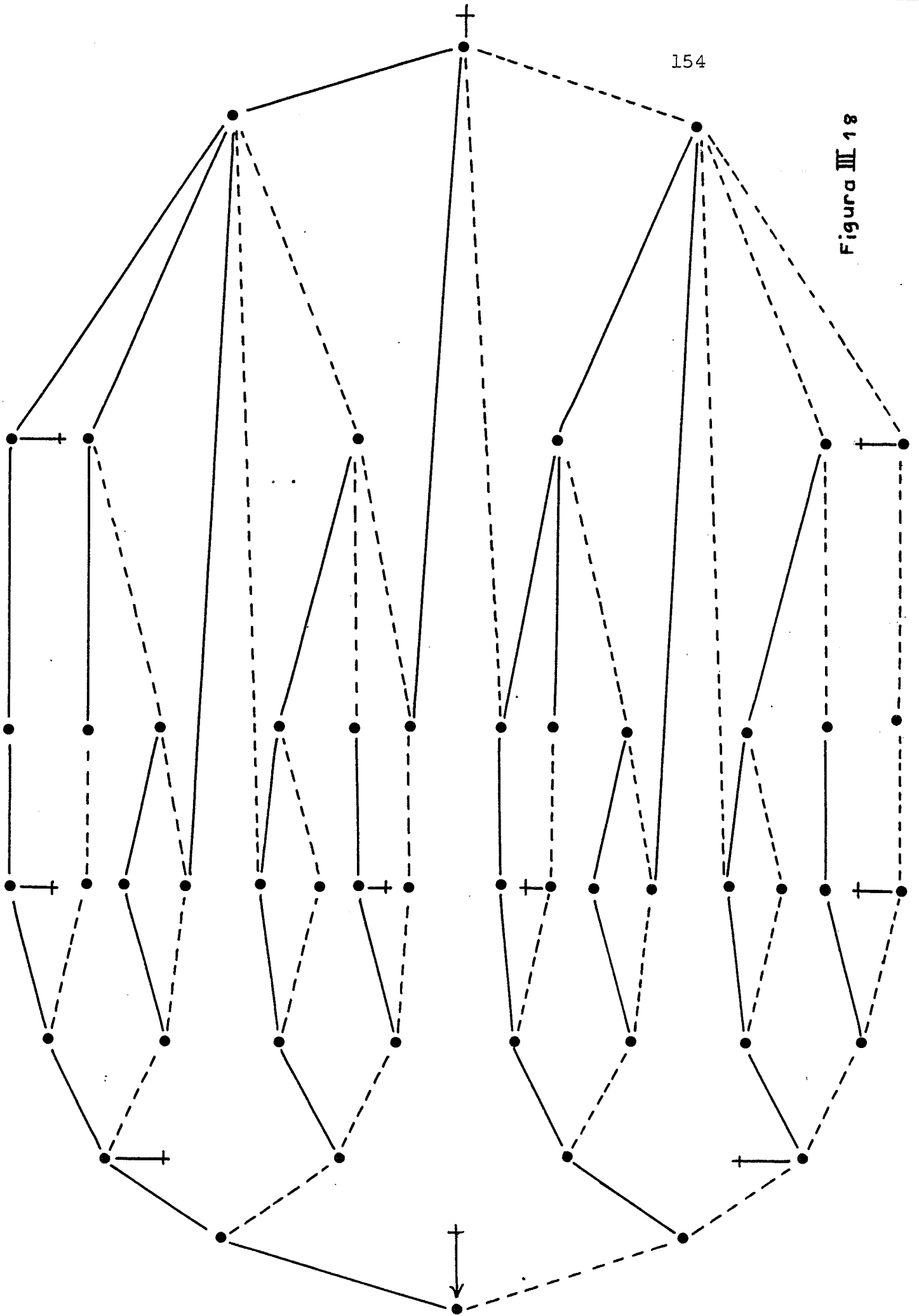


Figura III 18

Ejemplo 2.8 Vamos a calcular el índice inicial del lenguaje

$$\text{COPY}_2 = \{ w\bar{w} \mid w \in \{a,b\}^* , \bar{w} \text{ es una copia barrada de } w \}$$

(a) Inspirándonos en el autómata dado para $2n \text{ SYM}_2$, construimos el autómata $\mathcal{L}_n = \langle \{a, \bar{a}, b, \bar{b}\}, Q, q_-, q_+, \delta \rangle$ para $2n \text{ COPY}_2$ dado por:

$$Q = \{ (w, s) \mid w \in (n-1) \{a, b\}^* , s \in \{+, -\} \} \cup \{ (w, +) \mid w \in \{a, b\}^n \}$$

$$q_- = (\varepsilon, +) , \quad q_+ = \{ (\varepsilon, +), (\varepsilon, -) \}$$

La función de transición δ está definida por:

$$(w, +).u = (wu, +) , \quad u \in \{a, b\} , \quad |w| \leq n-1$$

$$(uw, +).\bar{u} = (w, -) , \quad u \in \{a, b\} , \quad \bar{u} \text{ copia barrada de } u$$

$$(uw, -).\bar{u} = (w, -)$$

La figura III.19 muestra \mathcal{L}_n .

$$\text{Tenemos que } ||Q|| = 2 \sum_{p=0}^{n-1} 2^p + 2^n = \Theta(2^n)$$

(b) Para demostrar que el autómata dado es pequeño, debemos modificar ligeramente la demostración dada en SYM_2 .

En este caso los factores izquierdos de COPY_2 , h son de la forma:

$$h = u_1 u_2 \dots u_p u_{p+1} \dots u_r \bar{u}_1 \bar{u}_2 \dots \bar{u}_p = g f \bar{g} \text{ con } g = u_1 u_2 \dots u_p,$$

$$f = u_{p+1} \dots u_r, \bar{g} = \bar{u}_1 \bar{u}_2 \dots \bar{u}_p.$$

En este caso definimos el peso por: $p(h) = u_{p+1} \dots u_r = f$

En el conjunto $2n \text{ COPY}_2$ pueden aparecer $n \{a, b\}^*$ pesos distintos.

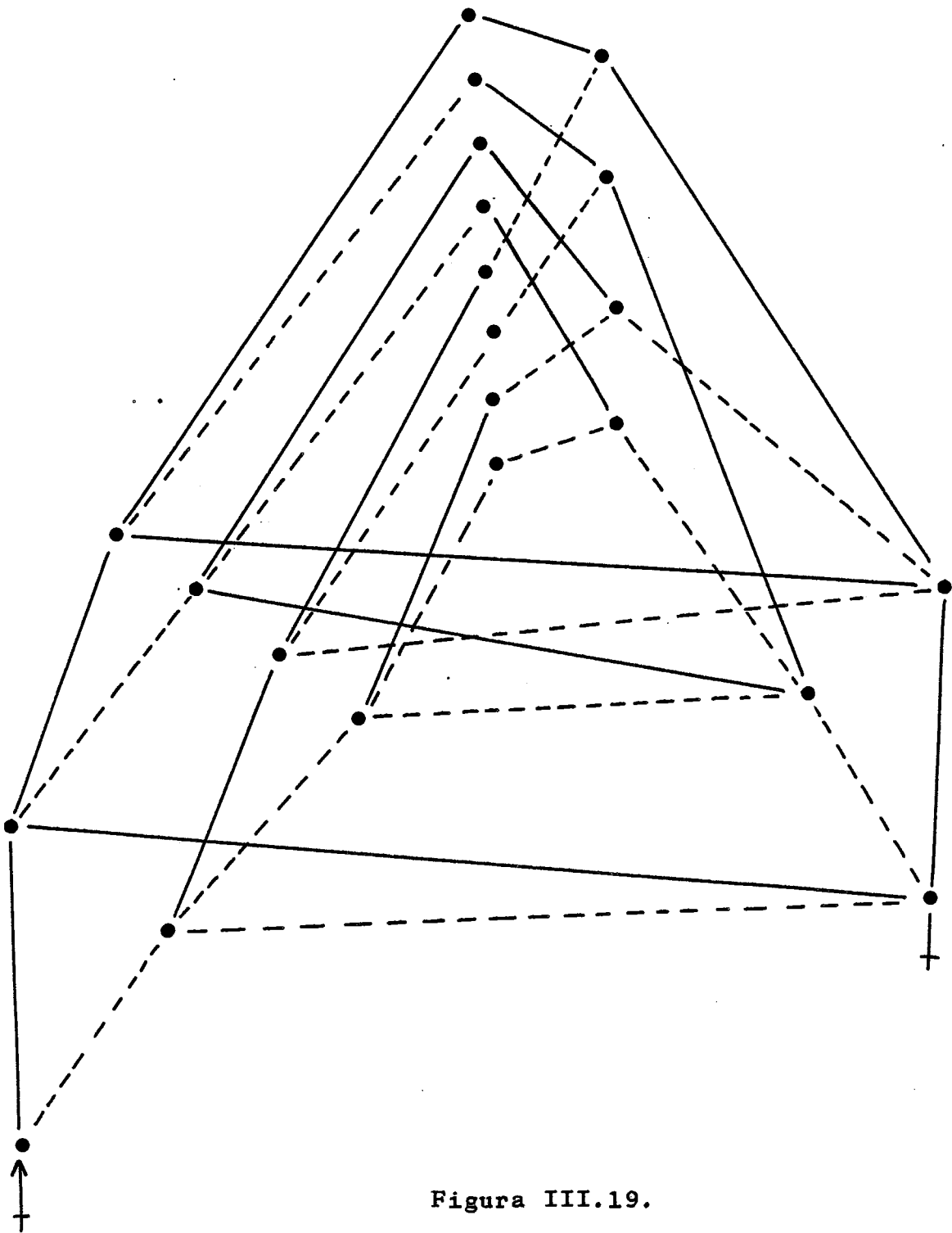


Figura III.19.

Considerando los pares de palabras $f, a^{n-|f|} f a^{n-|f|}$,
 $f \in (n-1) \{a, b\}^*$ tenemos que a todo peso de longitud menor
 que n corresponden dos pesos distintos. En consecuencia el au
 tómata \mathcal{L}_n es pequeño. Tenemos $\mu_{\text{COPY}_2}(n) = \Theta(2^{\frac{n}{2}})$.

(c) También para este caso tenemos una versión no determinista
 de autómata pequeño: $\mathcal{D}_n = \langle \{a, \bar{a}, b, \bar{b}\}, Q, q_-, q_+, \delta \rangle$

$$Q = \{ (w, s) \mid w \in (n-1) \{a, b\}^*, s \in \{+, -\} \} \cup \{ (w, -) \mid w \in \{a, b\}^n \}$$

$$q_- = (\varepsilon, +), \quad q_+ = \{ (\varepsilon, +), (\varepsilon, -) \}$$

La función de transición está dada por:

$$(w, +) \cdot u = \{ (wu, +), (wu, -) \}, \quad |w| \leq n-2$$

$$(w, +) \cdot u = (wu, -), \quad |w| = n-1$$

$$(uw, -) \cdot \bar{u} = (w, -)$$

La figura III.20 muestra 6 \mathcal{D}_n .

(d) De modo análogo podemos considerar $\text{COPY}_p, p > 1$, en este
 caso $\mu_{\text{COPY}_p}(n) = \Theta(p^{\frac{n}{2}})$

Notemos que $\text{COPY}_1 = \text{SYM}_1 = S_1$.

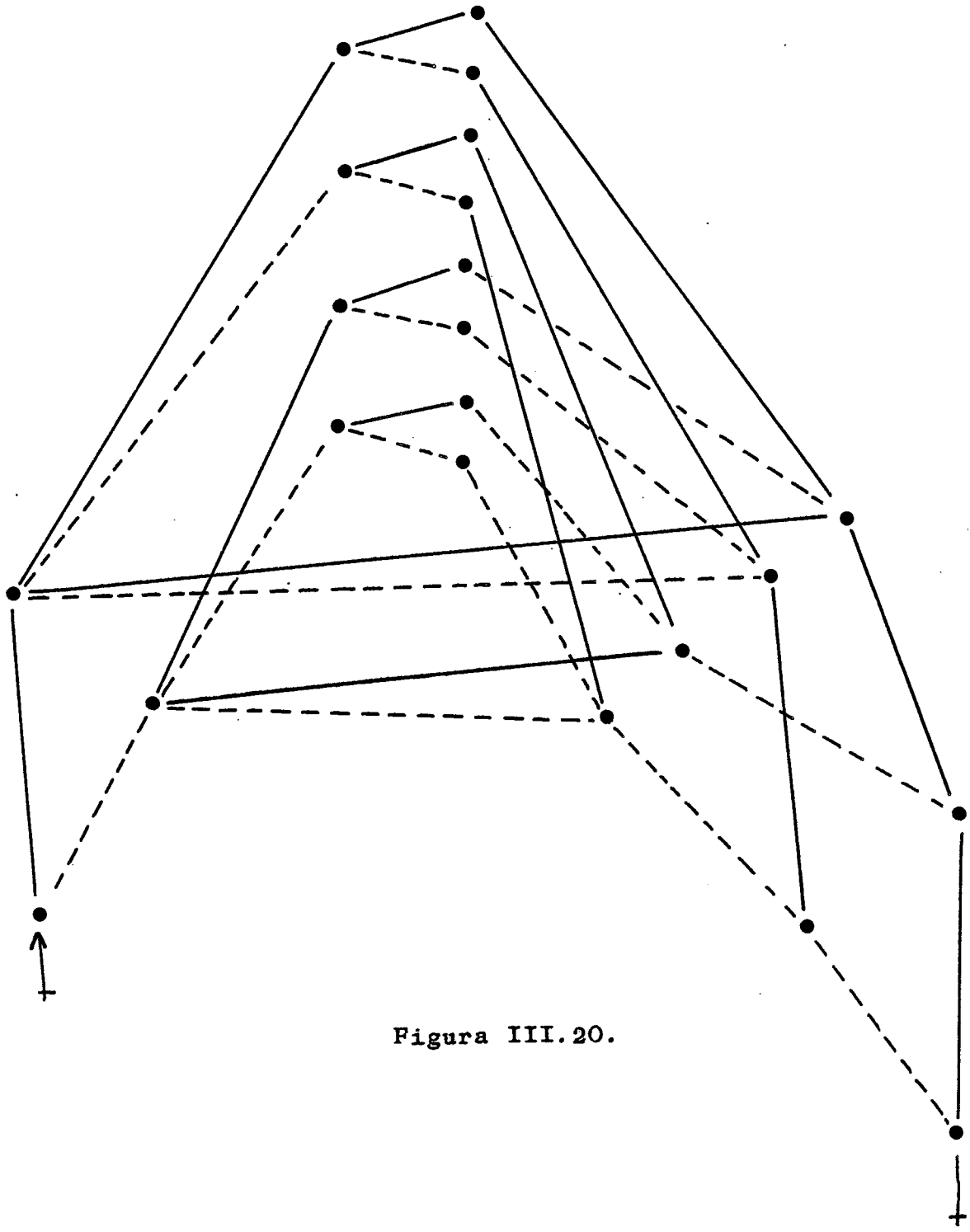


Figura III.20.

Ejemplo 2.9 Sea el lenguaje $SQR_2 = \{ ww \mid w \in \{a, b\}^* \}$, considerando SYM_2, PAL_2 y $COPY_2$ es evidente que

$$\mu_{SQR_2}(n) = \mu_{PAL_2}(n) = \Theta(2^{\frac{n}{2}})$$

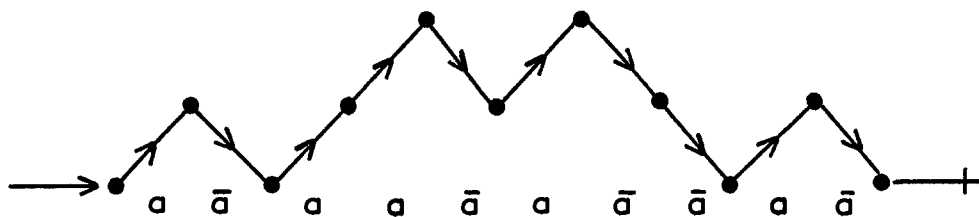
Ejemplo 2.10 Vamos a calcular el índice inicial del lenguaje de Dyck de una letra, notado D_1^* . Dicho lenguaje puede caracterizarse de múltiples maneras [6], por ejemplo:

(1) $D_1^* = \langle S \rightarrow aS\bar{a}S + \epsilon \rangle$

(2) Como la clase de ϵ en la congruencia Θ en $\{a, \bar{a}\}^*$ definida por $\Theta = \langle a\bar{a} = \epsilon \rangle$, es decir $D_1^* = [\epsilon]_{\Theta}$

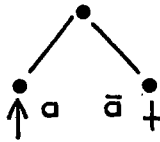
(3) Sea $m \in \{a, \bar{a}\}^*$ definimos $\|m\| = |m|_a - |m|_{\bar{a}}$. Tenemos que $D_1^* = \{ m \in \{a, \bar{a}\}^* \mid \|m\| = 0 \text{ y para todo } m' < m, \|m'\| \geq 0 \}$

La última definición admite, para toda palabra de Dyck una representación gráfica en términos de línea quebrada, por ejemplo:

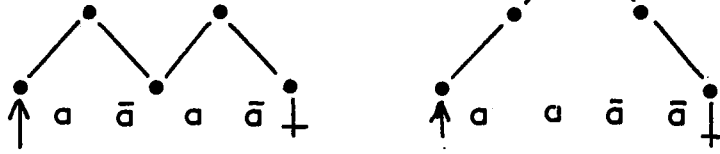


(a) Vamos a mostrar como se puede hallar de manera intuitiva un autómata para nD_1^* . Trataremos en principio el caso $n = 6$. Para ello comenzaremos por dibujar todas las palabras de longitud hasta 6.

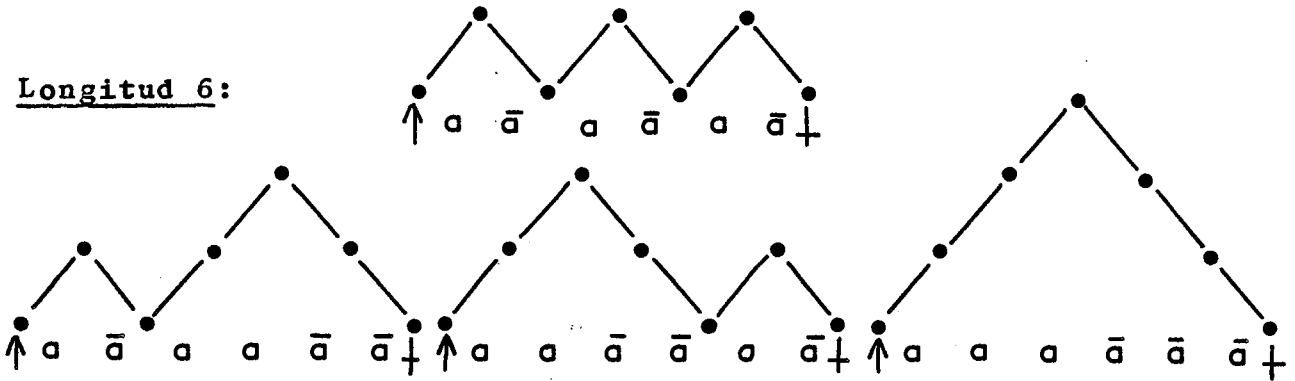
Longitud 2:



Longitud 4:



Longitud 6:



Verificamos que superponiendo todos estos dibujos obtenemos, figura III.21 un autómata que reconoce $6 D_2'$

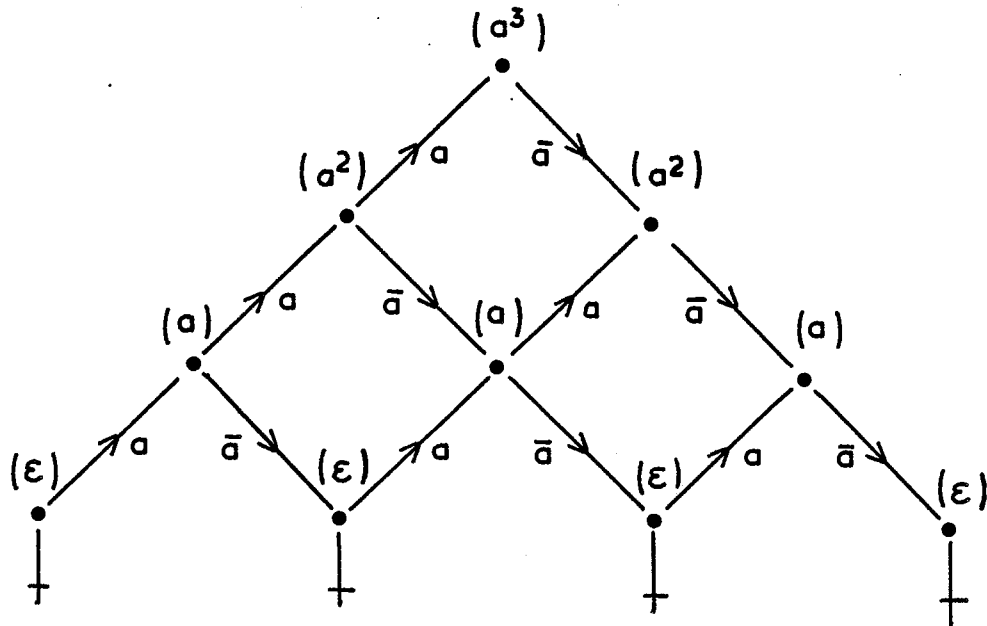


Figura III.21

A toda palabra $m \in \{a, b\}^*$ vamos a asociar su irreductible según la congruencia $\Theta = \langle a\bar{a} = \epsilon \rangle$, $\Theta(m)$. Dicho irreductible

se obtiene reescribiendo todo factor $a\bar{a}$ de m en \mathcal{E} . Cuando f es un factor izquierdo $\Theta(f)$ es de forma a^x . En la figura III.21 hemos escrito entre paréntesis los irreducibles asociados a cada factor izquierdo. Vemos que a todo estado corresponde un solo irreducible, demostraremos que esto es general en (c).

(b) Vamos ahora a generalizar el resultado anterior. Dado que $(2n + 1)D_1^{*} = 2nD_1^{*}$, trataremos solamente el caso $2nD_1^{*}$.

Consideremos el autómata $A_n = \langle \{a, \bar{a}\}, Q, q_-, q_+, \delta \rangle$, dado en la figura III.22 donde:

$$Q = \{ (x, 2p + x) \mid 0 \leq x \leq n, 0 \leq p \leq n - x \}$$

$$q_- = (0, 0), \quad q_+ = \{ (0, 2p) \mid 0 \leq p \leq n \}$$

La función de transición está dada por:

$$(x, y).a = (x + 1, y + 1) \quad ; \quad 2n > y + x$$

$$(x, y).\bar{a} = (x - 1, y + 1) \quad ; \quad x \geq 1$$

Vamos a demostrar que $L(A_n) = 2n D_1^{*}$.

La función de transición de dicho autómata está definida solamente para los factores izquierdos f de D_1^{*} tales que:

$$|f| + |\Theta(f)| \leq n, \text{ entonces:}$$

$$(0, 0).f = (|\Theta(f)|, |f|)$$

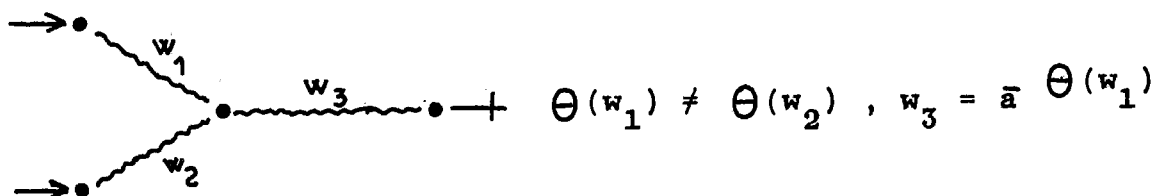
Tenemos en consecuencia $L(A_n) = 2n D_1^{*}$

Vamos a contar $||Q||$. Para cada $x \leq n$ tenemos $n + 1 - x$ estados

$$\text{distintos con lo que } ||Q|| = \sum_{x=0}^n (n + 1 - x) = \Theta(n^2).$$

(c) Vamos a demostrar que el autómata definido anteriormente es pequeño. Tenemos que a todo estado corresponde solo un irreducible. Si no se produce la contradicción dada en el siguiente

te esquema:



tenemos $w_1 w_3 \in D_1^*$, efectivamente $\Theta(w_1 w_3) \equiv \Theta(\Theta(w_1) \Theta(w_3)) \equiv$

$$\Theta(a \Theta(w_1) \bar{a} \Theta(w_1)) \equiv \varepsilon, w_2 w_3 \equiv \Theta(w_2 w_3) \equiv \Theta(a \Theta(w_2) \bar{a} \Theta(w_1)) \neq \varepsilon$$

Tenemos además que para cada irreductible $a^p, 0 \leq p \leq n$, existen $n + 1 - p$ estados distintos. Para demostrarlo basta considerar el conjunto de palabras de longitud máxima.

$$M_1 = \{ a^p (a\bar{a})^{n-p} a^p \mid 0 \leq p \leq n \}$$

Sea Q_p el conjunto de los estados de cualquier autómata pequeño los estados

$$E_p = Q_p \cdot \{ a^p (a\bar{a})^u \mid 0 \leq u \leq n - p \}$$

deben ser todos distintos, sino existiría un bucle, y tienen todos el mismo irreductible a^p . Tenemos $\|E_p\| = n - p + 1$.

Como E_p y E_q corresponden a irreductibles distintos $E_p \cap E_q = \emptyset$

si $p \neq q$ y en consecuencia $\sum_{p=0}^n \|E_p\| = \sum_{p=0}^n (n - p + 1) =$

$$= \Theta(n^2).$$

Entonces: $\mu_{D_1^*}(n) \leq \sum_{p=0}^n (n - p + 1) = \Theta(n^2).$

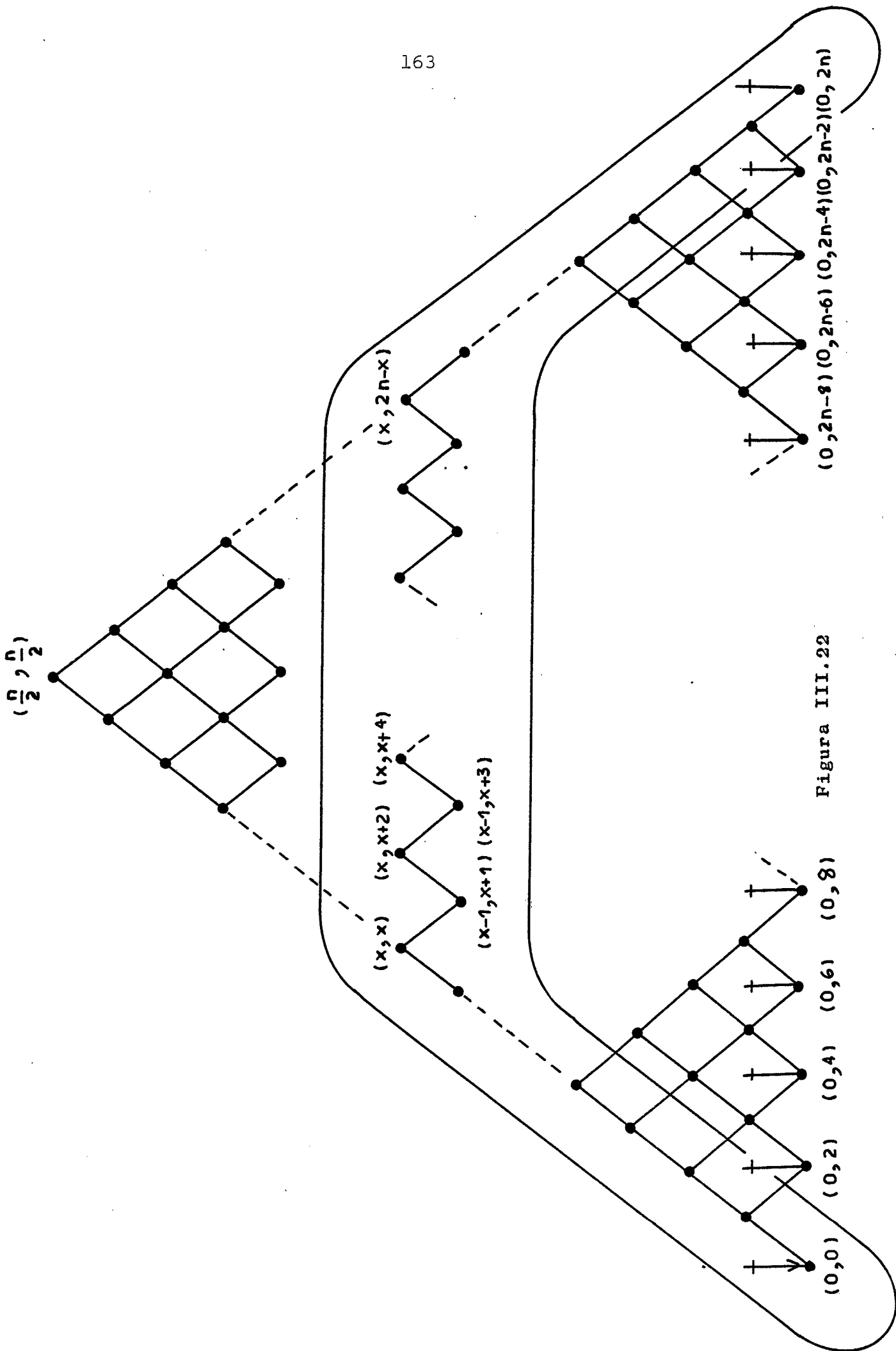


Figura III.22

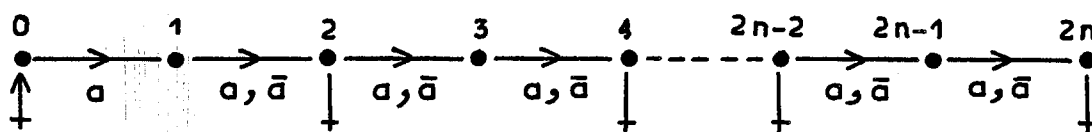
(d) El autómata dado en la figura III.22 puede obtenerse como intersección del pda que reconoce D_1^* esquematizado en:

$$(q, z).a = (q, za)$$

$$(q, a).a = (q, a^2)$$

$$(q, a).\bar{a} = (q, \epsilon)$$

que reconoce D_1^* por fondo de pila y del autómata finito siguiente:



Realizamos el cálculo para $n = 3$, figura III.23.

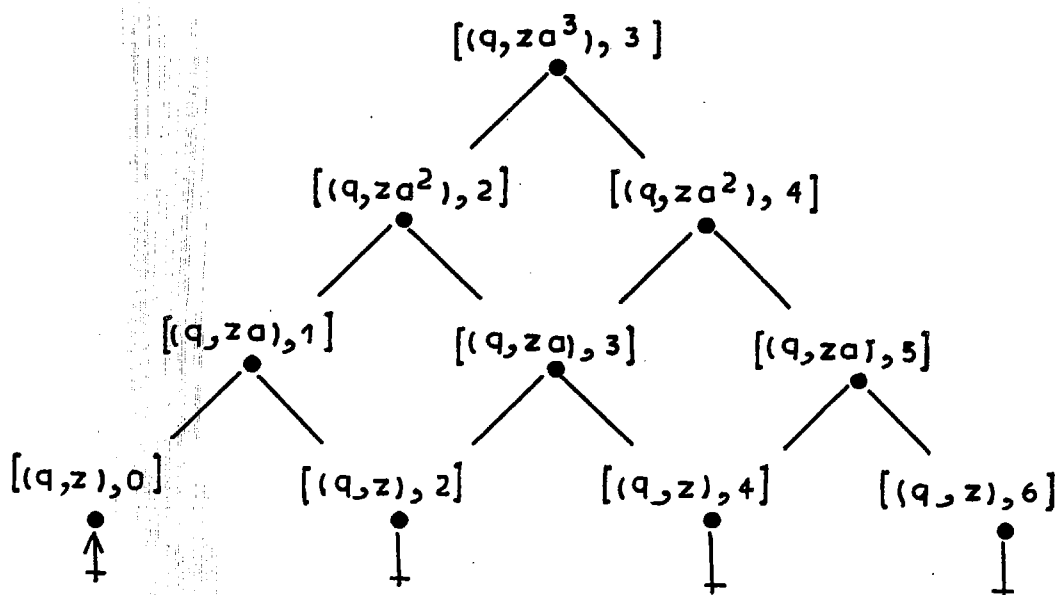


Figura III.23.

Ejemplo 2.11 Vamos a calcular el índice inicial de los factores izquierdos de D_1^* , que notaremos $FG(D_1^*)$.

Es fácil demostrar [23] que:

$$FG(D_1^*) = (a, aD_1^* \bar{a})^*$$

De manera intuitiva la demostración se basa en que todo factor izquierdo puede factorizarse de manera única en, figura III.24.

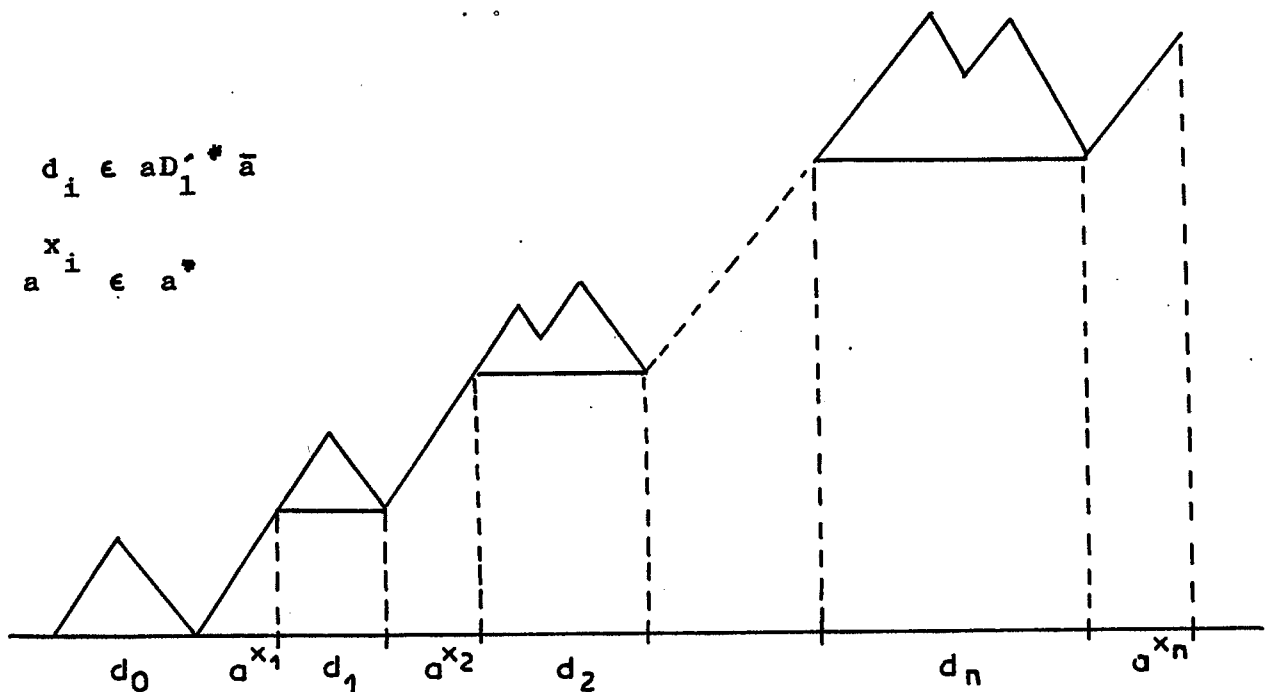


Figura III.24

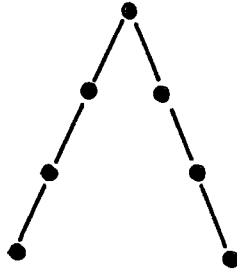
Vamos a dar seguidamente el cálculo de un autómata pequeño para $FG(D_1^*)$. Para ello vamos a representar todos los factores izquierdos posibles de longitud 6.

(1) Factor izquierdo sin ninguna ocurrencia de d :

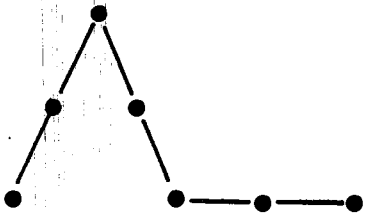


a_6

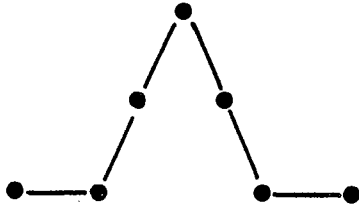
(2) Con un solo d :



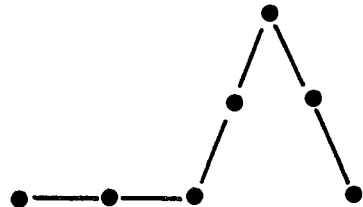
$(a^3 \bar{a}^3)$



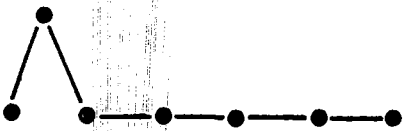
$(a^2 \bar{a}^2) a^2$



$a(a^2 \bar{a}^2)a$



$a^2(a^2 \bar{a}^2)$

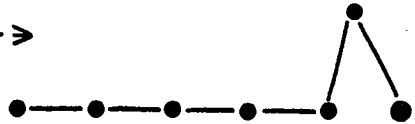


$(a \bar{a}) a^4$



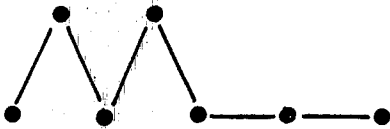
$a(a \bar{a}) a^3$

----->

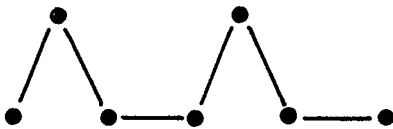


$a^4(a \bar{a})$

(3) Con dos d :



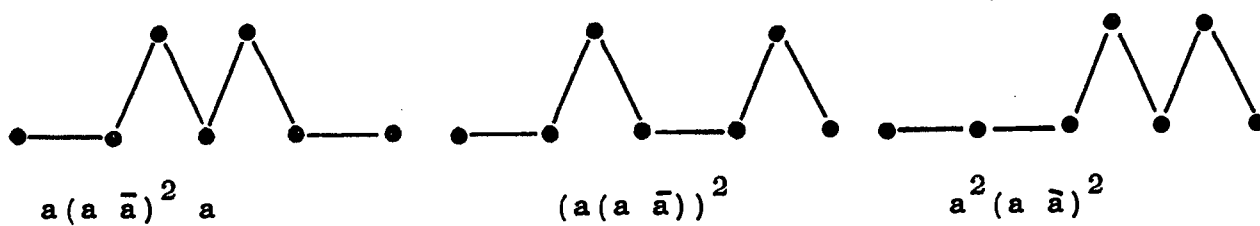
$(a \bar{a})^2 a^2$



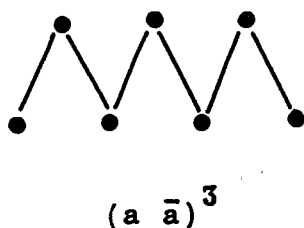
$((a \bar{a}) a)^2$



$(a \bar{a}) a a (a \bar{a})$



(4) Con 3 d:



Todas estas figuras se pueden superponer en el autómata, figura III.25

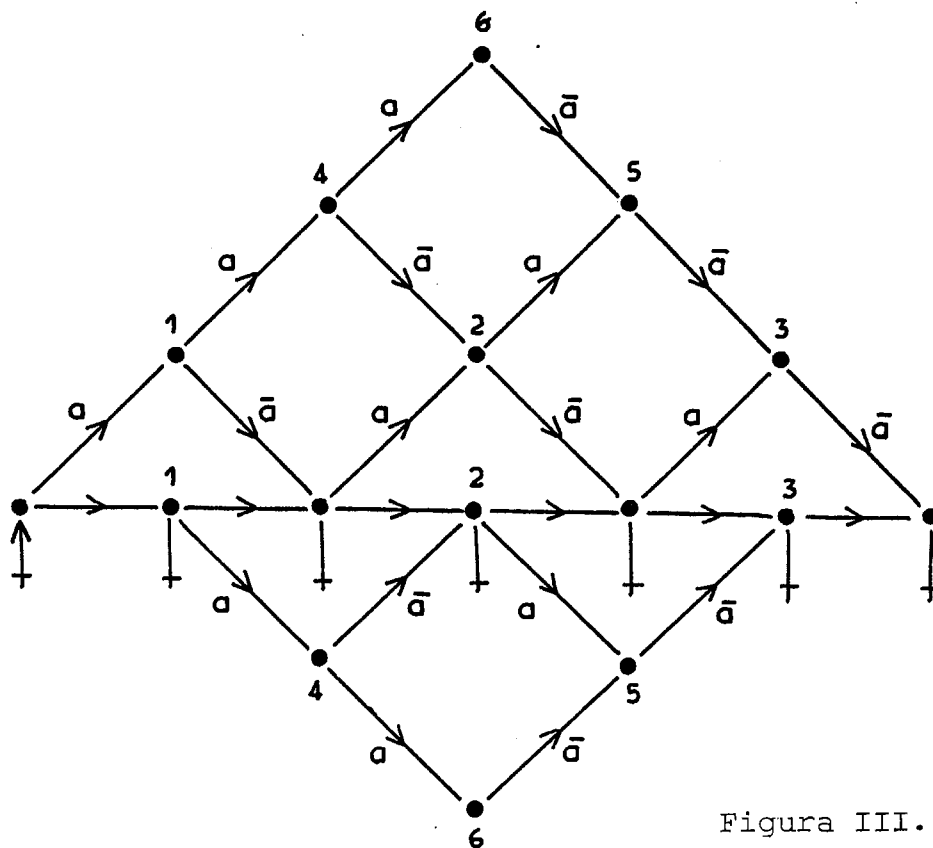


Figura III. 25.

En dicha figura se ha dado a los estados que pueden superponerse el mismo número. Tras realizar dicha superposición obtenemos el autómata, figura III.26.

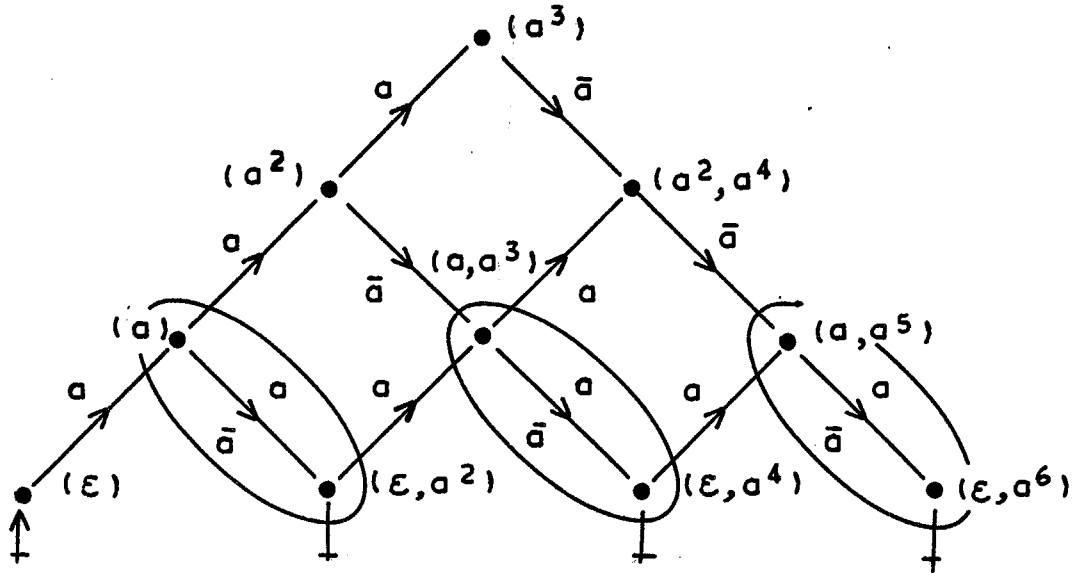


Figura III.26

El autómata así obtenido puede generalizarse para cualquier $2n$, obtenemos entonces la figura III.27.

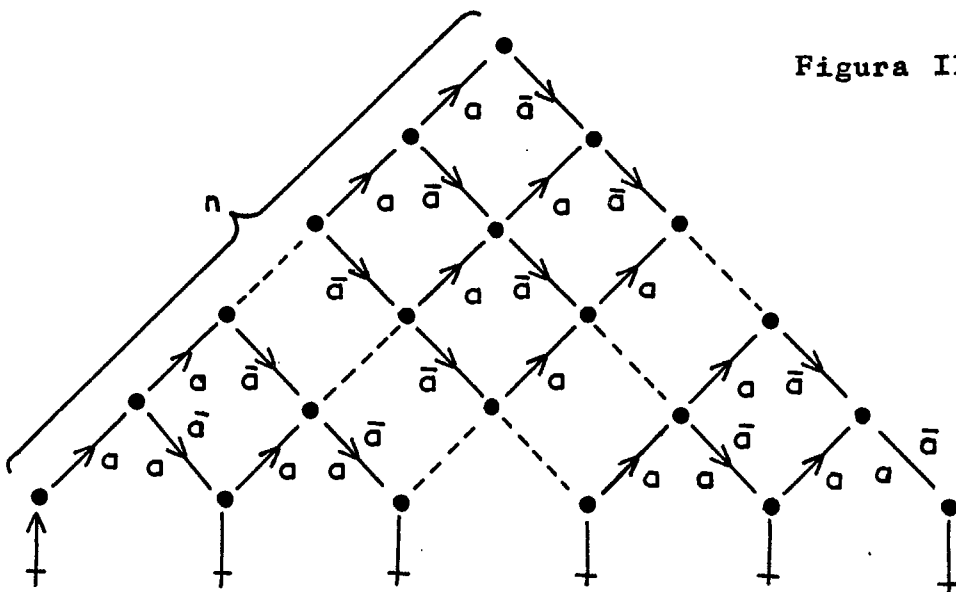


Figura III.27

Vemos en la figura III.26 que, a cada estado, podemos asociar más de un solo irreducible, con lo que el método desarrollado para demostrar la pequeñez del autómata dado para D_1^* , debe ser modificado.

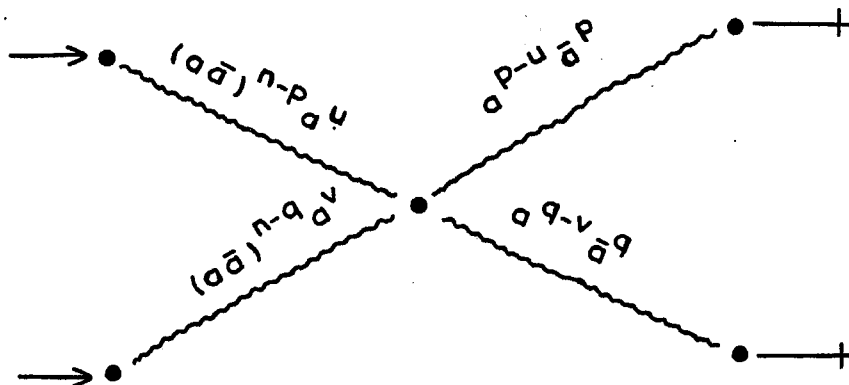
Para demostrar que todo autómata pequeño tiene al menos tantos estados como los de la figura III.27 vamos a considerar el siguiente conjunto de palabras de longitud máxima $2n$.

$$M_2 = \{ (a \bar{a})^{n-p} a^p \bar{a}^p \mid 0 \leq p \leq n \}$$

Sea Q_- el conjunto de estados iniciales de cualquier autómata pequeño. Estudiemos el cardinal del conjunto de estados.

$$E_2 = Q_- \cdot \{ (a \bar{a})^{n-p} a^x \mid 0 \leq p \leq n, 0 \leq x \leq p \}$$

Vamos a mostrar $||E|| = \sum_{p=0}^n (p+1) = \Theta(n^2)$. Caso de no ser así existe el esquema:



tenemos las igualdades $u + q + v = q$, $v + p - u = p$, es decir $u = v$. Supongamos $p > q$, consideremos la palabra:

$$m = (a \bar{a})^{n-q} a^v a^{p-u} \bar{a}^p, \quad |m| = |(a \bar{a})^{n-q} a^p \bar{a}^p| = 2n - 2q +$$

$+ 2p > 2n$, llegamos pues a una contradicción. Tenemos entonces que cada palabra $(a \bar{a})^{n-p} a^x$, $0 \leq p \leq n$, $0 \leq x \leq p$ dá lugar a un estado distinto, con lo que demostramos que el autómata de la figura III.27 es pequeño.

Notemos la semejanza entre los conjuntos:

$$M_1 = \{ a^p (a \bar{a})^{n-p} \bar{a}^p \mid 0 \leq p \leq n \}$$

$$M_2 = \{ (a \bar{a})^{n-p} a^p \bar{a}^p \mid 0 \leq p \leq n \}$$

empleados en calcular el índice inicial de $D_1'^*$ y de $FG(D_1'^*)$ respectivamente.

■

Ejemplo 2.12 Vamos a estudiar el lenguaje:

$$D_1 = \{ w \{a, \bar{a}\}^* \mid |w|_a = |w|_{\bar{a}} \} = \langle S \Rightarrow aS\bar{a}S + \bar{a}S aS + \epsilon \rangle$$

Un autómata pequeño para dicho lenguaje viene dado en la figura III.28.

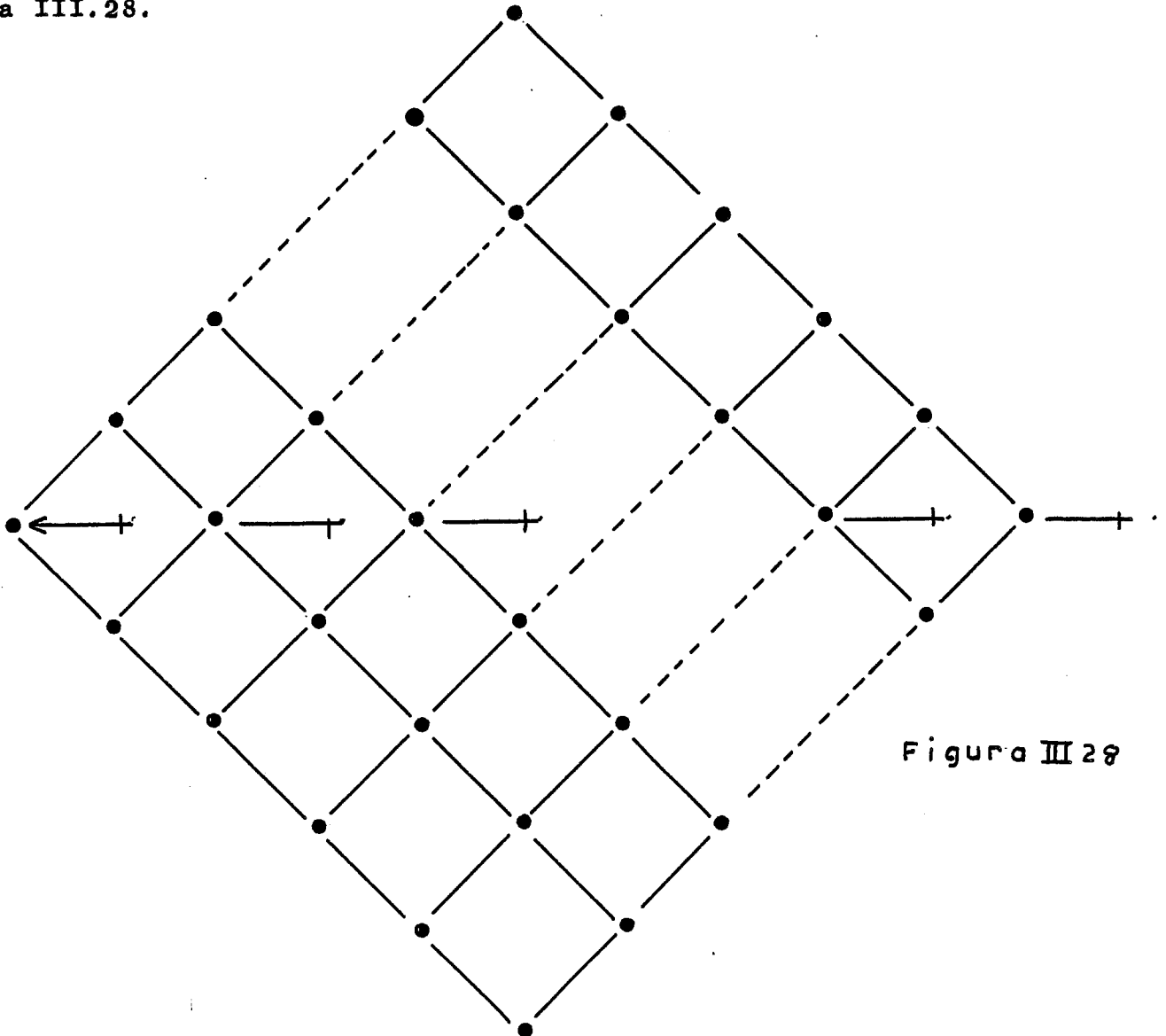


Figura III 28

En este caso los estados pueden codificarse por:

$$(x, 2p + x, s) , 0 \leq x \leq n , 0 \leq p \leq n - x , s \in \{+, -\}$$

Evidentemente $\mu_{D_1^*}(n) = \Theta(n^2)$

■

Ejemplo 2.13 Vamos a estudiar $L = \{w \in \{a, b\}^* \mid 2|w|_a = 3|w|_b\}$

Un autómata pequeño para dicho lenguaje está indicado en la figura III.29.

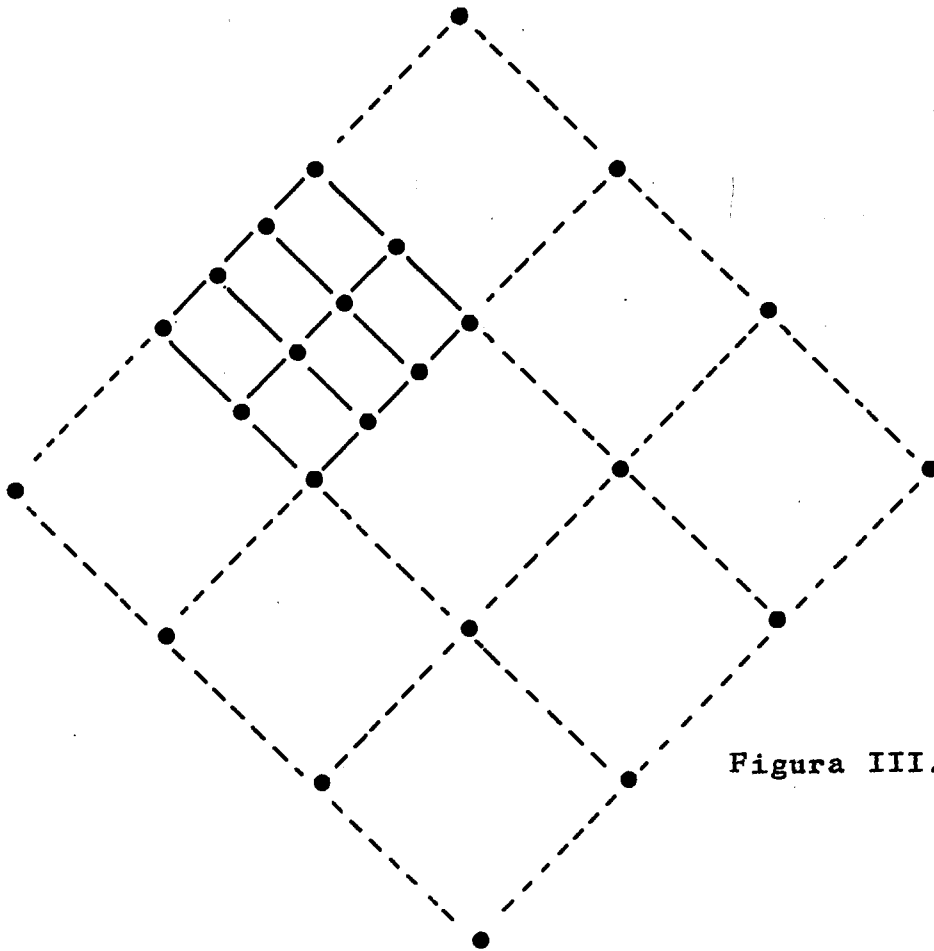


Figura III.29.

Tenemos $\mu_L(n) = \Theta(n^2)$

Ejemplo 2.14 Vamos a calcular el índice inicial del lenguaje de Dyck de 2 letras:

$$D_2'^* = \langle S \rightarrow a s \bar{a} s + b s \bar{b} s + \epsilon \rangle$$

Dicho lenguaje puede caracterizarse como la clase de la palabra vacía en la congruencia $\Theta = \langle a\bar{a} \equiv b\bar{b} \equiv \epsilon \rangle$ es decir

$$D_2'^* = [\epsilon]_{\Theta}$$

(a) Trataremos el caso $2nD_2'^*$. Consideremos el autómata A_n , descrito para $n = 3$ en la figura III.30, dado por:

$$A_n = \langle \{a, b, \bar{a}, \bar{b}\}, Q, q_-, q_+, \delta \rangle$$

$$Q = \{ (w, 2p + |w|) \mid w \in \{a, b\}^*, 0 \leq p \leq n - |w| \}$$

$$q_- = (\epsilon, \epsilon), \quad q_+ = \{ (\epsilon, 2p) \mid 0 \leq p \leq n \}$$

$$(w, y) \cdot u = (wu, y + 1), \quad 2n > |w| + y, \quad u \in \{a, b\}$$

$$(wu, y) \cdot \bar{u} = (w, y + 1), \quad u \in \{a, b\}$$

Cuando f es un factor izquierdo de $D_2'^*$ tenemos que su irreductible $\Theta(f)$ pertenece a $\{a, b\}^*$. La función de transición

δ está definida solamente cuando f es un factor izquierdo de $D_2'^*$ tal que $|f| + |\Theta(f)| \leq n$, en este caso $(\epsilon, 0) \cdot f =$

$$= (\Theta(f), |f|). \text{ Como consecuencia tenemos } L(A_n) = 2D_2'^*$$

Vamos a contar $||Q||$. Llamamos Q_x el conjunto:

$$Q_x = \{ (w, y) \in Q \mid |w| = x \}, \quad 0 \leq x \leq n$$

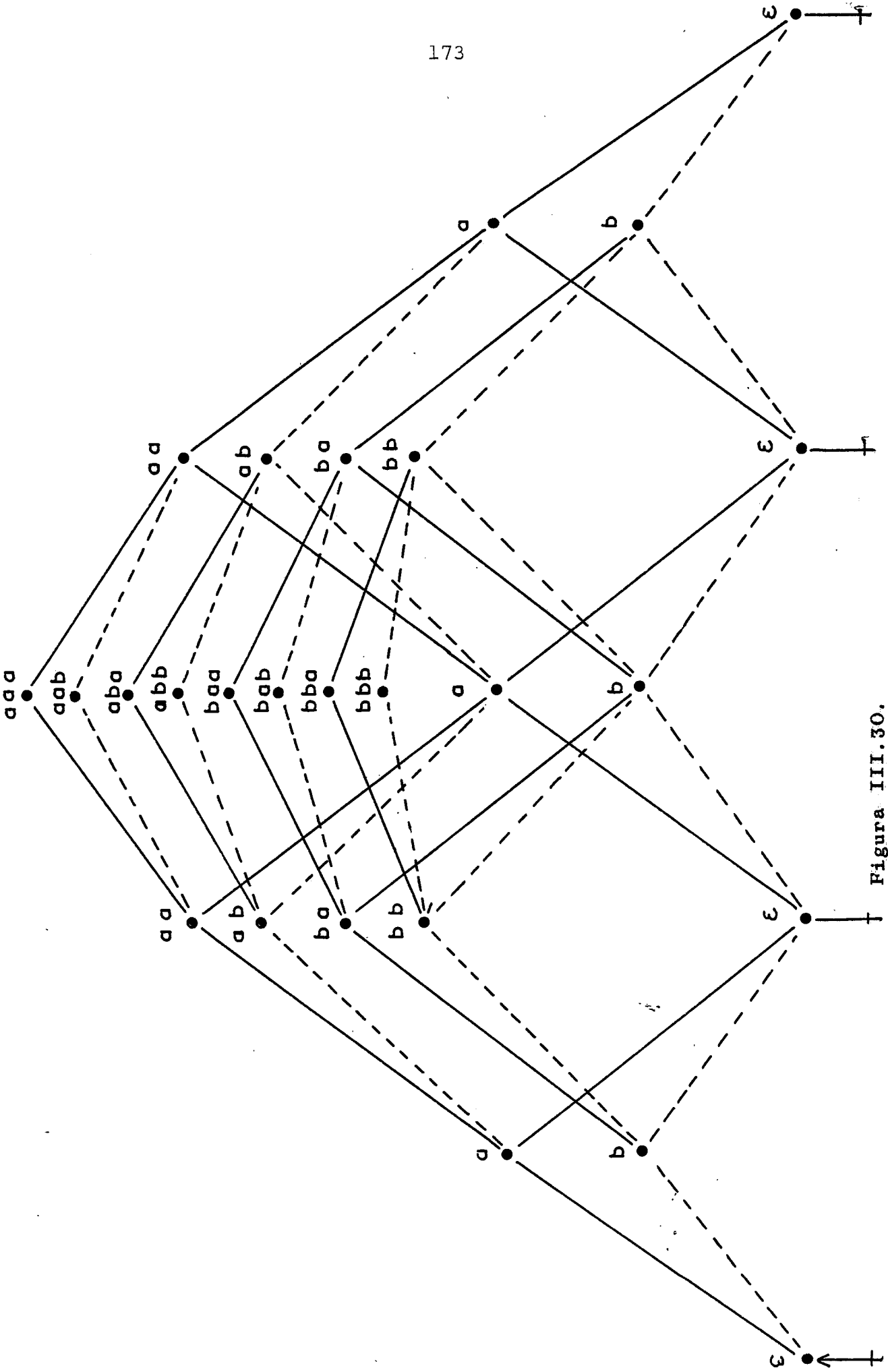


Figura III.30.

Evidentemente $\|Q_x\| = (n + 1 - x)2^x$, con lo que :

$$\|Q\| = \sum_{x=0}^n \|Q_x\| = \sum_{x=0}^n (n + 1 - x)2^x = \Theta(2^n)$$

(b) Vamos a demostrar que el autómata que acabamos de definir es pequeño. Tenemos, como precedentemente, que a cada estado corresponde un irreducible. Para un $0 \leq x \leq n$ dado, considerando las palabras:

$$M_3 = \{u_1 u_2 \dots u_p (a \bar{a})^{n-p} \bar{u}_p \dots \bar{u}_2 \bar{u}_1 \mid u_i \in \{a, b\}, 0 \leq p \leq n\}$$

podemos asociar a cada irreducible $u_1 u_2 \dots u_x, (n + 1 - x)$ esta dos distintos. Cualquier autómata pequeño para $2nD_2'^*$ tiene, al menos $\sum_{x=0}^n (n + 1 - x)2^x$ estados distintos.

Podemos concluir pues que: $\mu_{D_2'^*}(n) = \Theta(2^{\frac{n}{2}})$

(c) Si consideramos el lenguaje $D_p'^* = \langle S \rightarrow \sum_{i=1}^n a_i S \bar{a}_i S + \epsilon \rangle$

para $p > 1$ tenemos que:

$$\mu_{D_p'^*}(n) = \Theta(p^{\frac{n}{2}})$$

(d) Vamos a presentar otro método de cálculo del índice inicial basado en el autómata minimal infinito que reconoce $D_2'^*$. Dicho autómata tiene un número infinito de estados:

$$Q = \{ (w) \mid w \in \{a,b\}^* \}$$

siendo (ϵ) el único estado a la vez inicial y final.

La función de transición viene dada por:

$$(w).u = (wu) \quad , \quad u \in \{a,b\}$$

$$(wu).\bar{u} = (w) \quad , \quad u \in \{a,b\}$$

La figura III.31 muestra el principio de un tal autómata.

En cada estado se han marcado las longitudes de los caminos que a ellos pueden llegar y que pueden completarse en una palabra de longitud menor o igual a 8. Vemos entonces los distintos irreductibles y el número de veces que pueden aparecer en el proceso de $8D_2^*$. Vemos que para procesar una palabra de longitud 8 debemos recorrer 4 círculos partiendo del estado inicial y llegando al estado inicial del modo que sea. La relación entre la longitud n de la palabra y el número de círculos c recorridos es evidentemente $n = 2c$.

El número de estados de un autómata pequeño para $2cD_2^*$ vendrá dado por el número de estados, con sus respectivos, desdoblamientos que pueden aparecer en el autómata minimal.

En este caso dicho número es evidentemente:

$$\sum_{x=1}^{c+1} x 2^{c+1-x} = \sum_{z=0}^c (c+1-z) 2^z$$

Dado que $\sum_{x=1}^p x 2^{p-x} = \Theta(2^p)$ recuperamos el resultado dado

precedentemente.

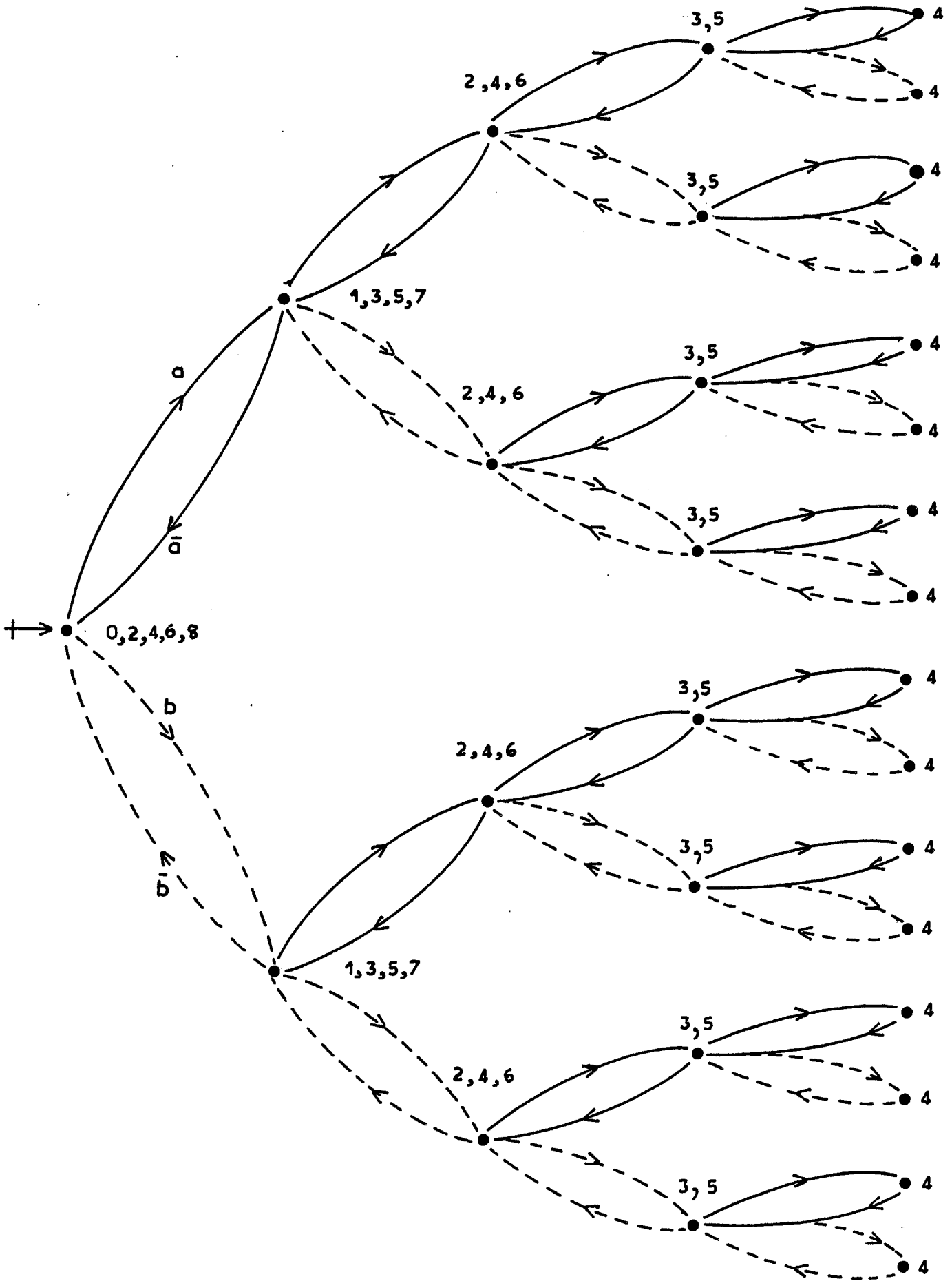


Figura III.31.

Ejemplo 2.15 Vamos a calcular el índice inicial de los factores izquierdos de $D_2'^*$, que notaremos $FG(D_2'^*)$.

Existe también en este caso una factorización análoga a la dada para $D_1'^*$:

$$FG(D_2'^*) = \{ a, b, aD_2'^* \bar{a}, bD_2'^* \bar{b} \}^*$$

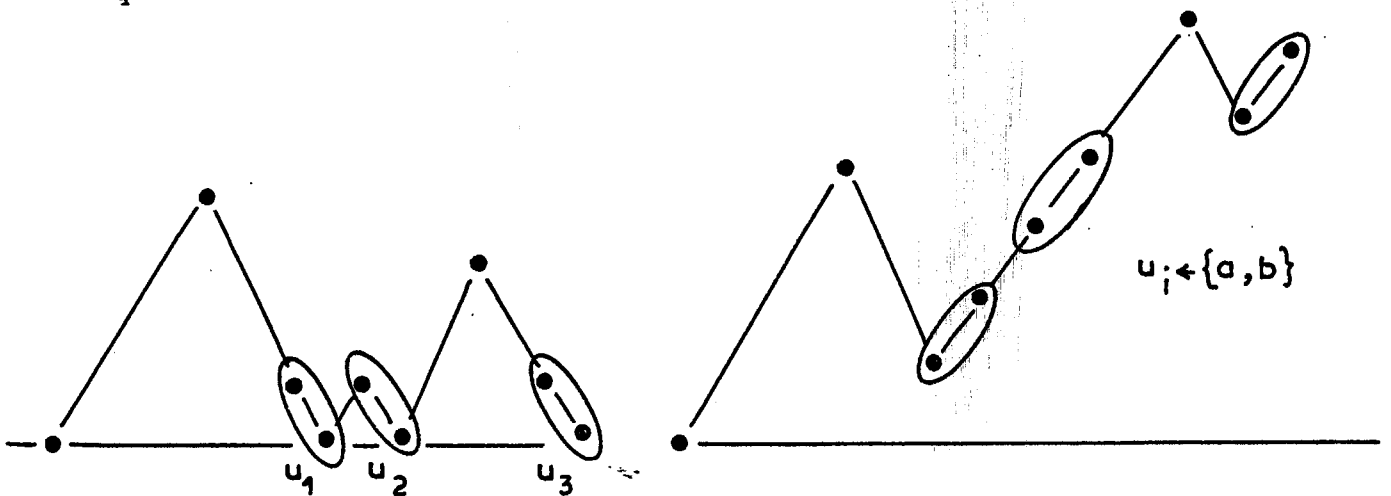
Sin embargo, a fin de hallar un autómata pequeño para $FG(D_2'^*)$, vamos a dar una propiedad de las palabras de Dyck que las relaciona con sus factores izquierdos, trataremos solo el caso par: Sea m un factor izquierdo de longitud par, existe una palabra de Dyck de la misma longitud, que factorizada en palabras primas:

$$d = u_1 d_1 \bar{u}_1 u_2 d_2 \bar{u}_2 \dots u_n d_n \bar{u}_n, \quad u_i \in \{a, b\}, d_i \in D_2'$$

satisface:

$$m \in d [u_i \leftarrow \{ u_i, a, b \}]$$

Esquemáticamente:



El autómata de la figura III.32 muestra un autómata para $6FG(D_2'^*)$. Este autómata es la generalización del autómata dado para $6FG(D_1'^*)$. La generalización de este autómata no presenta ninguna dificultad.

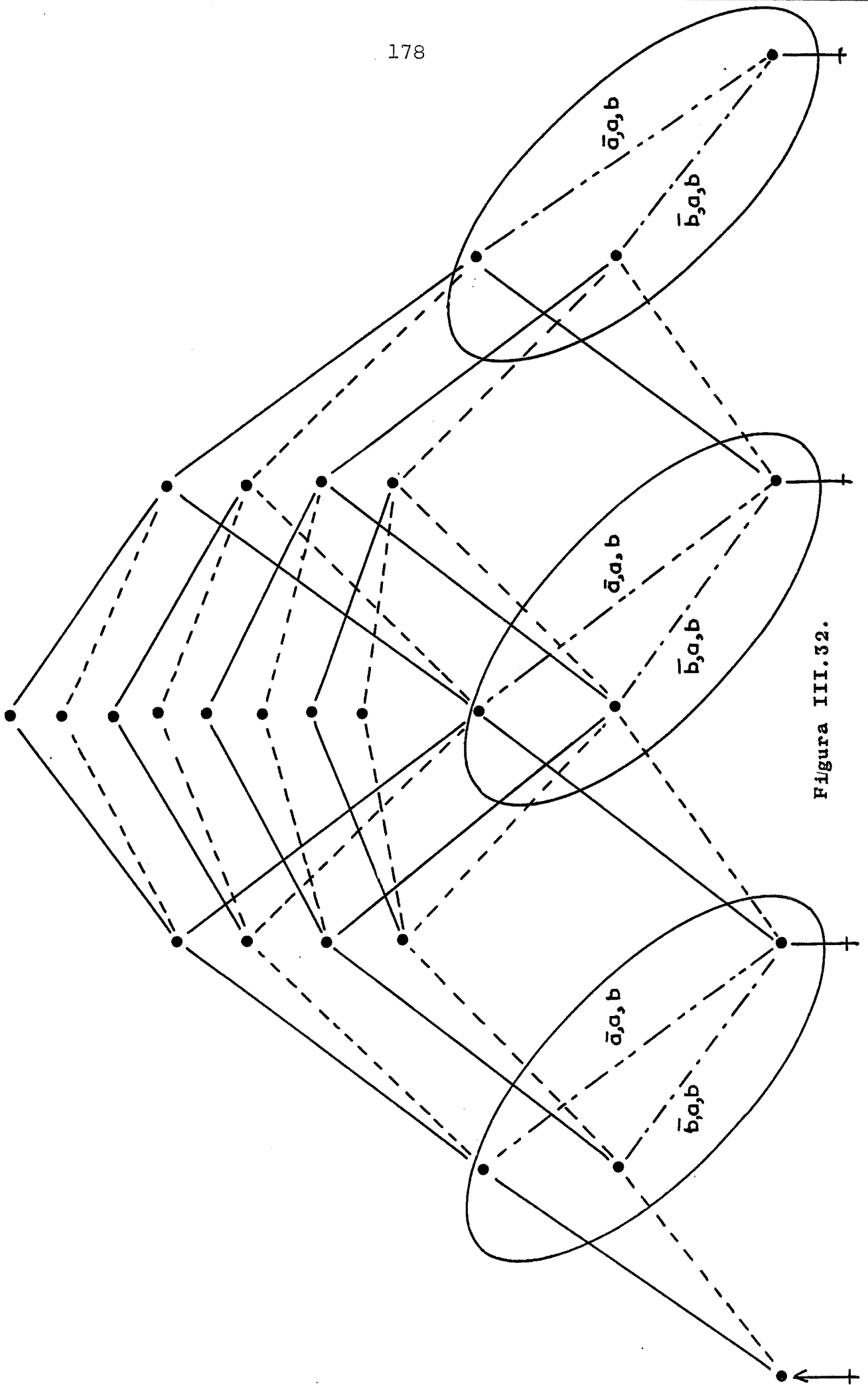


Figura III.32.

Para demostrar que dicho autómata es pequeño, vamos a considerar el conjunto:

$$M_3 = \left\{ (a \bar{a})^{n-p} u_1 u_2 \dots u_p \bar{u}_p \dots \bar{u}_2 \bar{u}_1 \mid u_i \in \{a, b\}, 0 \leq p \leq n \right\}$$

Utilizando este conjunto como en $FG(D_1^*)$ podemos llegar a separar todos los estados del autómata precedente.

$$\text{En consecuencia : } \mu_{FG(D_2^*)}(n) = \sum_{u=0}^p 2^u = \Theta(2^{\frac{n}{2}})$$

Ejemplo 2;16 Vamos a estudiar el índice inicial del lenguaje de Dyck completo de dos letras.

$$D_2^* = \langle S \rightarrow a\bar{a}S + b\bar{b}S + \bar{a}S\bar{a} + \bar{b}S\bar{b} + \epsilon \rangle =$$

$$[\epsilon] \langle a\bar{a} \equiv \bar{a}a \equiv b\bar{b} \equiv \bar{b}b \equiv \epsilon \rangle$$

Dado que los factores izquierdos de D_2^* cubren todo $X^* = \{a, \bar{a}, b, \bar{b}\}^*$ los irreducibles de los factores izquierdos vienen descritos por el conjunto:

$$\text{Irre} = X^* \setminus X^* (a\bar{a} + b\bar{b} + \bar{a}a + \bar{b}b) X^*$$

Un autómata pequeño para D_2^* está dado por:

$$A_n = \langle \{a, \bar{a}, b, \bar{b}\}, Q, q_-, q_+, \cdot \rangle$$

con:

$$Q = \{(w, 2p + |w|) \mid w \in \text{Irre}, 0 \leq p \leq n - |w|\}$$

$$q_- = (\epsilon, 0) \quad q_+ = \{(\epsilon, 2p) \mid 0 \leq p \leq n\}$$

Sea ρ la relación definida por: $(a, \bar{a}), (\bar{a}, a), (b, \bar{b}), (\bar{b}, b)$ entonces:

$$(wx, u) \cdot y = \begin{cases} \text{Si } (x, y) \in \rho & \text{entonces } (w, u + 1) \\ \text{sino} & (wxy, u + 1) \end{cases}$$

La figura III.33.a., muestra dicho autómata para $n = 2$.

Vamos a contar el número de estados utilizando el autómata minimal de D_2^* ; procedemos de modo análogo al empleado en $D_2'^*$, dicho proceso se halla esquematizado en la figura III.33.b., en la que tratamos, también, las palabras de longitud menor o igual a 8.

La relación entre c y n , viene dada, como anteriormente, por $c = n$.

Por inspección de la figura III.33.b. llegamos a:

$$\mu_{D_2}(2n) = 4 \left(\sum_{x=1}^c x 3^{c-x} \right) + 1 = \Theta \left(3^{\frac{n}{2}} \right)$$

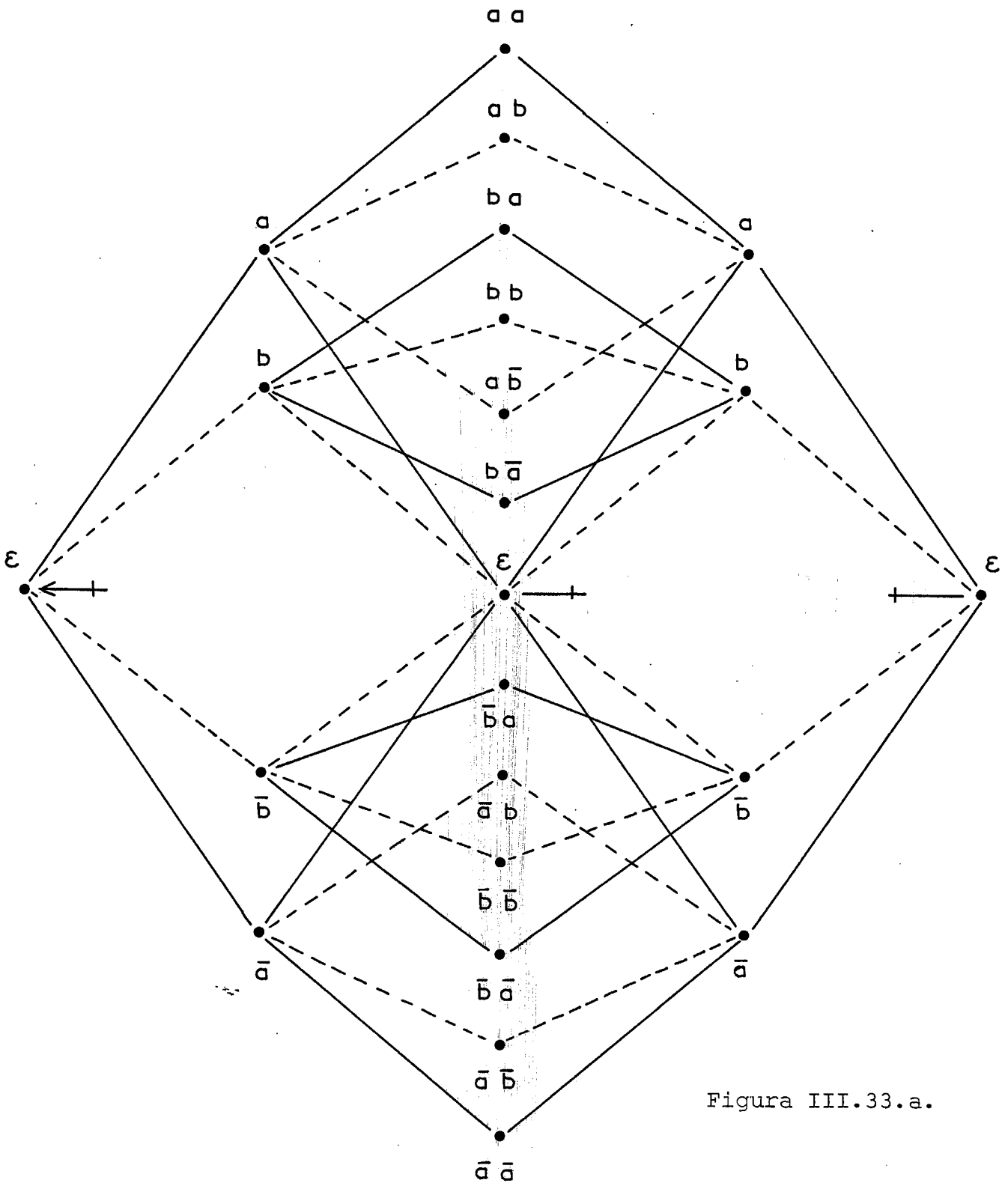


Figura III.33.a.

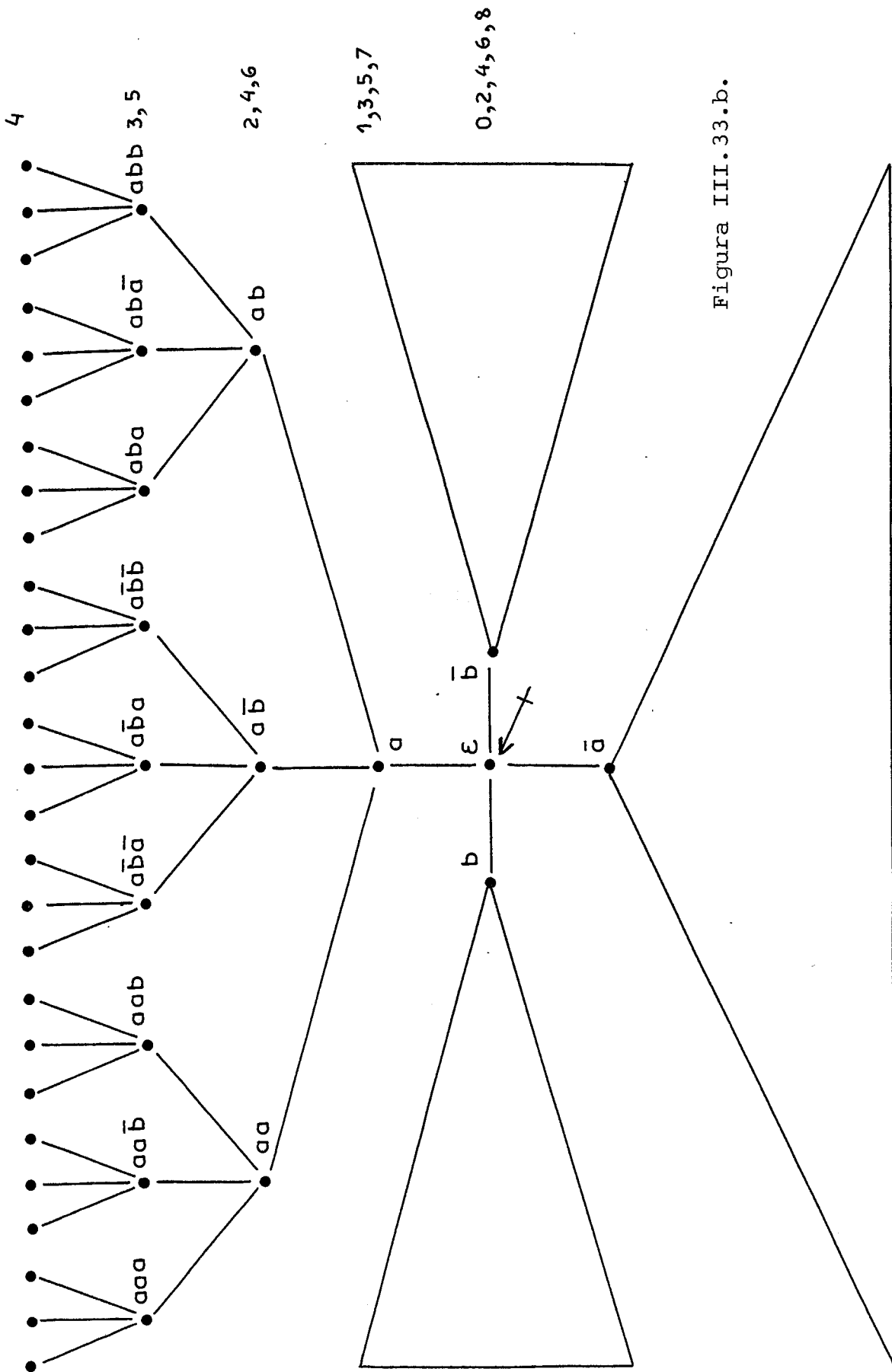


Figura III.33.b.

Ejemplo 2.17 Vamos a estudiar el comportamiento de FIFO_2 , [11]

Dicho lenguaje viene definido como la clase de \mathcal{E} en la congruencia α definida para palabras en $\{a, b, \bar{a}, \bar{b}\}^*$ por:

$$w_1 \equiv_{\alpha} w_2 \iff w_1 = ax\bar{a}y \text{ ó } w_1 = bx\bar{b}y, w_2 = xy, x \in \{a, b\}^*$$

(a) Un autómata pequeño para dicho lenguaje viene dado por:

$$A_n = \langle \{a, b, \bar{a}, \bar{b}\}, Q, q_-, Q_+, \delta \rangle$$

$$Q = \{(w, 2p + |w|) \mid w \in \{a, b\}^*, 0 \leq p \leq n - |w|\}$$

$$q_- = (\mathcal{E}, 0), \quad Q_+ = \{(\mathcal{E}, 2p) \mid 0 \leq p \leq n\}$$

$$(w, y) \cdot u = (wu, y + 1), \quad 2n > |w| + y, \quad u \in \{a, b\}$$

$$(uw, y) \cdot \bar{u} = (w, y + 1), \quad u \in \{a, b\}$$

En la figura III.34 damos A_n para $n = 3$.

Los mismos razonamientos dados en D_2^* permiten decir que

$$L(A_n) = 2n \text{ FIFO}_2.$$

$$\text{Evidentemente } \|Q\| = \sum_{x=0}^n (n+1-x)2^x = \Theta(2^n)$$

(b) Demostremos que el autómata dado anteriormente es pequeño.

Sabemos que todo irreducible de un factor izquierdo de FIFO_2 está en $\{a, b\}^*$. Notemos que tenemos la igualdad siguiente:

$$\begin{aligned} \{a, b\}^x &= \{uw \mid u \in \{a, b\}, w \in \{a, b\}^{n-1}\} = \\ &= \{\alpha(uwvu) \mid u, v \in \{a, b\}, w \in \{a, b\}^{n-1}\}. \quad (*) \end{aligned}$$

Sabemos que a todo estado podemos asociar un único irreducible, vamos a demostrar que al conjunto de irreducibles $\{a, b\}^x$

$0 \leq x \leq n$ podemos asociar $(n+1-x)2^x$ estados distintos.

Para ello consideremos los conjuntos de estados $Q_- \cdot X^x (XX)^i = Q_{x+2i}$ donde $X = \{a, b\}$, $0 \leq i \leq n-x$. En particular:

$$Q_x = Q_- \cdot X^x = \{q \cdot w \mid q \in Q_-, w \in X^x\}, \text{ estos estados tienen}$$

por irreducibles asociados el conjunto X^x , en consecuencia

$$\|Q_x\| \geq 2^x.$$

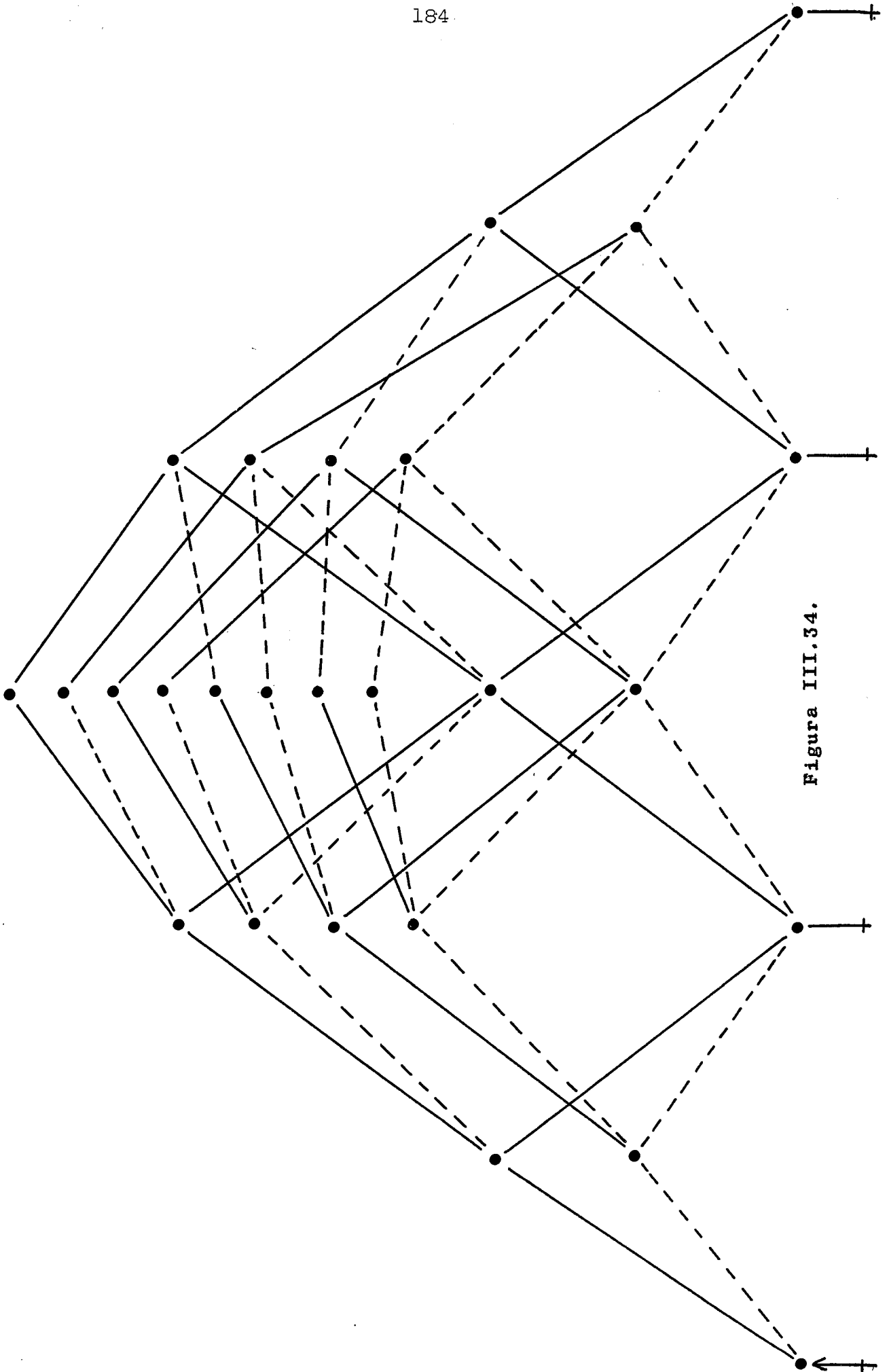


Figura III.34.

$$Q_{x+2} = Q_{-} \cdot X^x (X\bar{X}) = \{ q \cdot w u \bar{v} \mid q \in Q_{-}, w \in X^x, u \in \{a, b\}, \bar{v} \in \{\bar{a}, \bar{b}\} \}$$

en virtud de la igualdad (*) la función de transición

$q \cdot w u \bar{v}$ solo estará definida cuando $w = v w'$, con lo que Q_{x+2}

tiene también como irreducibles asociados el conjunto X^x , es
to es cierto para todo los Q_{x+2i} .

Considerando las longitudes de las palabras aceptadas por cada
 Q_{x+2i} es fácil demostrar que $Q_{x+2i} \cap Q_{x+2j} = \emptyset$ si $i \neq j$.

Al conjunto de irreducibles X^x tenemos como mínimo, asociado

un número de estados igual a $\sum_{i=0}^{n-x} \|Q_{x+2i}\| = (n+1-x)2^x$.

La figura III.35 muestra Q_{x+2i} para $x = 2$ en le caso $n = 5$.

En consecuencia:

$$\mu_{\text{FIFO}_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$$

(c) Notemos que $\text{FIFO}_1 = D_1^*$.

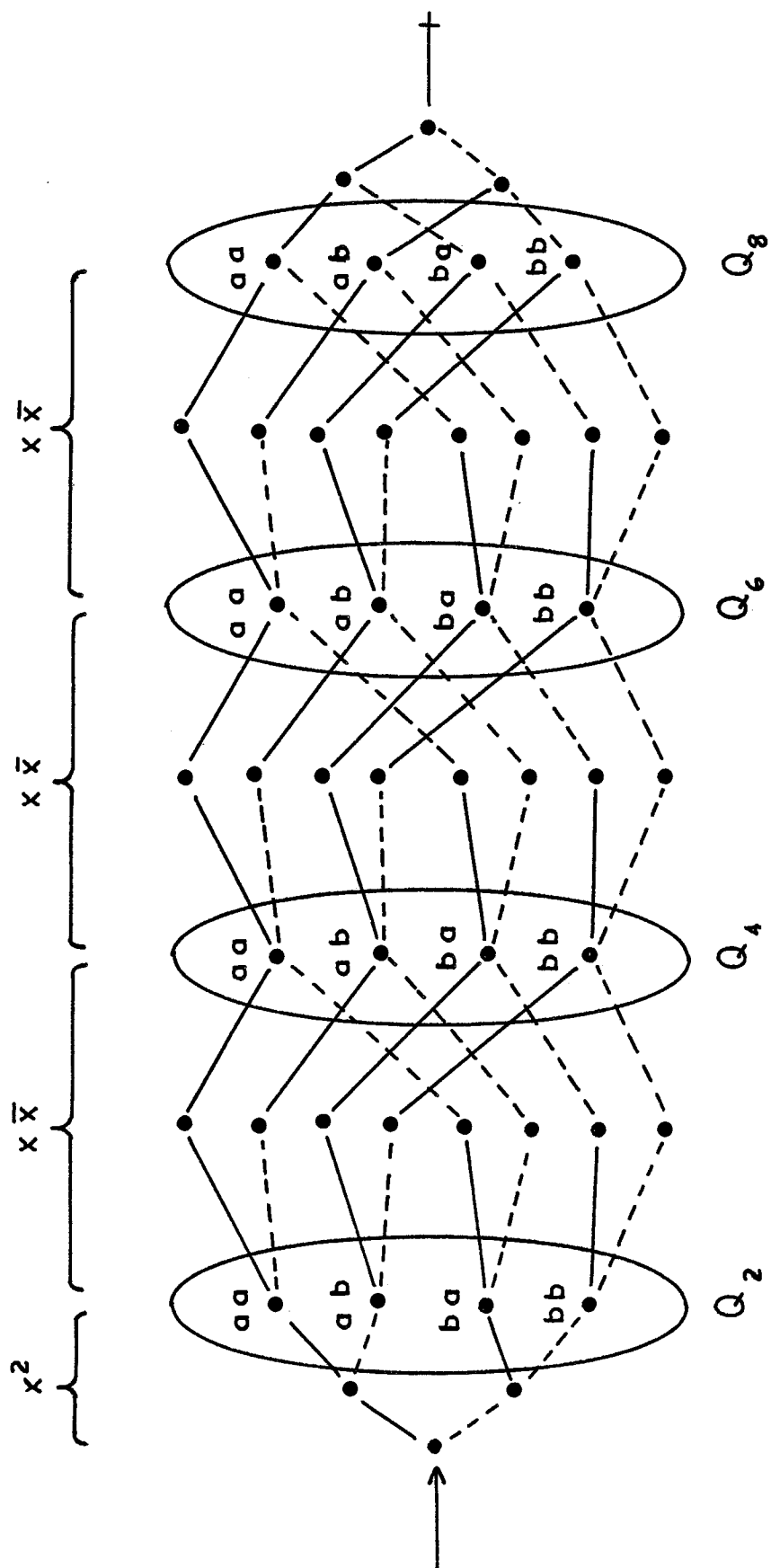


Figura III.35.

Ejemplo 2.18 Vamos ahora a estudiar el lenguaje $TWIN_2$ definido en [11] como la clase \mathcal{E} en la congruencia β definida en $\{a, b, \bar{a}, \bar{b}\}^*$ por:

$$w_1 \equiv w_2 \begin{cases} w_1 = axay \quad \text{ó} \quad w_1 = bxb y & , w_2 = xy & , x \in \{a, b\}^* \\ w_1 = auav \quad \text{ó} \quad w_1 = bubv & , w_2 = uv & , u \in \{\bar{a}, \bar{b}\}^* \end{cases}$$

Veamos la estructura de los irreducibles en los factores izquierdos de $TWIN_2$. Si una palabra de $X = a, b, a, b$ comienza por a , mientras no encontremos una a no podremos reducir, es decir, aX está contenido entre los irreducibles. Tenemos:

$$\text{Irre} = \bigcup_{x \in X} x(X \setminus x)^*$$

Un autómata pequeño para $2n$ $TWIN_2$ viene dado :

$$A_n = \langle X, Q, q_-, q_+, \cdot \rangle$$

con:

$$Q = \{(w, 2p + |w|) \mid w \in \text{Irre}, 0 \leq p \leq n - |w|\}$$

$$q_- = (\mathcal{E}, 0), \quad q_+ = \{(\mathcal{E}, 2p) \mid 0 \leq p \leq n\}$$

Tomemos la misma relación ρ de antes

$$\rho = \{(a, \bar{a}), (b, \bar{b}), (\bar{a}, a), (\bar{b}, b)\}$$

La función de transición está dada por:

$$(xw, u). y = \begin{cases} \text{Si } (x, y) \in \rho \text{ entonces } (w, u + 1) \\ \text{sino } (xwy, u + 1) \end{cases}$$

La figura III.36.a. da A_n para $n = 2$.

En la figura III.36.b. vemos que podemos utilizar el mismo tipo de análisis que el empleado en D_2^* y llegamos a:

$$\mu_{TWIN_2}(2n) = 4 \left(\sum_{x=1}^n x 3^{n-x} \right) + 1 = \Theta \left(3^{\frac{n}{2}} \right)$$

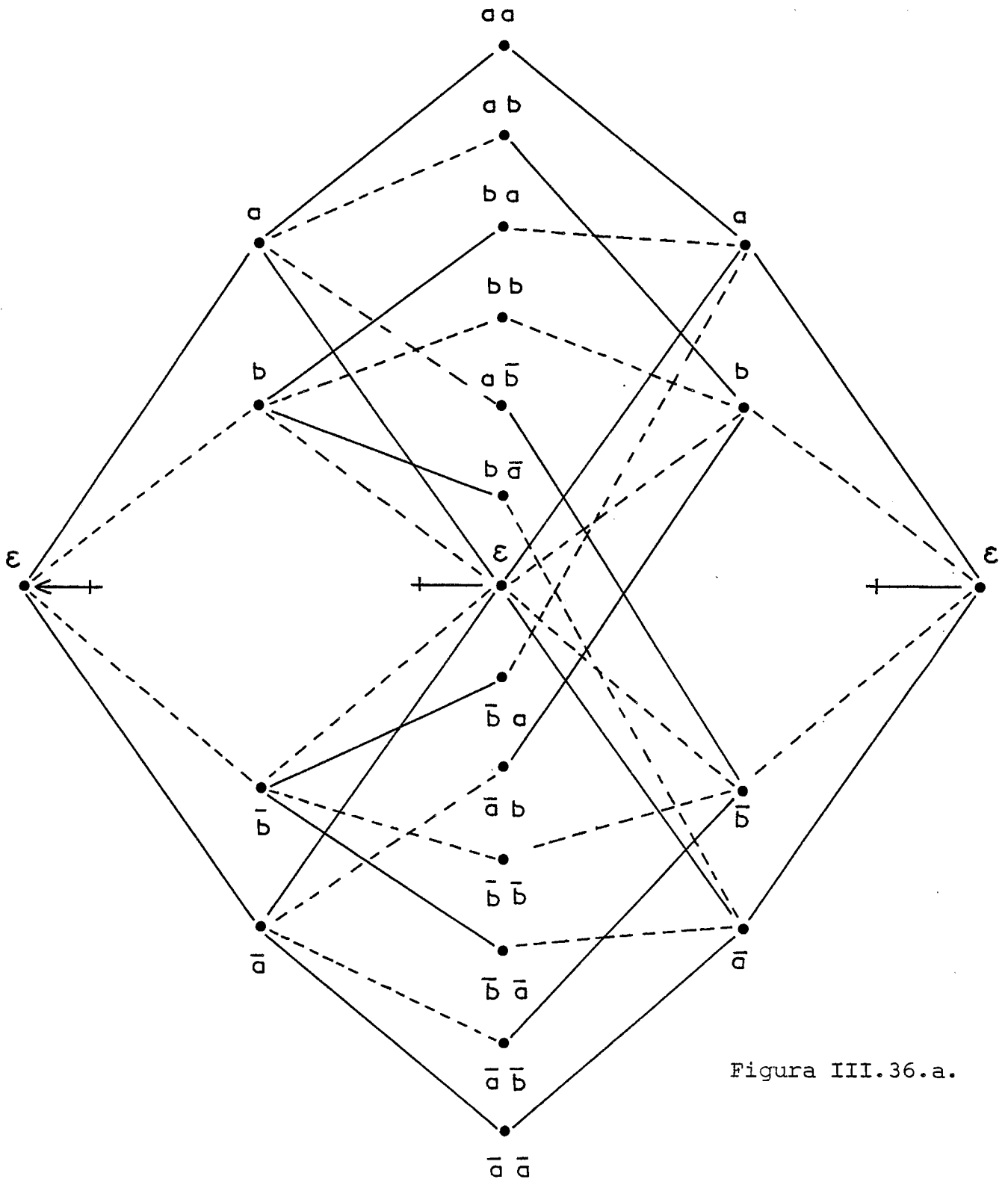


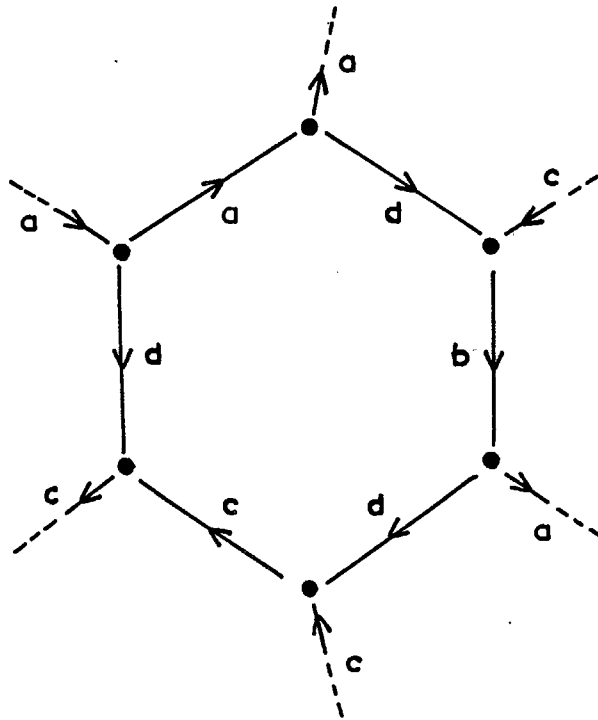
Figura III.36.a.

Ejemplo 2.19 Vamos a buscar el índice inicial del lenguaje E caracterizado por:

(1) $E = \langle S \rightarrow aSbSc + d \rangle$

(2) La clase de d en la congruencia γ dada en $X = \{a, b, c, d\}$ por $adbdc \stackrel{\gamma}{\equiv} d$.

(a) Vamos a calcular el índice inicial utilizando el autómata minimal infinito de dicho lenguaje. Contaremos los distintos estados que pueden aparecer con sus respectivas multiplicidades. En la figura III.37 hemos tratado el caso 17E. Vemos que en dicho autómata la celda básica es un exágono que puede esquematizarse:



La relación entre el número de exágonos y la longitud de la palabra viene dada por $n = 4e + 1$.

Por inspección vemos que el número de estados es:

$$\sum_{x=1}^{e+1} \{x2^{e+2-x} - (x-1)\}, \text{ dado que:}$$

$$\sum_{x=1}^p x2^{p-x} = \Theta(2^p) \text{ tenemos: } \mu_E(n) = \Theta(2^{\frac{n}{4}})$$

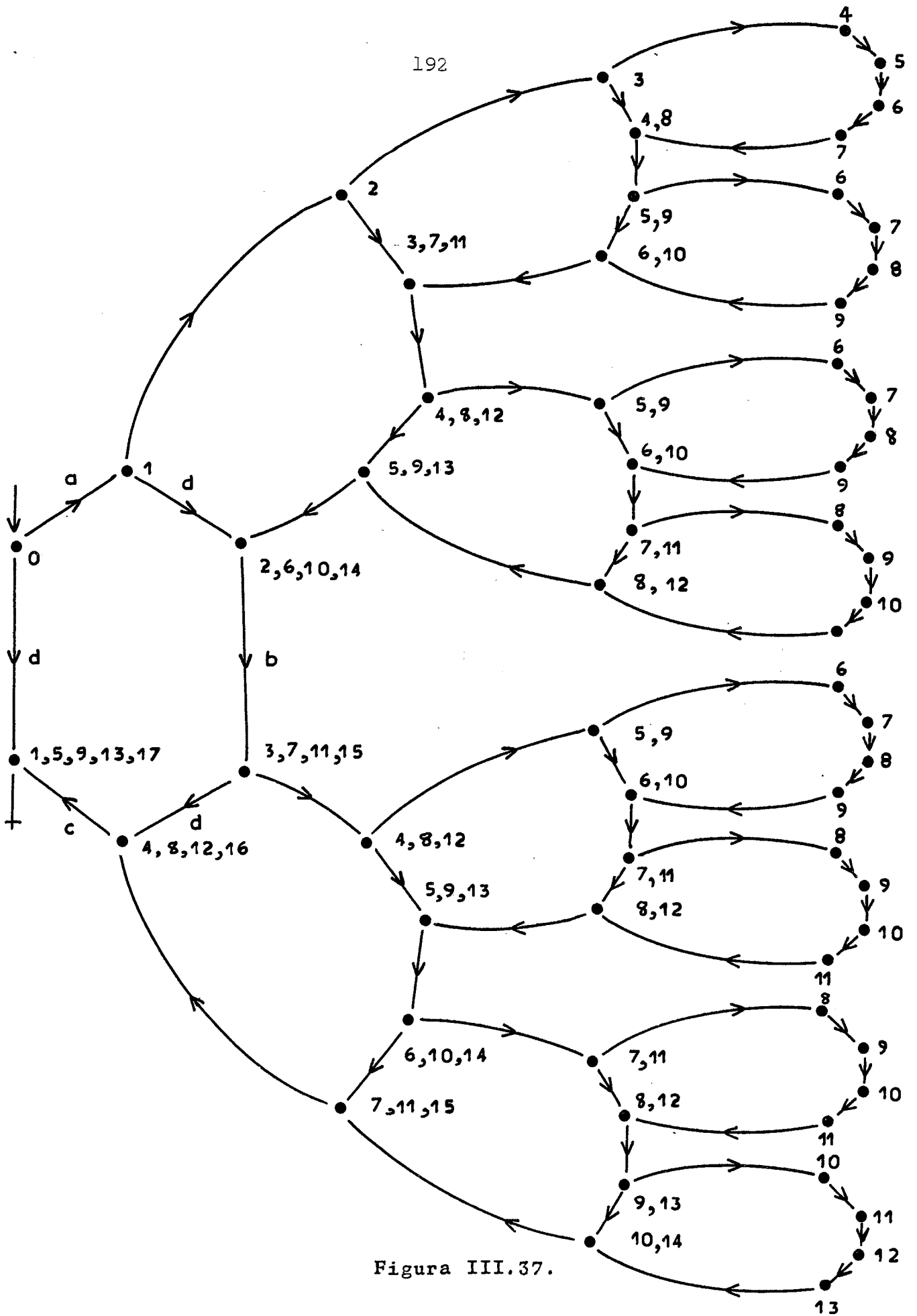


Figura III.37.

Ejemplo 2.20 Vamos a estudiar el índice inicial de un lenguaje dado por S.Greibach en [35], definido por:

$$L = \{ w \# a^{V(w)} \mid w \in 1(0+1)^* , V(w): \text{número en binario} \}$$

(a) En la figura III.38 presentamos un autómata para L , la generalización de dicho autómata no presenta dificultad.

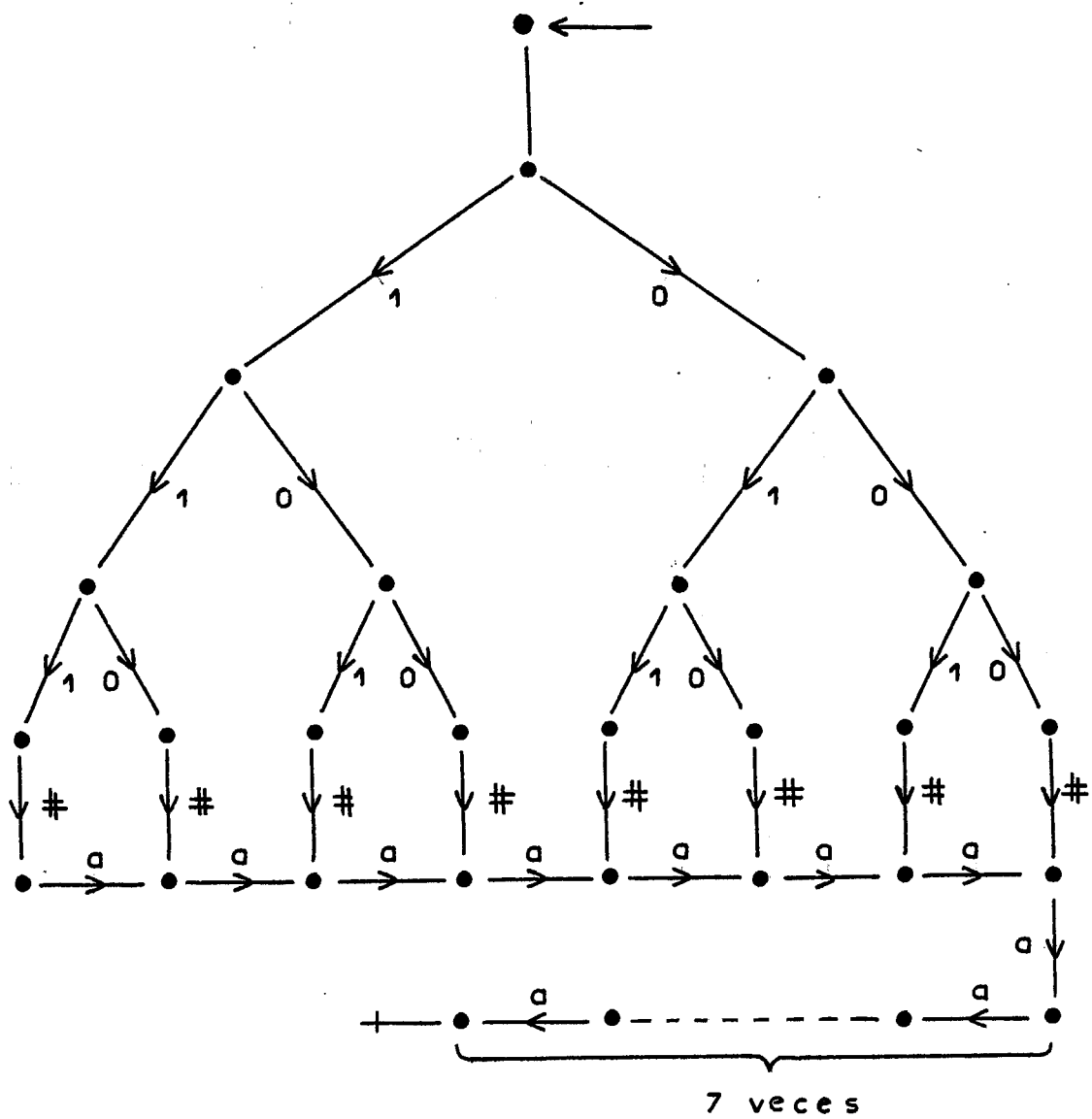


Figura III.38.

(b) Vamos a demostrar que L tiene un índice inicial lineal. Para ello estudiemos primero la longitud de $w \# a^{V(w)}$ en función de la longitud de w .

Sabemos que si $w \in 1(0+1)^{x-1}$ entonces $|w| = x$ y $2^{x-1} \leq V(w) \leq 2^x - 1$.

$$|w \# a^{V(w)}| = |w| + 1 + V(w) \leq x + 1 + 2^x - 1 = x + 2^x$$

$$|w \# a^{V(w)}| = |w| + 1 + V(w) \geq x + 1 + 2^{x-1}$$

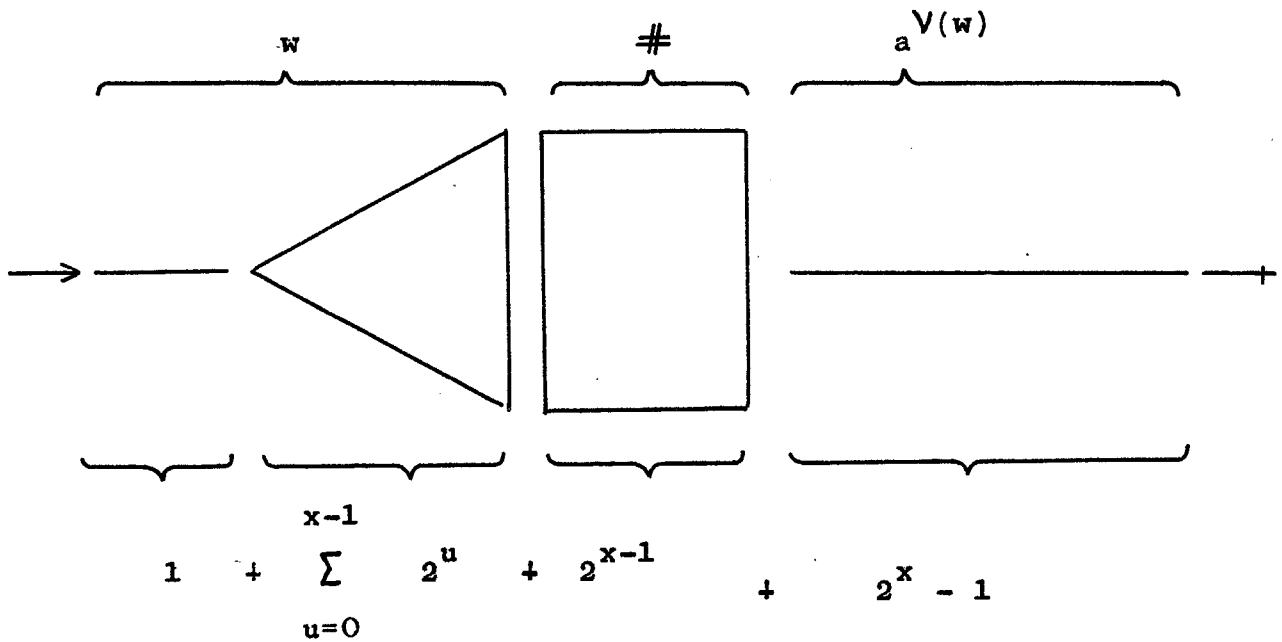
entonces:

$$x + 1 + 2^{x-1} \leq |w \# a^{V(w)}| \leq x + 2^x$$

en consecuencia:

$$|w \# a^{V(w)}| = \Theta(2^x)$$

Estimemos seguidamente el número de estados:



Tenemos que el número de estados es $\Theta(2^x)$.

Dado que tanto la longitud como el número de estados son $\Theta(2^x)$ tenemos:

$$\mu_L(n) = \Theta(n)$$

Las familias Pol μ y Exp μ

Considerando el comportamiento asintótico del índice inicial podemos definir nuevas familias de lenguajes.

Definición 2.1:

.Sea Pol μ la familia de lenguajes de índice inicial como máximo un polinomio, formalmente:

$$\text{Pol } \mu = \{ L \mid \exists k \text{ que satisface } \mu_L(n) = O(n^k) \}$$

.Sea Exp μ la familia de lenguajes de índice inicial como mínimo una exponencial, formalmente:

$$\text{Exp } \mu = \{ L \mid \exists k > 0 \text{ que satisface } \mu_L(n) = \Omega(2^{kn}) \}$$

Podemos resumir todos los ejemplos dados anteriormente en el siguiente teorema.

Teorema 2.1

(a) Los siguientes lenguajes son polinomiales y su índice inicial es:

$$\cdot S_1 = \{ a^n b^n \mid n \geq 0 \} \quad ; \quad \mu_{S_1}(n) = \Theta(n)$$

$$\cdot S_{>} = \{ a^n b^p \mid n \geq p \} \quad ; \quad \mu_{S_{>}}(n) = \Theta(n)$$

$$\cdot S_{<} = \{ a^n b^p \mid n \leq p \} \quad ; \quad \mu_{S_{<}}(n) = \Theta(n)$$

$$\cdot L = \{ a^n b^n c^n \mid n \geq 0 \} \quad ; \quad \mu_L(n) = \Theta(n^2)$$

$$\cdot \text{GOLD} = \{ a^{i_1} b a^{i_2} b \dots a^{i_p} b \mid p \geq 1 \ ; \ \exists j : 1 \leq j \leq p, i_j \neq j \}$$

$$\mu_{\text{GOLD}}(n) = \Theta(n)$$

- $D_1^* = \langle S \rightarrow aS\bar{a}S + \epsilon \rangle$; $\mu_{D_1^*}(n) = \Theta(n^2)$
- $FG(D_1^*) = \{ a, aD_1^* \bar{a} \}^*$; $\mu_{FG(D_1^*)}(n) = \Theta(n^2)$
- $D_1^* = \langle S \rightarrow aS\bar{a}S + \bar{a}S aS + \epsilon \rangle$; $\mu_{D_1^*}(n) = \Theta(n^2)$
- $L = \{ w \in \{a, b\}^* \mid 2|w|_a = 3|w|_b \}$; $\mu_L(n) = \Theta(n^2)$
- $L = \{ w \neq a^{V(w)} \mid w \in 1(0+1)^* , \forall(w) : \text{número binario} \}$
 $\mu_L(n) = \Theta(n)$

(b) Los siguientes lenguajes son exponenciales y su índice inicial es:

- $S_2 = SYM_2 = \langle S \rightarrow aS\bar{a} + bS\bar{b} + \epsilon \rangle$; $\mu_{SYM_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $PAL_2 = \langle S \rightarrow aSa + bSb + \epsilon \rangle$; $\mu_{PAL_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $COPY_2 = \{ w\bar{w} \mid w \in \{a, b\}^* ; \bar{w} \text{ es una copia barrada de } w \}$
 $\mu_{COPY_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $SQR_2 = \{ ww \mid w \in \{a, b\}^* \}$; $\mu_{SQR_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $D_2^* = \langle S \rightarrow aS\bar{a} + bS\bar{b} + \epsilon \rangle$; $\mu_{D_2^*}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $FG(D_2^*) = \{ a, b, aD_2^* \bar{a}, bD_2^* \bar{b} \}^*$; $\mu_{FG(D_2^*)}(n) = \Theta\left(2^{\frac{n}{2}}\right)$
- $D_2^* = \langle S \rightarrow aS\bar{a} + \bar{a}Sa + bS\bar{b} + \bar{b}Sb + \epsilon \rangle$
 $\mu_{D_2^*}(n) = \left(3^{\frac{n}{2}}\right)$

• Sea α la congruencia sobre $\{a, b, \bar{a}, \bar{b}\}^*$ por
 $w_1 \equiv_{\alpha} w_2 \iff w_1 = ax\bar{a}y \text{ ó } w_1 = bx\bar{b}y , w_2 = xy , x \in \{a, b\}^*$

entonces $FIFO_2 = [\epsilon]_{\alpha}$; $\mu_{FIFO_2}(n) = \Theta\left(2^{\frac{n}{2}}\right)$

. Sea β la congruencia definida por $\{a, b, \bar{a}, \bar{b}\}^*$ por:

$$w_1 \stackrel{\beta}{\equiv} w_2 \quad \left\{ \begin{array}{l} w_1 = ax\bar{a}y \text{ ó } w_1 = bx\bar{b}y \text{ ; } w_2 = xy \text{ , } x \in \{a, b\}^* \\ w_1 = \bar{a}xay \text{ ó } w_2 = \bar{b}xby \text{ ; } w_2 = xy \text{ , } x \in \{\bar{a}, \bar{b}\}^* \end{array} \right.$$

entonces $TWIN_2 = [\varepsilon]_\beta$; $\mu_{TWIN_2}(n) = \Theta\left(3^{\frac{n}{2}}\right)$

. $E = \langle S \rightarrow aSbSc + d \rangle$; $\mu_E(n) = \Theta\left(2^{\frac{n}{4}}\right)$

III.3. Índice inicial y su relación con las operaciones de lenguajes.

Vamos a estudiar en este apartado el comportamiento del índice inicial respecto a diversas operaciones entre lenguajes, por ejemplo:

- . operaciones racionales
- . operaciones booleanas
- . operaciones de trio y transducción fiel
- . operaciones de substitución y substitución sintáctica
- . operación de paso al centro

Obtendremos que el índice inicial se halla relacionado en ta les operaciones, excepto en la complementación y el paso al centro de manera polinomial.

Se estudiará en la mayoría de las operaciones precedentes la validez de los límites obtenidos, demostrándose su exactitud al menos asintóticamente.

Cuando aplicamos todos estos resultados a los lenguajes polinomiales obtenemos que dicha familia es un A.F.L. cerrado por intersección y abierto por complementación y paso al centro.

Operaciones racionales Vamos a estudiar el comportamiento del índice inicial con respecto a las operaciones racionales: unión, producto, estrella. Para ello estudiaremos previamente el comportamiento de los segmentos iniciales en tales operaciones.

Recordemos rápidamente algunas definiciones de autómatas [18]. Consideremos el autómata $A = \langle X, Q, Q_-, Q_+, \delta \rangle$ diremos que es accesible si $Q_- \cdot X^* = Q$, coaccesible si para todo q tenemos $q \cdot X^* \cap Q_+ \neq \emptyset$, de bucle w si existe q tal que $q \subset q \cdot w$. Vamos a dar seguidamente un lema de estandarización para los autómatas pequeños.

Lema 3.1 Consideremos L contenido en X^* , para todo n , existe un autómata pequeño A_n que reconoce nL y que satisface:

- (a) A_n es accesible y coaccesible
- (b) A_n no tiene bucles
- (c) Tiene un solo estado inicial que notaremos q_i
- (d) Si L no contiene ϵ tiene un solo estado final distinto de q_i que notaremos q_f . Si L contiene ϵ además de q_f , q_i es estado final.
- (e) No existe ningún arco (q, x, q_i) ni (q_f, x, q) .

Prueba Sea $B = \langle X, Q, Q_-, Q_+, \delta \rangle$ un autómata pequeño para nL vamos a construir a partir de B el autómata A_n .

(a) Si \mathcal{B} contiene algún estado que no es a la vez accesible y coaccesible podemos eliminarlo sin cambio el lenguaje engendrado. Dado que el número de estados es mínimo, la existencia de tales estados no es posible.

(b) Si siendo a la vez accesible y coaccesible existiese un bucle, dicho autómata reconocería un lenguaje infinito, hecho contradictorio dado que $||nL|| \leq ||X||^n$.

(c) Sea w_1 una palabra de longitud máxima en nL , en el autómata \mathcal{B} existe un camino (q_1, w_1, q_2) en que $q_1 \in Q_-$ y $q_2 \in Q_+$. Supongamos que exista un arco (q_3, x, q_1) , dado que \mathcal{B} es accesible debe existir un $q_4 \in Q_-$ tal que (q_4, w_2, q_3) .

Tenemos entonces que $w_2 x w_1$ es aceptada por el autómata y su longitud es estrictamente superior a w_1 , con lo que llegamos a una contradicción. Tenemos entonces que al estado q_1 no entra ningún arco.

Supongamos que q_1' es otro estado inicial, consideremos el conjunto de arcos que salen de dicho estado i.e. (q_1', y, q_1'') y para cada uno de ellos añadamos un arco tipo (q_1, y, q_1'') . Repitiendo dicha transformación para todo estado, inicial construimos un autómata con q_1 como único estado inicial. A partir de ahora diremos $q_1 = q_i$.

(d) Con las mismas notaciones que en (c), vemos que ningún arco puede salir de q_2 , sino por coaccesibilidad podemos encontrar a una palabra más larga que w_1 .

Sea q_2' un estado final distinto de q_2 . Si tenemos el arco

(q_2'', y, q_2') en \mathcal{B} añadiremos (q_2'', y, q_2) , repitiendo dicha operación podemos considerar q_2 como estado final en vez de q_2'' . No taremos $q_2 = q_f$

Si nL contiene ϵ añadiremos q_i como estado final.

(e) Por construcción no existen arcos (q, x, q_i) ni (q_f, x, q) .

En las demostraciones siguientes supondremos que los autómatas pequeños satisfacen las propiedades del lema precedente.

Lema 3.2 Sea L_1 y L_2 dos lenguajes en X_1^* y X_2^* respectivamente, tenemos:

$$\mu_{L_1 \cup L_2}(n) \leq \mu_{L_1}(n) + \mu_{L_2}(n) - 2$$

Prueba Evidentemente $n(L_1 \cup L_2) = nL_1 \cup nL_2$.

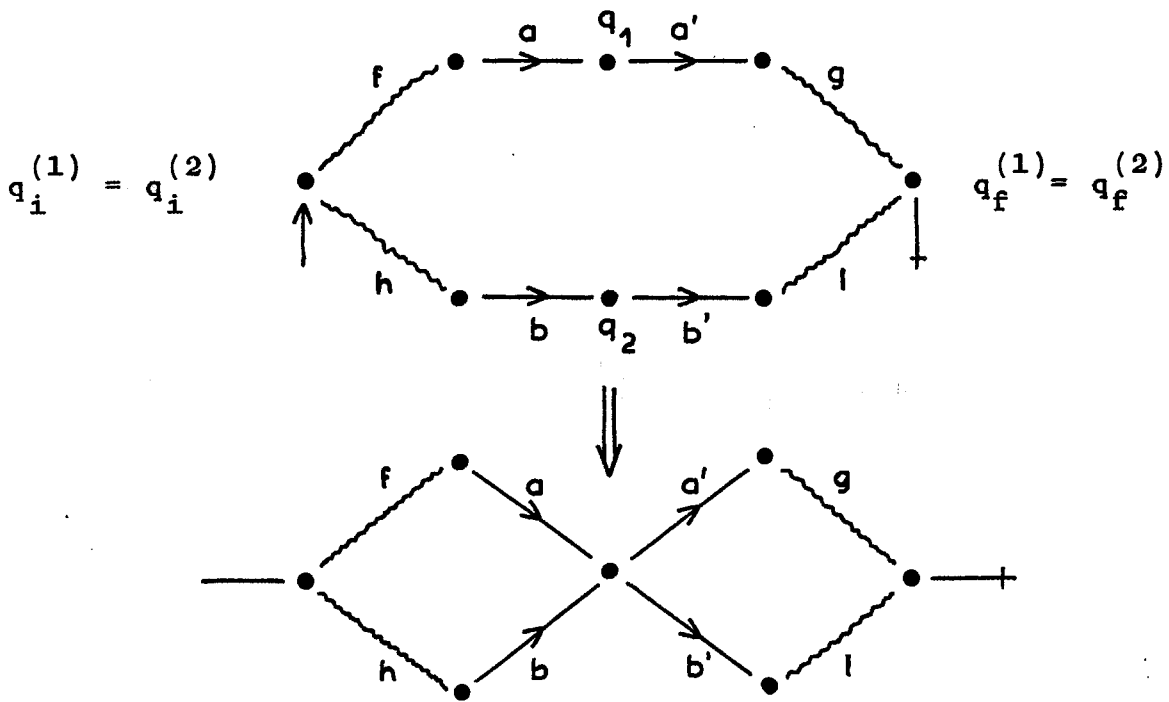
Sea $a_n^{(1)} = \langle X, Q^{(1)}, q_i^{(1)}, q_f^{(1)}, \delta^{(1)} \rangle$ un autómata pequeño para nL_1 y $a_n^{(2)} = \langle Y, Q^{(2)}, q_i^{(2)}, q_f^{(2)}, \delta^{(2)} \rangle$ un autómata pequeño para nL_2 . El autómata que se obtiene confundiendo $q_i^{(1)}$ con $q_i^{(2)}$ y $q_f^{(1)}$ con $q_f^{(2)}$ reconoce $nL_1 \cup nL_2$ y su número de estados es $\mu_{L_1}(n) + \mu_{L_2}(n) - 2$.

Hecho 3.1 Es importante saber si el límite obtenido para el comportamiento del índice inicial en la unión, es óptimo.

Para ello consideremos que si $X_1 \cap X_2 = \emptyset$ entonces:

$$\mu_{L_1 \cup L_2}(n) = \mu_{L_1}(n) + \mu_{L_2}(n) - 2$$

En efecto; supongamos que podamos confundir q_1 en $a_n^{(1)}$ con q_2 en $a_n^{(2)}$, tendremos entonces:



Por construcción $faa'g \in nL_1$, $hbb'l \in nL_2$, pero

$$fab'l \notin nL_1 \cup nL_2$$



Vamos ahora a estudiar la operación producto.

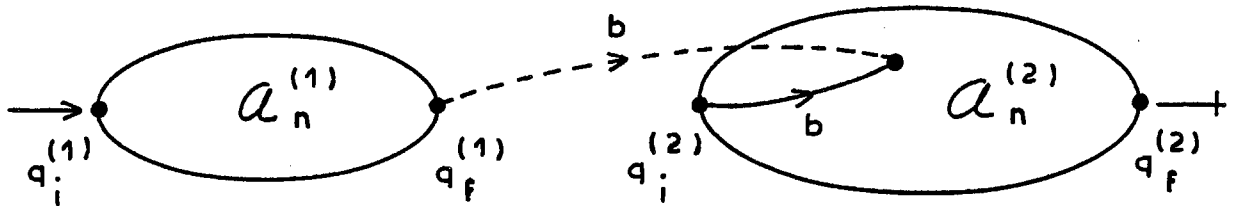
Lema 3.3 Sean L_1 y L_2 dos lenguajes en X_1^* y X_2^* respectivamente, tenemos:

$$\mu_{L_1 \cdot L_2}(n) \leq (n+1)(\mu_{L_1}(n) + \mu_{L_2}(n))$$

Prueba Sea $w \in n(L_1 \cdot L_2)$, w puede factorizarse en $w = w_1 \cdot w_2$ con $w_1 \in L_1, w_2 \in L_2$ con $|w_1| \leq n, |w_2| \leq n$. En consecuencia:

$$n(L_1 \cdot L_2) \subset nL_1 \cdot nL_2$$

Realizemos la construcción esquematizada en la figura siguiente para el caso en que L_1 no contiene ϵ (Cuando $\epsilon \in L$, hay pequeñas variaciones):



El autómata así construido reconoce $nL_1 \cdot nL_2$ y tiene $\mu_{L_1}(n) + \mu_{L_2}(n)$ estados.

Notemos que $n(L_1 \cdot L_2) = nL_1 \cdot nL_2 \cap n(X_1 \cup X_2)^*$.

El lenguaje $n(X_1 \cup X_2)^*$ es reconocido por un autómata a $(n+1)$ estados. Sabemos construir un autómata que reconoce $n(L_1 \cdot L_2)$ con $(n+1)(\mu_{L_1}(n) + \mu_{L_2}(n))$ estados.

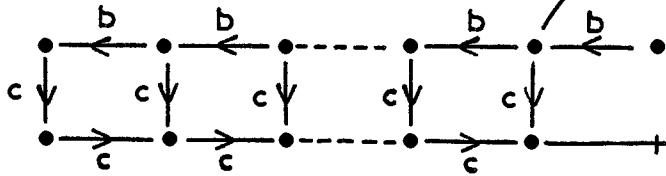
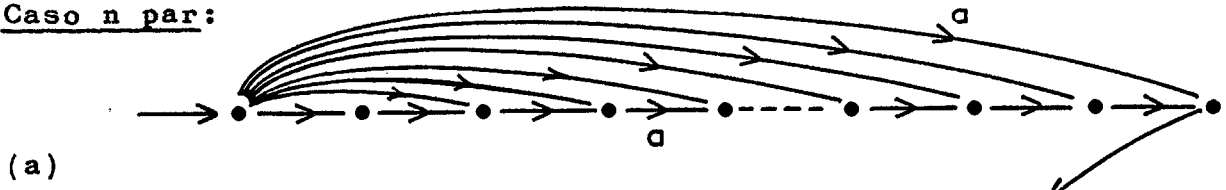
Hecho 3.2 Vamos a estudiar la optimalidad del límite precedente. Para ello vamos a analizar con todo detalle un caso concreto.

Sea $L_1 = a^+$ y $L_2 = \{ b^n c^n \mid n > 0 \}$. La figura III.39(a) muestra la construcción del lema 3.3 para dichos lenguajes. A fin de encontrar una aproximación de un autómata pequeño para $L_1.L_2$ es preciso hacer la intersección de los autómatas (a) y (b) de la figura III.39, las partes (c) y (d) de dicha figura muestran el resultado de dicha intersección. Evidentemente el autómata obtenido tiene $\Theta(n^2)$ estados. En consecuencia podemos afirmar que $\mu_{L_1.L_2}(n) = O(n^2)$ estados.

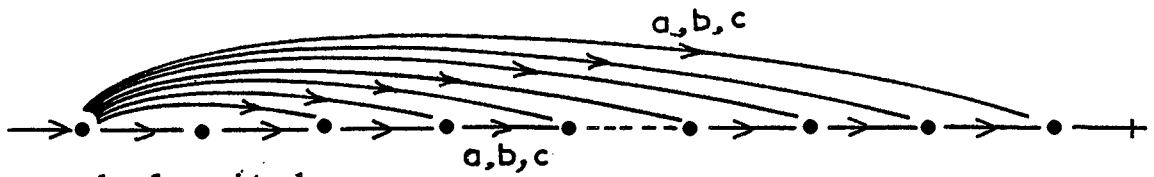
La figura III.40 (a) muestra que el autómata obtenido está lejos del óptimo cuando tenemos un número ^{par} de letras a. En dicha figura se han encerrado en un círculo todos los estados que pueden confundirse. La parte (b) muestra el resultado de dicha minimización. Un resultado análogo existe cuando tratamos con un número impar de letras a. La figura (c) muestra el autómata obtenido cuando "solapamos" de modo conveniente los autómatas minimizados correspondientes al caso par y al caso impar. Vamos a demostrar que el autómata (c) es un autómata pequeño para $L_1.L_2$. Para ello vamos a considerar las palabras de longitud máxima:

$$M = \left\{ a^{2j} b^{\frac{n}{2}-j} c^{\frac{n}{2}-j} \mid 0 < j < \frac{n}{2} \right\}$$

Caso n par:



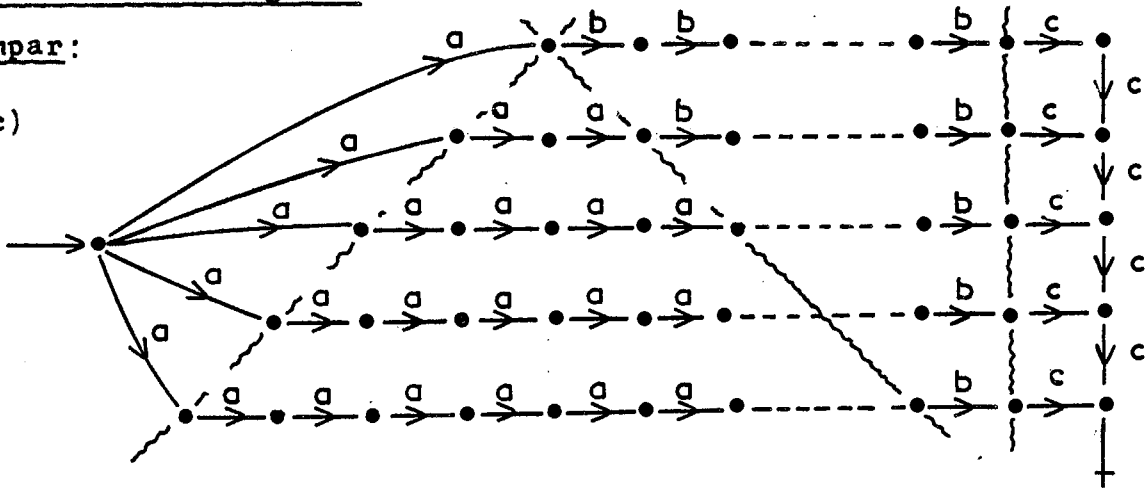
(b)



Palabras de longitud

impar:

(c)



Palabras de longitud

par:

(d)

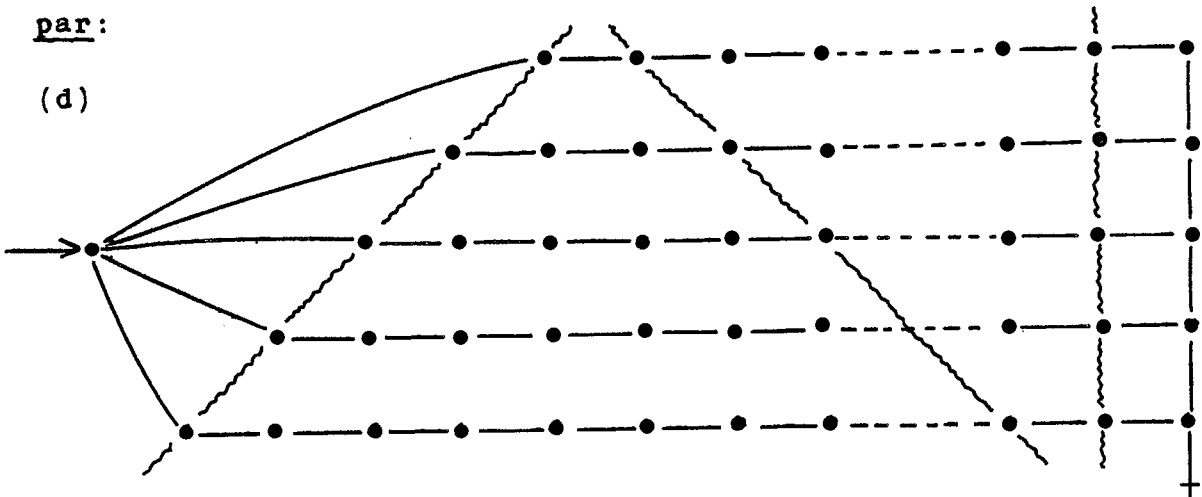


Figura III.39

Notemos que si $a^u b^v c^v \in M$ entonces $u + 2v = n$.

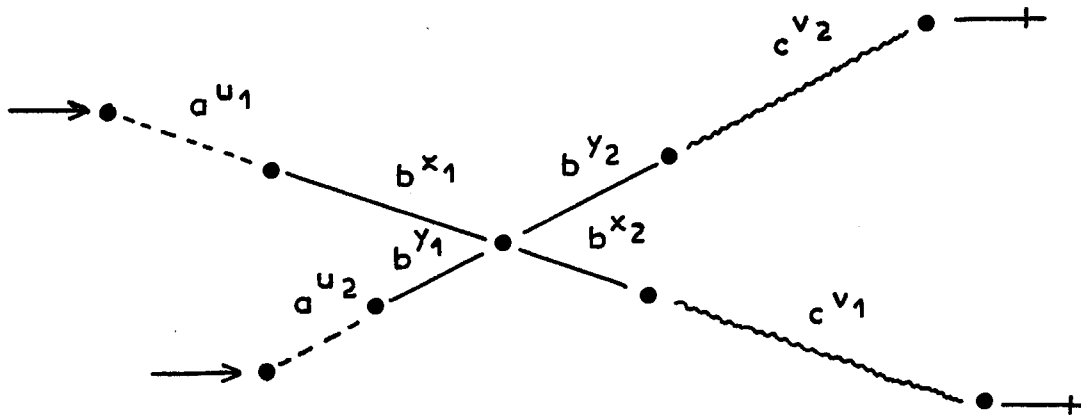
Vamos a considerar el conjunto de estados que intervienen en las computaciones de las letras b en las palabras de M ; es decir:

$$E = Q_- \cdot \left\{ a^{2j} b^x \mid 0 < j < \frac{n}{2}, 0 \leq x \leq \frac{n}{2} - j \right\}$$

Vamos a demostrar que no podemos confundir dos estados de E .

Supongamos que podamos confundir q y q' que intervienen en el

proceso de $m_1 = a^{u_1} b^{v_1} c^{v_1}$ y $m_2 = a^{u_2} b^{v_2} c^{v_2}$, respectivamente. Tenemos entonces el esquema siguiente:



tenemos el conjunto de igualdades:

$$\begin{array}{lll} u_1 + 2v_1 = n & x_1 + x_2 = v_1 & x_1 + y_2 = v_2 \\ u_2 + 2v_2 = n & y_1 + y_2 = v_2 & y_1 + x_2 = v_1 \end{array}$$

como consecuencia $x_1 = y_1$.

Supongamos $u_1 < u_2$ entonces $v_1 > v_2$, consideremos la palabra

$m_3 = a^{u_2} b^{y_1} b^{x_2} c^{v_1} = a^{u_2} b^{v_1} c^{v_1}$, dicha palabra es demasiado larga, llegamos pues a una contradicción. Tenemos entonces que:

$$\mu_{L_1 \cdot L_2}^{(n)} \geq \|E\| = \sum_{j=0}^{n/2} \left(\frac{n}{2} - j \right) = \Omega(n^2)$$

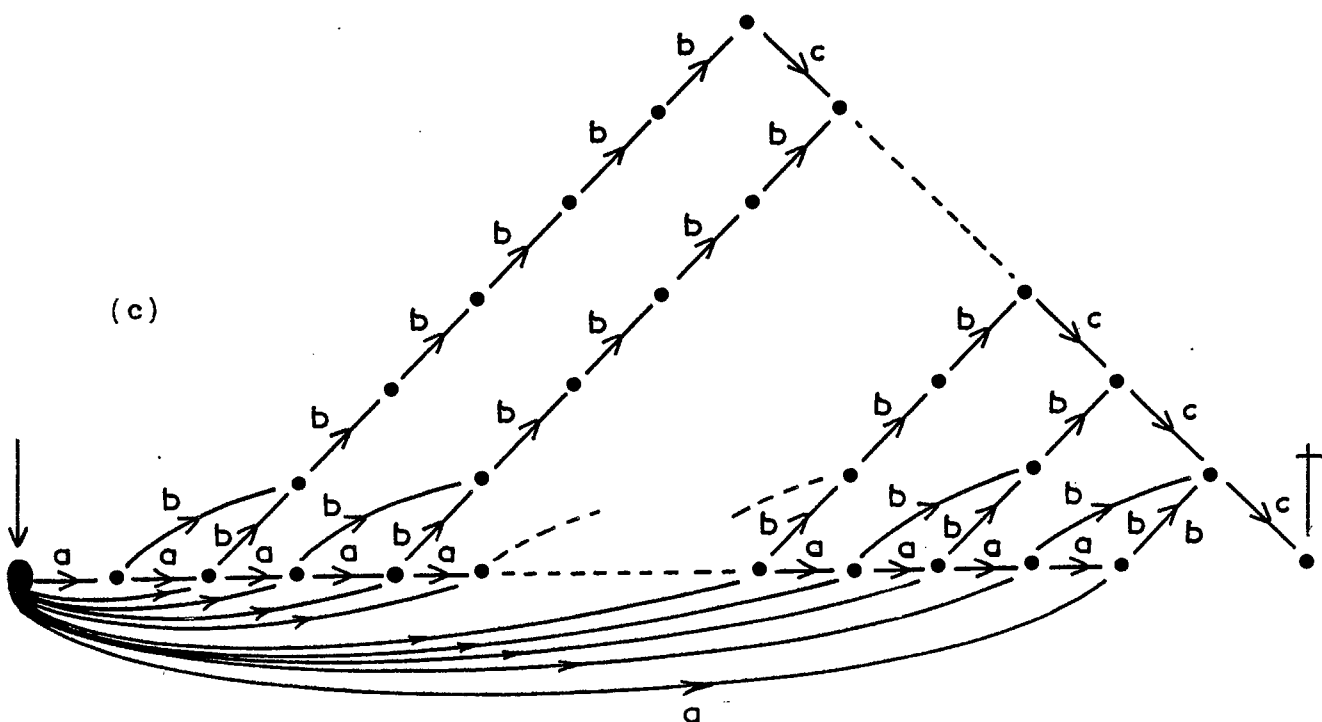
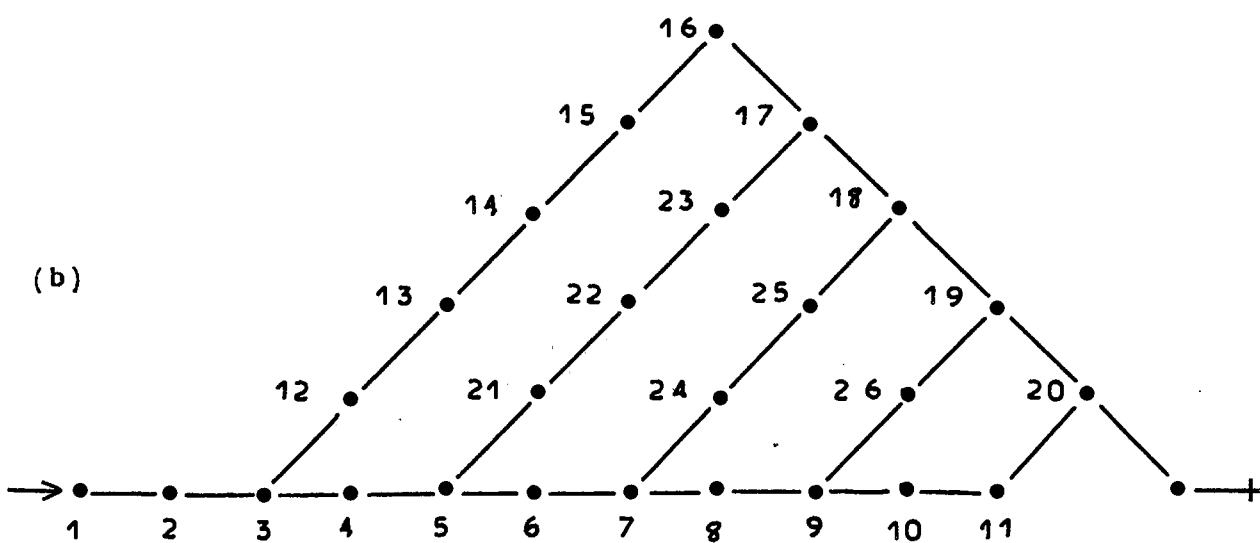
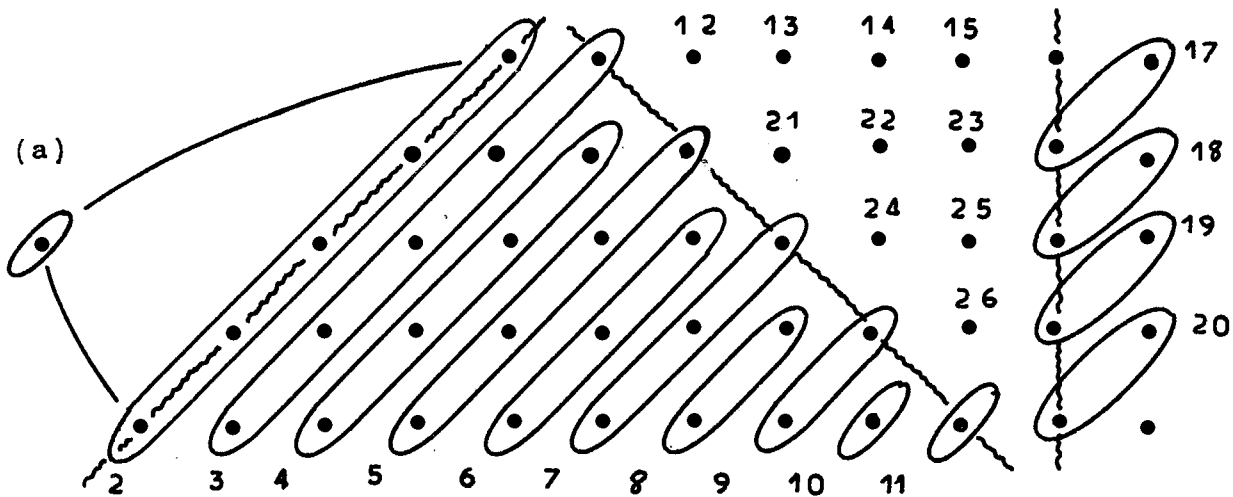


Figura III.40

Vemos entonces que si bien el método dado en el lema 3.3 es demasiado grosero para permitir hallar, en general, autómatas pequeños para el producto, existen casos en los que el resultado hallado es en orden de magnitud correcto. Es entonces inútil intentar hallar resultados más precisos en el caso general para la concatenación.

Un resultado asintóticamente correcto se obtendría también tomando: $L_1 = \{ a^n b^n \mid n > 0 \}$, $L_2 = \{ c^n d^n \mid n > 0 \}$

La figura III.41 muestra un autómata pequeño para $L_1 \cdot L_2$.

En consecuencia:

$$\mu_{L_1 \cdot L_2}(n) = \Theta(n^2)$$

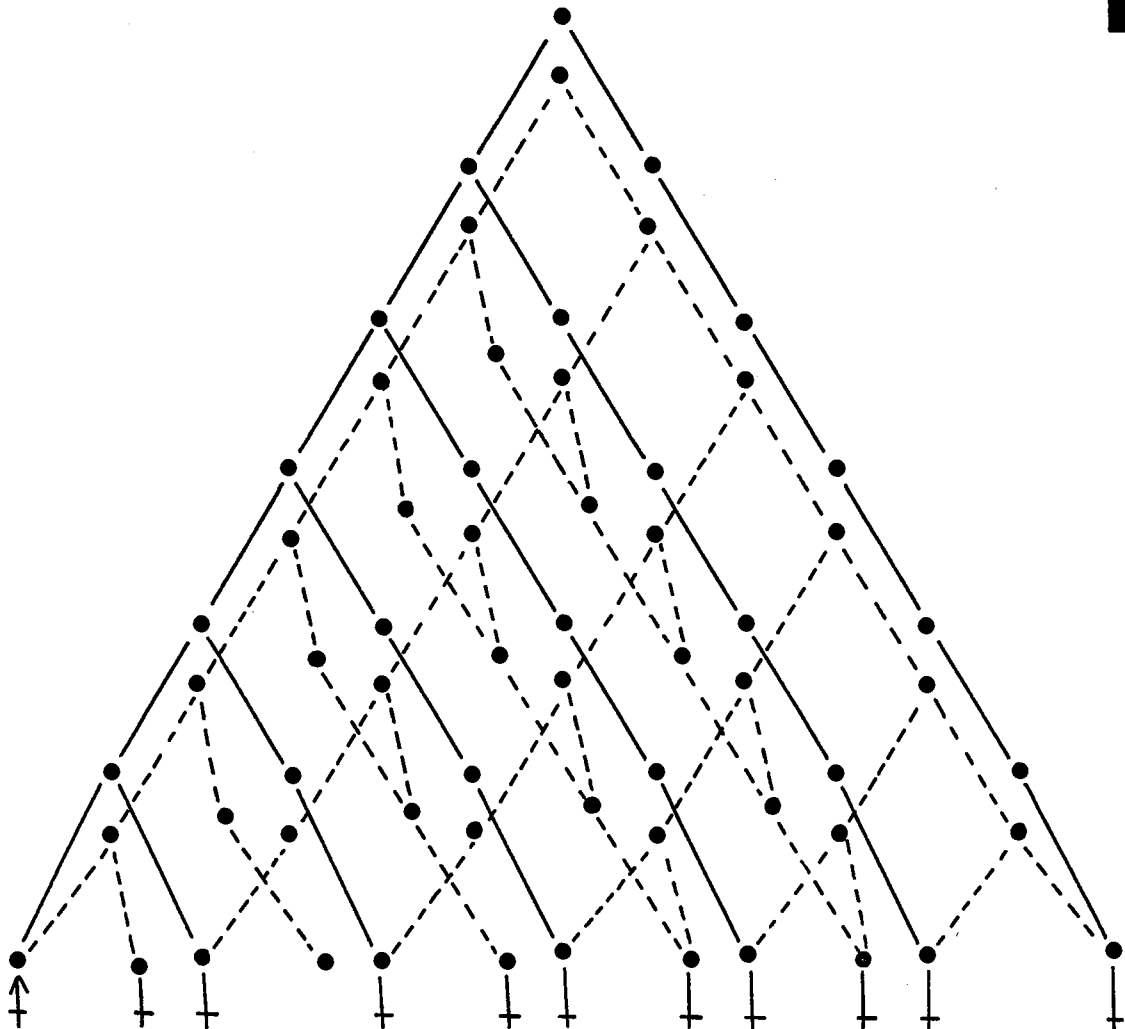


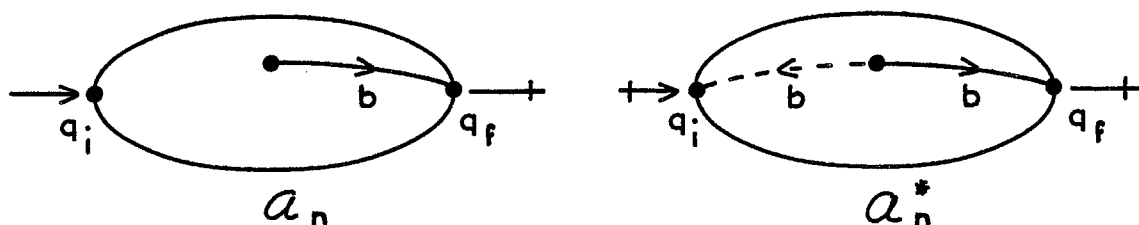
Figura III 41

Lema 3.4 Sea L un lenguaje en X^* , su índice inicial satisfaca:

$$\mu_{L^*}(n) \leq (n+1) \mu_L(n)$$

Prueba Tenemos que $n(L^*) \subset (nL)^*$. Efectivamente, sea $w \in n(L^*)$, w se factoriza en $w = w_1 w_2 \dots w_k$ con $w_i \in L$ y $|w_i| \leq n$.

En este caso podemos suponer que L no contiene ϵ , dado que en el caso contrario ya lo obtendremos al hacer la estrella. Realizemos la construcción esquematizada en la figura siguiente:



El nuevo autómata tiene el mismo número de estados y reconoce $(nL)^*$. Dado que $n(L^*) = (nL)^* \cap nX^*$, tenemos:

$$\mu_{L^*}(n) < (n+1) \mu_L(n)$$

Hecho 3.3 Para estudiar la optimalidad del límite precedente vamos a estudiar el índice inicial de $L^* = \{a^n b^n \mid n \geq 0\}^*$. Dado que $\mu_L(n) = \Theta(n)$, ejemplo 2.1, tenemos que $\mu_{L^*}(n) = O(n^2)$.

Vamos a demostrar que este límite es el mejor posible, para ello basta demostrar que $\mu_{L^*}(n) = \Omega(n^2)$. La figura III.42. nos indica un autómata pequeño para $n(L^*)$, $n = 6$, dicho autómata tiene $\Theta(n^2)$ estados. Para demostrar formalmente que todo autómata pequeño tiene $\Omega(n^2)$ basta aplicar las técnicas del hecho precedente, tomando como conjunto M, el conjunto:

$$M = \left\{ (ab)^j a^{\frac{n}{2}-j} b^{\frac{n}{2}-j} \mid 0 \leq j \leq \frac{n}{2} \right\}$$

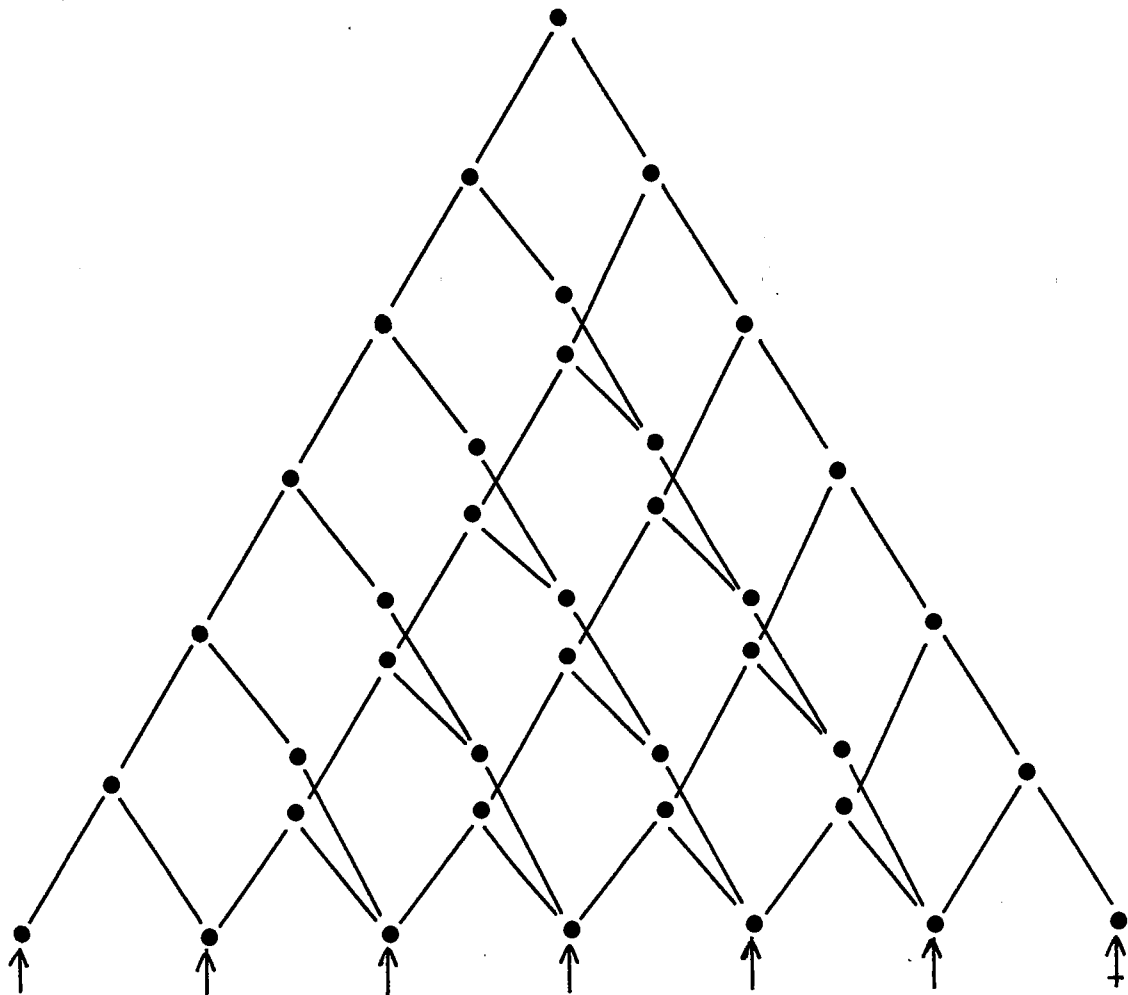


Figura III.42

Operaciones booleanas Recordemos que la unión ha sido tratada en las operaciones racionales, lema 3.2. Debemos tratar entonces la intersección y la complementación.

Vamos a tratar en primer lugar la intersección.

Lema 3.5 Sean L_1 y L_2 dos lenguajes en X^* , tenemos:

$$\mu_{L_1 \cap L_2}(n) \leq \mu_{L_1}(n) \cdot \mu_{L_2}(n)$$

Cuando L_2 es un lenguaje racional R existe una constante k tal que:

$$\mu_{L_1 \cap R}(n) \leq k \cdot \mu_{L_1}(n)$$

Prueba Tenemos que $n(L_1 \cap L_2) = nL_1 \cap nL_2$. El límite

$$\mu_{L_1 \cap L_2}(n) \leq \mu_{L_1}(n) \cdot \mu_{L_2}(n)$$

expresa el número de esta

dos de un autómata que es intersección de dos otros.

Notemos que tenemos también $n(L_1 \cap R) = nL_1 \cap R$. Dado que R es racional existe un autómata con k estados que lo describe.

La regla de intersección nos da entonces:

$$\mu_{L_1 \cap R}(n) \leq k \cdot \mu_{L_1}(n)$$



Vamos a tratar seguidamente la complementación.

Lema 3.6 Sea L un lenguaje en X^* tenemos que:

$$\mu_{X^* \setminus L}(n) \leq (n+1) 2^{\mu_L(n)}$$

Prueba Evidentemente $n(X^* \setminus L) \subset X^* \setminus nL$. Para hallar un autó-
mata que reconozca $X^* \setminus nL$ a partir de un autómata pequeño
para nL , necesitamos determinar dicho autómata y cambiar los
estados finales en no finales y viceversa. Dicho proceso nos ha

ce pasar de $\mu_L(n)$ a $2^{\mu_L(n)}$ estados.

Como de costumbre $n(X^* \setminus L) = (X^* \setminus nL) \cap nX^*$, con lo que:

$$\mu_{X^* \setminus L}(n) \leq (n+1) 2^{\mu_L(n)}$$

Hecho 3.4 Vamos a dar un ejemplo de lenguaje, en el que el pa-
so al complementario, hace pasar el índice inicial de un poli-
nomio a una exponencial.

Sea L el lenguaje en $\{a, b\}^*$ definido por:

$$L = \{w \mid w \in \{a, b\}^* \text{ y } w \text{ no es palíndromo}\}$$

Evidentemente $\{a, b\}^* \setminus L = \text{PAL}_2$. El lenguaje L puede reexpresar-
se como:

$$L = (\{a, b\}^2)^* \{a, b\} \cup \{w_1 w_2 \mid w_1 w_2 \in \{a, b\}^*, |w_1| = |w_2|, w_1 \neq \tilde{w}_2\}$$

El índice inicial de $(\{a, b\}^2)^* \{a, b\}$ es lineal.

Para calcular el índice del segundo lenguaje de L notemos que
las palabras $w_1 w_2$ pueden factorizarse como $w_1 w_2 = m_1 u m_2 v m_3$,

$m_1, m_3 \in \{a, b\}^*$, $|m_1| = |m_3|$; $m_2 \in (\{a, b\}^2)^*$; $u, v \in \{a, b\}$

$u \neq v$, En el ejemplo 1.1 hemos visto que las palabras de dicho tipo pueden reconocerse con un autómata de $O(n^2)$ estados.

Tenemos en consecuencia que $\mu_L(n) = O(n^3)$. Por otra parte hemos visto que en el ejemplo 2.7 que $\mu_{PAL_2}(n) = 2^{n/2}$.

Operaciones de trio y transducción fiel

Recordemos que las operaciones de trio son: morfismo alfabético ϵ -limitado, morfismo alfabético inverso e intersección con los lenguajes racionales [4].

Diremos que un morfismo es alfabético si la imagen de una letra es una letra ó ϵ . A fin de poder estudiar el comportamiento del índice inicial en la operación de morfismos necesitamos conocer su comportamiento en dicho morfismo, ello es posible en dos casos:

(a) Conocemos el comportamiento borrador de ϵ sobre L (seguidamente trataremos este caso).

(b) L tiene propiedades combinatorias fuertes (lemas de interacción), trataremos el caso L algebraico en el capítulo siguiente.

Demos una definición precisa de comportamiento borrador.

Definición 3.1 Dada una función creciente $\alpha : \mathbb{N} \rightarrow \mathbb{N}$, un morfismo alfabético $\varphi : X^* \rightarrow Y^*$ y un lenguaje $L \subset X^*$, diremos que φ es α -limitado en L si existe $k \in \mathbb{R}_+$ que satisface:

$$k \cdot \alpha(|\varphi(w)|) \geq |w|$$

para cualquier palabra w de L .

Podemos expresar ahora el comportamiento del índice inicial en un morfismo α -limitado.

Lema 3.7 Sea $\varphi: X^* \rightarrow Y^*$ un morfismo α -limitado sobre $L \subset X^*$ existe una constante $k \in \mathbb{R}_+$, tal que el índice inicial de $\varphi(L)$ satisface:

$$\mu_{\varphi(L)}(n) \leq (n+1) \mu_L(k \cdot \alpha(n))$$

Prueba En el caso α -limitado, los segmentos iniciales satisfacen:

$$n \varphi(L) \subset \varphi(k \cdot \alpha(n)L)$$

Efectivamente sea $h \in n \varphi(L)$ entonces $|h| \leq n$, para cualquier w que cumpla $h = \varphi(w)$ sabemos:

$k \cdot \alpha(|\varphi(w)|) \geq |w|$; $k \cdot \alpha(|h|) \geq |w|$, como α es una función creciente $\alpha(|h|) \leq \alpha(n)$ con lo que $k \cdot \alpha(n) \geq |w|$, desigualdad que también podemos escribir $w \in k \cdot \alpha(n)L$.

Podemos escribir $n \varphi(L) = nY^* \cap \varphi(k \cdot \alpha(n)L)$ y en consecuencia:

$$\mu_{\varphi(L)}(n) \leq (n+1) \mu_L(k \cdot \alpha(n))$$

Vamos a considerar tres casos particulares.

(a) Caso $\varphi : X^* \rightarrow Y^*$ es ε -limited. Sabemos que esto significa que existe una constante $q \in \mathbb{N}$, tal que, el número de letras consecutivas borradas por φ es como máximo q .

Sea $h = \varphi(w)$, $|h| = n$, en este caso w es de la forma:

$w = w_0 x_1 w_1 x_2 w_2 \dots x_{n-1} w_{n-1} x_n w_n$, con $|w_i| \leq q$, $\varphi(w_i) = \varepsilon$, entonces

$w \leq q(n+1) + n$ con lo que podemos tomar $\alpha(n) = q(n+1) + n$ obteniendo:

$$\mu_{\varphi(L)}(n) \leq (n+1) \mu_L(q(n+1) + n)$$

Notemos que cuando $q = 0$ tenemos un morfismo estrictamente alfabético, en este caso $n \varphi(L) = \varphi(nL)$ y $\mu_{\varphi(L)} \leq \mu_L(n)$.

(b) Diremos que un morfismo $\varphi : X^* \rightarrow Y^*$ es pol-limitado cuando existen $k \in \mathbb{R}_+$, $p \in \mathbb{N}_+$ tales que $\alpha(n) = k |n|^p$, en este caso:

$$\mu_{\varphi(L)}(n) \leq (n+1) \mu_L(k \cdot n^p)$$

(c) Diremos que un morfismo $\varphi : X^* \rightarrow Y^*$ es exp-limitado cuando existe $c \in \mathbb{R}_+$ tal que: $\alpha(n) = 2^{c \cdot n}$, en este caso:

$$\mu_{\varphi(L)}(n) \leq (n+1) \mu_L(2^{c \cdot n})$$

Hecho 3.5 Vemos que en el caso exp-limitado podemos pasar de un lenguaje polinomial a un lenguaje exponencial; vamos a mostrar un ejemplo:

Consideremos w en $1(0+1)^*$ y notemos $V(w)$ el entero del cual w es su expresión en binario. Definimos entonces L como una variación del lenguaje dado en el ejemplo 2.20:

$$L = \left\{ w\bar{w} \# V(w) \mid w \in 1(0+1)^*, \bar{w} \text{ copia barrada de } w \right\}$$

Sea φ el morfismo definido por:

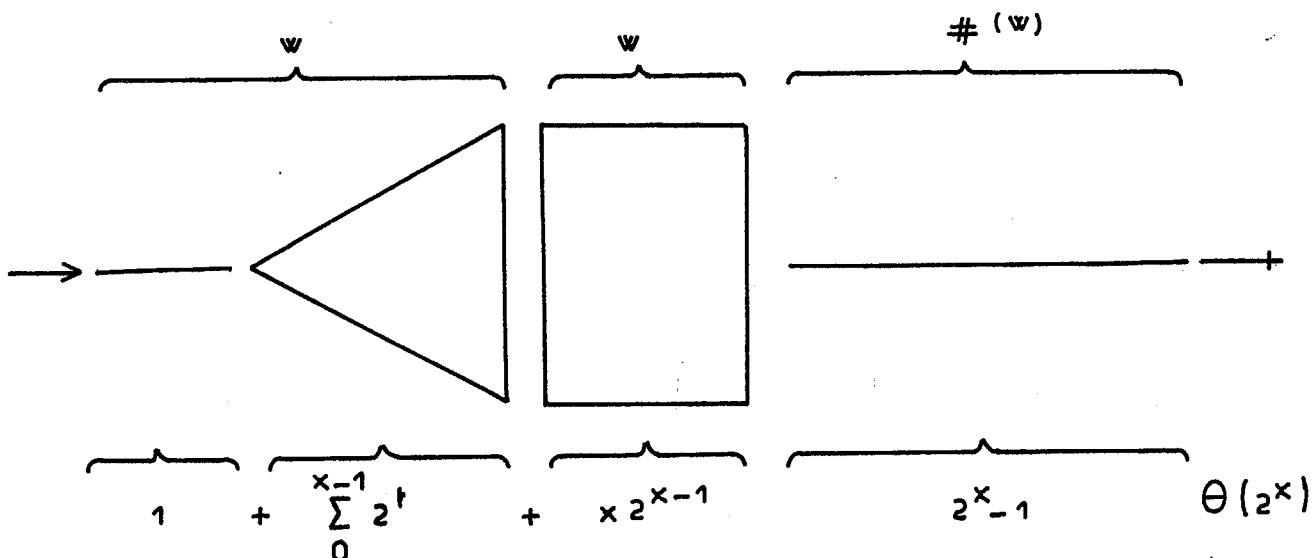
$$\varphi = \langle 0 \rightarrow 0 ; 1 \rightarrow 1 ; \bar{0} \rightarrow \bar{0} ; \bar{1} \rightarrow \bar{1} ; \# \rightarrow \varepsilon \rangle$$

entonces $\varphi(L) = \{ w\bar{w} \mid w \in 1(0+1)^+ , \bar{w} \text{ copia barrada de } w \}$

Veamos que φ es exp-limitado sobre L , $\varphi(w\bar{w}\#V(w)) = w\bar{w}$, sea $w \in 1(0+1)^{x-1}$, $|w| = x$, $2x + 2^{x-1} \leq |w\bar{w}\#V(w)| \leq 2x + 2^x - 1$ tenemos entonces $2^{x-1} \leq |w\bar{w}\#V(w)| \leq 2^{2x}$ o sea:

$$2^{|\varphi(w)|-1} \leq |w\bar{w}\#V(w)| \leq 2^{2|\varphi(w)|}$$

Esquematicemos un autómata para nL en la figura siguiente:



dado que $w\bar{w}\#V(w) = \Theta(2^x)$ tenemos $\mu_L(n) = \Theta(n)$, por otra parte $\mu_{\varphi(L)}(n) = \Theta(2^{n/2})$.

Vemos pues que el morfismo precedente nos ha transformado un lenguaje de índice lineal en un lenguaje exponencial.

■

Vamos seguidamente a estudiar la operación de morfismo alfabético inverso.

Lema 3.8 Sea $\varphi : X^* \longrightarrow Y^*$ un morfismo alfabético, tenemos:

$$\mu_{\varphi^{-1}(L)}^{(n)} \leq (n+1) \mu_L^{(n)}$$

Prueba En este caso tenemos $n \varphi^{-1}(L) \subset \varphi^{-1}(nL)$, efectivamente, $w \in \varphi^{-1}(h)$ como φ es alfabético $|w| \geq |h|$.

Tenemos entonces $n \varphi^{-1}(L) = nX^* \cap \varphi^{-1}(nL)$.

Para hallar un autómata para $n \varphi^{-1}(L)$ tomamos el autómata para nL , cambiamos de nombre a los arcos como el morfismo inverso nos indica, añadimos en cada estado bucles de aquellas letras que por φ dan ε , y calculamos las buenas longitudes haciendo la intersección con el autómata que reconoce nX^* .

Hecho 3.6 Mostremos que el anterior límite es óptimo en orden de magnitud.

$\varphi = \langle a \rightarrow \varepsilon; b \rightarrow b; c \rightarrow c \rangle$, $L = \{b^n c^n \mid n \geq 0\}$
 tenemos $\varphi^{-1}(L) = a^* (ba^*)^n (ca^*)^n$. Sea $R = a^* b^* c^*$, $\varphi^{-1}(L) \cap R = a^* L$. Tenemos:

$$\mu_{\varphi^{-1}(L)}^{(n)} = \Theta(\mu_{\varphi^{-1}(L) \cap R}^{(n)}) = \Theta(n^2)$$

La operación de intersección con un lenguaje racional ha sido tratada al estudiar las operaciones booleanas.

La definición de trio coincide con la de cono racional fiel [4].

Un cono racional fiel es una familia de lenguajes cerrada por transducción racional fiel. Recordemos que una transducción racional fiel $\tau^f: X^* \rightarrow Y^*$ es una aplicación de X^* en $\mathcal{P}(Y^*)$ definida por:

$$\tau^f(f) = \psi(\varphi^{-1}(f) \cap R)$$

en la que Z es un alfabeto finito, φ, ψ dos morfismos alfabéticos de Z^* en X^* y Y^* respectivamente y R un lenguaje racional y además el número de letras consecutivas de imagen ε por ψ en una palabra de R está acotado.

Reuniendo los lemas precedentes tenemos que el índice inicial pasa de manera agradable en las transducciones fieles, tenemos el teorema.

Teorema 3.1 Sea $\tau^f: X^* \rightarrow Y^*$ una transducción racional fiel, existen k_1 y k_2 tales que:

$$\mu_{\tau^f(L)}(n) \leq k_2 \cdot (n+1) \cdot (\mu_L(k_1 \cdot (n+1) + n))$$

Prueba Como τ^f es una transducción fiel, se factoriza en:

$$\tau^f(L) = \psi(\varphi^{-1}(L) \cap R) \text{ con } \psi, \varphi \text{ alfabéticos, dado que}$$

ε -limitado existe una constante k_1 que limita el número de letras consecutivas borradas en R , k_2 será el número de estados de un autómata que reconoce R .

Estudiemos la evolución del segmento inicial $n \tau^f(L)$, aplicando los lemas precedentes tenemos:

$$\begin{aligned} n \tau^f(L) &= n \psi(\varphi^{-1}(L) \cap R) \subset \psi(\{k_1(n+1)+n\} \{\varphi^{-1}(L) \cap R\}) = \\ &= \psi(\{k_1(n+1)+n\} \varphi^{-1}(L) \cap R) \subset \psi(\varphi^{-1}(\{k_1(n+1)+n\}L) \cap R) = \\ &= \tau^f(\{k_1(n+1)+n\}L), \text{ con lo que:} \end{aligned}$$

$$n \tau^f(L) = n Y^* \cap \tau^{(f)}(\{k_1(n+1) + n\}L)$$

Expresando la evolución del índice inicial en los segmentos iniciales llegamos a:

$$\mu_{\tau^f(L)}^{(n)} \leq k_2 \cdot (n+1) \cdot \mu_L(k_1(n+1) + n)$$

Operación de sustitución

Vamos a tratar seguidamente las operaciones de sustitución sin táctica y sustitución propia.

Recordemos rápidamente la definición de sustitución sintáctica

[6] $L_1 \uparrow L_2$ de L_2 en L_1 definida por : $L_1 \subset X_1^*$, $L_2 \subset X_2^*$,

$$X_1 \cap X_2 = \emptyset.$$

$$L_1 \uparrow L_2 = \{a_1 u_1 a_2 u_2 \dots a_n u_n \mid a_i \in X_1, u_i \in L_2, a_1 a_2 \dots a_n \in L_1\}$$

El comportamiento del índice inicial en dicha sustitución viene expresado en:

Lema 3.9 Sea $L_1 \subset X_1^*$, $L_2 \subset X_2^*$, $X_1 \cap X_2 = \emptyset$ tenemos:

$$\mu_{L_1 \uparrow L_2}^{(n)} \leq (n+1) \cdot \mu_{L_1}^{(n)} \cdot \mu_{L_2}^{(n)}$$

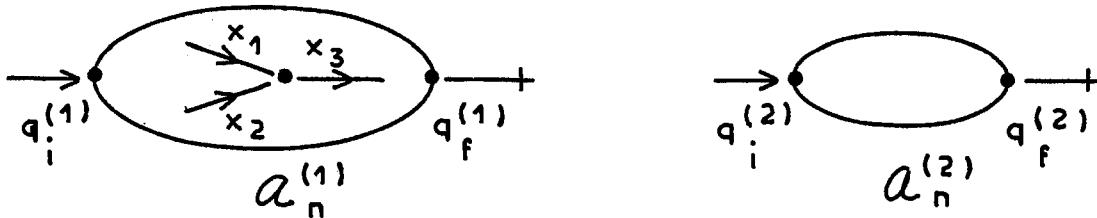
Prueba Veamos que $n(L_1 \uparrow L_2) \subset nL_1 \uparrow nL_2$. Sea $w \in n(L_1 \uparrow L_2)$, w

se factoriza de manera única en $a_1 u_1 a_2 u_2 \dots a_p u_p$ con

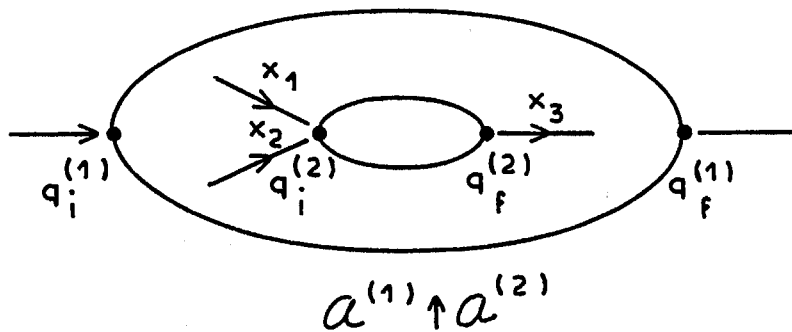
$$|a_1 u_1 a_2 u_2 \dots a_p u_p| = |a_1 a_2 \dots a_p| + \sum_{i=1}^p |u_i| \leq n, \text{ entonces}$$

$$|a_1 a_2 \dots a_p| \leq n \text{ y } |u_i| \leq n, \text{ con lo que } w \in nL_1 \uparrow nL_2.$$

Realizaremos la construcción dada en el esquema siguiente, el caso $L_2 \subset X^+$, cuando $\varepsilon \in L_2$ hay pequeñas variaciones.



tenemos:



El autómata $a_n^{(1)} \uparrow a_n^{(2)}$ tiene $\mu_{L_1}(n) \cdot \mu_{L_2}(n)$ estados

Dado que $n(L_1 \uparrow L_2) = (nL_1 \uparrow nL_2) \cap n(X_1 \cup X_2)^*$ llegamos a

$$\mu_{L_1 \uparrow L_2}(n) \leq (n+1) \cdot \mu_{L_1}(n) \cdot \mu_{L_2}(n)$$



Vamos a tratar seguidamente la substitución propia, $\sigma: X \rightarrow \mathcal{P}(Y^+)$ en la que a cada $a \in X$ le asociamos un lenguaje $\sigma(a)$ en Y^+ .

Tenemos entonces:

$$\sigma(L) = \{ u_1 u_2 \dots u_n \mid \text{existe } a_1 a_2 \dots a_n \in L \text{ y } u_i \in \sigma(a_i) \text{ para todo } i \} .$$

Antes de estudiar el comportamiento de $\sigma(L)$ en el índice inicial vamos a dar el siguiente lema de normalización.

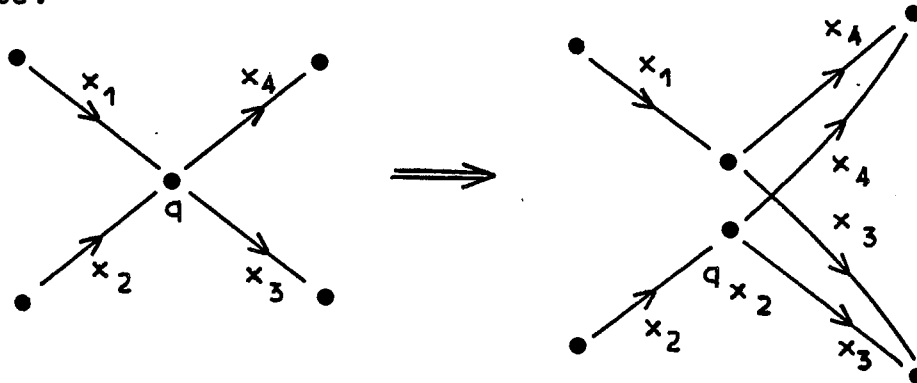
Lema 3.10 Sea L un lenguaje en X^* . Existe para $n \in \mathbb{N}$ un autómata que satisface:

- (a) Existe un único estado q_i sin arcos incidentes. Dicho estado es inicial.
- (b) Para todo estado $q \neq q_i$ existe un único arco incidente.
- (c) Este autómata tiene como máximo $\|X\| \mu_L(n)$ estados.

Prueba

(a) Ha sido demostrada en la parte (e) del lema 3.1.

(b) Si no, realizemos la construcción esquematizada en la figura siguiente:



(c) Realizando la construcción precedente en un autómata pequeño, multiplicamos como máximo por $\|X\|$ el número de estados.

El comportamiento del índice inicial viene dado en:

Lema 3.11 Sea $\sigma: X \rightarrow \mathcal{P}(Y^+)$ una substitución propia, tenemos que:

$$\mu_{\sigma(L)}(n) \leq \|X\| (n+1) \mu_L(n) \max \{ \mu_{\sigma(a)}(n) \mid a \in X \}$$

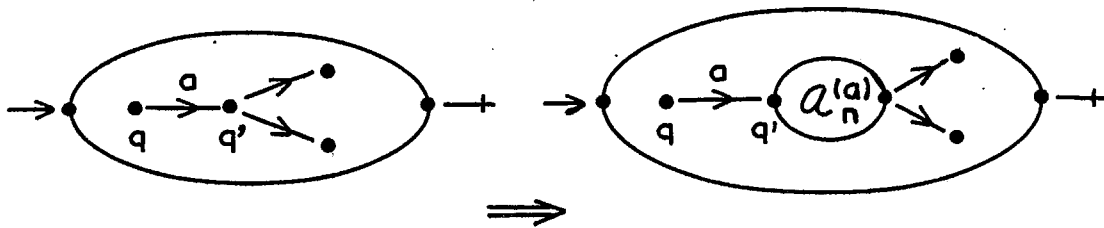
Prueba Notemos σ_n la substitución $\sigma_n(a) = n \sigma(a)$, tenemos que $n \sigma(L) \subset \sigma_n(nL)$, efectivamente sea $w \in n \sigma(L)$, dado que

σ es propia w se factoriza en $w = u_1 u_2 \dots u_r$ con $r \leq n$ con $x_1 x_2 \dots x_r \in L$ y $u_i \in \sigma(x_i)$, $|u_i| \leq n$.

Para hallar un límite al índice inicial procedamos de la forma siguiente: Sea $\mathcal{A}_n^{(1)}$ un autómata pequeño para nL , utilizando el

lema precedente llegamos a $\mathcal{B}_n^{(1)}$ con $\|X\| \mu_L(n)$ estados.

Sea $\mathcal{A}_n^{(a)}$, $a \in X$ un autómata pequeño para $\sigma_n^{(a)}$, realizemos la construcción:



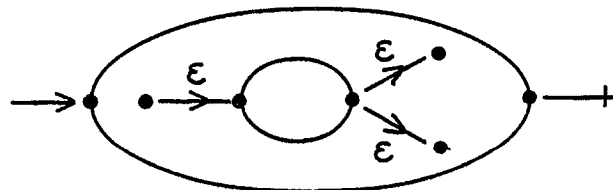
El nuevo autómata reconoce el lenguaje:

$$\{ a_1 u_1 a_2 u_2 \dots a_n u_n \mid a_1 a_2 \dots a_n \in nL \text{ y } u_i \in n \sigma(a_i) \}$$

Su número de estados está limitado por:

$$\|X\| \mu_L(x) \max \{ \mu_{\sigma(a)}(n) \mid a \in X \}$$

Para obtener un autómata que reconozca $\sigma_n(nL)$ realizemos el sistema de transición [3 7] esquematizado por:



Dicho sistema de transición puede ser transformado en un autó-
mata sin aumentar su número de estados.

Para obtener $n \bar{C}(L)$ realizemos la habitual intersección con nY^* .

Operación de paso al centro

Recordemos rápidamente el centro de un lenguaje dada por L. Boas-
son y M. Nivat en [9]: sea L un lenguaje en X^* , defini-
mos su centro $C(L)$ por:

$$C(L) = \{ w \in X^* \mid \text{Card}(wX^* \cap L) = \infty \}$$

Dicha operación y su relación con diversas familias de lenguajes
ha sido estudiada en [10, 23, 24, 43]. Vamos a demostrar aho-
ra que la familia de los lenguajes polinomiales no está cerrada
por paso al centro. Para ello vamos a considerar una variación
del lenguaje dado en 2.20. Sea L el lenguaje definido sobre

$$\{ 0, 1, \bar{0}, \bar{1}, \# \} \text{ por:}$$

$$L = \{ d \# \vee(\varphi(d)) \mid d \in \langle S \rightarrow OS\bar{0}S + 1S\bar{1}S + \varepsilon \rangle ;$$

$$\varphi = \langle 0 \rightarrow 0; 1 \rightarrow 1; \bar{0} \rightarrow 0; \bar{1} \rightarrow 1 \rangle \}$$

evidentemente $L \in \text{Pol } \mu$

Vamos a calcular el centro de dicho lenguaje.

Lema 3.12 Sea $D_2^* = \langle S \rightarrow OS\bar{0}S + 1S\bar{1}S + \varepsilon \rangle$, entonces

$$C(L) = \text{FG}(D_2^*)$$

Prueba Consideremos los factores izquierdos posibles de
 $d \# \vee(\varphi(d))$, hay de dos tipos.

. El factor izquierdo contiene al menos una letra $\#$. En este caso
es de la forma $d \#^r$ con $r < \vee(\varphi(d))$.

Podemos completar dicho factor como máximo con $\forall(\varphi(d)) - r$ letras \neq , en consecuencia dicha palabra no pertenece al centro.

. El factor izquierdo no contiene \neq . En este caso es un factor izquierdo f de $D_2'^*$, dicho factor puede completarse con $\widetilde{\Theta}(f)(0\bar{0})^*$, en consecuencia f pertenece al centro.

Hemos visto en el ejemplo 2.15 que $FG(D_2'^*) \in \overline{\text{Exp } \mu}$. En consecuencia los lenguajes polinomiales no están cerrados por paso al centro.

Considerando conjuntamente los lemas anteriormente citados llegamos respecto a la familia Pol μ a:

Teorema 3.1 La familia de los lenguajes de índice inicial como máximo polinomial es un A.F.L. cerrado por intersección y abierto por complementación y paso al centro.



CAPITULO IV. INDICE INICIAL Y FAMILIAS CLASICAS DE
LENGUAJES.

IV.1 Lenguajes polinomiales algebraicos y a multi-
contador.

IV.II Lenguajes exponenciales y generadores fieles.



En este capítulo estudiamos el índice inicial con relación a las familias de lenguajes. Trataremos separadamente los lenguajes polinomiales y exponenciales.

En el caso polinomial separaremos el caso algebraico y el caso no algebraico. Demostraremos que los lenguajes polinomiales algebraicos son un full-A.F.L no principal cerrado por substitución que contiene los lenguajes a contador iterado.

Mostraremos; que en el caso no algebraico; los lenguajes polinomiales contienen varias familias bien conocidas de lenguajes entre los multicontadores. Entre ellas, encontramos los lenguajes de etiquetas ϵ -libres de redes de Petri, los lenguajes de Szilard de las gramáticas algebraicas, y los multicontadores que trabajan en tiempo casi-real o polinomial.

Relacionaremos, finalmente, los lenguajes de índice inicial con los generadores fieles de distintas familias de lenguajes, como por ejemplo los lenguajes algebraicos, lineales, Post y Single-Reset.

IV.1. Lenguajes polinomiales algebraicos y a multicontadores.

Esta sección está dividida en dos subsecciones. En la primera trataremos el caso algebraico, y en la segunda el caso no algebraico.

Lenguajes algebraicos polinomiales.

En este apartado estudiamos los lenguajes algebraicos polinomiales y mostramos que contienen una subfamilia bien conocida de lenguajes algebraicos.

En el caso algebraico, podemos estudiar el comportamiento del índice inicial en transducciones cualquiera, y no solamente en el caso de transducciones fieles. A este fin estudiamos el índice inicial con relación a un morfismo cualquiera. A este fin, también, introducimos el árbol de los ancestros que modaliza el mínimo de nudos que son necesarios, a fin de dar cuenta de las variables que no se borran por el morfismo. El lema de Ogden, nos permite entonces acotar el comportamiento del índice inicial en el morfismo. Podemos a continuación expresar un paso en las transducciones racionales. Concluiremos que los lenguajes polinomiales son un full-A.F.L. Estudiaremos seguidamente su comportamiento en la substitución. Encontramos una cadena creciente de lenguajes polinomiales. Podemos decir que dichos lenguajes polinomiales son un full-A.F.L., no principal, cerrado por substitución. Dado que D_1^{1*} es polinomial, concluimos que los lenguajes a contador iterado son también polinomiales. Es una cuestión abierta, saber si son lenguajes polinomiales algebraicos.

Comenzamos por estudiar el índice inicial en el morfismo. Para ello, estudiaremos el comportamiento de los segmentos iniciales de los lenguajes algebraicos en un morfismo cualquiera.

A fin de poder dar cuenta, en el árbol de derivación, de las variables que dan letras que no se borran por el morfismo, introducimos el árbol de los ancestros. Dado, en un dominio de árbol, un determinado subconjunto de hojas, el árbol de los ancestros nos da "el mí-

nimo de nodos" que sirven para enlazar estas hojas formalmente.

Definición 1.1 Sea D un dominio de árbol, sean d_1 y d_2 en D , definimos el ancestro de d_1 y d_2 , notado $\text{anc}(d_1, d_2)$ por:

$\text{anc}(d_1, d_2)$ = factor izquierdo común a d_1 y d_2 de longitud máxima.

Sea $H = \{h_1, h_2, \dots, h_n\}$ un subconjunto de hojas de D , definimos el conjunto de ancestros de H por:

$$\text{Anc } H = \bigcup_{i, j \in \{1; \dots; n\}} \text{anc}(h_i, h_j)$$

Vamos a transformar el conjunto $\text{Anc } H$ en una arborescencia ordenada, definiendo la función sucesor por:

Definición 1.2

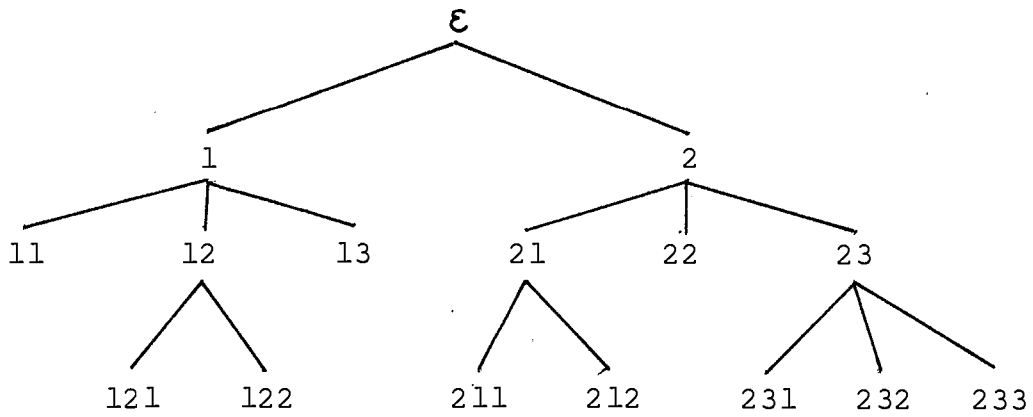
(1) Raíz: Es el elemento $r \in \text{Anc } H$ que tiene longitud máxima.

(2) Función sucesor: Sea $s \in \text{Anc } H$, diremos que $s' \in \text{Anc } H$ es un sucesor de s o $s' \in \Gamma(s)$ ssi, el factor izquierdo más largo de s' de longitud estrictamente inferior a s' contenido en $\text{Anc } H$ es s .

(3) Orden en $\Gamma(s)$: Sea s y s' en $\Gamma(s)$, diremos que $s < s'$ ssi s es mas pequeño que s' en el orden militar.

Evidentemente el conjunto $\langle \text{Anc } H, \Gamma \rangle$ es una arborescencia ordenada. Vamos seguidamente a dar un ejemplo de aplicación de estos conceptos.

Ejemplo 1.1 Consideremos el dominio del árbol:



en que hemos tomado como conjunto H el subconjunto de hojas siguiente:

$$H = \{ 11, 211, 212, 233 \}$$

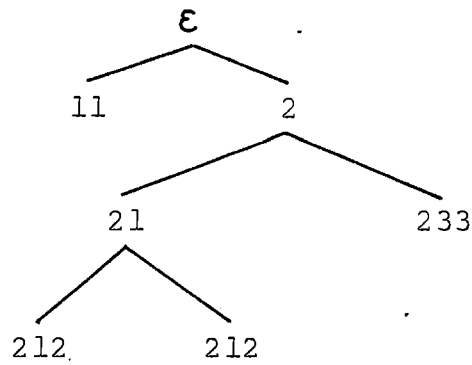
Calculemos Anc H :

			233	233
		212	2	212
	211	21	2	211
11	ε	ε	ε	11
11	211	212	233	

tenemos entonces:

$$\text{Anc } H = \{ \epsilon, 11, 2; 21; 211; 212; 233 \}$$

con la relación Γ tenemos:



Remarquemos que $\langle \text{Anc } H, \Gamma \rangle$ es una arborescencia ordenada; pero no un dominio de árbol.

Vamos a dar seguidamente un lema que nos permite relacionar el número de nodos internos con el número de hojas en una arborescencia ordenada, en que cada nodo tiene al menos dos sucesores. Dicho lema es una generalización del que existe para árboles binarios, que dice que el número de nodos internos es el número de hojas menos 1.

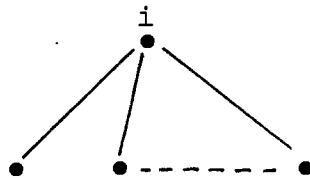
Tenemos:

Lema 1.1 Sea A una arborescencia ordenada, en la que todo nudo interno tiene al menos dos sucesores. Sea h el número de hojas, e i el número de nudos internos; tenemos:

$$i \leq h - 1$$

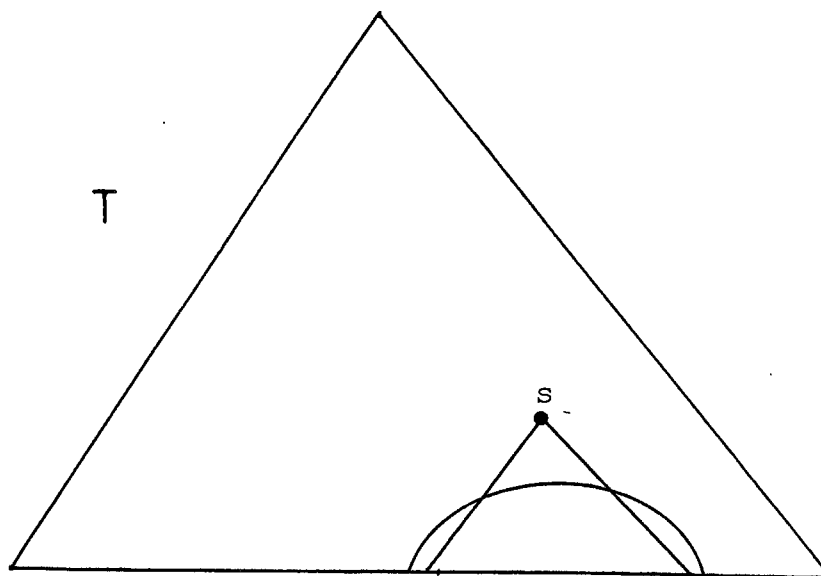
Prueba Razonemos por inducción sobre el número de nudos internos.

. Cuando $i = 1$ las únicas arborescencias posibles tienen la forma:



en las que $h = \|\Gamma(i)\| \geq 2$; con lo que $i \leq h - 1$.

. Sea T una arborescencia con i_T nodos internos, tomemos en T un nodo s tal que todos sus descendientes sean hojas. Eliminemos dichas hojas de T , el nuevo árbol T' que se obtiene viene esquematizado por:



El árbol T' satisface:

$$i_{T'} = i_T - 1$$

$$h_{T'} = h_T + 1 - \|\Gamma(s)\|$$

Sabemos por hipótesis de inducción que $i_{T'} \leq h_{T'} - 1$, con lo que:

$$i_T - 1 \leq h_T + 1 - \|\Gamma(s)\| - 1, \quad i_T \leq h_T - \|\Gamma(s)\| + 1$$

dado que $\|\Gamma(s)\| \geq 2$ obtenemos:

$$i_T \leq h_T - 1$$

Aplicamos el lema precedente a $\langle \text{Anc } H, \Gamma \rangle$ en:

Lema 1.2 Los nudos internos $\text{Int } H = \text{Anc } H \setminus H$ satisfacen:

- (1) $s \in \text{Int } H$ entonces $\|\Gamma(s)\| \geq 2$
- (2) $\|\text{Int } H\| \leq \|H\| - 1$

Prueba

(1) Razonemos por el absurdo. Sea $s \in \text{Int } H$ con $\|\Gamma(s)\| < 2$.

Hay dos casos a tratar:

Si $\Gamma(s) = \emptyset$, dicho nodo es una hoja con lo que $s \in H$, tenemos por tanto una contradicción.

Si $\|\Gamma(s)\| = 1$, existe s' único con $\Gamma(s') = s'$. Dado que $s \in \text{Anc } H$ existen $h_1, h_2 \in H$ con $s = \text{anc}(h_1, h_2)$; con $h_1 = s\alpha$, $h_2 = s\beta$. Dado que $s' \in \Gamma(s)$ tenemos $h_1 = s'\alpha'$ y $h_2 = s'\beta'$ con $|s'| > |s|$. Tenemos entonces que s no es el factor izquierdo de h_1 y h_2 de longitud máxima, llegamos, pues, a una contradicción.

(2) La prueba es evidente utilizando el lema precedente.



Los lemas y definiciones precedentes, tienen por objeto el permitirnos tratar el comportamiento de los segmentos iniciales de los lenguajes algebraicos, con respecto a los morfismos. Veremos que en este caso no es necesario el comportamiento -limitado del morfismo, a fin de obtener buenos límites. Formalmente:

Lema fundamental 1.3 Consideremos el lenguaje algebraico

$L = L(G)$ en que $G = \langle X, V, P \rangle$ es una gramática reducida. Notemos:

$$\delta = \max \{ |f| \mid (v, f) \in P \} ; p = \|\Gamma(v)\|$$

Sea $\varphi : X^* \rightarrow Y^*$ un morfismo; se cumple:

$$n \varphi(L) \subset \varphi(knL)$$

donde $k = 2p \delta^{p+1}$

Prueba Sea $w \in L(G)$; demostraremos que si $\varphi(w)$ es demasiado corta con respecto a w existe w' más corta con $\varphi(w') = \varphi(w)$. Para ello sea T un árbol de derivación para w . Vamos a tratar las distintas variables que pueden aparecer en T . En la figura 1.1, mostramos esquemáticamente las distintas etapas del proceso.

Trataremos separadamente las distintas ocurrencias de variables en T según que correspondan o no a miembros izquierdos de reglas, que sólo contienen letras de X que se borran por φ .

Comenzamos por un algoritmo de marcage de T .

(a) Trataremos las ocurrencias de variables v , tales que $\Gamma(v)$ contiene al menos una letra no borrrable por φ .

Tenemos aquí dos subcasos:

(a.1) Consideremos las ocurrencias de variables v en T que satisfacen:

. $\Gamma(v)$ contiene al menos una letra que no se borra por φ .

. Para toda ocurrencia de $v' \in \Gamma(v)$ el subárbol de T de raíz v' tiene por frontera una palabra que se borra por φ .

Sea el número de ocurrencias de variables en T que satisfacen

(a). Formemos el árbol de ancestros A_1 tomando como hojas las ocurrencias de tales variables. Dicho árbol describe el número de nodos mínimo para ligar las variables más bajas de T (entre sus

⊙ significa una letra
que no se borra por ψ

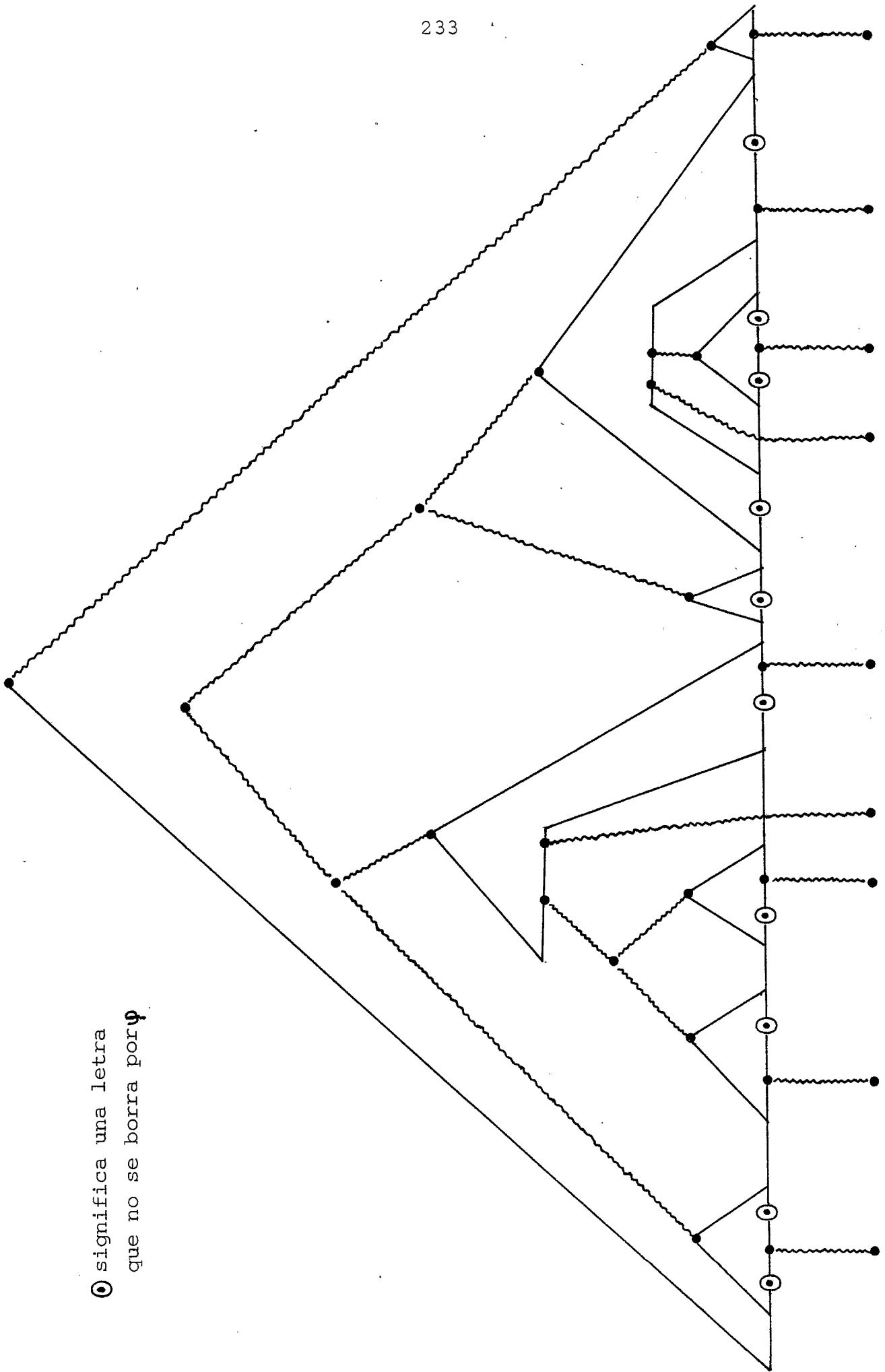


Figura 1.1.

descendientes no hay variables que no se borran) que no se borran por φ .

De acuerdo con el lema precedente, el número de nodos internos es, como máximo, $r - 1$. El número de arcos de A_1 es como máximo:

$$2r - 2$$

Trataremos ahora el segundo caso de (a):

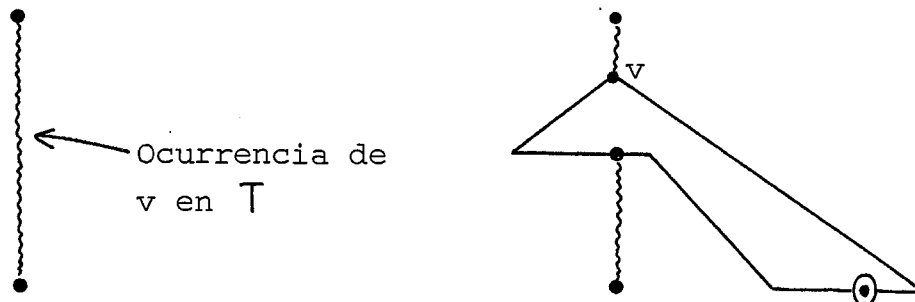
(a.2) Consideremos las ocurrencias de variables v en T que satisfacen:

- . $\Gamma(v)$ contiene al menos una letra que no se borra por φ .
- . Existe $v' \in \Gamma(v)$ tal que el subárbol de T de raíz v' tiene por frontera una palabra que no se borra por φ .

Vemos que la segunda condición puede reescribirse diciendo que

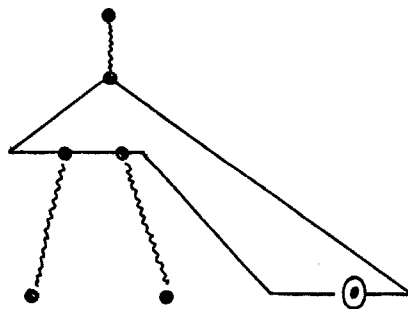
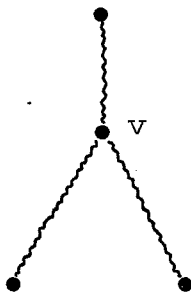
$\Gamma^+(v)$ contiene una variable de tipo (a.1).

Vamos a transformar A_1 a fin de dar cuenta de estas nuevas variables. Tenemos que dichas variables se insertan en el árbol de los ancestros A_1 , mediante un proceso esquematizado por:



Vemos que en este caso se separa el arco precedente en dos.

Nos falta por tratar el caso:



Tenemos que en este caso no se aumenta el número de arcos.

Tras este proceso, hemos obtenido un nuevo árbol A_2 .

Sea s el número de ocurrencias de variables de tipo (a.2), el número de arcos en A_2 es como máximo:

$$2r + s - 2$$

Esquemáticamente A_2 describe las variables que dan letras que no se borran por φ y el mínimo número de arcos que se necesitan para unirlos.

Recordemos que \mathbb{T} contiene un segundo tipo de variables; en efecto.

(b) Consideremos las ocurrencias de variables v en \mathbb{T} , tales que el subárbol de \mathbb{T} de raíz v , da una palabra que se borra por φ .

Esquematzaremos las variables de tipo (b) por:



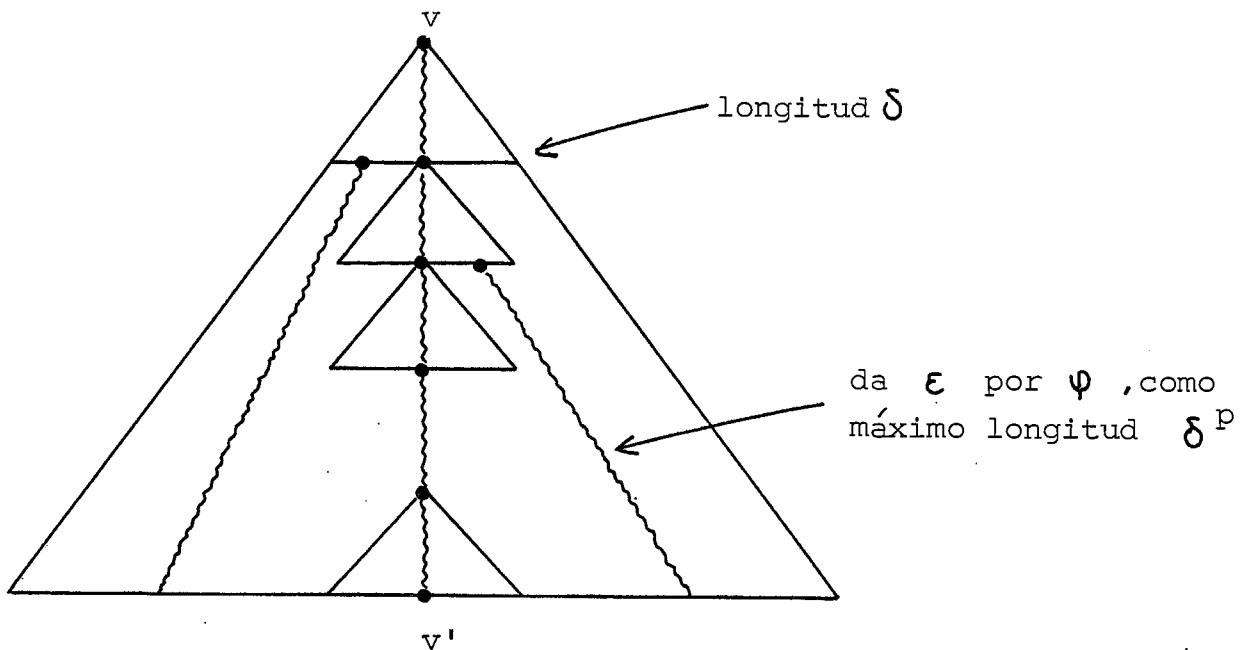
Vamos ahora a reducir el tamaño de \mathbb{T} , de manera que obtengamos una palabra corta que tenga la misma imagen que w .

Damos ahora el proceso de reducción de \mathbb{T} .

(1) Comencemos por tratar las variables de tipo(b). Utilizando el lema de interacción, siempre podemos transformar dicho árbol en otro, cuya altura es como máximo p , sin cambiar el resultado del morfismo. Tenemos entonces, que la longitud de la frontera del árbol de una variable (b), puede siempre suponerse limitada por:

$$\delta^p$$

(2) Vamos a tratar las partes de Γ que corresponden a arcos en Δ_2 . Esquemáticamente:



Siempre podemos limitar el número de nodos en el camino de v a v' a p . Con lo que dichas partes, pueden reducirse a palabras de longitud limitada por:

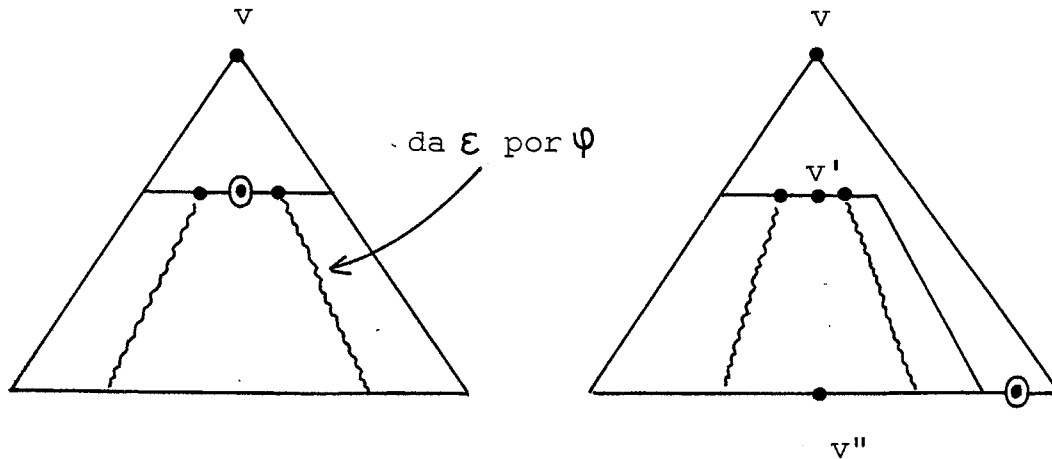
$$p \cdot \delta \cdot \delta^p = p \delta^{p+1}$$

Dado que el número total de arcos internos es $2r - s - 2$, podemos considerar que la parte de la palabra limitada a esta parte de árbol

bol tiene como máximo la longitud:

$$(2r - s - 2) \cdot p \cdot \delta^{p+1}$$

(3) Tratemus finalmente las variables de tipo (a). El caso más desfavorable es que cada variable de una sola letra que no se borra. Tenemos entonces $r + s = n$ en la que $n = |\varphi(w)|$. Esquemáticamente tenemos:



Podemos considerar que dichos subárboles, engendran palabras cuya longitud esta limitada por $\delta \cdot \delta^p = \delta^{p+1}$. Dado que como máximo hay n , tenemos $n \delta^{p+1}$.

Obtenemos como resultado que, para obtener una imagen de longitud n , siempre podemos partir de una palabra de longitud:

$$(2r - s - 2)p \delta^{p+1} + n \delta^{p+1}$$

Dado que $2r - s - 2 \leq 2r - 2s - 2 = n - 2$ tenemos:

$$(2r - 1 - 2)p \delta^{p+1} \leq (n - 2)p \delta^{p+1} \leq np \delta^{p+1}$$

con lo que

$$(2r - s - 2)p \delta^{p+1} + n \delta^{p+1} \leq 2np \delta^{p+1}$$



El lema precedente nos permite dar un teorema de paso del índice inicial a través de cualquier transducción algebraica, cuando consideramos lenguajes algebraicos.

Teorema 1.1 Sea L un lenguaje algebraico y τ una transducción racional, existen k_1 y k_2 tales que:

$$\mu_{\tau(L)}(n) \leq k_1 \cdot (n + 1) \cdot \mu_L(k_2 \cdot n)$$

Prueba Basta considerar la caracterización de Nivat; de las transducciones racionales a partir de φ^{-1} , ψ , $\cap R$, y el paso de los segmentos iniciales a tales operaciones.

Dicho teorema nos permite estudiar los lenguajes algebraicos de índice inicial polinomial, en términos de full-A.F.L. Considerando los resultados obtenidos en el capítulo precedente, podemos escribir:

Lema 1.4 Los lenguajes polinomiales algebraicos, son un full-A.F.L.

Recordemos que llamamos cono racional a una familia cerrada por la operación de transducción racional. Llamaremos lenguajes a con- tador restringido, notado Rocl, al cono racional mas pequeño que con-

tiene D_1^* , formalmente escribiremos $\text{RoCl} = \mathcal{L}(D_1^*)$, [6]
 Hemos visto en el capítulo precedente que $\mu_{D_1^*}(n) = \Theta(n^2)$,
 con lo que podemos afirmar:

Lema 1.5 Los lenguajes a contador restringido son algebraicos polinomialmente, formalmente $\text{RoCl} \subset \text{Pol}\mu$.

■

Diremos que un lenguaje L es generador de RoCl ssi $L \in \text{RoCl}$ y $\text{RoCl} = \mathcal{L}(L)$. Entre los generadores de RoCl contamos D_1^* , D_1' , D_1 , ε , [6]. En el lema siguiente, vamos a mostrar un generador de RoCl a través de transducciones racionales que conservan la longitud de las palabras; tenemos:

Lema 1.6 [18, 42] : Sea L un lenguaje a contador restringido, existen dos morfismos g, h estrictamente alfabéticos ($\Psi(x) \subset Y$) y un racional R que satisfacen:

$$L = g(h^{-1}(\tilde{L}') \cap R)$$

en que $\tilde{L}' = \langle S \rightarrow aSbS + \varepsilon \rangle (b + \varepsilon)$

Dicho lema nos permite dar un límite al índice inicial de los lenguajes de RoCl , en efecto.

Lema 1.7 Si L es un lenguaje a contador restringido, tenemos que

$$\mu_L(n) = O(n^2)$$

Prueba Evidentemente $\mu_{L'}(n) = O(n^2)$. Dado que:
 $L = g(h^{-1}(L') \cap R)$ tenemos $\mu_L(n) = O(n^2)$



Vamos ahora a tratar el comportamiento del índice inicial, respecto a la substitución en el caso de los lenguajes algebraicos. Recordemos, que una substitución es un morfismo $\sigma: X^* \rightarrow \mathcal{P}(Y^*)$. En el capítulo precedente hemos tratado el comportamiento del índice inicial en el caso de las substituciones ϵ -libre. Veamos ahora, que en el caso de lenguajes algebraicos, podemos suprimir la hipótesis ϵ -libre; tenemos:

Lema 1.8 Sea $\sigma: X^* \rightarrow \mathcal{P}(Y^*)$ y L un lenguaje algebraico, existe k tal que:

$$\mu_{\sigma(L)}(n) \leq (n+1)(1 + \max \{ \mu_{\sigma(x)}(kn) \}) \mu_L(kn)$$

Prueba Sea $\bar{X} = \{ \bar{x} \mid x \in X \}$ una copia de X . Definamos

$\hat{\sigma}(x): X^* \rightarrow \mathcal{P}(\bar{X} \cup Y)^*$ por $\hat{\sigma}(x) = \bar{x} \sigma(x)$. Vemos que la substitución $\hat{\sigma}$ es ϵ -libre. Consideremos el morfismo

$$\psi: \langle \bar{x} \rightarrow \epsilon; y \rightarrow y \mid \bar{x} \in \bar{X}, y \in Y \rangle$$

Evidentemente $\sigma(L) = \psi(\hat{\sigma}(L))$.

Consideremos el comportamiento de los segmentos iniciales en $\sigma(L)$. Tenemos $n \sigma(L) = n \psi(\hat{\sigma}(L))$, de acuerdo con el lema fundamental, existe una constante k que satisface:

$n \sigma(L) \subset \psi(kn \hat{\sigma}(L))$. Si definimos $\hat{\sigma}_r(x) = r \hat{\sigma}(x)$, podemos escribir $n \sigma(L) \subset \psi(\hat{\sigma}_{kn}(kn))$, con lo que llegamos a:

$$\mu_{\sigma(L)}(n) = \psi(\hat{\sigma}_{kn}(kn)) \cap nY^*$$

Recordando las construcciones dadas en el capítulo precedente para la substitución, tenemos:

$$\mu_{\sigma(L)}(n) \leq (n+1) \cdot \max \{ \mu_{\hat{\sigma}(x)}(kn) \mid x \in X^* \} \cdot \mu_L(kn), \text{ dado}$$

que $\max \{ \mu_{\hat{\sigma}(x)}(kn) \mid x \in X^* \} = 1 + \max \{ \mu_{\sigma(x)}(kn) \mid x \in X^* \}$

llegamos al resultado deseado. ■

Vamos a calcular el índice inicial de los lenguajes a contador iterado, notados Fcl [6]. Sea \mathcal{M} una familia de lenguajes, recordemos que definimos una \mathcal{M} substitución σ , como una substitución en la que $\sigma(x) \in \mathcal{M}$ para cualquier x .

Podemos entonces definir:

$$\mathcal{L} \square \mathcal{M} = \{ \sigma(L) \mid L \in \mathcal{L}, \sigma: X^* \rightarrow Y^*, \sigma \text{ es una } \mathcal{M} \text{ substitución} \}$$

Si substituimos en una misma familia, podemos construir $\mathcal{L} \square \mathcal{L}$, notando dicha familia por $\mathcal{L} \bar{\mathcal{L}}$, podemos construir

$$\mathcal{L}^n = \mathcal{L} \square \mathcal{L}^{n-1} = \mathcal{L} \bar{\mathcal{L}}^{n-1}$$

y finalmente $\mathcal{L} \bar{\mathcal{L}}^* = \bigcup_{n \geq 1} \mathcal{L}^n$. Cuando aplicamos este proceso a RoCl, obtenemos Fcl = RoCl RoCl^{*}. Dado que RoCl son lenguajes polinomiales, los lemas precedentes nos permiten enunciar:

Lema 1.9 Los lenguajes algebraicos polinomiales son un full-A.F.L cerrado por substitución, que contiene Fcl. ■

Vamos ahora a atacar el problema de la no principalidad de los lenguajes polinomiales algebraicos.

Como paso preliminar, vamos a demostrar el lema siguiente, de naturaleza combinatoria.

Lema 1.10 [51] Sea el alfabeto $X = \{a_1, a_2, \dots, a_k\}$, consideremos el lenguaje $L_k = a_1^+ a_2^+ \dots a_k^+$, se cumple:

- $\| L_k \cap X^n \| = \binom{n-1}{k-1}$
- $\| L_k \cap nX^* \| = \binom{n}{k}$
- $\| FG(L_k \cap nX^*) \| = \binom{n+1}{k}$

Prueba (1) El lenguaje $L_k = a_1^+ a_2^+ \dots a_k^+$ puede ser descrito por medio de la siguiente serie en variables no conmutativas

$$L_k = \frac{a_1}{1-a_1} \cdot \frac{a_2}{1-a_2} \cdot \dots \cdot \frac{a_k}{1-a_k}$$

La serie generatriz del número de palabras para una longitud determinada, se obtiene aplicando el morfismo:

$$a_i \longrightarrow x, \quad a_i \in X$$

obtenemos entonces la serie "conmutativa" siguiente:

$$L_k(x) = \left(\frac{x}{1-x} \right)^k = \sum l_n^{(k)} x^n$$

en la que $l_n^{(k)}$ es el número de palabras de longitud n . Evidentemente:

$$l_n^{(k)} = [x^n] \left(\frac{x}{1-x} \right)^k = [x^{n-k}] (1-x)^{-k}$$

Recordemos que la regla del binomio nos dice:

$$[x^p] (1-x)^{-k} = \frac{k(k-1)\dots(k+p-1)}{p!} = \binom{k+p-1}{p}$$

En consecuencia:

$$l_n^{(k)} = \binom{n-1}{n-k} = \binom{n-1}{k-1}$$

(2) Si queremos contar $l_{\leq n}^{(k)} = \|\mathbb{L}_k \cap nX^*\|$ basta hacer la su-

$$\text{ma: } l_{\leq n}^{(k)} = l_k^{(k)} + l_{k+1}^{(k)} + \dots + l_n^{(k)} = \sum_{k \leq p \leq n} \binom{p-1}{k-1} = \binom{n}{k}$$

Podemos, también obtener, este resultado con la ayuda de series generatrices. Para ello, vamos a contar el número de palabras de longitud n del lenguaje $L_k \cdot z^*$, en el que z es una nueva letra. Tenemos:

$$L_k z^* = \frac{a_1}{1-a_1} \cdot \frac{a_2}{1-a_2} \dots \frac{a_k}{1-a_k} \cdot \frac{1}{1-z}$$

Utilizando el mismo morfismo que antes, obtenemos:

$$L_k z^*(x) = \left(\frac{x}{1-x} \right)^k \frac{1}{1-x} = \sum_{l \leq n} l_n^{(k)} x^n$$

con lo que:

$$\begin{aligned} l_{\leq n}^{(k)} &= [x^n] \left(\frac{x}{1-x} \right)^k \frac{1}{1-x} = [x^{n-k}] (1-x)^{-(k+1)} = \\ &= [x^{n-k}] (1-x)^{-(k+1)} = \binom{k+n-k}{n-k} = \binom{n}{n-k} = \binom{n}{k} \end{aligned}$$

(3) Vamos finalmente a contar $g_n^{(k)} = \|\text{FG}(\mathbb{L}_k \cap nX^*)\|$.

A fin de contar los factores izquierdos de palabras de longitud máxima n de L_k describimos el lenguaje G_k de manera no ambigua

por:

$$\begin{aligned} G_k &= \frac{z^k}{1-z} + \frac{a_1}{1-a_1} \cdot \frac{z^{k-1}}{1-z} + \dots + \frac{a_1}{1-a_1} \cdot \frac{a_2}{1-a_2} \cdot \frac{z^{k-2}}{1-z} + \dots \\ &+ \frac{a_1}{1-a_1} \cdot \frac{a_2}{1-a_2} \dots \frac{a_k}{1-a_k} \cdot \frac{1}{1-z} \end{aligned}$$

Se trata del lenguaje de los factores izquierdos de L_k , en el que hemos añadido un número de letras z , mayor o igual al número posible de letras que no aparecen en el factor. Se trata de un "llenado" que satisface:

$$g_n^{(k)} = \text{n}^\circ \text{ de palabras de } G_k \text{ de longitud } n = \| \text{FG}(L_k \cap nX^*) \| .$$

Considerando el mismo morfismo que antes, tenemos:

$$g_n^{(k)} = [x^n] \left(\frac{x^k}{1-x} + \frac{x^k}{(1-x)^2} + \dots + \frac{x^k}{(1-x)^{k+1}} \right)$$

Sumando la progresión geométrica obtenemos:

$$g_n^{(k)} = [x^{n-k}] \frac{1-(1-x)^{k+1}}{x(1-x)^{k+1}} = [x^{n-k}] (1-x)^{-(k+1)} - 1 =$$

$$= \begin{cases} 0 & \text{si } n < k \\ \binom{n+1}{k} & \text{si } n \geq k \end{cases}$$



El cálculo precedente, se halla relacionado a problemas combinatorios bien conocidos. Recordemos que llamamos k -composición de n a una sucesión de enteros (x_1, x_2, \dots, x_k) con $x_i > 0$, $1 \leq i \leq k$, que satisface $\sum_{j=1}^k x_j = n$. Sea $\mathcal{C}_{k,n}$ el conjunto de todas las k composiciones de n . Evidentemente tenemos:

$$\| L_k \cap X^n \| = \| \mathcal{C}_{k,n} \| = \binom{n-1}{k-1}$$

Dicha igualdad viene del hecho que:

$$\mathcal{C}_k = \bigcup_n \mathcal{C}_{k,n} \simeq \underbrace{N^+ \times N^+ \times \dots \times N^+}_{k \text{ veces}}$$

con lo que existe una biyección evidente entre los elementos de

$$\mathcal{C}_k \text{ y } L_k.$$

Queremos ahora estudiar el comportamiento asintótico de $\|FG(L_k \cap nX)\|$ en función de n . Para ello recordemos que:

$$(n)_k = n(n-1)(n-2)\dots(n-k+1)$$

puede expresarse con la ayuda de los números de Stirling de primera especie S_k^p por $[40]_k$

$$(n)_k = \sum_{x=1}^k S_k^x n^x$$

con lo que $\binom{n}{k} = \frac{(n)_k}{k!} = \frac{1}{k!} \sum_{x=1}^k S_k^x n^x = \Theta(n^k)$

dado que $S_k^k = 1$.

Tenemos entonces que $\|FG(L_k \cap nX^*)\| = \Theta((n+1)^k) = \Theta(n^k)$.

Vamos a mostrar en el lema siguiente una sucesión de lenguajes

P_k , que satisfacen $\mu_{P_k}(n) = \Theta(n^k)$.

Lema 1.11 Consideremos el alfabeto infinito $A = \{a_1, a_2, \dots, a_n, \dots\}$

la sucesión de lenguajes para $k \geq 1$.

$$P_k = \left\{ a_1^{n_1} a_2^{n_2} \dots a_k^{n_k} \# \bar{a}_k^{n_k} \dots \bar{a}_2^{n_2} \bar{a}_1^{n_1} \mid n_i \geq 1, 1 \leq i \leq k \right\}$$

Satisfacen $\mu_{P_k}(n) = 2 \binom{n+1}{k} = \Theta(n^k)$

Prueba (a) Vamos a dar en este apartado un autómata para los segmentos iniciales de P_k . Notemos que toda palabra de P_k es impar.

Vamos a reconocer segmentos de la forma $(2n+1)P_k$ con $n \geq k$, dado que la palabra más corta es:

$$a_1 a_2 \dots a_k \# \bar{a}_k \dots \bar{a}_2 \bar{a}_1$$

Notemos que un autómata para $(2n+1)P_k$ debe poder leer las palabras:

$$a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} \in FG(L_k \cap nX^*)$$

y leer a un estado que es normal caracterizado por $\langle n_1, n_2, \dots, n_i, + \rangle$ en que + significa que aún no hemos leído #. Análogamente la lectura de una palabra:

$$a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} a_i^{n'_i} a_{i+1}^{n_{i+1}} \dots a_k^{n_k} \# \bar{a}_k^{n_k} \dots \bar{a}_{i+1}^{n_{i+1}} \bar{a}_i^{n'_i}$$

conducirá a un estado $\langle n_1, n_2, \dots, n_i, \dots \rangle$ en que significa que ya hemos leído #, notemos que dicho estado memoriza que aún falta por leer la parte

$$\bar{a}_i^{n_i} \dots \bar{a}_2^{n_2} \bar{a}_1^{n_1}$$

Notemos además que todo factor izquierdo $a_1^{n_1} a_2^{n_2} \dots a_i^{n_i}$ debe poder completarse en una palabra de L_k de longitud menor o igual a n. Dado que la manera más económica de hacerlo es:

$$a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} a_{i+1} \dots a_{k-1} a_k$$

Los coeficientes $n_j, 1 \leq j \leq i$ deben satisfacer

$$\sum_{1 \leq j \leq i} n_j \leq n - (k - i)$$

Tras estas consideraciones, definamos formalmente el autómata

$$A_k = \langle X_k, Q_k, q_0, q_+ \rangle \quad \text{por:}$$

$$X_k = \{a_1, a_2, \dots, a_k\}$$

$$Q_k = \{q_0, q_+\} \cup \{ \langle n_1, n_2, \dots, n_i, s \rangle \mid s \in \{+, -\}, i \leq k, 1 \leq j \leq i, n_j \geq 1, \sum_{1 \leq j \leq i} n_j \leq n - (k - i) \}$$

Vamos seguidamente a definir la función de transición T_k .

Hagamos notar que la función de transición está definida si y solo

si el estado de llegada existe.

La inicialización está descrita por:

$$q_0 \cdot a_1 = \langle 1, + \rangle$$

Si la parte leída de la palabra no es demasiado larga, podemos, ya continuar a leer la misma letra, ya leer la letra siguiente; es decir:

$$\langle n_1, n_2, \dots, n_i, + \rangle \cdot a_i = \langle n_1, n_2, \dots, n_i + 1, + \rangle$$

$$\langle n_1, n_2, \dots, n_i, + \rangle \cdot a_{i+1} = \langle n_1, n_2, \dots, n_i, 1, + \rangle$$

Cuando ya se ha leído una letra n_k , tenemos derecho a leer , esto nos determina el paso de las letras borradas.

$$\langle n_1, n_2, \dots, n_k, + \rangle \cdot \# = \langle n_1, n_2, \dots, n_k, - \rangle$$

Cuando estamos leyendo letras borradas, hace falta leer la letra que corresponde con el n_i más a la derecha; es decir:

$$\langle n_1, n_2, \dots, n_i, - \rangle \cdot \bar{a}_i = \langle n_1, n_2, \dots, n_i - 1, - \rangle$$

$$\langle n_1, n_2, \dots, n_{i-1}, 1, - \rangle \cdot \bar{a}_i = \langle n_1, n_2, \dots, n_i, - \rangle$$

Llegamos al estado final por:

$$\langle 1, - \rangle \cdot \bar{a}_1 = q_+$$

Indiquemos como funciona T_k sobre las palabras de $(2n + 1)P_k$.

Si se satisface:

$$q_0 \cdot w_1 = \langle n_1, n_2, \dots, n_i, - \rangle$$

La única posibilidad es:

$$w_1 = a_1^{n_1} a_2^{n_2} \dots a_i^{n_i}$$

Por otra parte tenemos que si:

$$\langle n_1, n_2, \dots, n_i, - \rangle \cdot w_2 = q_+$$

La palabra w_2 es de la forma:

$$w_2 = \bar{a}_1^{n_1} \dots \bar{a}_2^{n_2} \bar{a}_1^{n_1}$$

Notemos que A_k es determinista, accesible y coaccesible. Vemos que todo cálculo en dicho autómata se factoriza de modo único en:

$$q_- \xrightarrow{w_1} \langle n_1, n_2, \dots, n_k, + \rangle \xrightarrow{\#} \langle n_1, n_2, \dots, n_k, - \rangle \xrightarrow{w_2} q_+$$

con lo que:

$$f = w_1 \# w_2 = a_1^{n_1} a_2^{n_2} \dots a_k^{n_k} \# \bar{a}_k^{n_k} \dots \bar{a}_2^{n_2} \bar{a}_1^{n_1}, \quad \sum_{i=1}^k n_i \leq n$$

Es decir, $L(A_k) = (2n + 1)P_k$. Sabemos además que

$$\|Q\| = 2 \|FG(L_k \cap nX^*)\|$$

(b) Vamos a demostrar en esta segunda parte, que el autómata que hemos definido precedentemente, es pequeño. Para ello vamos a mostrar que cualquier autómata que acepte el lenguaje $(2n + 1)P_k$ tiene al menos

$$2 \|FG(L_k \cap nX^*)\|$$

Sea $B_k = \langle X_k, Q, Q_-, Q_+, T \rangle$ un autómata pequeño para $(2n + 1)P_k$.

La pequeñez de B_k impone la accesibilidad y coaccesibilidad de todos los estados. Vamos a razonar por reducción al absurdo. La prueba tiene lugar en tres pasos.

(1) Vamos a demostrar que todos los factores izquierdos de $L_k \cap nX$ dan estados distintos. Formalmente consideremos el conjunto de estados

$$Q_1 = Q_- \cdot FG(L_k \cap nX^*)$$

queremos demostrar que:

$$\|Q_1\| = \|FG(L_k \cap nX^*)\|$$

Supongamos que el cardinal de Q_1 es más pequeño, esto significa que existen w_1, w_2 , $w_1 \neq w_2$ en $FG(L_k \cap nX^*)$ con $Q_-.w_1 \cap Q_-.w_2 \neq \emptyset$.

Las palabras w_1 y w_2 tienen la forma:

$$\left. \begin{aligned} w_1 &= a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} \\ w_2 &= a_1^{n'_1} a_2^{n'_2} \dots a_j^{n'_j} \end{aligned} \right\} \text{con } i \neq j \text{ ó } \exists 1 \leq l \leq j \text{ que } n_l \neq n'_l$$

Comencemos con $i \neq j$. Supongamos $i < j$. Sabemos que existen q_1 y q_2 en Q_- con $q_1 \cdot w_1 = q_2 \cdot w_2$. Por coaccesibilidad, existe:

$$w \in a_j^* a_{j+1}^+ \dots a_k^+ \# \bar{a}_k^+ \dots a_1^+$$

que satisface $q_2 \cdot w_2 w \in Q_+$. En consecuencia w tiene la forma:

$$w = a_j^{p_j} a_{j+1}^{p_{j+1}} \dots a_{j_k}^{p_{j_k}} a_{j_k}^{p_{j_k}} \dots \bar{a}_{j+1}^{p_{j+1}} \bar{a}_j^{p_j+n'_j} \dots \bar{a}_2^{n'_2} \bar{a}_1^{n'_1}$$

con $p_j \geq 0$, $p_l > 0$ si $j+1 \leq l \leq k$.

Llegamos entonces a una contradicción. Por una parte, la palabra $w_1 w$ tiene la forma:

$$\begin{aligned} w_1 w &= a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} a_j^{p'_j} a_{j+1}^{p_{j+1}} \dots a_{j_k}^{p_{j_k}} \# \bar{a}_{j_k}^{p_{j_k}} \dots \bar{a}_{j+1}^{p_{j+1}} \bar{a}_j^{p_j+n'_j} \dots \\ &\dots \bar{a}_2^{n'_2} \bar{a}_1^{n'_1} \end{aligned}$$

con $p_j + n'_j > p_j$ y en consecuencia $w_1 w_2 \notin (2n+1)P$; pero por otra parte $q_1 \cdot w_1 w \in Q_+$.

Trataremos el caso $i = j$. En este caso:

$$w_1 = a_1^{n_1} \dots a_2^{n_2} \dots a_i^{n_i}$$

$$w_2 = a_1^{n'_1} \dots a_2^{n'_2} \dots a_i^{n'_i} \quad \text{con } n_1 \neq n'_1$$

La coacesibilidad de $q_2 \cdot w_2$ implica que existe w que satisface $q_2 \cdot w_2 w \in Q_+$, siendo dicha palabra de la forma:

$$w = a_i^{p_i} a_{i+1}^{p_{i+1}} \dots a_{i_k}^{p_{i_k}} \# \bar{a}_{i_k}^{p_{i_k}} \dots \bar{a}_{i+1}^{p_{i+1}} \bar{a}_i^{p_i+n'_i} \dots \bar{a}_1^{n'_1} \dots \bar{a}_1^{n'_1}$$

Tenemos entonces que en la palabra $w_1 w$ los coeficientes de a_1 y \bar{a}_1 no coinciden, con lo que llegamos a una contradicción.

(2) Vamos a tratar en esta segunda etapa, el conjunto de estados

$$Q_2 = Q_- \cdot \left\{ a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} a_{i+1} \dots a_k \# \bar{a}_k \dots \bar{a}_{i+1} \mid a_1^{n_1} a_2^{n_2} \dots \dots a_i^{n_i} \in \text{FG}(L_k \cap nX^*) \right\}$$

Argumentos análogos a los empleados en el caso (a) nos permiten mostrar:

$$\|Q_2\| = \left\{ a_1^{n_1} \dots a_i^{n_i} a_{i+1} \dots a_k \# \bar{a}_k \dots \bar{a}_{i+1} \mid a_1^{n_1} \dots a_i^{n_i} \in \text{FG}(L_k \cap nX^*) \right\}$$

(3) Queremos mostrar finalmente que $Q_1 \cap Q_2 = \emptyset$. Supongamos que exista $q \in Q_1 \cap Q_2$. Tenemos entonces que existen:

$$w_1 = a_1^{n_1} a_2^{n_2} \dots a_i^{n_i}$$

$$w_2 = a_1^{n'_1} a_2^{n'_2} \dots a_j^{n'_j} a_{j+1} \dots a_k \# \bar{a}_k \dots \bar{a}_{i+1}$$

$q_1, q_2 \in Q_-$ con $q_1 w_1 = q_2 w_2$. La palabra w_2 puede completarse con una palabra w de la forma:

$$w = \bar{a}_j^{n'_j} \dots \bar{a}_2^{n'_2} \bar{a}_1^{n_1}$$

Tenemos entonces de la palabra $w_1 w$ que es aceptada por el autó-mata:

$$w_1 w = a_1^{n_1} a_2^{n_2} \dots a_i^{n_i} \bar{a}_j^{n'_j} \dots \bar{a}_2^{n'_2} \bar{a}_1^{n_1}$$

no pertenece a $(2n + 1)P_k$ dado que al menos falta $\#$.

Dado que existe la igualdad:

$$\begin{aligned} & \left\| a_1^{n_1} \dots a_i^{n_i} a_{i+1} \dots a_k \# \bar{a}_k \dots \bar{a}_{i+1} \mid a_1^{n_1} \dots a_i^{n_i} \in FG(L_k \cap nX^*) \right\| = \\ & = \left\| FG(L_k \cap nX^*) \right\| \end{aligned}$$

y dado que $Q_1 \cap Q_2 = \emptyset$ tenemos que cualquier autómata pequeño para P_k tiene al menos $2 \left\| FG(L_k \cap nX^*) \right\|$.

Considerando (a) y (b) conjuntamente llegamos a:

$$\mu_{P_k}(2n + 1) = 2 \left\| FG(L_k \cap nX^*) \right\| = 2 \binom{n+1}{k} = \Theta(n^k)$$

■

Notemos que podemos obtener fácilmente que $P_k \in \text{Pol } \underline{\mu}$ para todo $k \geq 1$. Para ello notemos que dichos lenguajes pueden ser contruidos por substitución como sigue:

Definamos el lenguaje:

$$S_{k+1} = \left\{ a_{k+1}^n \# \bar{a}_{k+1}^n \mid n \geq 1 \right\}$$

y la substitución:

$$\sigma_{k+1} = \langle \# \rightarrow S_{k+1} ; a_i \rightarrow a_i ; \bar{a}_i \rightarrow \bar{a}_i \mid 1 \leq i \leq k \rangle$$

tenemos entonces que:

$$P_1 = S_1$$

$$P_{k+1} = \sigma_{k+1}(P_k) , \quad k \leq 1$$

Dado que $\mu_{S_1}(n) = \Theta(n)$ y que las operaciones de substitución conservan el carácter polinomial, obtenemos:

$$P_k \in \text{Pol}\mu, \quad k \geq 1$$

La ventaja de las construcciones del lema precedente, residen en que nos permiten situar P_k dentro de la jerarquía de los polinomios. En particular, nos permiten mostrar que los distintos grados de polinomios, forman familias estrictamente incluidas las unas en las otras.

La sucesión de lenguajes P_k nos permite demostrar el carácter no principal de $\text{Pol}\mu$.

Antes de proceder a la demostración, recordemos algunas notaciones. Sea L un lenguaje, notaremos $\mathcal{F}(L)$ la clausura de L por las operaciones de full-A.F.L., notaremos $\mathcal{C}(L)$ su clausura por las operaciones de cono racional. El teorema siguiente [6] debido a Ginsburg y Greibach, nos permite hallar un generador de cono a partir de un generador de full-A.F.L.

Teorema 1.2 Se cumple que:

$$\mathcal{F}(L) = \mathcal{C}((L \#)^*)$$



Vamos finalmente a demostrar la no principalidad de $\text{Pol}\mu$; tenemos:

Lema 1.12 La familia de los lenguajes polinomiales algebraicos son un full A.F.L., no principal.

Prueba Razonemos por reducción al absurdo. Supongamos que existe $L \in \text{Pol}\mu$ que satisface $\mathcal{F}(L) = \text{Pol}\mu$. Sabemos que existe k tal

que $\mu_L(n) = O(n^k)$.

Utilizando el teorema precedente tenemos que $\mathcal{L}((L\#)^*) = \underline{\text{Pol}} \mu$.

Se cumple $\mu_{(L\#)^*}(n) = O(n^{k+1})$.

El teorema de Elgolt y Mezei, nos permite decir que para cualquier

$L' \in \underline{\text{Pol}} \mu$ existe una transducción racional τ que satisface

$L' = \tau((L\#)^*)$.

Utilizando el teorema de paso del índice inicial, a través de las transducciones, obtenemos:

$$\mu_{L'}(n) \leq k_1(n+1) \cdot \mu_{L\#^*}(k_2 n)$$

con lo que $\mu_{L'}(n) = O(n^{k+2})$.

Tenemos entonces que el lenguaje P_{k+3} no puede ser obtenido a partir de L via las operaciones de full-A.F.L., llegamos, pues, a una contradicción.



Recapitulando los lemas y teorema definidos en los párrafos precedentes, obtenemos un teorema que nos permite caracterizar los lenguajes algebraicos de índice inicial polinomial en términos de familias abstractas de lenguajes, y situarlos respecto a las diferentes subfamilias algebraicas. Tenemos:

Teorema 1.3 Los lenguajes algebraicos de índice inicial polinomial son un full-A.F.L. no principal, cerrado por substitución, que contiene full-A.F.L. de los lenguajes a contador iterado.



Podemos esquematizar la situación en la figura 1.2:

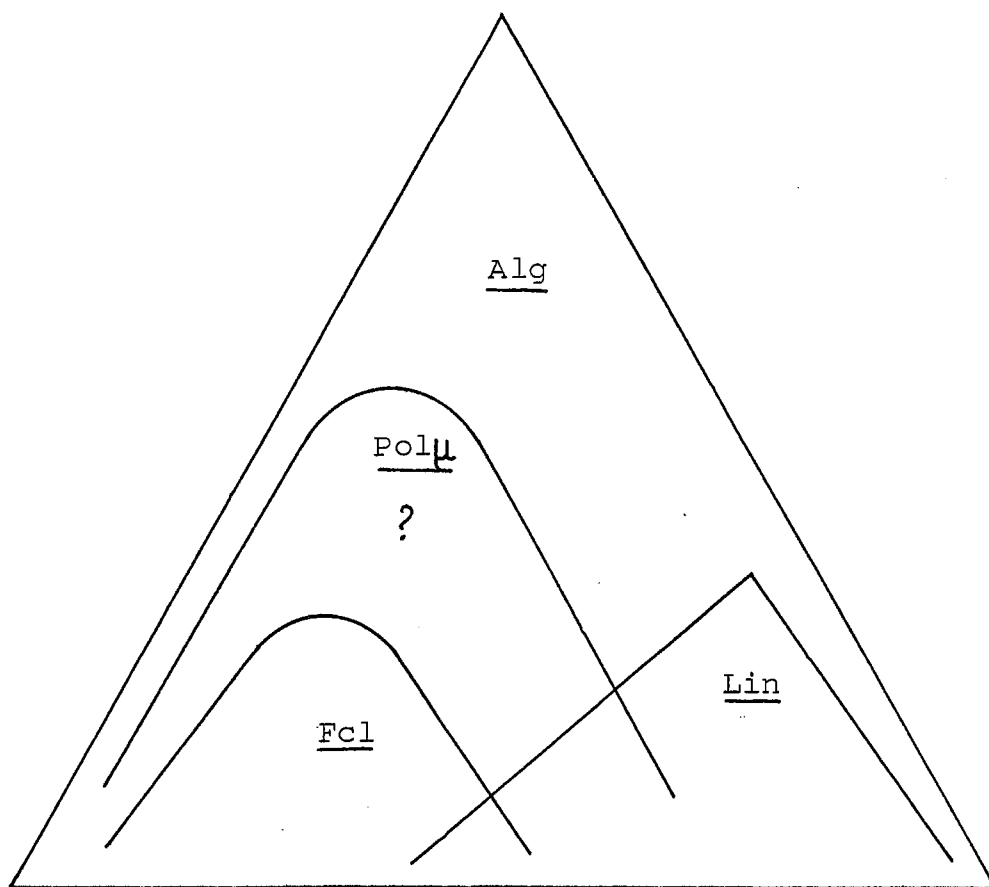


Figura 1.2

Queda abierto el problema de saber si la inclusión de Fcl en Polμ es o no estricta. A este respecto, conjeturamos:

Conjetura 1.1 La inclusión de los lenguajes a contador iterado, en los lenguajes polinomiales algebraicos, es estricta.

Notemos que hay lenguajes lineales exponenciales, por ejemplo S_2 .

Lenguajes a multicontador polinomiales.

En esta sección, vamos a estudiar la relación de los lenguajes polinomiales con respecto a diversas sub-familias bien conocidas, definidas a partir de multicontadores of-line.

La primera familia que vamos a estudiar, es la de los multicontadores que trabajan en tiempo casi real, parcialmente ciegos. Dichos multicontadores, son mucho más conocidos como lenguajes de redes de Petri con etiquetajes ϵ -libres, llamados en la literatura conjuntos de secuencias de computación \mathcal{LSS}

Estudiamos las propiedades de cierre de dichos lenguajes. Mostramos especial interés por las redes de Petri con una sola plaza, mostrando que en este caso obtenemos siempre lenguajes algebraicos. Mostramos, además, que hay lenguajes de estructura simple que no se pueden obtener con redes de Petri a una plaza. Damos finalmente la prueba del teorema bien conocido, que caracteriza \mathcal{LSS} a partir de D_1^* a partir de propiedades de cierre. Dicho teorema nos permite demostrar que los lenguajes de \mathcal{LSS} tienen índice inicial polinomial.

Como caso especial de \mathcal{LSS} estudiamos los lenguajes de Szilard, o lenguajes de etiquetas de los lenguajes algebraicos.

Mostramos que también dichos lenguajes son de índice inicial polinomial.

Damos, finalmente, la caracterización de los multicontadores que trabajan en tiempo casi real y en tiempo polinomial, y mostramos que dichos lenguajes son de índice inicial polinomial.

Comenzamos por dar la definición de multicontadores of-line y de los lenguajes por ellos aceptados [21 , 35] .Notemos Z los enteros, si $x \in Z$, diremos $\text{sgn}(x) = 1, 0, -1$, según que sea positivo, nulo o negativo.

Definición [35] 1.3 Una máquina $M = \langle X, Q, q_0, Q_+, \delta \rangle$ diremos que es un k-contador si X es un alfabeto finito de entrada, Q es el conjunto de estados, q_0 es el estado inicial, Q_+ es el subconjunto de los estados finales y la función de transición es una aplicación de $Q \times \{ X + \epsilon \} \times \{ 1, 0, -1 \}^k$ en las partes de $Q \times \{ 1, 0, -1 \}^k$

Llamaremos descripción instantánea ID $(q, w, y_1, y_2, \dots, y_k)$ a un triplete de $Q \times X^* \times Z^k$.

Si $(q', v_1, \dots, v_k) \in \delta (q, a, u_1, u_2, \dots, u_k)$ y

$(q, aw, y_1, y_2, \dots, y_k)$ es una ID y $\text{sgn}(u_i) = \text{sgn}(y_i), 1 \leq i \leq k$ podemos escribir $(q, aw, y_1, \dots, y_k) \vdash (q', w, y_1 v_1, \dots, y_k + v_k)$.

Si $a = \epsilon$ hemos realizado un ϵ -movimiento. Notaremos \vdash^* el cierre reflexivo transitivo de la relación antes definida.

Llamaremos $L(M)$ al lenguaje:

$$L(M) = \left\{ w \in X^* \mid (q_0, w, 0, 0, \dots, 0) \vdash^* (q', \epsilon, 0, 0, \dots, 0), q' \in Q_+ \right\}$$

La definición siguiente sirve para dar restricciones al tiempo de cálculo de los multicontadores.

Definición 1.4

. Diremos que un multicontador es casitiemporeal de retraso d (quasirealtime of delay d) si siempre que:

$$(q, \epsilon, y_1, \dots, y_k) \stackrel{n}{\vdash} (q', \epsilon, y'_1, \dots, y'_k) \text{ tenemos } n \leq d$$

. Diremos que un multicontador M acepta con tiempo polinomial k , si para todo $w \in L(M)$ existe un cálculo :

$$(q_0, w, 0, 0, \dots, 0) \stackrel{n}{\vdash} (q, \epsilon, 0, 0, \dots, 0), \quad q \in Q_+ \text{ con } n = O(|w|^k).$$

Damos ahora dos familias que vamos a emplear seguidamente:

Definición 1.5 Llamaremos:

Contador (ctr) = $\{ L \mid \text{Existe un multicontador casitiemporeal que acepta } L \}$

Contador (pol) = $\{ L \mid \text{Existe un multicontador que acepta } L \text{ con tiempo polinomial.} \}$

Además de limitar el tiempo de cálculo, podemos también restringir los contenidos y acceso de la memoria. Podemos, por ejemplo, impedir que la memoria contenga números negativos, y forzar a las transiciones a ignorar el contenido de la memoria. Estas consideraciones, nos conducen a definir los multicontadores parcialmente ciegos (partially blind), notados p -ciegos.

Definición 1.6 Diremos que un multicontador es p -ciego ssi para todo $q \in Q$ y $a \in X \cup \epsilon$ se cumple:

$$(1) \quad \delta(q, a, u_1, \dots, u_k) = \emptyset \text{ si } u_i = -1, \quad 1 \leq i \leq k$$

$$(2) \quad \delta(q, a, u_1, \dots, u_k) = \delta(q, a, v_1, \dots, v_k) \text{ para todo } u_i, v_i \text{ en } \{0, 1\}, \quad 1 \leq i \leq k$$

Podemos ahora definir:

Definición 1.7

p-ciego(ctr) = $\{ L \mid \text{existe un multicontador casitiemporeal, p-ciego que acepta } L. \}$

Como caso especial de multicontadores, comenzaremos por estudiar los lenguajes de las redes de Petri [4 8]. Para ello, comencemos por recordar la definición clásica de red marcada de Petri y de secuencias de disparo:

Definición [4 9] 1.8 Llamaremos red de Petri marcada a un sextúplete:

$$R = \langle P, T, I, O, \mu_0, F \rangle$$

en la que:

- . El conjunto de plazas es $P = \{ p_1, p_2, \dots, p_j, \dots, p_q \}$
- . El conjunto de las transiciones es $T = \{ t_1, t_2, \dots, t_i, \dots, t_r \}$
- . La relación entre las plazas y las transiciones está especificada por:

$$I : T \rightarrow \mathbb{N} \times P, \quad O : T \rightarrow \mathbb{N} \times P$$

Si $I(t_i) = \langle n_1 p_1, n_2 p_2, \dots, n_j p_j, \dots, n_q p_q \rangle$ notaremos $I(t_i, p_j) = n_i$

Si $O(t_i) = \langle n'_1 p_1, n'_2 p_2, \dots, n'_j p_j, \dots, n'_q p_q \rangle$ notaremos $O(t_i, p_j) = n'_i$

. Llamaremos marcaje μ a todo vector de \mathbb{N}^q .

. Diremos que la transición t_i está dispuesta (enabled) en μ si:

$$\mu(p_j) \geq I(t_i, p_j), \quad 1 \leq j \leq q$$

Cuando la transición t_i se dispara (fires) notaremos $\delta(\mu, t_i) = \mu'$

en qué:

$$\mu'(p_j) = \mu(p_j) - I(t_i, p_j) + O(t_i, p_j)$$

. Considerando el cierre reflexivo transitivo de δ llamaremos conjunto de secuencias de disparo a:

$$T(R) = \{ w \in T^* \mid \delta(\mu_0, w) \in F \}$$

Podemos ahora definir los lenguajes asociados a las redes de Petri por:

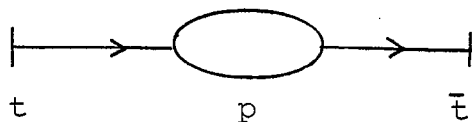
Definición [48] 1.9 Llamamos conjunto de secuencias de computación \mathcal{LSS} (computation sequences sets) a la familia:

$$\mathcal{LSS} = \{ L \mid \exists R, \sigma \text{ estrictamente alfabético, tales que } L = \sigma(T(R)) \}$$

Vamos a dar seguidamente ejemplos de lenguajes en \mathcal{LSS}

Ejemplo 1.2 El lenguaje de Dyck restringido sobre una letra, pertenece a \mathcal{LSS} , formalmente $D_1^* = \langle S \rightarrow tS \bar{t}S + \epsilon \rangle \in \mathcal{LSS}$

Para ello consideremos la red de Petri R siguiente:



en la que $\mu_0 = \langle 0 \rangle$ y $F = \{ \langle 0 \rangle \}$

Sea $w \in \{t, \bar{t}\}^*$ definimos $\|w\| = |w|_t - |w|_{\bar{t}}$, notemos

$$L = \{ w \in \{t, \bar{t}\}^* \mid \forall w' < w \text{ tenemos } \|w'\| \geq 0 \}$$

Vamos a demostrar:

$$\delta(\mu_0, w) = \langle n \rangle \text{ ssi } w \in L \text{ y } \|w\| = n$$

(a) Demostremos que $w \in L$ y $\|w\| = p$, implica $\delta(\mu_0, w) = \langle p \rangle$ razonando sobre la longitud de las palabras.

. Cuando $w = \varepsilon$ es verdad dado que $\delta(0, \varepsilon) = \langle 0 \rangle$.

. Sea w de longitud $n + 1$, tenemos $w = w'u$ con $u \in \{t, \bar{t}\}$ con $w' \in L$, supongamos $\|w'\| = q$, por hipótesis de recurrencia

$$\delta(\mu_0, w') = \langle q \rangle$$

Tratemos $u = t$, en este caso $\|w\| = \|w'\| + 1 = q + 1$ y

$$\delta(\mu_0, w't) = \delta(\delta(\mu_0, w')t) = \delta(\langle q \rangle, t) = \langle q + 1 \rangle. \text{ Trate-}$$

mos $u = \bar{t}$, en este caso $\|w\| = \|w'\| - 1$, dado que $w \in L$, tenemos

$$\|w'\| = q > 0 \text{ y } \delta(\mu_0, w'\bar{t}) = \delta(\langle q \rangle, \bar{t}) = \langle q - 1 \rangle$$

(b) Demostremos $\delta(\mu_0, w) = \langle p \rangle$ implica $w \in L$ y $\|w\| = p$; razonando sobre el número de transición.

. Cuando el número de transición es 0, tenemos $w = \varepsilon$ y $\langle p \rangle = \mu_0 = 0$ con lo que se satisface la hipótesis.

. Trataremos el caso $n + 1$ transiciones, tenemos la factorización

$$\delta(\mu_0, w'u) = \delta(\delta(\mu_0, w'), u)$$

dado que w' se obtiene con n transiciones, tenemos

$$w' \in L \text{ y } \|w'\| = \delta(\mu_0, w') = \langle p \rangle$$

Cuando $u = t$, $\|w\| = \langle p \rangle + 1$. Dado que los factores izquierdos de w son $w't$ más los de w' , obtenemos que $w \in L$.

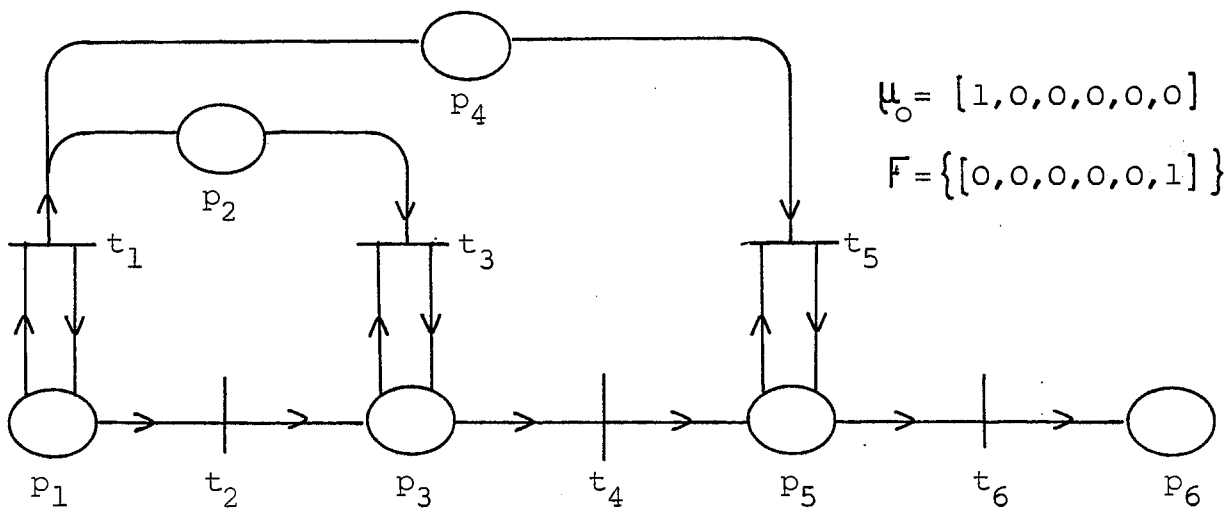
Cuando $u = \bar{t}$ tenemos $\|w\| = \langle p \rangle - 1$ con $p > 0$, también en este caso $w \in L$.

Dado que [6], que $w \in D_1^*$ ssi $w \in L$ y $\|w\| = 0$ obtenemos:

$$D_1^* = T(R)$$

Este ejemplo nos muestra que \mathcal{LSS} contiene algunos lenguajes algebraicos; el ejemplo siguiente muestra que contiene también lenguajes no algebraicos.

Ejemplo 1.3 Consideremos la red de Petri R y el morfismo σ siguientes:



$$\sigma = \langle t_1 \rightarrow a; t_2 \rightarrow e; t_3 \rightarrow b; t_4 \rightarrow e; t_5 \rightarrow c; t_6 \rightarrow e \rangle$$

$$\{a^n e b^n e c^n \mid n \geq 1\} = \sigma(T(R))$$

Posteriormente daremos una prueba formal de esta igualdad.

Damos sin demostración, la proposición siguiente, que caracteriza las secuencias de computación en términos de multicontadores. Un esquema de la prueba se halla en [35 , pag 321].

Lema 1.13 $\mathcal{LSS} = p\text{-ciego}(\text{ctr})$

Vamos a estudiar seguidamente algunas propiedades de la familia \mathcal{LSS} . Dichas propiedades de cierre se hallan estudiadas en [48,36]

Lema 1.14 Los lenguajes racionales, se hallan contenidos en \mathcal{LSS} .

Prueba Consiste en simular un autómata a partir de una red de Petri. Sea L reconocido por un autómata determinista

$$= \langle \{x_1, \dots, x_n\}, \{q_1, \dots, q_p\}, q_1, \{q_1^!, \dots, q_1^!\}, \dots \rangle$$

Construyamos una red de Petri R en que a cada estado de \mathcal{A} asociamos una plaza $\{p_1, \dots, p_n\}$. A cada arco dado por la función de transición (q_i, x_r, q_j) corresponde una transición $t_{i,j}^r$ con $I(t_{i,j}^r) = p_i$ y $O(t_{i,j}^r) = p_j$. El marcaje inicial μ_1 es

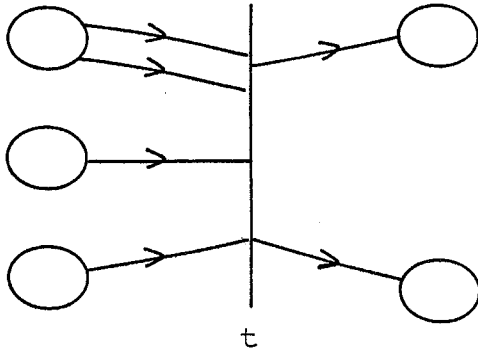
$\mu_1 = (1, 0, \dots, 0)$, a cada estado final $q_i^!$ asociamos un marcaje $\mu_i^! = (0, 0, \dots, 1, \dots, 0)$ en el que 1 se encuentra en la posición i . Considerando $\sigma : T \rightarrow \{x_1, \dots, x_n\}$ definido por

$$\sigma(t_{i,j}^r) = x_r \text{ obtenemos } \sigma(T(R)) = L$$

Vamos a estudiar las propiedades de cierre con relación al morfismo.

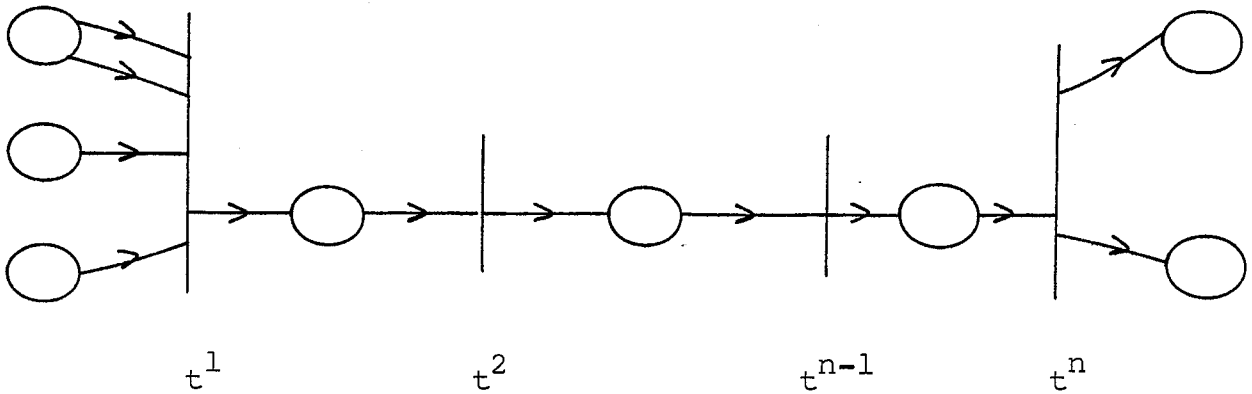
Lema 1.15 $\mathcal{E}SS$ es estable por morfismo \mathcal{E} -libre.

Prueba El esquema siguiente muestra esquemáticamente la construcción a realizar.



Con $\sigma : T \longrightarrow X, \quad \varphi : X \longrightarrow Y^+$
 $\sigma(t) = x \quad \varphi(x) = y_1 y_2 \cdots y_n$

La transición precedente se separa en



Definamos ahora en la nueva red obtenida el etiquetaje:

$$\sigma'(t^1) = y_1, \quad \sigma'(t^2) = y_2, \quad \dots, \quad \sigma'(t^n) = y_n$$



Vamos a tratar seguidamente la operación de morfismo.

Lema [36] 1.16 es estable por morfismo inverso.

Prueba Sea $\varphi : X^* \rightarrow Y^*$ un morfismo. Sea $R = \langle P, T, I, O, \mu, F \rangle$ una red de Petri de etiquetaje $\sigma : T \rightarrow Y$.

Vamos a construir una nueva red R' y un nuevo etiquetaje σ' , que satisfagan $\varphi^{-1}(\sigma(L(R))) = \sigma'(L(R'))$.

Sea $f = t_1 t_2 \dots t_n \in T^*$ tal que $\sigma(f) = \sigma(t_1 t_2 \dots t_n) = \varphi(x)$ con $x \in X$. En este caso vamos a asociar en R' una sola transición que notaremos $[f]$ a esta sucesión de transiciones que "vienen de x ". Formalmente $R' = \langle P, T', I', O', \mu', F \rangle$ con

$$T' = \{ [f] \mid f \in T^* \text{ y } \exists x \in X \text{ con } \sigma(f) = \varphi(x) \}$$

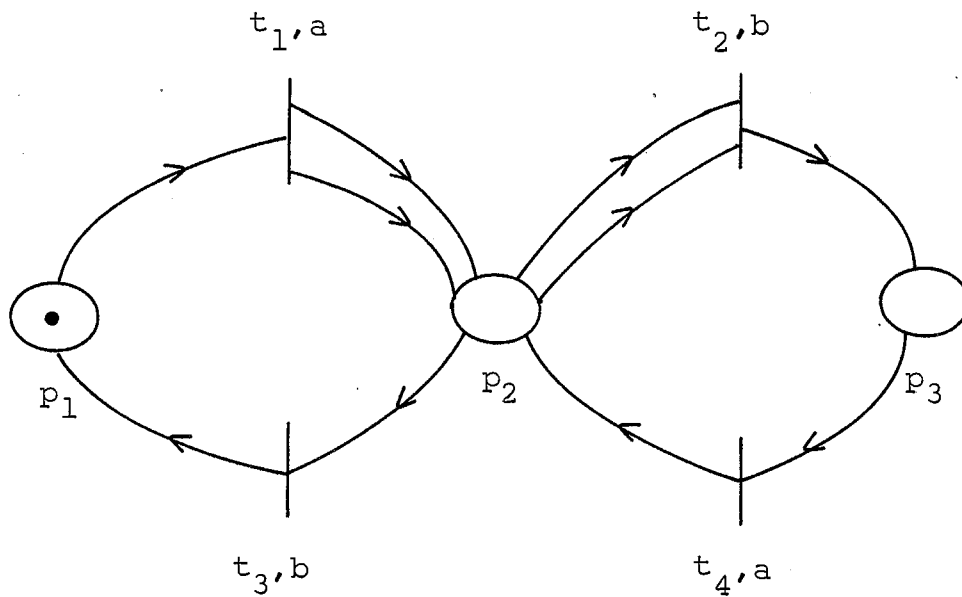
La función input y output estarán definidas de manera a simular la secuencia de transiciones f en un marcaje. Es decir, $I'([f])$ y $O'([f])$ deben definirse de modo a conservar $\mu \cdot f = \mu'$, entonces $\mu \cdot [f] = \mu'$ cualquiera que sea μ .

Vamos finalmente a definir el nuevo marcaje por:

$$\sigma'([f]) = x \text{ ssi } \varphi(x) = \sigma(f).$$

Demos seguidamente un ejemplo de tales construcciones.

Ejemplo 1.4 Consideremos la red de Petri.



$$\mu_0 = \langle 1, 0, 0 \rangle, \quad F = \{ \langle 1, 0, 0 \rangle \}$$

Consideremos el etiquetaje $\sigma = \langle t_1 \rightarrow a; t_2 \rightarrow b; t_3 \rightarrow b; t_4 \rightarrow a \rangle$

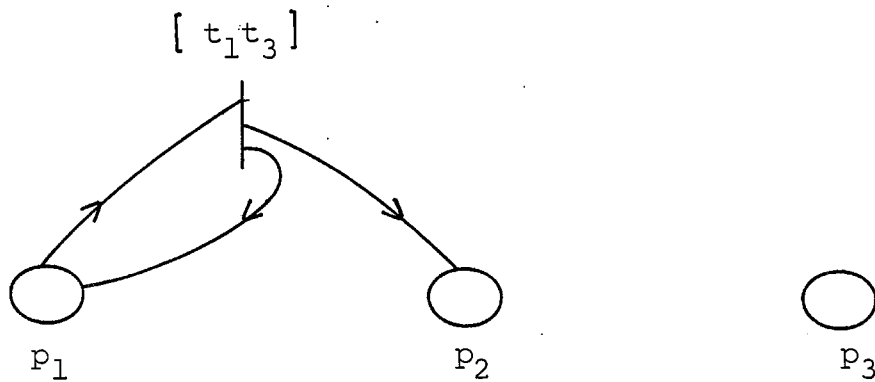
Sea ψ el morfismo $\psi = \langle x \rightarrow ab; y \rightarrow aa; z \rightarrow ba; r \rightarrow \epsilon \rangle$

Vamos a estudiar la sucesión de transiciones que pueden dar la imagen de una letra.

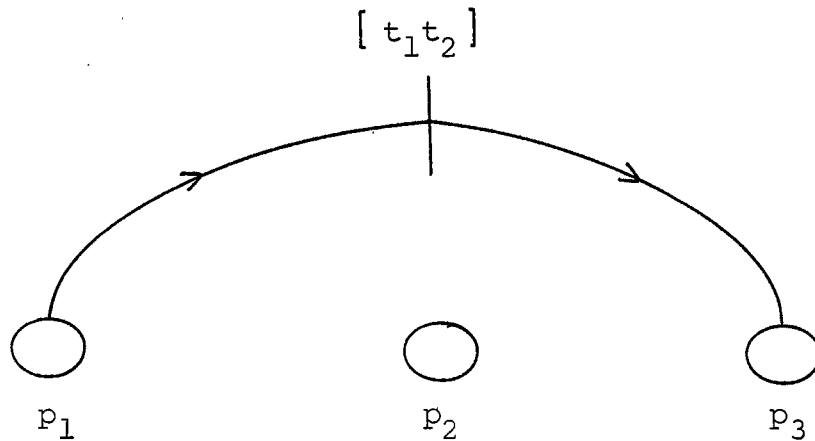
		Nueva transición
$\psi(x) = ab$	$t_1 t_3$ $t_1 t_2$	$[t_1 t_3]$ $[t_1 t_2]$
$\psi(y) = aa$	$t_1 t_1$ $t_1 t_4$ $t_4 t_1$ $t_4 t_4$	$[t_1 t_1]$ $[t_1 t_4]$ $[t_4 t_1]$ $[t_4 t_4]$
$\psi(z) = ba$	$t_3 t_1$ $t_2 t_1$	$[t_3 t_1]$ $[t_2 t_1]$
$\psi(r) = \epsilon$		$[\epsilon]$

Vamos a hallar las nuevas funciones input y output. Trataremos cada nueva transición separadamente.

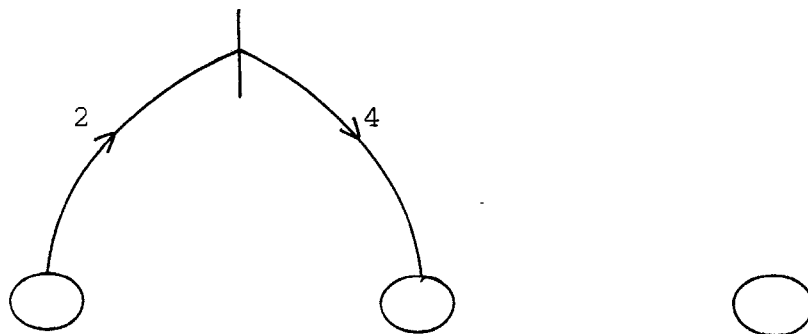
(1) Trataremos $[t_1 t_3]$. Cuando se dispara t_1 quita una marca de p_1 y pone dos en p_2 . Cuando se dispara p_2 quita una marca en p_2 y la pone en p_1 . Tenemos entonces que $[t_1 t_3]$ quita y pone una marca en p_1 y pone una marca en p_2 . Es decir:



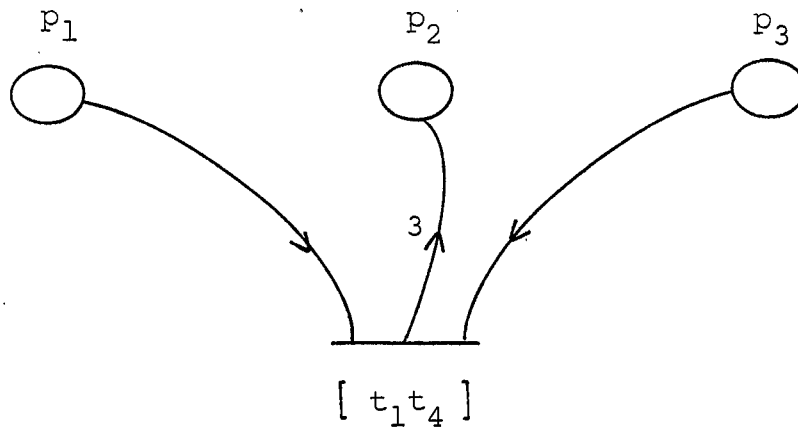
(2) Trataremos $[t_1 t_2]$.



(3) Trataremos $[t_1 t_1]$



(4) Tratemus $[t_1 t_4]$

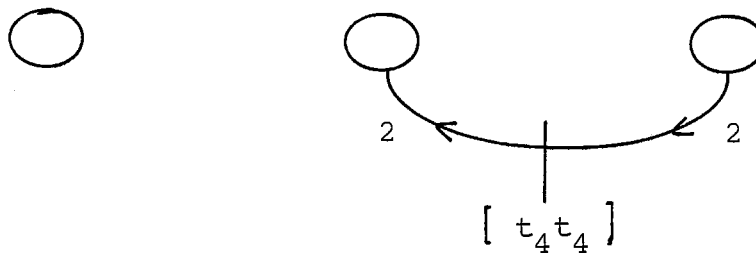


(5) Tratemus $[t_4 t_1]$.Dado que las transiciones t_4 y t_1 son independientes, $[t_4 t_1]$ y $[t_1 t_4]$ tienen las mismas funciones input y output.

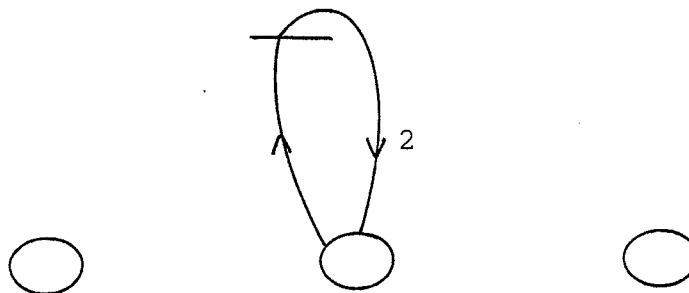
Como además, $\sigma (t_4 t_1) = \sigma (t_1 t_4) = \psi (z) = aa$, se va a cumplir que $\sigma '(t_4 t_1) = \sigma '(t_1 t_4) = z$. Es decir, ambas transiciones van a ser en todos los aspectos indistinguibles.

Podemos, entonces, confundirlas en una sola.

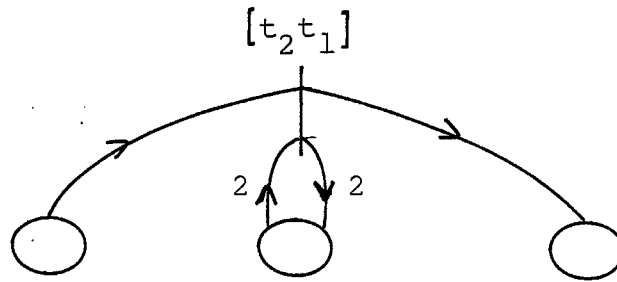
(6) Tratemus $[t_4 t_4]$



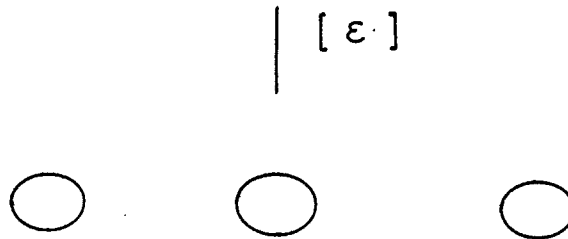
(7) Tratemus $[t_3 t_1]$: $[t_3 t_1]$



(8) Tratemos $[t_2 t_1]$



(9) Tratemos $[\epsilon]$. No tiene ni precondiciones ni postcondiciones.



Obtenemos finalmente la red dada en la figura 1.3.

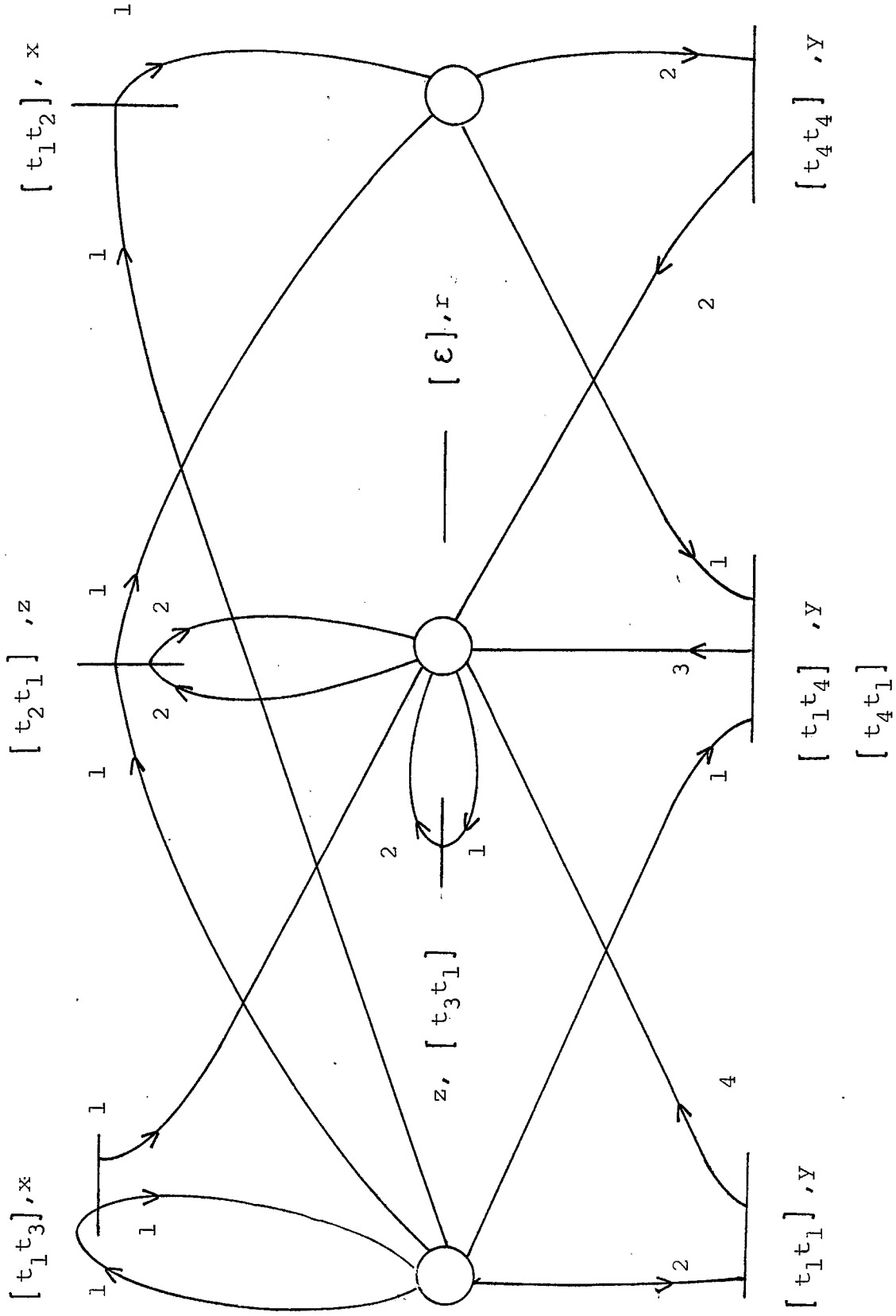


Figure 1.3

Vamos a tratar seguidamente la intersección.

Lema [49] 1.17 \mathcal{LSS} es estable por intersección.

Prueba Sea $R_1 = \langle P_1, T_1, I_1, O_1, \mu_1, F_1 \rangle$ con etiquetaje σ_1 y

$R_2 = \langle P_2, T_2, I_2, O_2, \mu_2, F_2 \rangle$ con etiquetaje σ_2 .

Tomamos $P_1 \cap P_2 = \emptyset$ y $T_1 \cap T_2 = \emptyset$.

Si $\sigma_1(t_i) = \sigma_2(t_j)$ con $t_i \in T_1$ y $t_j \in T_2$ creamos una nueva transición $t'_{ij} \in T_3$.

$I_3(t'_{ij}) = I_1(t_i) + I_2(t_j)$ y $O_3(t'_{ij}) = O_1(t_i) + O_2(t_j)$.

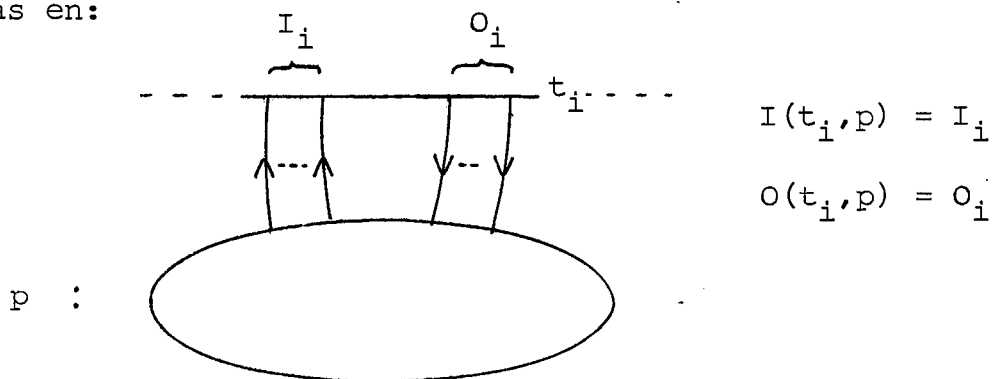
Entonces $R_3 = \langle P_1 \cup P_2, T_3, I_3, O_3, \mu_3, F_3 \rangle$. En que μ_3 restringido a P_1 es μ_1 y restringido a P_2 es μ_2 .

De modo análogo definimos F_3 . Definimos σ_3 por $\sigma_3(t'_{ij}) = \sigma_1(t_i) = \sigma_2(t_j)$.

Tenemos $\sigma_3(T(R_3)) = \sigma_1(T(R_1)) \cap \sigma_2(T(R_2))$

■

Queremos hallar un teorema de caracterización de \mathcal{LSS} en forma de operaciones de cierre. Dicho proceso nos conduce a estudiar las redes de Petri con una sola plaza. Dichas redes se hallan esquematizadas en:



Se cumple [14] que todos estos lenguajes son algebraicos, mas específicamente tenemos:

Lema 1.18 Sea R una red de Petri con una plaza, tenemos $T(R) \in \text{Roc1}$

Prueba Vamos a mostrar un autómata a pila, con un solo símbolo de pila z que lo reconoce.

Sea $R = \langle T, \{p\}, I, O, \mu_0, F \rangle$ con $T = \{t_1, \dots, t_r\}$, $\mu_0 \langle p \rangle$,

$F = \{ \langle f_1 \rangle, \langle f_2 \rangle, \dots, \langle f_l \rangle \}$ notaremos $I(t_i, p) = I_i$,

$O(t_i, p) = O_i$.

Definimos el autómata a pila $\mathcal{A} = \langle T, \{z\}, Q, q_0, Q_+, \mathcal{Z} \rangle$, con

z como único símbolo de pila, en el que:

$$Q = \{i, l, a\} \cup \bigcup_{\{i | I_i > 0\}} \{t_i^{[r]} \mid 1 \leq r \leq I_i\} \cup \bigcup_{\{i | \langle f_i \rangle \in F\}} \{f_i^{[r]} \mid 0 \leq r \leq f_i\}$$

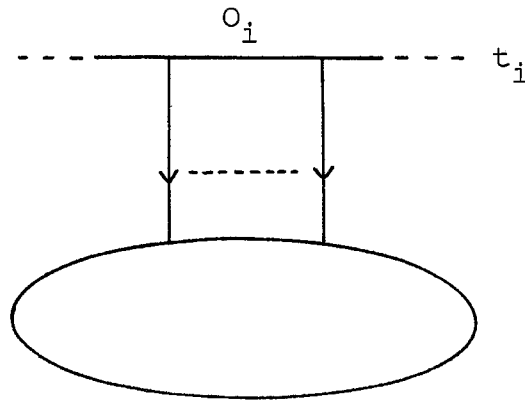
El estado "i" va a significar el estado inicial, "l" va a ser el estado de lectura y "a" va a ser el estado de aceptación, "f" va a ser el estado final. Tomaremos $q_0 = i, Q_+ = \{a\}$

La función de transición \mathcal{Z} está definida.

. Inicialización. A partir del estado inicial ponemos el marcaje inicial $\mu_0 \langle p \rangle$ en la pila, situándonos en el estado de lectura en el que podemos comenzar a leer transiciones:

$$(i, z) \xrightarrow{\epsilon} (l, z^{p+1})$$

. Supongamos que leemos la transición t_i que cumple $I_i = 0$. En este caso se halla esquematizado en:

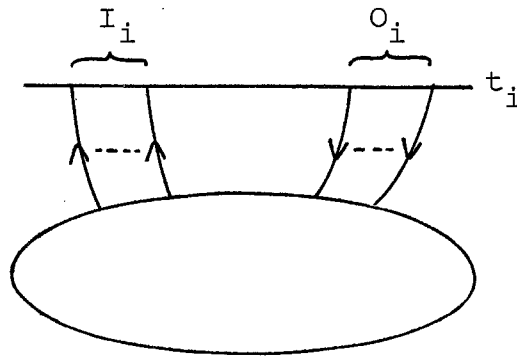


En este caso el número de marcas aumenta de O_i . Esto nos induce en

a la regla:

$$(1, z) \xrightarrow{t_i} (1, z^{O_i+1})$$

Supongamos que leemos t_i con $I_i > 0$. En este caso se halla esquematizado en:



Vamos a utilizar los estados $t_i^{[r]}$ $1 \leq r \leq I_i$ para depilar z^{I_i} símbolos, formalmente:

$$(1, z) \xrightarrow{t_i} (t_i^{[1]}, \varepsilon)$$

$$(t_i^{[1]}, z) \xrightarrow{\varepsilon} (t_i^{[2]}, \varepsilon)$$

$$(t_i^{[2]}, z) \xrightarrow{\varepsilon} (t_i^{[3]}, \varepsilon)$$

$$(t_i^{[I_i-1]}, z) \xrightarrow{\varepsilon} (t_i^{[I_i]}, \varepsilon)$$

Si el cálculo es correcto debe quedar como mínimo en la pila, el z que corresponde al fondo de pila, es decir, la pila no debe es-

tar vacía, tras esto añadimos las z^{0_i} que corresponden a t_i :

$$(t_i^{[1]}, z) \xrightarrow{\epsilon} (1, z^{0_i+1})$$

. Tras haber leído las transiciones, debemos verificar que en la pila quedan el buen número de marcas. Para ello, seleccionamos de manera no determinista un $\langle f_i \rangle$ al que corresponde un estado

$f_i^{[f_i]}$ y depilamos contando; formalmente:

$$(1, z) \xrightarrow{\epsilon} (f_i^{[f_i]}, z)$$

$$(f_i^{[f_i]}, z) \xrightarrow{\epsilon} (f_i^{[f_i-1]}, \epsilon)$$

$$(f_i^{[1]}, z) \xrightarrow{\epsilon} (f_i^{[0]}, \epsilon)$$

Tras esto depilamos el fondo de pila con "a".

$$(f_i^{[0]}, z) \xrightarrow{\epsilon} (a, \epsilon)$$

Evidentemente $\Gamma(R)$ es el lenguaje aceptado con pila vacía, tomando "a" como estado final.

Notemos \mathcal{LSS}_1 el conjunto de secuencias de computación que se pueden generar con una red de Petri a una plaza, tenemos:

Corolario 1.1 $\mathcal{LSS}_1 \subset \text{Roc1}$

Prueba Evidente, dado que Roc1 es estable por morfismo.

Vamos a mostrar ahora que $\mathcal{ESS}_1 \not\subseteq \text{Roc1}$. Vemos que si bien $D_1^* \in \mathcal{ESS}_1$, existen lenguajes de estructura "muy simple" que no pertenecen a \mathcal{ESS}_1 . Veamos por ejemplo que no podemos obtener lenguajes con "poca conmutación local".

Lema 1.19 [20] Se cumple $(ab)^* \notin \mathcal{ESS}_i$.

Prueba Vamos a razonar por reducción al absurdo, demostrando que debe existir una transición que decrece el número total de marcas.

Dicha transición podrá conmutar con la siguiente, obteniendo una palabra con dos letras iguales seguidas.

Formalmente, sean $R = \langle \{p\}, T, I, O, \mu_0, F \rangle$ y σ tales que

$$\sigma(T(R)) = (ab)^* .$$

Llamemos M al número máximo de marcas que hay en F , es decir:

$$M = \{ \max \mu \mid \mu \in F \}$$

Llamemos D_i a la oscilación máxima de marcas, tras el disparo de una transición, es decir: $D_i = O_i - I_i$, notemos:

$$D = \max \{ |D_i| \mid t_i \in T \}$$

Vamos a demostrar que existe un r y una secuencia de transiciones

$$W = t_{i_1} t_{i_2} \dots t_{i_{r-1}} t_{i_r} \text{ con } \delta(\mu_0, w) \in F \text{ y } \exists 1 \leq j \leq r \text{ con}$$

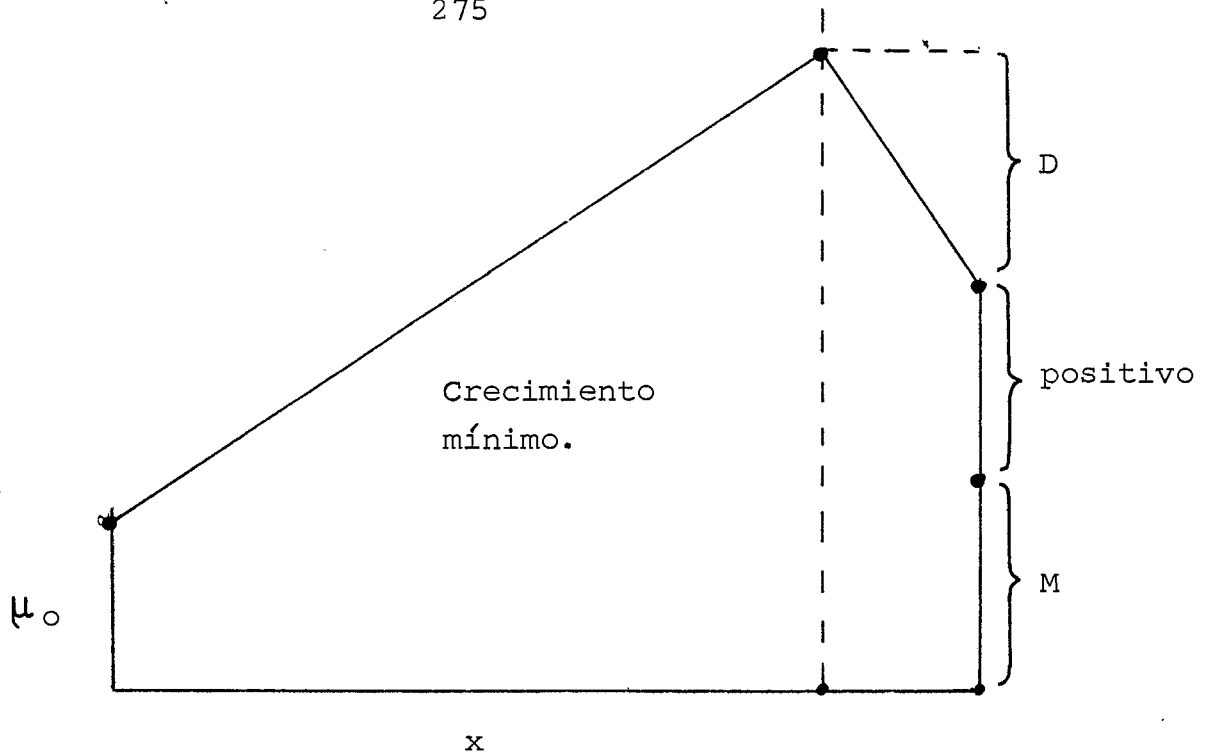
$$D_{i_j} \leq 0 .$$

Si una tal sucesión no existe, todas las sucesiones de transiciones aumentan el número de marcas, es decir, $D_i > 0$ para $t_i \in T$;

tomemos una sucesión de transiciones de longitud más larga que

$$M + D - \mu_0 .$$

La figura siguiente esquematiza la evolución de las marcas.



Tomemos $x > H + D - \mu_0$. Toda sucesión de transiciones de longitud $x + 1$ con todos los $D_i > 0$ crecientes hasta x , jamás podrá dar un marcaje en F .

Sea $t_1 t_2 \dots t_r t_{r+1}$ con $r \geq x$, sabemos que existe $1 \leq i \leq r$ con $D_i \leq 0$. La sucesión $t_1 \dots t_{i+1} t_i t_{i+2} \dots t_r$ también puede obtenerse; conduce al mismo marcaje final y las transiciones t_1 y t_{i+2} dan la misma letra por σ . Hemos obtenido una sucesión de transiciones con dos letras a o b seguidas.

Tenemos, entonces, una contradicción. ■

Veamos que de modo intuitivo, podemos decir que con una sola plaza no podemos "contar", es decir:

Ejemplo 1.5 $S_1 = \{a^n b^n \mid n \geq 0\} \notin \mathcal{LSS}_1$

Razonemos por reducción al absurdo. Supongamos que existe una red de Petri R con una sola plaza y un etiquetaje σ que cumple $S_1 = \sigma(\Gamma(R))$.

En dicha red debe existir una sucesión:

$$S = \{t_1 t_2 \dots t_n t'_{n+1} \dots t'_{2n} \mid n \geq 0\}$$

que satisface $\delta(\mu_0, t_1 t_2 \dots t_n t'_{n+1} \dots t'_{2n}) \in F$ y

$\sigma(t_1 \dots t_n t'_{n+1} \dots t'_{2n}) = a^n b^n$. Tenemos entonces que el conjunto

$$S' = \{\delta(\mu_0, t_1 t_2 \dots t_n) \mid \exists t'_{n+1} \dots t'_{2n} \text{ con } t_1 t_2 \dots t_n t'_{n+1} \dots t'_{2n} \in S\}$$

no es un conjunto acotado. De lo contrario, existen dos sucesiones de transiciones que satisfacen:

$$\delta(\mu_0, t_1 t_2 \dots t_n) = \delta(\mu_0, t_1 t_2 \dots t_{n'})$$

con $n' \neq n$ y en consecuencia obtenemos también la sucesión de disparos $t_1 t_2 \dots t_n t'_{n'+1} \dots t'_{2n'}$, que nos permite obtener $a^n b^{n'}$ con $n \neq n'$.

Dado que S' no está acotado, existe $t_1 t_2 \dots t_n t'_{n+1} \dots t'_{2n}$ en S con

$$\delta(\mu_0, t_1 t_2 \dots t_{n-1}) > I_{n+1} \text{ con lo que la sucesión :}$$

$t_1 t_2 \dots t_{n-1} t'_{n+1} t'_{n+2} \dots t'_{2n}$ también pertenece a la secuencia de disparos, con lo que hemos obtenido la palabra $a^{n-1} b a b^{n-1}$.

Veamos seguidamente que tampoco podemos obtener lenguajes con "demasiadas" estrellas.

Ejemplo 1.6 $a^*b^* \notin \mathcal{LSS}_1$.

Razonemos por reducción al absurdo. Las sucesiones de transiciones deben ser de la forma $t_1 t_2 \dots t_p t'_1 t'_2 \dots t'_q \in \mathcal{T}(\mathbb{R})$ con

$$\sigma(t_1 t_2 \dots t_p t'_1 t'_2 \dots t'_q) = a^p b^q$$

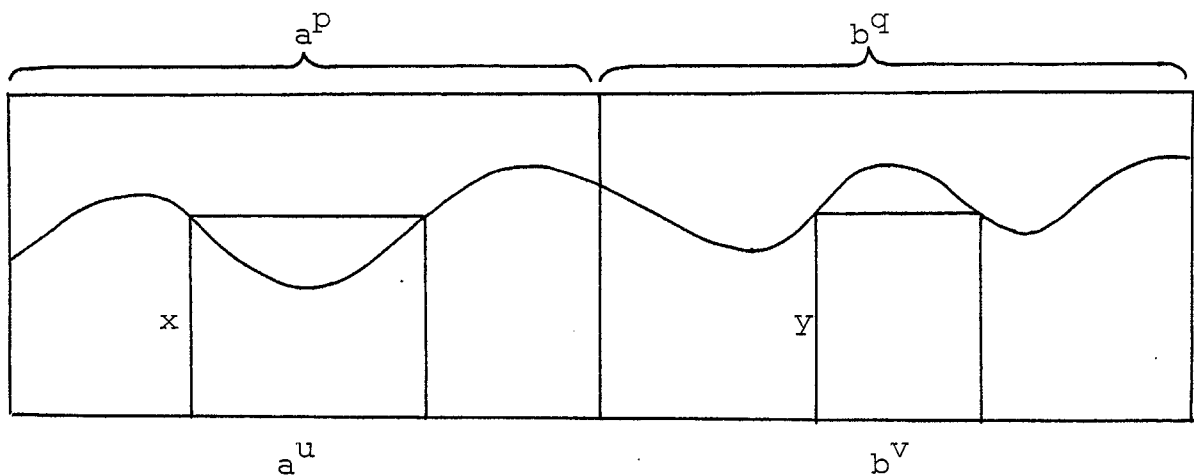
Tratemos tres casos separadamente.

(a) El número de marcas entre las transiciones que dan a, no está acotado. Formalmente:

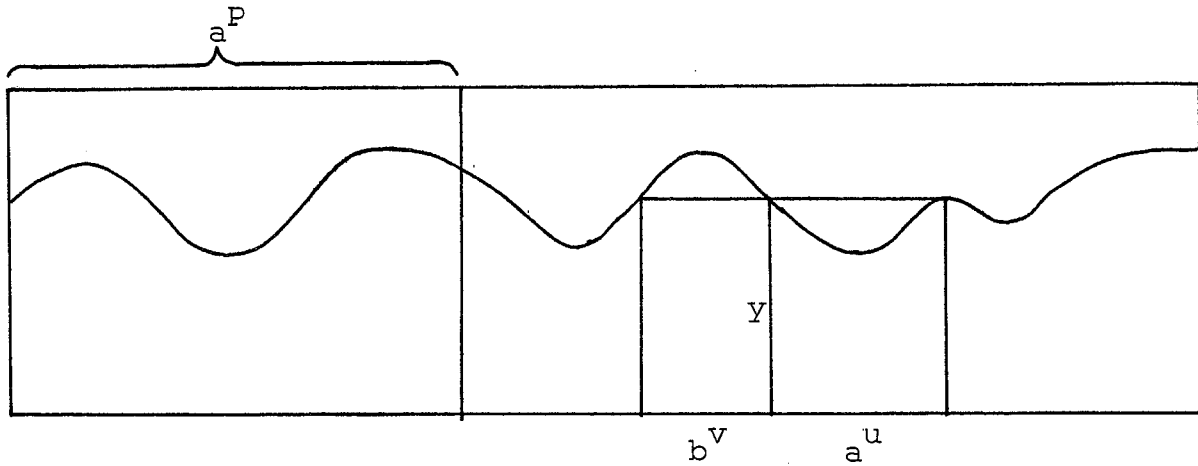
$\{ \delta(\mu_0, t_1 t_2 \dots t_i) \mid \exists t_{i+1} \dots t_p t'_1 \dots t'_q \text{ con } t_1 \dots t_i t_{i+1} \dots t_p t'_1 \dots t'_q \in \mathcal{T}(\mathbb{R}) \}$ no está acotado. Podremos, entonces, elegir $i < p$, e intercalar letras b, obteniendo una palabra en $a^* b^+ a^+ b^*$

(b) Recíprocamente el número de marcas entre las transiciones que dan b no está acotado. En este caso podremos intercalar letras a, entre letras b.

(c) Ambos tipos de sucesiones de transiciones, tienen un número acotado de marcas. En este caso, dado que $a^* b^*$ es infinito, para palabras con gran número de letras a y b, debe producirse la situación esquematizada en la figura.



En que $\delta(x, a^u) = x$ y $\delta(y, b^v) = y$. Si $x \leq y$ podemos también obtener la palabra:



Si $y < x$ podemos poner b^v a la izquierda de a^n . En ambos casos obtenemos una palabra que no está en a^*b^* .

Los ejemplos siguientes nos indican como deben ser los lenguajes de \mathcal{LSS}_1 contenidos en los acotados. Llamemos $\mathcal{L}_k = \mathcal{LSS}_1 \cap a_1^* a_2^* \dots a_k^*$. La proposición siguiente nos dice que \mathcal{L}_k puede contener "como máximo una estrella".

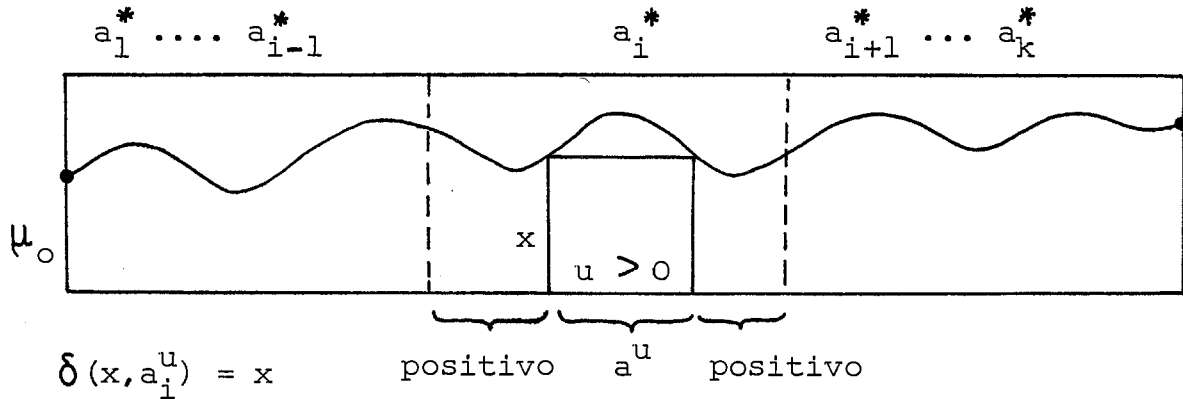
Lema 1.20 Sea $L \in \mathcal{L}_k$, se cumple que L es racional y que existe como máximo un único $1 \leq i \leq k$, tal que el conjunto

$$\{x_i \mid a_1^{x_1} a_2^{x_2} \dots a_i^{x_i} \dots a_k^{x_k} \in L\} \text{ es infinito.}$$

Prueba Sea R una red de Petri con una sola plaza para L . Si los

marcajes que aparecen al disparar las distintas transiciones no están acotados, podremos mezclar, cuando el número de marcas sea grande, las distintas letras, obteniendo palabras que no están en $a_1^* a_2^* \dots a_k^*$.

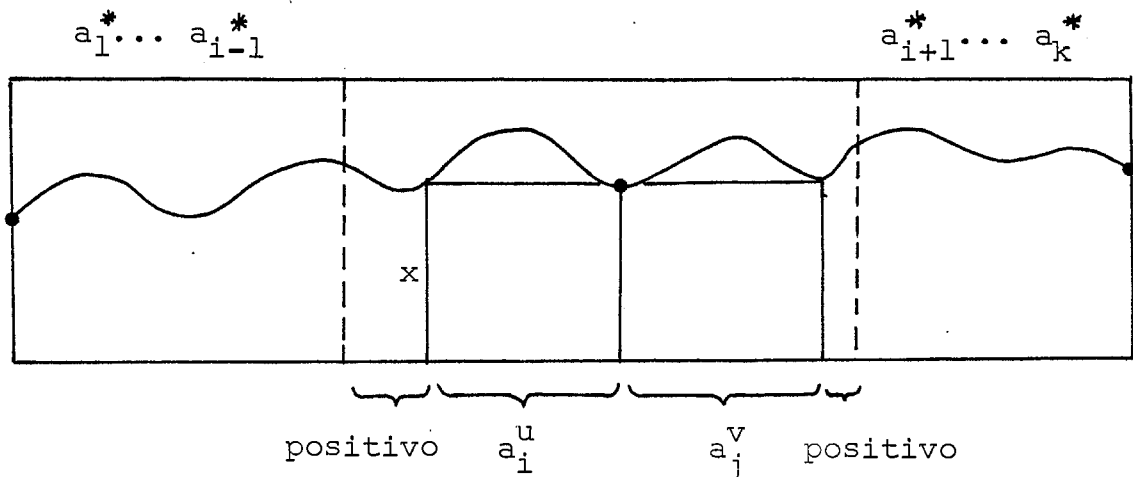
Si existen palabras que contienen un número arbitrariamente grande de a_i , debe contener una repetición; es decir:



Supongamos que existe una sucesión de palabras que contienen un número creciente de a_i , y otra sucesión que contiene un número creciente de a_j con $a_j \neq a_i$.

La primera sucesión genera una repetición; pongamos $\delta(x, a_i^u) = x$, la segunda también $\delta(y, a_j^v) = y$.

Supongamos $x \geq y$. Podemos obtener:



Vamos seguidamente a caracterizar la familia \mathcal{ESS} en términos de operaciones de cierre. Para ello vamos a introducir varias clases de operaciones de cierre.

- $\hat{\mathcal{H}}$ es la clase de todos los morfismos.
- \mathcal{H} es la clase de todos los morfismos ε -libres.
- \mathcal{H}^{-1} es la clase de todos los morfismos inversos.
- \wedge es la intersección de familias de lenguajes.

$$\mathcal{L}_1 \wedge \mathcal{L}_2 = \{ L_1 \cap L_2 \mid L_1 \in \mathcal{L}_1 \text{ y } L_2 \in \mathcal{L}_2 \}$$

Vamos a recordar seguidamente un teorema debido a S. Greibach [33] que afirma que en Rocl podemos considerar solamente morfismos directos ε -libres.

Teorema 1.4 Los lenguajes a contador restringidos satisfacen:

$$\underline{\text{Rocl}} = (\hat{\mathcal{H}}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}})_{D_1^*} = (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}})_{D_1^*}$$

podemos ahora enunciar el teorema de caracterización de \mathcal{ESS} .

Teorema [14,35,39] 1.5 La familia \mathcal{ESS} satisface:

$$\mathcal{ESS} = (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}}, \wedge)_{D_1^*} = \mathcal{H}(\wedge \underline{\text{Rocl}})$$

Prueba Vamos a demostrar:

$$(\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}}, \wedge)_{D_1^*} \subset \mathcal{ESS} \subset \mathcal{H}(\wedge \underline{\text{Rocl}}) \subset (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}}, \wedge)_{D_1^*}.$$

Comencemos por ver $(\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}}, \wedge)_{D_1^*} \subset \mathcal{ESS}$. Tenemos que $D_1^* \in \mathcal{ESS}$. Además \mathcal{ESS} es cerrado por morfismo ε -libre, morfismo inverso, contiene los racionales y es cerrado por intersec-

ción.

. Veamos que $\mathcal{E}SS \subset \mathcal{H}(\wedge \underline{RoCl})$. Sea R una red de Petri, demostremos que $\dot{T}(R) \subset \wedge \underline{RoCl}$. El proceso a seguir es expresar R como intersección de redes de Petri de una sola plaza.

Sea $R = \langle P, T, I, O, \mu_0, F \rangle$, $P = \{p_1, p_2, \dots, p_i, \dots, p_q\}$. Consideremos la restricción de R a la plaza p_i ;

$$R_{p_i} = \langle \{p_i\}, T, I_{p_i}, O_{p_i}, \mu_{p_i}, F_{p_i} \rangle.$$

Si $I(t_j) = \langle n_1 p_1, \dots, n_i p_i, \dots, n_q p_q \rangle$ entonces

$$I_{p_i}(t_j) = \langle 0, \dots, 0, n_i p_i, 0, \dots, 0 \rangle, \text{ análogamente}$$

$$O_{p_i}(t_j) = \langle 0, \dots, 0, n_i' p_i, 0, \dots, 0 \rangle.$$

Los marcajes satisfacen, si $\mu_0 = (\mu_1, \mu_2, \dots, \mu_j, \dots, \mu_q)$

entonces $\mu_{p_i} = (0, \dots, 0, \mu_i, 0, \dots, 0)$ de modo análogo definimos

F_{p_i} .

Dado que R_{p_i} sólo tiene una plaza $\dot{T}(R_{p_i}) \in \underline{RoCl}$. Considerando la definición de intersección de redes de Petri llegamos a:

$$R = \bigcap_{p \in P} R_p$$

podemos entonces afirmar $\dot{T}(R) = \bigcap_{p \in P} \dot{T}(R_p)$, condición que se traduce en $\dot{T}(R) \subset \wedge \underline{RoCl}$.

Vamos a mostrar finalmente que:

$$\mathcal{H}(\wedge \underline{RoCl}) \subset (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{Rat}, \wedge) D_1^*.$$

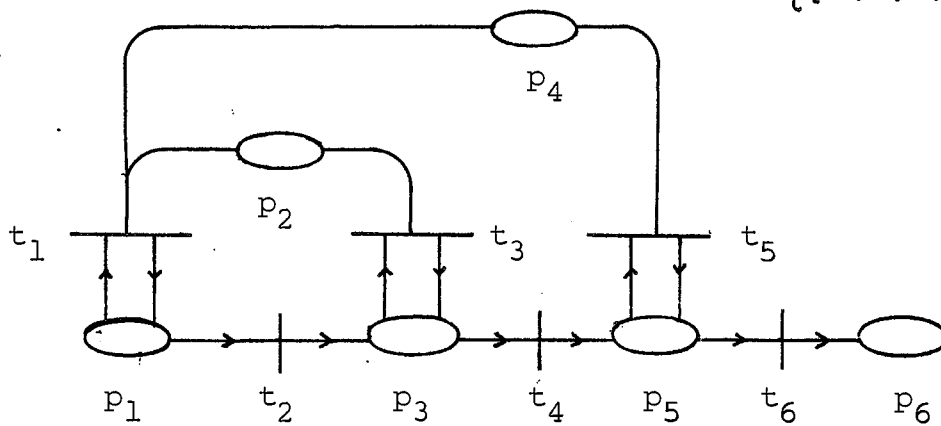
Esto es evidente dado que:

$$\mathcal{H}(\wedge \underline{RoCl}) = \mathcal{H}(\wedge (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{Rat}) D_1^*)$$

El procedimiento de reducir R a intersección de redes de Petri monoplazas, nos permite hallar de modo formal, en algunos casos, el lenguaje engendrado por R, como vemos en el ejemplo siguiente:

Ejemplo 1.6

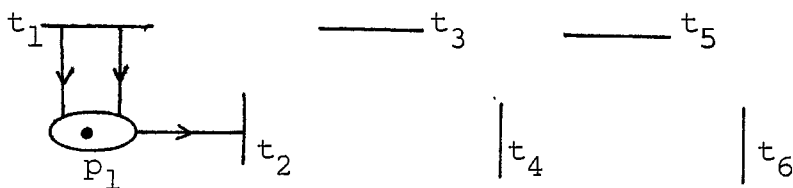
$$\mu_0 = [1, 0, 0, 0, 0, 0,] \\ F = \{ [0, 0, 0, 0, 0, 1] \}$$



$$\sigma = \langle t_1 \rightarrow a; t_2 \rightarrow e; t_3 \rightarrow b; t_4 \rightarrow e; t_5 \rightarrow c; t_6 \rightarrow e \rangle$$

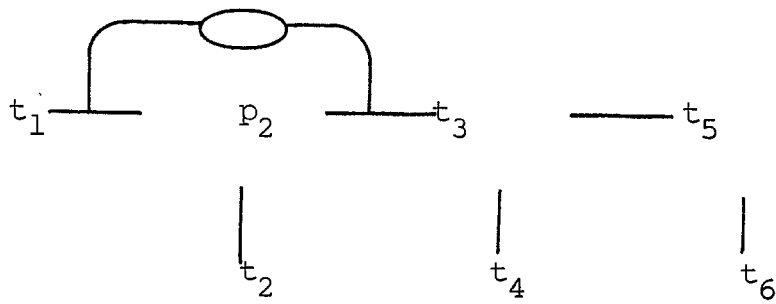
Utilizando el proceso de reducción a una plaza, obtenemos las seis redes elementales siguientes:

(1) Red Rp₁:



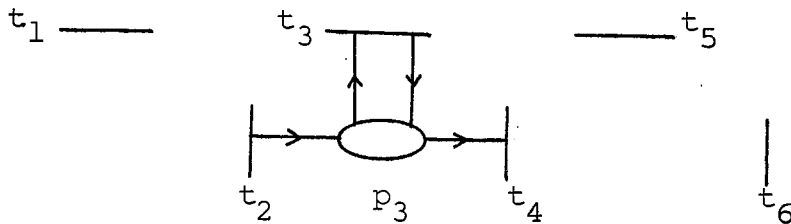
$$T(Rp_1) = t_1^* t_2 \sqcup \{ t_3, t_4, t_5 \}^*$$

(2) Red Rp₂ :



$$T(Rp_2) = \langle s \rightarrow t_1 s t_2 s + \epsilon \rangle \sqcup (t_2, t_4, t_5, t_6)^*$$

(3) Red Rp₃:

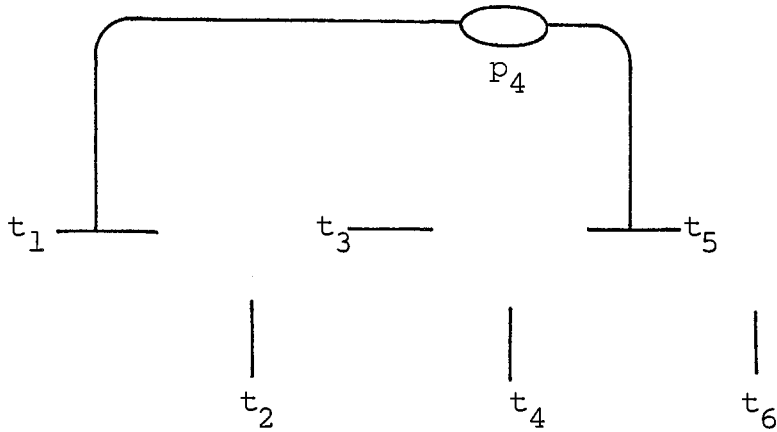


Notemos que podemos disparar t_3 siempre que exista al menos una marca en p_3 .

Para solucionar este problema, vamos a considerar los factores de Dyck primeros, en los que introduciremos t_3 y elevaremos con t_2 y t_4 , a fin de exigir que exista una marca en p_3 . Formalmente:

$$T(Rp_3) = \{ t_2 \{ \langle s \rightarrow t_2 s t_4 s + \epsilon \rangle \sqcup t_3^* \} t_4 \}^* \sqcup \{ t_1, t_5, t_6 \}^*$$

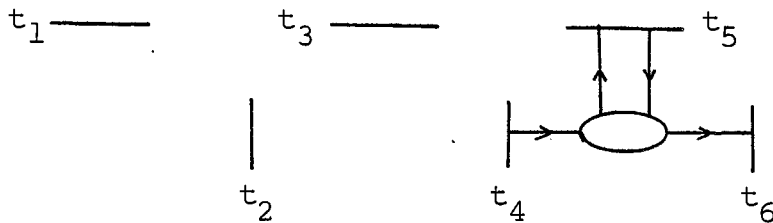
(4) Red Rp₄:



Obtenemos el mismo tipo de lenguaje que con p_2 :

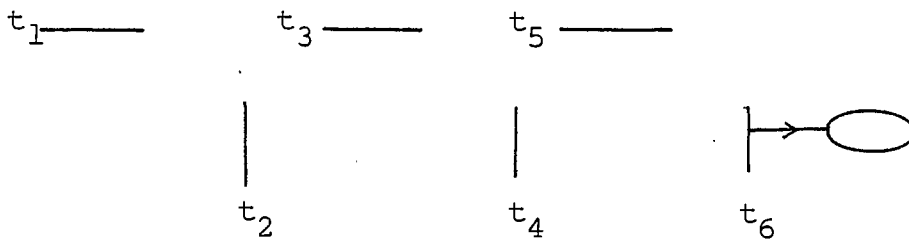
$$\mathbb{T}(Rp_4) = \langle S \rightarrow t_1 S t_5 S + \epsilon \rangle \sqcup (t_2, t_3, t_4, t_6)^*$$

(5) Red Rp_5 :



$$\mathbb{T}(Rp_5) = \{ t_4 \{ \langle S \rightarrow t_4 S t_6 S + \epsilon \rangle \sqcup t_5^* \} t_6 \}^* \sqcup \{ t_1, t_2, t_3 \}^*$$

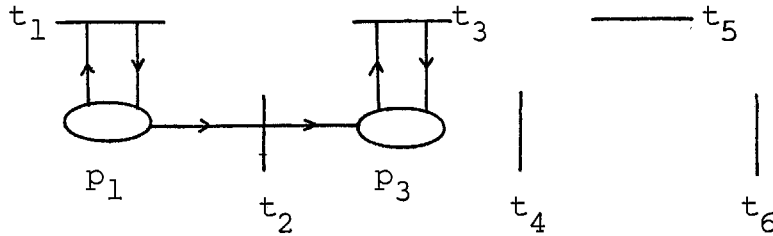
(6) Red Rp_6 :



Tomamos como marcaje final una sola marca en p_6 . Entonces

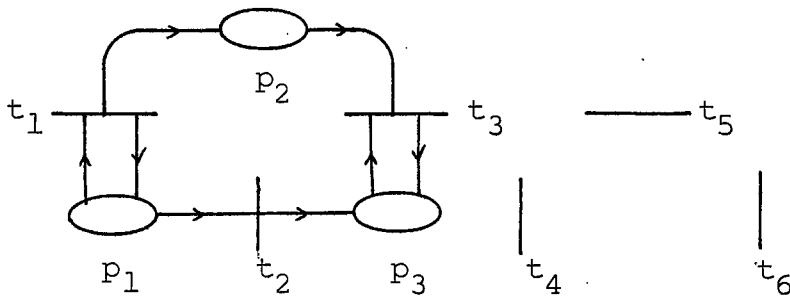
$$\overline{T}(Rp_6) = t_6$$

Realizamos ahora $Rp_1 \cap Rp_3$. Obtenemos:



$$\begin{aligned} \overline{T}(Rp_1 \cap Rp_3) &= \overline{T}(Rp_1) \cap \overline{T}(Rp_3) = \\ &= \{t_1^* t_2 \sqcup \{t_3, t_4, t_5\}^*\} \cap \{t_2 \{ \langle S \rightarrow t_2^S t_4^S + \epsilon \rangle \sqcup t_3^* \}^* t_4\}^* \sqcup \\ &\quad \{t_1, t_5, t_6\}^* \} = t_1^* t_2 t_3^* t_4 \sqcup \{t_5, t_6\}^* \end{aligned}$$

Realicemos ahora la intersección con Rp_2 , obtenemos:



El lenguaje engendrado es: $\overline{T}(Rp_1 \cap Rp_2 \cap Rp_3) =$

$$\begin{aligned} &= \{t_1^* t_2 t_3^* \sqcup \{t_5, t_6\}^*\} \cap \{ \langle S \rightarrow t_1^S t_3^S + \epsilon \rangle \sqcup (t_2, t_4, t_5, t_6)^* \} \\ &= \{t_1^n, t_2 t_3^n \mid n \geq 0\} \sqcup \{t_5, t_6\}^* \end{aligned}$$

Añadiendo las redes Rp_5 y Rp_6 obtenemos $\{t_1^n t_2 t_3^n \mid n \geq 0\} t_4 t_5^* t_6$

Añadiendo Rp_4 , exigimos que haya tantas transiciones t_1 como t_5 con lo que llegamos a: $\{t_1^n t_2 t_3^n t_4 t_5^n t_6 \mid n \geq 0\}$. Utilizando el etiquetaje σ obtenemos $\{a^n e b^n e c^n e \mid n \geq 1\}$

El último teorema admite como corolario.

Corolario 1.2 Los lenguajes en \mathcal{ESS} tienen índice inicial polinomial.

Un ejemplo interesante de lenguajes \mathcal{ESS} , lo constituyen los lenguajes de Szilard [14]. Dada una gramática G , llamamos lenguaje de Szilard $Sz(G)$ al lenguaje de etiquetas de derivaciones de las palabras de G . Cuando G es algebraica $Sz(G)$ pertenece a \mathcal{ESS} .

Damos ahora la construcción debida a Crespi-Regtrizzi y Mandrioli, que permite asociar a una gramática algebraica G una red R_G .

Definición 1.10 Sea $G = \langle X, V, P, v_1 \rangle$ una gramática algebraica en que hemos numerado las reglas de P , $t_i : v_j \rightarrow w_i$, $1 \leq i \leq r$, en que $r = \text{Card } P$. Asociamos a G la red $R_G = \langle P, T, I, O, \mu_0, F \rangle$ en que $P = V$, las transiciones son $T = \{ t_i \mid t_i \text{ es etiqueta de una regla de } G \}$. La función input está dada por:

$$I(t_i) = \langle 0, \dots, 0, 1v_j, 0, \dots, 0 \rangle$$

la función output de t_i está dada por:

$$O(t_i) = \langle n_1^i v_1, n_2^i v_2, \dots, n_q^i v_q \rangle \quad \text{en que}$$

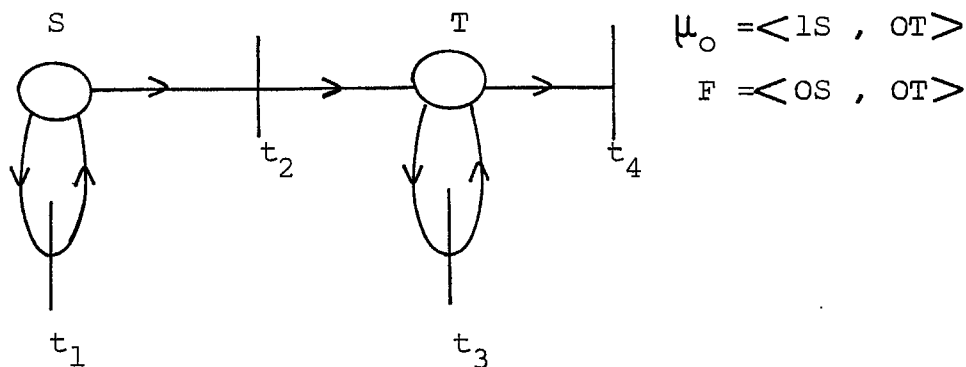
$$|w_i|_{v_j} = n_j^i \quad \text{para } 1 \leq j \leq q$$

Damos seguidamente un ejemplo de tal construcción.

Ejemplo 1.7 Consideremos la gramática

$$G = \langle t_1: S \rightarrow aSb ; t_2: S \rightarrow T ; t_3: T \rightarrow aT ; t_4: T \rightarrow \epsilon ; \text{axioma } S \rangle$$

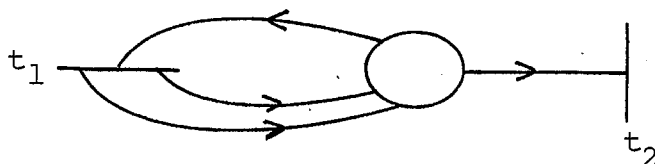
la red R_G asociada es:



Evidentemente $Sz(G) = T(R_G) = t_1^* t_2 t_3^* t_4$

Ejemplo 1.8 Trataremos el lenguaje de Dyck a una letra:

$D_1^* = \langle t_1 : S \rightarrow xSxS ; t_2 : S \rightarrow \epsilon ; S \text{ axioma} \rangle$. La red R_G es:



Tenemos:

$$Sz(G) = \{ w \in \{t_1, t_2\}^* \mid |w|_{t_1} + 1 = |w|_{t_2} \text{ y para todo } w' \leq w \text{ tenemos } |w'|_{t_1} \geq |w'|_{t_2} \}$$

Si definimos $\|w\| = |w|_{t_1} - |w|_{t_2}$ podemos reescribir $Sz(G)$ como

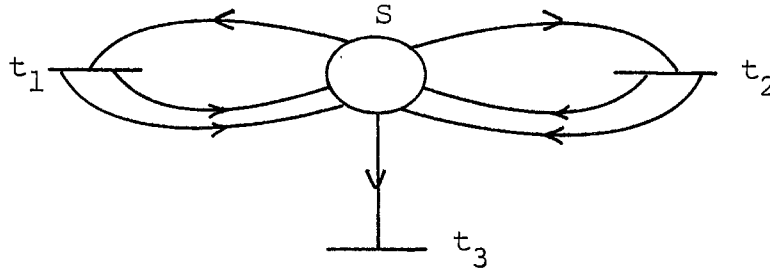
$$Sz(G) = \{ w \in \{t_1, t_2\}^* \mid \|w\| = -1 \text{ y para todo } w' \leq w \text{ tenemos } \|w'\| \geq 0 \}$$

Dicho lenguaje es conocido en la literatura con el nombre de lenguaje de Lukasiewicz, es algebraico, se cumple:

$$Sz(G) = \langle S \rightarrow t_1 SS + t_2 \rangle$$

Ejemplo 1.9 Tratamos ahora como ejemplo el lenguaje de Dyck a dos letras:

$$D_2^* = \langle t_1: S \rightarrow xSxS ; t_2: S \rightarrow ySyS ; t_3: S \rightarrow \epsilon ; \text{axioma } S \rangle$$



Tenemos:

$$Sz(G) = \{ w \in \{t_1, t_2, t_3\}^* \mid |w|_{t_1} + |w|_{t_2} + 1 = |w|_{t_3} \text{ y para todo } w' \neq w \text{ tenemos } |w'|_{t_1} + |w'|_{t_2} \geq |w'|_{t_3} \}$$

Dicho lenguaje es algebraico y está engendrado por:

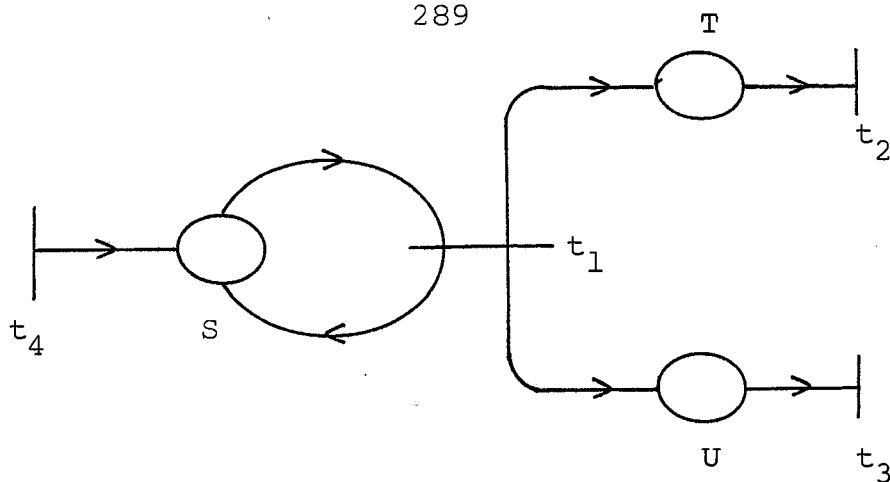
$$Sz(G) = \langle S \rightarrow t_1 SS + t_2 SS + \epsilon \rangle$$

Existen sin embargo lenguajes algebraicos cuyos lenguajes de Szilard no son algebraicos; damos seguidamente un ejemplo:

Ejemplo 1.10 Consideremos la gramática:

$$G = \langle t_1: S \rightarrow aSbT dU ; t_2: T \rightarrow c ; t_3: U \rightarrow d ; t_4: S \rightarrow g ; \text{axioma } S \rangle$$

Tiene como red de Petri asociada:



Tenemos que:

$$\text{Sz}(G) = \{ w \in \{t_1, t_2, t_3\}^* t_4 \{t_2, t_3\}^* \mid |w|_{t_1} = |w|_{t_2} = |w|_{t_3} ;$$

$$w' \not\leq w \text{ tiene } |w'|_{t_2} \leq |w'|_{t_1} ; |w'|_{t_3} \leq |w'|_{t_1} \}$$

Consideremos el morfismo:

$$\psi : \langle t_1 \rightarrow x ; t_2 \rightarrow \bar{x} ; t_4 \rightarrow \varepsilon ; t_3 \rightarrow \varepsilon \rangle$$

$$\chi : \langle t_1 \rightarrow x ; t_2 \rightarrow \varepsilon ; t_3 \rightarrow \bar{x} ; t_4 \rightarrow \varepsilon \rangle$$

$$\text{Sz}(G) = \chi^{-1}(D_1^*) \cap \psi^{-1}(D_1^*) \cap \{t_1, t_2, t_3\}^* t_4 \{t_2, t_3\}^*$$

Respecto a la estructura de los lenguajes de Szilard de los lenguajes algebraicos, tenemos:

Lema [14] 1.21 El lenguaje de Szilard de una gramática algebraica pertenece a $\bigwedge \text{Rocl}$.

Prueba Evidente, observando que los marcajes que aparecen en R_G modalizan las variables que aparecen en cada etapa de la derivación.

Dicho lema admite como corolario:

Corolario 1.3 Los lenguajes de Szilard de los lenguajes algebraicos, tienen índice inicial polinomial.

De modo análogo a $\mathcal{E}SS$ existen también, teoremas de caracterización de Contador(ctr) y Contador(pol), en términos de Rocl. A este fin notemos:

$$\mathcal{H}^{pol}(\mathcal{L}) = \{ \varphi(L) \mid L \in \mathcal{L} \text{ y existe } k \text{ y } p \text{ tales que para todo } w \in L \text{ se cumple } k \mid |\varphi(w)|^p \geq |w| \}$$

Se cumplen las caracterizaciones siguientes:

Teorema [34,35] 1.6

$$\underline{\text{Contador(ctr)}} = \mathcal{H}(\wedge \underline{\text{Fcl}}); \underline{\text{Contador(pol)}} = \mathcal{H}^{pol}(\wedge \underline{\text{Fcl}}) \text{ en que } \underline{\text{Fcl}} = (\mathcal{H}, \mathcal{H}^{-1}, \wedge \underline{\text{Rat}}, \cup, \dots, +) D_1^*$$

Considerando el comportamiento del índice inicial con respecto a las distintas operaciones, llegamos a:

Lema 1.22 Las familias Contador(ctr) y Contador(pol) están contenidas en Pol μ .

IV.2. Lenguajes exponenciales y generadores fieles.

En esta corta sección, vamos a analizar la relación que existe entre los lenguajes exponenciales y los generadores fieles de varias familias bien conocidas de lenguajes.

Para ello, comenzaremos por ver que una transducción racional fiel, no puede transformar un lenguaje polinomial en uno exponencial.

Seguidamente recordaremos de modo rápido, la definición de las familias Alg, Lin, Post, Single-Reset, Q, BNP, Multi-Reset en términos de autómatas. Daremos sus caracterizaciones en términos de propiedades de cierre, cosa que nos conduce a definir sus respectivas familias de generadores fieles. Mostraremos, finalmente, que dichos generadores tienen índice inicial exponencial.

Recordemos rápidamente la definición de lenguajes exponenciales.

$$\overline{\text{Exp}} \mu = \{ L \mid \exists \alpha \in \mathbb{R}_+ \text{ con } \mu_L(n) = \Omega(2^{\alpha n}) \}$$

El lema siguiente expresa el hecho que una transducción fiel no puede transformar un índice inicial polinomial en uno exponencial.

Lema 2.1 Sea L' un lenguaje de índice inicial exponencial y una transducción fiel, si $L' = \tau(L)$ entonces L es un lenguaje de índice inicial exponencial.

Prueba Si τ es una transducción fiel, existe k_1 y k_2 que satisfacen $\mu_{L'}(n) \leq k_1(n+1) \mu_L(k_2(n+1)+1)$. Se cumple:

$$\mu_L(k_2(n+1)+1) = \Omega\left(\frac{1}{n} \mu_{L_1}(n)\right).$$

Dado que $\mu_{L_1}(n) = \Omega(2^{\alpha n})$ y $n = O(2^{\frac{\alpha}{2} n})$ tenemos

$$\mu_L(k_2(n+1)+1) = \Omega(2^{\frac{\alpha}{2} n}).$$

Vamos a recordar rápidamente la definición de varias familias de lenguajes [11] en términos de los autómatas que las reconocen.

Definición 2.1

- (1) Llamamos Alg a la familia de los lenguajes algebraicos. Dichos lenguajes son aceptados por un autómata a pila que trabaja en tiempo real.
- (2) Llamamos Lin a la familia de los lenguajes lineales. Dichos lenguajes son aceptados por un autómata a pila, en tiempo real con un solo pico.
- (3) Llamamos Post a la familia de los lenguajes de Post. Dichos lenguajes son aceptados por un autómata a pila en tiempo real.
- (4) Llamamos Single-Reset a los lenguajes aceptados por un autómata a pila con una sola reinicialización.
- (5) Llamamos Q a la clase de lenguajes aceptados por una máquina de Turing no determinista, con varias bandas en tiempo real.
- (6) Llamamos BNP a la clase de lenguajes aceptados por una máquina de Turing no determinista, con varias bandas, con un número acotado de vueltas atrás, en tiempo real.
- (7) Llamamos Multi-Reset a la clase de lenguajes reconocidos en

tiempo real por una máquina no determinista, con un número finito de bandas de reinicialización.

Vamos a recordar seguidamente el concepto de generador fiel.

Definición 2.2 Sea \mathcal{L} una familia de lenguajes. Diremos que

$L \in \mathcal{L}$ es un generador fiel si:

$$\mathcal{L} = \{ \tau(L) \mid \tau \text{ es una transducción fiel} \}$$

En este caso diremos que \mathcal{L} es el cono fiel engendrado por L

y notaremos $\mathcal{L} = \mathcal{L}^f(L)$.

Llamaremos $\text{Gen}^f(\mathcal{L})$ al conjunto de generadores fieles de \mathcal{L} .

Vamos seguidamente a caracterizar las anteriores familias en términos de generadores fieles, o en términos de operaciones a partir de familias más elementales. Tenemos:

Teorema 2.1 [11] Las familias precedentes admiten la caracterización:

- (1) $\underline{\text{Alg}} = \mathcal{L}^f(D_2)$
- (2) $\underline{\text{Lin}} = \mathcal{L}^f(\underline{\text{PAL}})$
- (3) $\underline{\text{Post}} = \mathcal{L}^f(\text{FIFO}_2)$
- (4) $\underline{\text{Single-Reset}} = \mathcal{L}^f(\text{COPY}_2)$
- (5) $\underline{\text{Q}} = \mathcal{H}(\wedge \underline{\text{Alg}})$
- (6) $\underline{\text{BNP}} = \mathcal{H}(\wedge \underline{\text{Lin}})$
- (7) $\underline{\text{Multi-Reset}} = \mathcal{H}(\wedge \underline{\text{Single-Reset}})$

El teorema siguiente expresa que si una familia contiene un lenguaje exponencial, sus generadores deben ser exponenciales.

Teorema 2.2 Sea \mathcal{L} una familia que satisface $\mathcal{L} \cap \overline{\text{Exp}}_\mu \neq \emptyset$ y $\text{Gen}^f(\mathcal{L}) \neq \emptyset$, entonces se cumple $\text{Gen}^f(\mathcal{L}) \subset \overline{\text{Exp}}$

Prueba Sea L un lenguaje exponencial de \mathcal{L} . Sea G un generador fiel de \mathcal{L} . Sabemos que existe τ fiel con $L = \tau(G)$, el lema precedente nos permite concluir que G es exponencial. ■

Dicho teorema admite como consecuencia.

Corolario 2.1 Todo generador fiel de las familias Alg, Lin, Post, y Single-Reset tiene índice inicial exponencial.

Prueba Sabemos que D_2^* , PAL, FIFO₂, COPY₂, son lenguajes de índice inicial exponencial. El teorema precedente, nos permite obtener el resultado. ■

Cuando consideramos la clausura por intersección de los conos precedentes, obtenemos los mismos resultados respecto a los nuevos generadores obtenidos.

CAPITULO V. LENGUAJES DE ALTURA DE PILA SUBLINEAL.

V.1 Índice de pila, lenguajes p-sublineales.

V.1.1 Primeras definiciones y ejemplos.

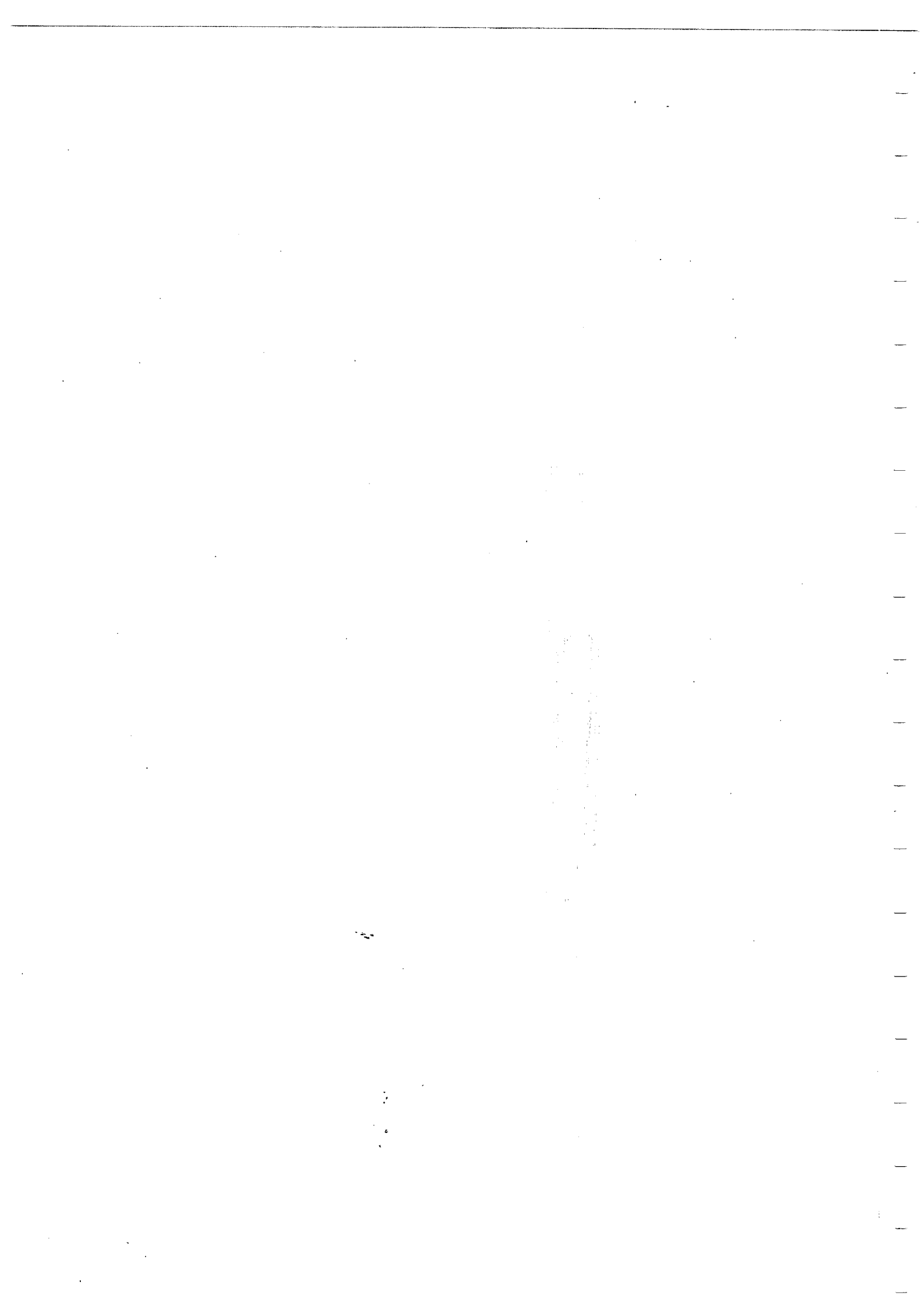
V.1.2 Lenguajes $p\text{-}\overline{\text{Root}(q)}$ y $p\text{-}\overline{\text{Log}}$.

V.2. Operaciones entre lenguajes y cadenas de
full-A.F.L.

V.2.1 Operaciones entre lenguajes.

V.2.2 Lemas de iteración y límites inferiores.

V.2.3 Cadena de full-A.F.L.



En este capítulo, estudiamos los lenguajes algebraicos que admiten una altura de pila que es una función sublineal de la longitud de la palabra.

En una primera sección, damos las primeras definiciones y ejemplos. En la segunda sección, vemos que con este método podemos definir una jerarquía infinita de full-A.F.L., estrictamente incluidos los unos en los otros.

V.1 Índice de pila, lenguajes p-sublineales.

Esta sección se halla dividida en dos subsecciones. En la primera subsección, se da la definición de una nueva medida de complejidad, la p-medida, para lenguajes algebraicos. La p-medida es una función de complejidad uniforme, que se basa en la altura de pila de un autómata que reconoce un lenguaje dado. Sea L un lenguaje algebraico, notaremos p_L su p-medida.

Definimos, seguidamente, la familia de los lenguajes p-sublineales, formada por todos los lenguajes algebraicos que admiten un autómata a pila, en el que para cada palabra existe un cálculo válido, en que la altura máxima de la pila es una función sublineal de la longitud de la palabra.

En este primer apartado, damos varios ejemplos de lenguajes cuya p-medida es la raíz cuadrada; entre ellos encontramos el lenguaje de Godstine, con sus numerosas variaciones y el lenguaje de Patterson.

En la segunda subsección, estudiamos las familias $p\text{-}\overline{\text{Root}(q)}$ y $p\text{-}\overline{\text{Log}}$. La familia $p\text{-}\overline{\text{Root}(q)}$ contiene aquellos lenguajes cuya altu-

ra de pila está acotada por la raíz q -ésima.

La obtención de lenguajes $p\text{-}\overline{\text{Root}}(q)$, se basa en la generalización del lenguaje de Godstine. Para ello, comenzamos por definir un orden total en el conjunto acotado de q letras:

$$B_q = (q-1)^* \dots 1^* 0^* \setminus \varepsilon$$

dicho orden total nos da una función sucesor suc_p , que nos permite generalizar el lenguaje de Godstine. Dado que el número de elementos de B_q , de longitud n , es del orden n^{q-1} y el lenguaje construido se basa en listar sucesivamente las distintas palabras de B_q , obtenemos el resultado.

Notemos, en particular, que el lenguaje así obtenido es lineal y a contador con un solo pico. Pertenece, entonces, a la intersección de Rocl y Lin.

Existe otro método para obtener lenguajes en $p\text{-}\overline{\text{Root}}(q)$, que consiste en implicar distintos lenguajes de Godstine. La definición de tales lenguajes se debe a Autebert, Beauquier, Boasson y Lattenx. Trataremos, finalmente, un lenguaje $p\text{-}\overline{\text{Log}}$. En este caso la función sucesor, es la dada por los números binarios.

V.1.1 Primeras definiciones y ejemplos

Queremos estudiar los lenguajes algebraicos con respecto a la cantidad de memoria que necesita un autómata a pila (p.d.a.), para reconocerlos con pila vacía.

Consideremos un p.d.a. \mathcal{A} . Consideremos w tal que $w \in L(\mathcal{A})$, notaremos $\mathcal{C}(w)$ el conjunto de cálculos válidos para w . Sea $c(w) \in \mathcal{C}(w)$, notaremos $p_c(w)$ la altura máxima alcanzada por la

pila durante el cálculo c de w . Notaremos finalmente:

$$p(w) = \min \{ p_c(w) \mid c(w) \in C(w) \}$$

El valor $p(w)$ mide el mínimo de pila necesario para calcular w con el p.d.a. a .

Dado un p.d.a. con X como alfabeto de entrada, llamaremos índice de pila a la función:

$$p_a(n) = \max \{ p(w) \mid w \in L(a) \cap nX^* \}$$

Dicha función mide la cantidad de pila necesaria para reconocer una palabra cualquiera de longitud como máximo n .

Definición 1.1 Diremos que un lenguaje algebraico L , tiene índice de pila $p_L(n)$, si existe un autómata a pila a que lo reconoce, tal que $p_L(n) = p_a(n)$.

En el lema siguiente vamos a dar las propiedades más básicas de la función $p_L(n)$.

Lema 1.1 Sea L un lenguaje algebraico, tenemos:

- (1) $p_L(n) \leq p_L(n+1)$
- (2) $p_L(n) = O(1)$ ssi L es racional.
- (3) $p_L(n) = O(n)$

Prueba

(1) Evidentemente la función p_L es creciente por construcción.

(2) Si la cantidad de memoria está limitada por una constante, el número de configuraciones de pila que pueden aparecer es finito, y en consecuencia el p.d.a. es una máquina de estados finitos.

(3) Consideremos una gramática bajo forma normal de Greibach, que genera L . El autómata a pila asociado es sin estados ni ϵ -transiciones, con lo que la altura de la pila está limitada por una fun-

ción lineal de la longitud de la palabra.

El lema precedente nos conduce a definir dos tipos de lenguajes, según el crecimiento asintótico de la pila.

Recordemos que $f(n) = o(g(n))$ ssi $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$.

Definición 1.2

(1) Diremos que un lenguaje L es p-sublineal ssi $p_L(n) = o(n)$, llamaremos $p\text{-}\overline{\text{Sub}}$ la familia de los lenguajes sublineales, es decir:

$$p\text{-}\overline{\text{Sub}} = \{ L \mid p_L(n) = o(n) \}$$

Entre las funciones sublineales más conocidas tenemos las raíces y los logaritmos. Trataremos separadamente estos casos.

Comenzaremos por tratar las raíces cuadradas.

Lenguajes p-Root(2)

Definición 1.3 Diremos que un lenguaje algebraico L es un lenguaje $p\text{-}\overline{\text{Root}}(2)$ ssi existe un autómata a pila a que satisface:

- (1) $L = L(a)$
- (2) $p_a(n) = O(n^{\frac{1}{2}})$

La familia $p\text{-}\overline{\text{Root}}(2)$ contiene todos los lenguajes algebraicos para los que existe un autómata a pila, que acepta las palabras del lenguaje, en un cálculo cuya altura máxima de pila está limitada por la raíz cuadrada de la longitud de la palabra. Notemos que pedimos que exista un cálculo limitado por la raíz, no que todo cálculo esté limitado por la raíz.

En los párrafos siguientes, damos numerosos ejemplos de lenguajes en $p\text{-}\overline{\text{Root}}(2)$.

Ejemplo 1.1 Consideremos el lenguaje de Goldstine, definido por:

$$\text{Gold} = \left\{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_r} b \mid r \geq 1, \exists j, 1 \leq j \leq r, \text{ que } i_j \neq j \right\}$$

Esquematicemos el funcionamiento de un autómata a pila para Gold.

El verificar que la palabra leída pertenece a $(a^*b)^*$ es un trabajo de autómata finito, se realiza sin problema.

Si $w \in \text{Gold}$ debe existir un mal bloque, es decir, un grupo $a^{i_j} b$ con $i_j \neq j$.

Notemos:

$$m_{j-1} = a^{i_1} b a^{i_2} b \dots a^{i_{j-1}} b$$

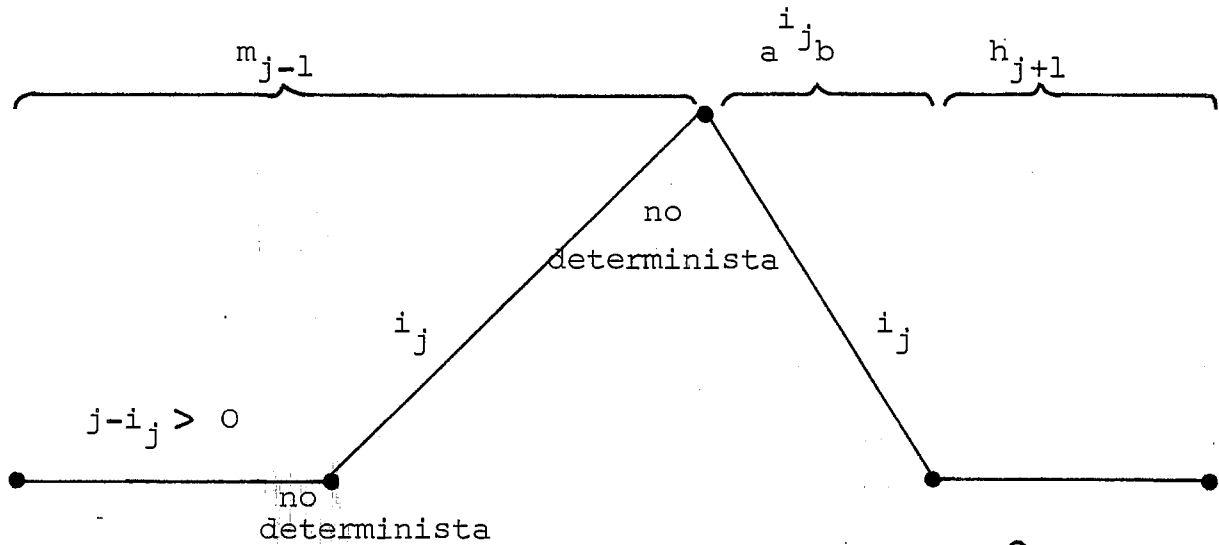
$$h_{j+1} = a^{i_{j+1}} b a^{i_{j+2}} b \dots a^{i_r} b$$

$$\text{tenemos que: } m = m_{j-1} a^{i_j} b h_{j+1}$$

Hay dos casos a tratar: $j > i_j$, $j < i_j$.

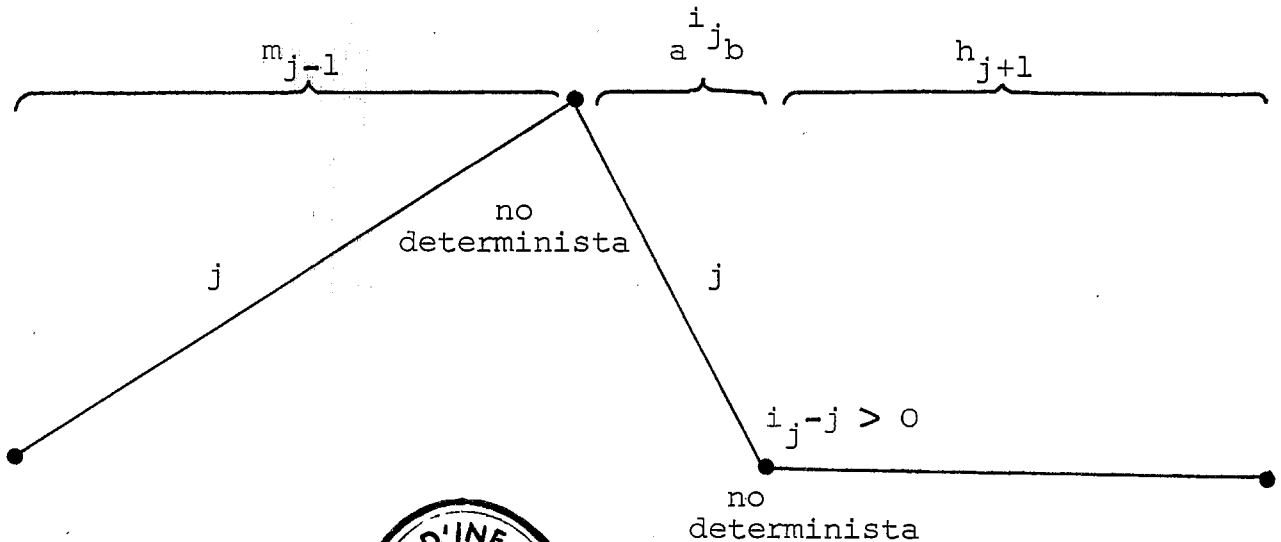
Caso $j > i_j$ En este caso el autómata a pila comenzará a leer un número de bloques superior o igual a 1, sin empilar nada. De manera no determinista, decretará que ha leído $j - i_j$ bloques, y a partir de este momento comenzará a empilar solamente las letras b, hasta

que de manera no determinista decreta que ha llegado al bloque i_j . En este momento testará que el número de letras a del bloque, es igual a la altura de pila. Caso de ser así, se pondrá en estado de aceptación y seguirá leyendo la palabra hasta terminarla. El cálculo precedente puede esquematizarse en:



Notemos que para este cálculo c_1 , la pila es $p_{c_1}(w) = \Theta(i_j)$.

Caso $j < i_j$ Con las mismas notaciones que en el párrafo precedente, el cálculo viene esquematizado por:



Para este cálculo c_2 , tenemos $p_{c_2}(w) = \Theta(j)$.

Tenemos, entonces, que si para una palabra $w \in (a^*b)^*$ existe un bloque $a^{i_j}b$ con $i_j \neq j$, existe un cálculo c que satisface:

$$p_c(w) = \Theta(\min\{i_j, j\})$$

La pila más pequeña para una palabra w dada, será:

$$p(w) = \Theta(\min\{\min\{i_j, j\} \mid a^{i_j}b \text{ con } i_j \neq j\})$$

Intuitivamente podemos decir que el autómata testa el "bloque más pequeño".

Notemos que toda palabra w de Gold, se factoriza de modo único en:

$$w = aba^2ba^3b \dots a^{j-1}ba^{i_j}bh_{j+1}$$

en que hemos puesto en evidencia el factor izquierdo más largo formado de buenos bloques. Podemos, entonces, afirmar que :

$$p_c(w) = O(\min(j-1, i_j))$$

Dado que $|w| > |aba^2b \dots a^{j-1}b| = \Theta(j^2)$, obtenemos que para cualquier palabra de Gold, existe un cálculo c que satisface:

$$p_c(w) = O(|w|^{\frac{1}{2}})$$

podemos, entonces, concluir que $p_{\text{Gold}}(n) = O(\sqrt{n})$.

Podemos preguntarnos si la raíz es el menor de los límites posibles para el autómata descrito. Para ello, vamos a dar una sucesión de palabras w_j en las que se cumple $p(w_j) = \Omega(j^2)$ y $|w_j| = \Theta(j^2)$.

Para obtener w_j , basta tomar en las definiciones precedentes

$h_{j+1} = \varepsilon$ y $i_j = j - 1$, tenemos:

$$w_j = aba^2ba^3b \dots a^jba^j b.$$

Podemos, entonces, afirmar que el autómata a pila descrito precedentemente, lo notaremos por \mathcal{A} , satisface:

$$p_a(n) = \Theta(n^{\frac{1}{2}})$$

Es, sin embargo, un problema abierto, saber si para Gold, existe algún autómata cuya altura de pila es asintóticamente más pequeño que la raíz.

Ejemplo 1.2 Podríamos estar tentados a creer que el autómata dado precedentemente, es el único que realiza la raíz. Esto es falso, como vemos seguidamente.

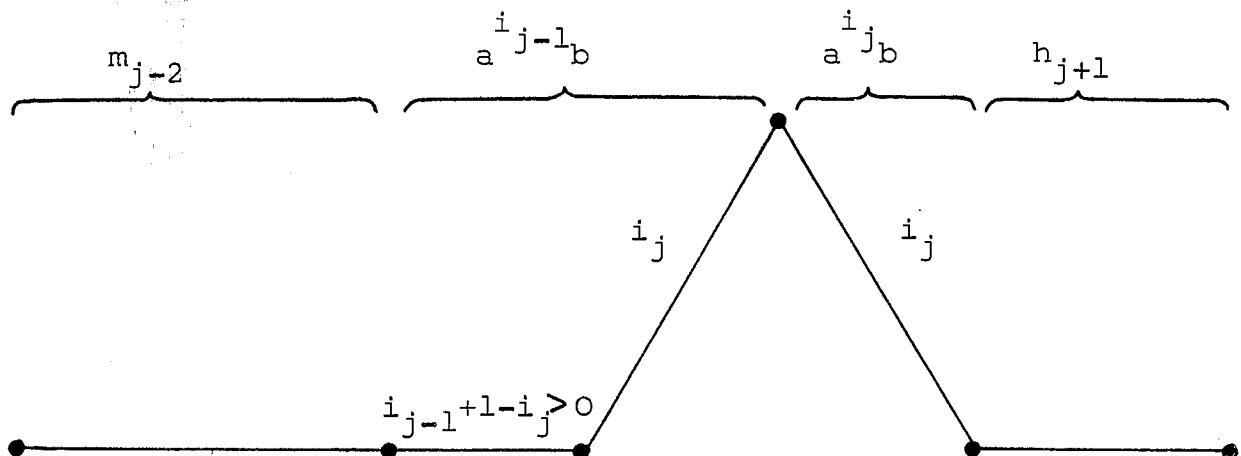
El lenguaje Gold, admite la escritura siguiente:

$$\text{Gold} = (\epsilon + aa^+)b(a^*b)^* \cup \{aba^{i_2}b \dots a^{i_{j-1}}ba^{i_j}b \dots a^{i_r}b \mid r \geq 1, \exists 2 \leq j \leq r, i_j \neq i_{j-1} + 1\}$$

Consideremos el p.d.a. que testa de manera no determinista si el número de letras a de dos bloques consecutivos $j-1, j$ satisfacen $i_j \neq i_{j-1} + 1$.

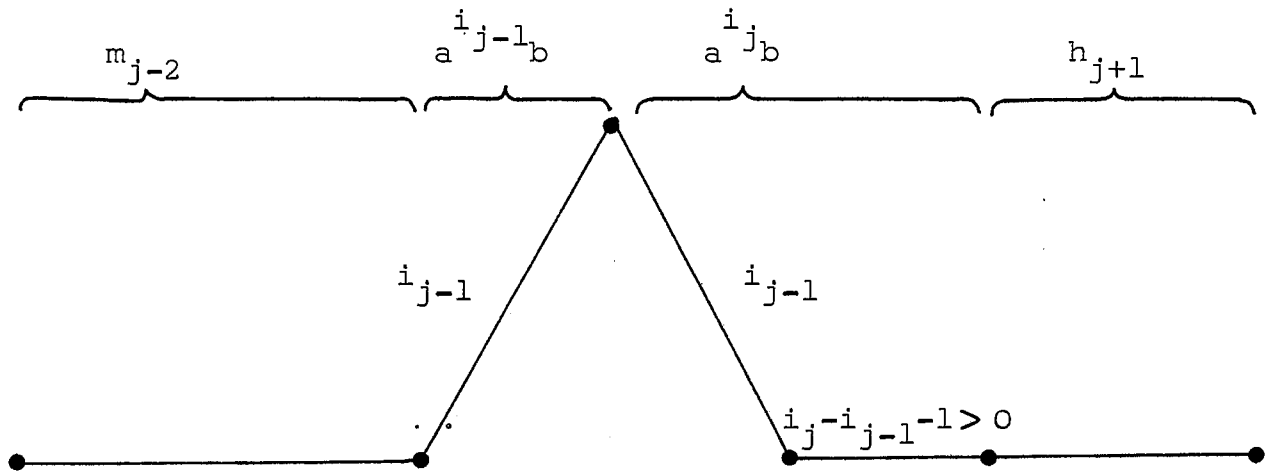
También, aquí, se dan dos casos:

Caso $i_j < i_{j-1} + 1$



Sea c dicho cálculo, tenemos $p_c(w) = \Theta(i_j)$

Caso $i_j > i_{j-1} + 1$



Sea c dicho cálculo, tenemos $p_c(w) = \Theta(i_{j-1})$

Si en una palabra w existen dos bloques consecutivos $a^{i_{j-1}b} a^{i_j b}$ con $i_{j-1} + 1 \neq i_j$, existe un cálculo c con

$$p_c(w) = \Theta(\min\{i_{j-1}, i_j\})$$

Como en el caso precedente, las palabras que nos dan pila máxima son:

$$w_j = aba^2ba^3b \dots a^jba^j b$$

con lo que el autómata descrito precedentemente es también de altura raíz. ■

Remarca 1.1 Hemos visto que el lenguaje de Goldstine puede reconocerse de dos modos distintos:

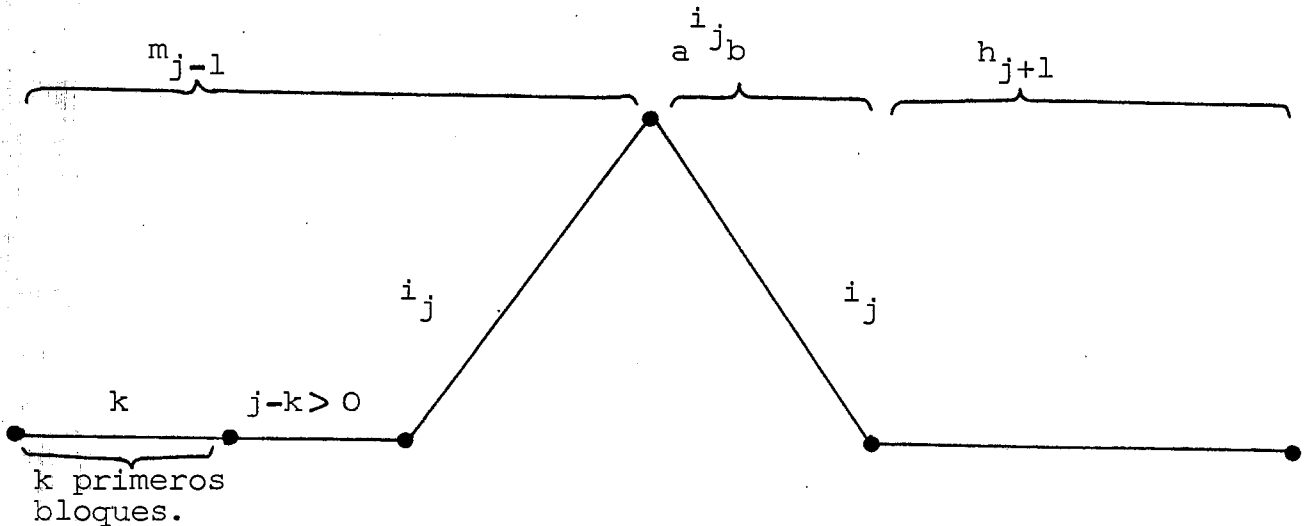
- (1) Testando que el número de bloque no coincide con el número de letras a del bloque (Empilando las b y depilando con las a).
- (2) Testando que el número de letras a en dos bloques sucesivos no se corresponden (Empilando las del primer bloque y depilando las del segundo).

Ejemplo 1.3 Sea k positivo, consideremos el lenguaje:

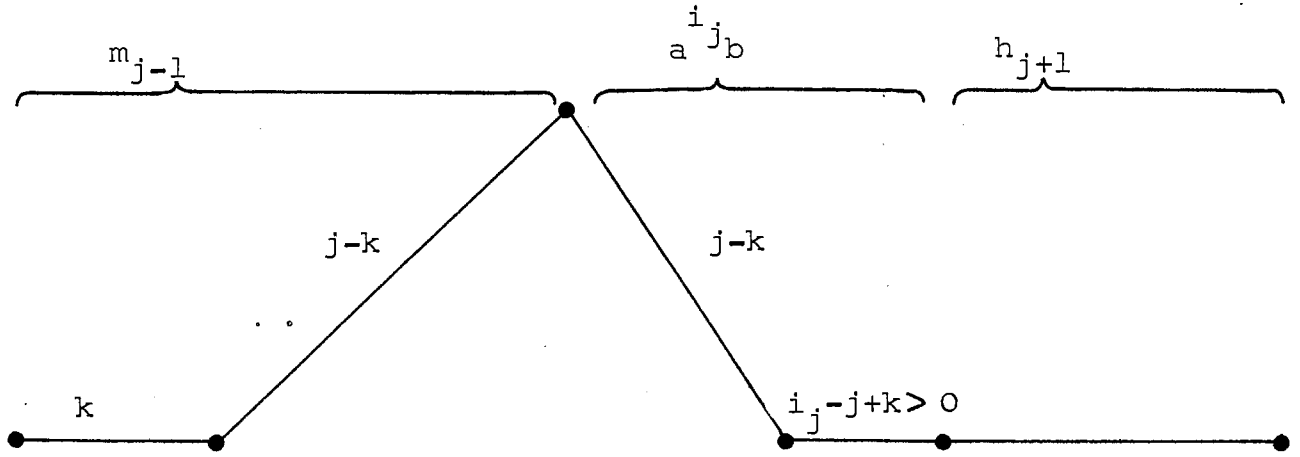
$$L_k = \{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_r} b \mid r \geq k \exists j, k \leq j \leq r \text{ que } i_j \notin \{j-k, \dots, j+k\} \}$$

Consideremos un autómata a pila que testa de manera no determinista la existencia de un mal bloque $a^{i_j} b$, es decir, de un bloque que satisface $i_j < j - k$, $i_j > j + k$. El autómata va a tratar cada caso separadamente.

Caso $i_j < j - k$



Caso $i_j > j - k$



Evidentemente $p_{L_k}(n) = O(n^{\frac{1}{2}})$

Ejemplo 1.4 El lenguaje de Goldstine es, prácticamente, el complementario de los factores izquierdos de la palabra infinita hacia la derecha

$$m = aba^2ba^3b \dots a^j b \dots$$

En vez de ello, podemos considerar el lenguaje dado por los complementarios de la palabra espejo de m , es decir:

$$\dots a^j b \dots a^3 b a^2 b a b = m$$

Tenemos entonces:

$$\hat{\text{Gold}} = \{ a^{i_r} b a^{i_{r-1}} b \dots a^{i_{r-j}} \dots a^{i_1} b \mid r > 0, \exists 0 \leq j \leq r-1 \text{ que } i_{r-j} \neq r-j \}$$

Si $w \in \hat{\text{Gold}}$, dicha palabra debe contener un mal bloque, es decir un bloque $a^{i_{r-j}} b$ con $i_{r-j} \neq r-j$.

$$\text{Notemos: } h_{j-1} = a^{i_r} b a^{i_{r-1}} b \dots a^{i_{r-j+1}} b$$

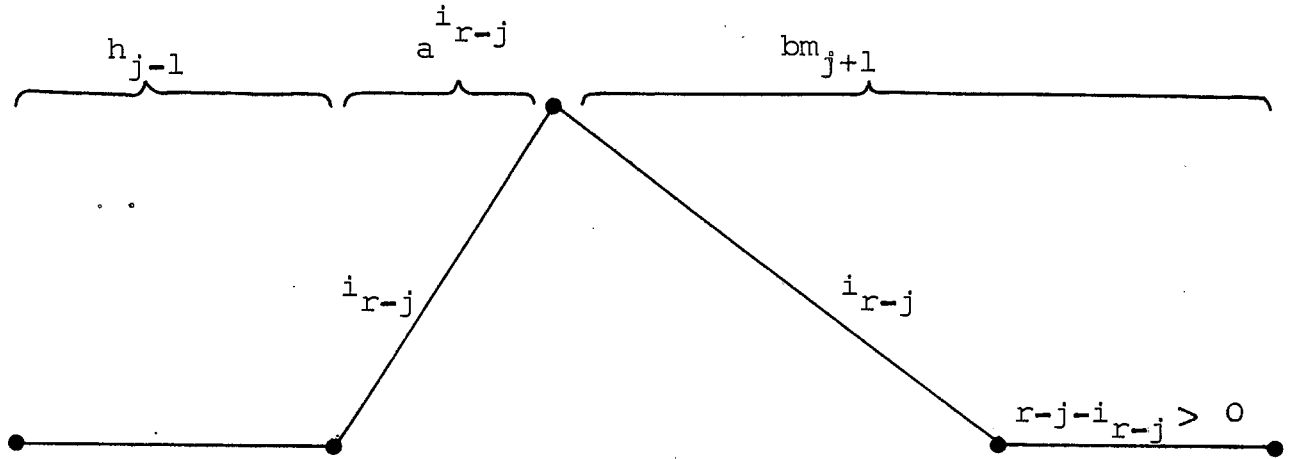
$$m_{j+1} = a^{i_{r-j-1}} b \dots a b$$

Hay dos casos a tratar $i_{r-j} < r-j$ y $i_{r-j} > r-j$

Caso $i_{r-j} < r-j$ El autómata a pila leerá h_{j-1} , sin modificar la pila. Detectará de manera no determinista que $a^{i_{r-j}} b$ es la mala transición y que $i_{r-j} < r-j$. En este caso empilará todas las letras a ; empila un total de i_{r-j} letras.

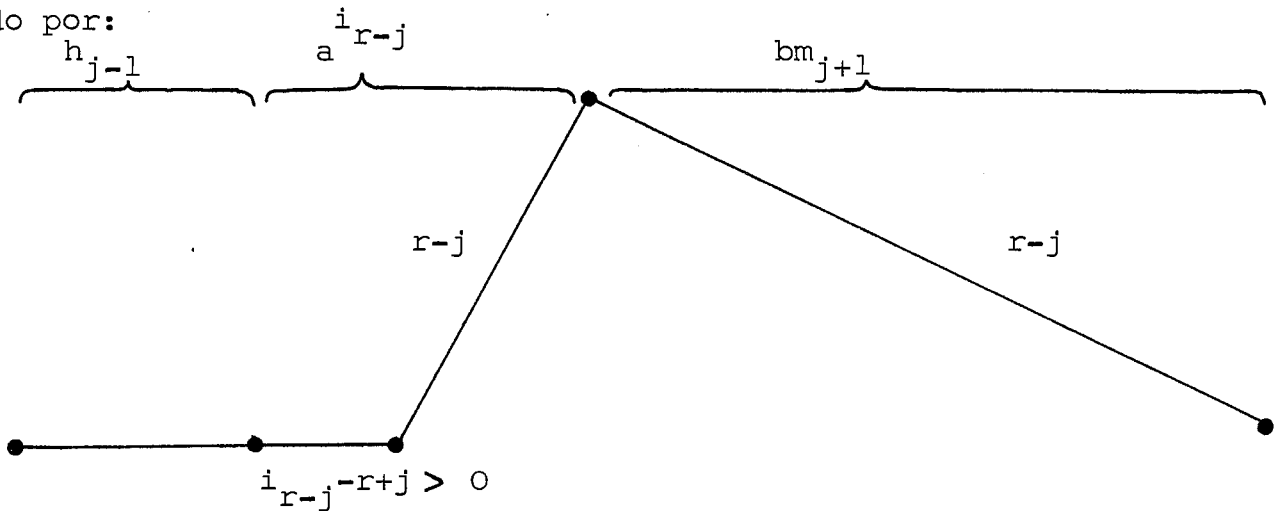
Tras esto comienza a leer bm_{j+1} , depilando cada vez que encuentra una letra b. El número total de letras b que se encuentran a la derecha de $a^{i_{r-j}}$ es $r - j$, contando la letra b del bloque $a^{i_{r-j}}b$.

Tras vaciar la pila, el autómata testará que aún le queda alguna b para leer. Este cálculo c puede esquematizarse en:



En este caso $p_c(w) = \Theta(i_{r-j})$

Caso $i_{r-j} > r - j$ En este caso el esquema del cálculo c viene dado por:



En este caso $p_c(w) = \Theta(r - j)$

Tenemos, entonces, que si en una palabra $w \in (a^*b)^*$, hay un bloque

$a^{i_{r-j}}$ con $i_{r-j} \neq r - j$, existe un cálculo c que satisface

$$p_c(w) = \Theta(\min(i_{r-j}, r - j))$$

La pila más pequeña para w será:

$$p(w) = \Theta(\min\{\min\{i_{r-j}, r - j\} \mid a^{i_{r-j}}b \text{ con } i_{r-j} \neq r - j\})$$

Toda palabra de $\hat{\text{Gold}}$ se factoriza de modo único

$$w = h_{j-1} a^{i_{r-j}} b a^{r-j-1} b a^{r-j-2} b \dots ab$$

en la que hemos puesto de manifiesto el factor derecho más largo formado de buenas transiciones. Se cumple que:

$$p_c(w) = O(\min(i_{r-j}, r - j))$$

Dado que w satisface

$$|w| \leq |a^{i_{r-j}} b a^{i_{r-j}-1} b a^{i_{r-j}-2} b \dots ab| = \Theta(j^2)$$

obtenemos para el autómata definido precedentemente, notado

$$p_a(w) = O\left(w^{\frac{1}{2}}\right)$$

con lo que $p_{\hat{\text{Gold}}}(n) = O\left(n^{\frac{1}{2}}\right)$

Si queremos demostrar que para el autómata \mathcal{A} el comportamiento asintótico es óptimo, basta tomar:

$$w_r = a^r b a^{r-1} b a^{r-2} b \dots ab$$

Ejemplo 1.5 Podemos considerar conjuntamente Gold y $\hat{\text{Gold}}$ en el lenguaje siguiente:

$$L = (a^*b^*c)^* \setminus \{ a^r c a^{r-1} b c a^{r-2} b^2 c \dots a^{r-j} b^j c \dots c b^r c \mid r \geq 2 \}$$

Sabemos que $w \in (a^*b^*c)^r$ es una palabra de L ssi contiene un mal bloque, es decir, un bloque $a^{u_{r-j}} b^{v_j} c$ que satisface:

$$\neg (u_{r-j} = r - j) \wedge (v_j = j) \text{ es decir} \\ (u_{r-j} \neq r - j) \vee (v_j \neq j).$$

Caso $u_{r-j} \neq r - j$ La palabra w se factoriza en:

$$h_{j-1} = a^{u_{r-j}} b^{v_0} c a^{u_{r-1}} b^{v_1} c a^{u_{r-2}} b^{v_2} c \dots a^{u_{r-j+1}} b^{j-1} c \\ m_{j+1} = a^{u_{r-j-1}} b^{v_{j+1}} c \dots c a^{u_0} b^{v_r} c$$

En este caso el autómata funciona de modo análogo al de Gold, ignorando las letras b , empilando las letras a de $a^{u_{r-j}}$ y depilando con las ocurrencias de b . Sea c un tal cálculo, se satisface:

$$p_c(w) = \Theta(\min(u_{r-j}, r - j))$$

Caso $v_j \neq j$ La palabra w se factoriza en:

$$\bar{m}_{j-1} = a^{u_{r-j}} b^{v_0} c a^{u_{r-1}} b^{v_1} c \dots a^{u_{r-j+1}} b^{j-1} c \\ \bar{h}_{j+1} = a^{u_{r-j-1}} b^{v_{j+1}} c \dots c a^{u_0} b^{v_r} c$$

En este caso el autómata ignora las ocurrencias de a , y trata las de b y c de un modo análogo al autómata esquematizado en Gold. Empila las letras c hasta llegar al bloque $a^{u_{r-j}} b^{v_j} c$, depilando con las ocurrencias de a en dicho bloque. Sea c dicho cálculo, tenemos:

$$p_c(w) = \Theta(\min(v_j, j))$$

Toda palabra $w \in (a^*b^*c)^r$ admite las dos factorizaciones siguientes.

Consideremos la factorización que hemos puesto en evidencia el factor derecho más largo de buenas transiciones entre los bloques de a .

$$w = h_{j-1} a^{u_{r-j}} b^{v_j} c \dots ca^3 b^{v_{r-3}} ca^2 b^{v_{r-2}} cab^{v_{r-1}} cb^{v_r} c$$

Consideremos la factorización en que hemos indicado el factor izquierdo más largo de buenas transiciones entre bloques de b.

$$w = a^{u_r} ca^{u_{r-1}} bca^{u_{r-2}} b^2 c \dots a^{u_{r-i}} b^{v_i} c h_{i+1}$$

Evidentemente:

$$p(w) = O(\min\{u_{r-j}, r-j, v_i, i\})$$

Dado que la palabra w satisface

$$|w| = O(\max\{(r-j)^2, i^2\}) = O(\max\{j^2, i^2\})$$

Podemos concluir que

$$p_a(w) = O(|w|^{\frac{1}{2}})$$

con lo que $p_L(n) = O(n^{\frac{1}{2}})$.

El límite de la raíz cuadrada es óptimo para el autómata definido precedentemente, para ello basta tomar:

$$w_j = a^j ca^{j-1} bca^{j-2} b^2 c \dots a^3 b^{j-3} ca^2 b^{j-2} cab^{j-1} cb^j cb^j c$$

Ejemplo 1.6 Analicemos el siguiente lenguaje debido a Mike Paterson [3]

$$\text{Pat} = (a^*b)^* \setminus \{(a^k b)^k \setminus k \geq 0\}$$

dicho lenguaje puede reescribirse como:

$$\text{Pat} = \{a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_k} b \mid k \geq 1, \exists 1 \leq j \leq k; i_j \neq k\}$$

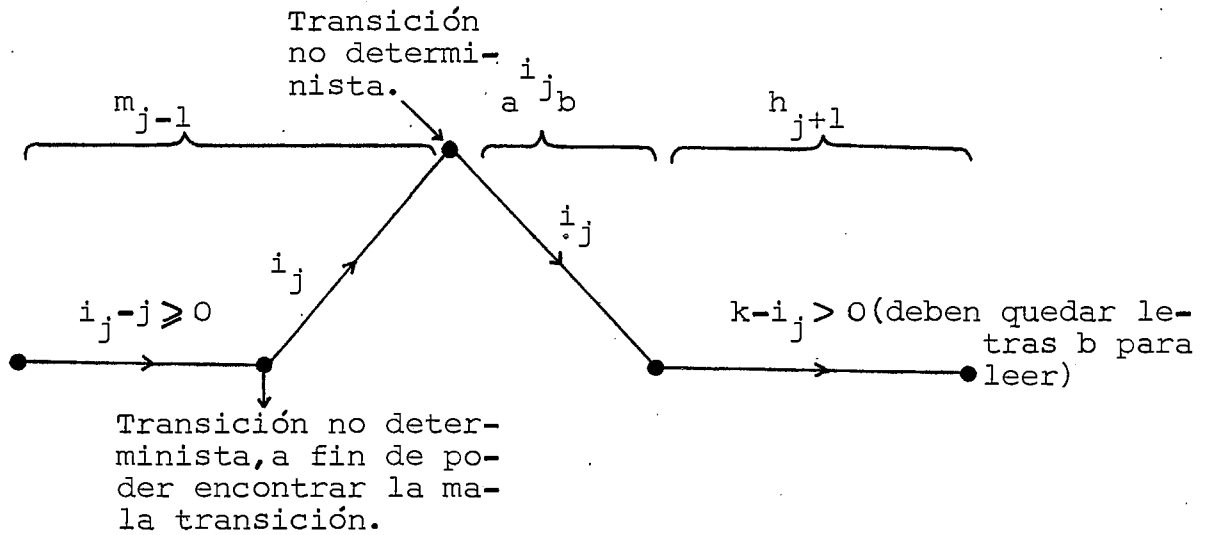
Sea $w = a^{i_1} b \dots a^{i_j} b \dots a^{i_k} b \in L$, existe $i_j \neq k$. Hay dos casos a tratar.

(a) Caso $i_j < k$ Elijamos el i_j mas pequeño posible.

Entonces $w = a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_k} b$ satisface $i_j \geq i_l, l = 1, \dots, k$ con lo que $|w| \geq i_j \cdot k \geq (i_j)^2$.

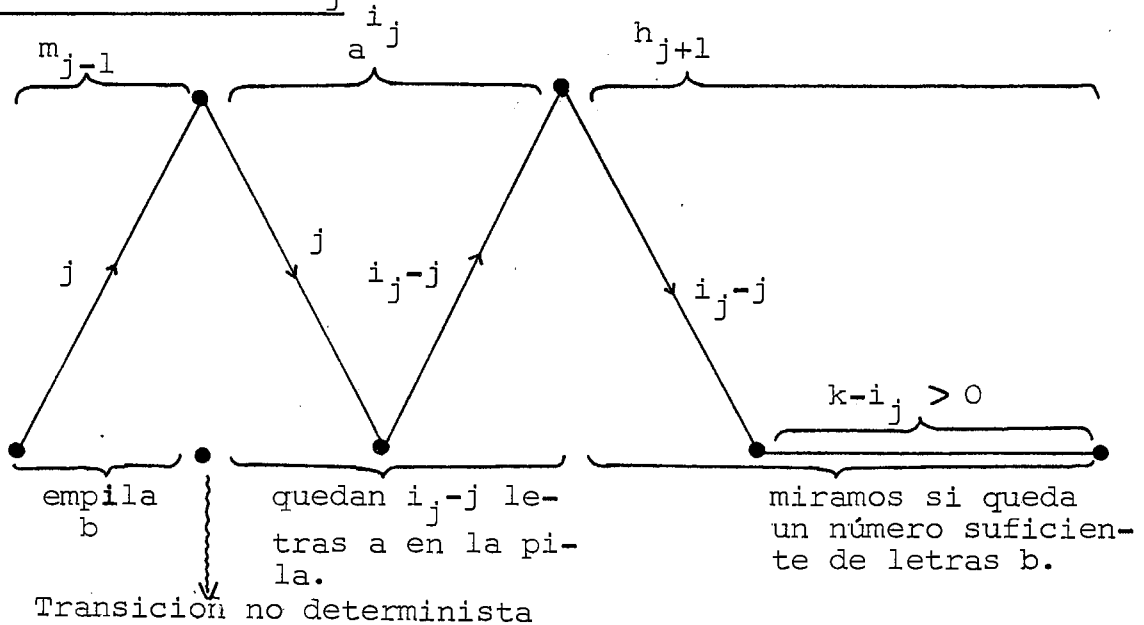
Separemos este caso en dos subcasos, segun $j > i_j$ ó $j < i_j$.

Primer subcaso $j \geq i_j$ En este caso el cálculo del autómata viene esquematizado en:



En este caso $p(w) = \Theta(i_j)$.

Segundo subcaso $j < i_j$ El cálculo viene esquematizado en:

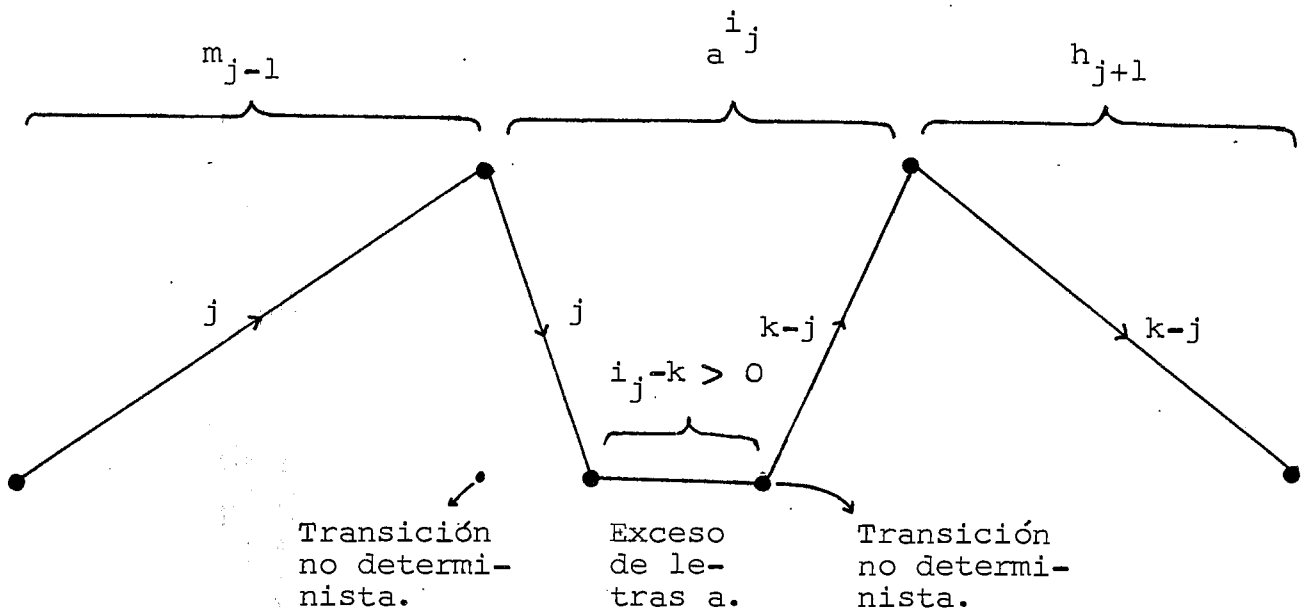


En este caso $p(w) = \max \{j, i_j - j\} = O(i_j)$

(b) Caso $i_j > k$ Buscamos el i_j más pequeño, entonces

$w = a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_k} b$ satisface $i_l > k, l = 1, \dots, k$ con lo que $|w| \geq i_j \cdot k \geq k^2$.

En este caso el cálculo viene esquematizado por:



En este caso $p(w) = \max \{j, j - k\} \leq k$

Tenemos entonces que $p_L(n) = O(\sqrt[n]{n})$

Vamos a generalizar los resultados precedentes a raíces de cualquier grado.

V.1.2. Lenguajes $p\text{-}\overline{\text{Root}}(q)$ y $p\text{-}\overline{\text{Log}}$.

Lenguajes $p\text{-}\overline{\text{Root}}(q)$ Vamos a estudiar la existencia de lenguajes no racionales cuya altura de pila está limitada por la raíz q -ésima de la longitud.

Definición 1.4 Diremos que un lenguaje L es $p\text{-}\overline{\text{Root}}(q)$, ssi existe un autómata a pila que lo reconoce con

$$p \ a^{(n)} = O(n^{\frac{1}{q}})$$

Los párrafos siguientes tienen por objeto mostrar lenguajes $p\text{-}\overline{\text{Root}}(q)$. Recordemos que el primer lenguaje que hemos encontrado en $p\text{-}\overline{\text{Root}}(2)$ es el lenguaje de Goldstine. Con pequeñas variaciones dicho lenguaje puede definirse fundamentalmente de dos maneras:

(a) Exigiendo una mala sucesión de bloques. Es decir:

$$\{ a^i b a^{i_2} b \dots a^i b a^{i_{j+1}} b \dots a^i b \mid \exists j, i_{j+1} \neq i_j \}$$

(b) Exigiendo la aparición de un bloque que no coincide con su rango. Es decir:

$$\{ a^i b a^{i_2} b \dots a^i b \dots a^i b \mid \exists j, i_j \neq j \}$$

Vamos a generalizar cada uno de estos dos métodos.

La generalización del primer método nos conduce a generalizar la noción de sucesor sobre un conjunto acotado. El segundo método nos conduce a realizar imbricaciones de lenguajes de Goldstine.

I. Generalización de la función sucesor

Recordemos que el lenguaje de Goldstine se construye a partir de la palabra infinita

$$m = aba^2ba^3b \dots a^j b \dots$$

Si definimos la función sucesor por $\text{suc}(a^p) = a^{p+1}$, podemos reescribir dicha palabra como

$$m = ab \text{suc}(a)b \text{suc}^2(a)b \dots \text{suc}^j(a)b \dots$$

El crecimiento de m con relación a la función sucesor viene expresado por:

$$m_j = ab \text{suc}(a)b \text{suc}^2(a)b \dots \text{suc}^j(a)b$$

$$|m_j| = \Theta(j^2)$$

$$|\text{suc}^j(a)| = \Theta(j)$$

con lo que $|m_j| = \Theta(|\text{suc}^j(a)|^2)$ es dicha igualdad la que nos permite demostrar que

$$\{ abwb \dots w_j b \dots w_r b \mid w_i \in a^*, \exists j \text{ con } \text{suc}(w_{j-1}) \neq w_j \}$$

es un lenguaje con una pila en raíz cuadrada.

A fin de generalizar el lenguaje precedente, debemos por lo tanto

(a) Definir una nueva palabra infinita, generalizando la función sucesor, y estudiar su crecimiento.

(b) Definir el lenguaje asociado a esta nueva palabra.

1. Construcción de suc_p

Vamos a generalizar la función sucesor tomando como base un len-

guaje acotado, sea:

$$B_{p+1} = p^*(p-1)^* \dots 1^* 0^* \setminus \epsilon, \quad p \geq 0$$

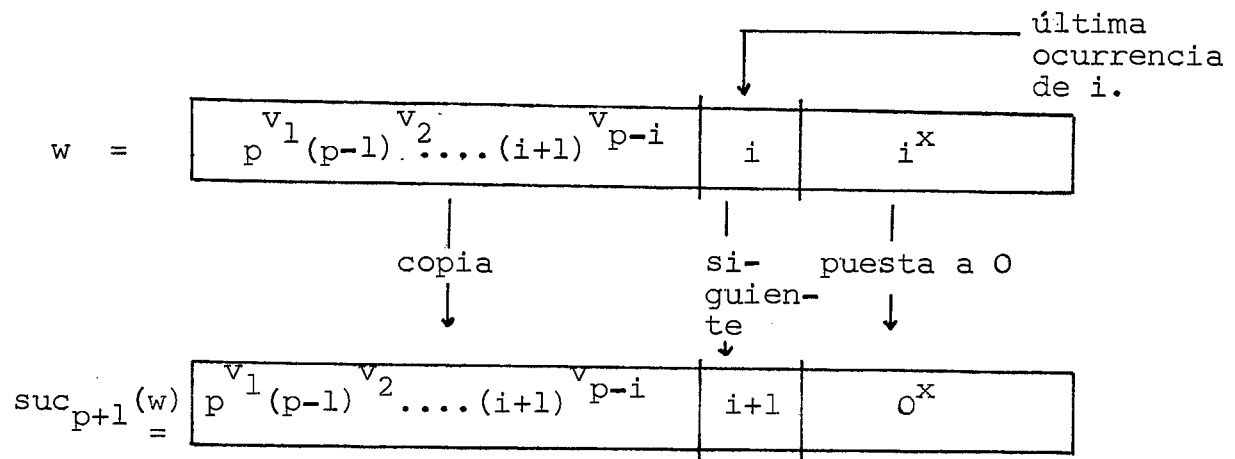
definamos sobre B_{p+1} la función sucesor por:

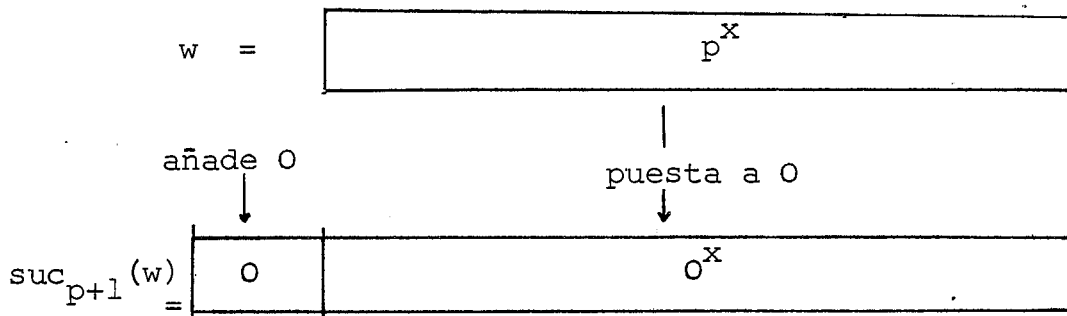
Definición 1.5 Consideremos la función $\text{suc}_{p+1}: B_{p+1} \longrightarrow B_{p+1}$

definida por:

- (1) Si $w \in B_{p+1}$ se factoriza en $w = w_{i+1} i^x$, $x > 0$, $0 \leq i < p$, $w_{i+1} \in p^*(p-1)^* \dots (i+1)^*$, entonces $\text{suc}_{p+1}(w) = w_{i+1} (i+1) 0^x$
- (2) Si $w \in B_{p+1}$ se factoriza en $w = p^x$, $x > 0$ entonces $\text{suc}_{p+1}(w) = 0^{x+1}$.

Mostramos, en la figura siguiente, como funciona la función suc_{p+1} .





A fin de estudiar el crecimiento de m_p con respecto a suc_p , vamos primeramente a contar cuantas palabras hay en B_p de una longitud dada.

Lema 1.2 Sea $B_p = (p-1)^* (p-2)^* \dots 0^*$, tenemos

$$\| \{w \mid w \in B_p \text{ y } |w| = n\} \| = \Theta(n^{p-1})$$

Prueba Sean u_r, u_{r-1}, \dots, u_1 fijos, pertenecientes a $\{0, 1, \dots, p-1\}$ con $u_r > u_{r-1} > \dots > u_1$. Sabemos que:

$$\| \{w \mid w \in u_r^+ u_{r-1}^+ \dots u_1^+, |w| = n\} \| = \binom{n-1}{r-1} = \Theta(n^{r-1})$$

dado que a cada palabra w podemos asociar la partición del entero n en r partes.

El número total de grupos de r letras distintas entre un total de p es:

$$\| \{(u_1, u_2, \dots, u_r) \mid u_1 > u_2 > \dots > u_r\} \| = \binom{p}{r}$$

Tenemos entonces que

$$\begin{aligned} & \| \{w \mid w \in B_p, \exists i_1, \dots, i_r \text{ con } w \in u_{i_1}^+ \dots u_{i_r}^+ \text{ y } |w| = n\} \| \\ &= \binom{p}{r} \binom{n-1}{r-1} = \Theta(n^{r-1}) \end{aligned}$$

Para hallar $\| \{w \mid w \in B_p \text{ y } |w| = n\} \|$ hay que hacer la suma a todos los $1 \leq r \leq p$ posibles, es decir:

$$\sum_{r=1}^p \binom{p}{r} \binom{n-1}{r-1} = \Theta(n^{p-1})$$

Podemos ahora estudiar el crecimiento de m con respecto a la función sucesor.

Lema 1.3 La palabra:

$$m_{p,j} = Ob \operatorname{suc}_p^1(0)b \operatorname{suc}_p^2(0)b \operatorname{suc}_p^3(0)b \dots b \operatorname{suc}_p^j(0)b$$

satisface:

$$|m_{j,p}| = \Theta(j \sqrt[j]{j}) \text{ y } |\operatorname{suc}_p^j(0)| = \Theta(\sqrt[j]{j})$$

Prueba Comencemos por demostrar $|\operatorname{suc}_p^j(0)| = \Theta(\sqrt[j]{j})$. Elijamos j que cumpla $|\operatorname{suc}_p^{j+1}(0)| > |\operatorname{suc}_p^j(0)| = n$. En este caso:

$$j = \|nB_p\| = \|\{w \in B_p \mid |w| \leq n\}\|$$

pero $\|nB_p\| = \sum_{j=1}^p \Theta(n^{j-1}) = \Theta(n^p)$ con lo que $j = \Theta(n^p)$, podemos entonces escribir

$$|\operatorname{suc}_p^{\Theta(n^p)}(0)| = n, \text{ o bien, } |\operatorname{suc}_p^j(0)| = \Theta(\sqrt[j]{j})$$

Vamos ahora a estimar $|m_{j,p}|$. Tenemos

$$|m_{j,p}| = j|b| + \sum_{i=0}^j |\operatorname{suc}_p^i(0)|$$

Podemos estimar la suma $\sum_{i=0}^j |\operatorname{suc}_p^i(0)|$ sumando por bloques, es de-

cir:

$$\begin{aligned} \sum_{i=0}^j | \text{suc}_p^i(0) | &= \sum_{i=1}^j \theta(\sqrt[i]{j}) \cdot i \cdot \| \{ w \in B_p \mid |w| = i \} \| \\ &= \sum_{i=1}^j \theta(\sqrt[i]{j}) \cdot i \cdot i^{p-1} = (\theta(\sqrt[j]{j}))^{p+1} = \theta(j\sqrt[j]{j}) \end{aligned}$$

Notemos que podemos, también, estimar la suma precedente con la ayuda de la integral $\int_0^j x \sqrt[p]{x} dx$, tenemos

$$\int_0^j x \sqrt[p]{x} dx = \theta(j\sqrt[j]{j})$$

2. Construcción de Gold(q)

En este párrafo vamos a generalizar el lenguaje de Goldstine a partir de la palabra infinita m .

$$m_q = 0b \text{suc}_q(0)b \text{suc}_q^2(0)b \dots$$

definida precedentemente. Para ello habrá que analizar la forma de las palabras w_1, w_2 , con $w_2 \neq \text{suc}_q(w_1)$.

Esto nos conducirá a definir el lenguaje Trans_q , formado de las malas transiciones posibles. A partir de Trans_q podemos fácilmente definir $\text{Gold}(q)$ y demostrar que $\text{Gold}(q) \in \overline{p\text{-Root}(q)}$.

Lema 1.4 Sean w_1, w_2 pertenecientes a B_{p+1} .

(a) Cuando w_1 se factoriza:

$$w_1 = w_{i+1} i i^x, \quad w_{i+1} \in p^* \dots (i+1)^*, \quad 0 \leq i < p,$$

$x > 0$

tenemos que $w_2 \neq \text{suc}_{p+1}(w_1)$ en los dos casos siguientes:

$$(a.1) \quad w_2 = w_l l^y, \quad w_l \in p^* \dots l^*, \quad l > j, \quad y > 0 \quad \text{con}$$

$$(j^y \neq 0^x) \vee (i + 1 \neq l) \vee (w_{i+1} \neq w_l)$$

$$(a.2) \quad w_2 = l^y, \quad l \in \{0, 1, \dots, p\}, \quad y > 0$$

(b) Cuando w_1 se factoriza $w_1 = p^x$, w_2 debe ser de la forma $w_2 \neq 0^{x+1}$ para satisfacer $w_2 \neq \text{suc}_{p+1}(w_1)$

Prueba Evidente a partir de la definición de la función suc_{p+1} .

Remarca 2.1 Sea X un alfabeto finito y sea $\# \in X$, recordemos que el lenguaje

$$L = \{f\#g \mid f, g \in X^*, f \neq g\}$$

es un lenguaje a contador y a un pico ($L \in \text{RoCl} \cap \text{Lin}$).

Para ello notemos que L puede expresarse como la unión de:

$$L_1 = \{f\#g \mid f, g \in X^*, |f| \neq |g|\}$$

$$L_2 = \{X^n x_1 X^p \# X^n x_2 X^q \mid x_1 \neq x_2, n, p, q \geq 0\}$$

Lema 1.5 Para el lenguaje siguiente

$$\text{Trans}_{p+1} = \{w_1 b w_2 b \mid w_1, w_2 \in B_{p+1}, w_2 \neq \text{suc}_{p+1}(w_1)\}$$

Existe un autómata a pila, a un pico y a contador que satisface:

$$p(w_1 b w_2 b) = 0 \quad (\min \{|w_1|, |w_2|\})$$

Prueba El caso (a) del lema precedente se escinde en cuatro casos

distintos:

$$L_1 = \{ w_{i+1} i^x b w_l l^y b \mid l > j, j^y \neq 0^x \}$$

Al llegar a i^x empilamos $\min(x, y)$ símbolos; al llegar a j^y depilamos y testamos $j^y \neq 0^x$.

$$\text{pila}_{L_1} = \min(x, y)$$

$$L_2 = \{ w_{i+1} i^x b w_l l^y b \mid l > j, l \neq i+1 \}$$

L_2 es un lenguaje racional. Luego $\text{pila}_{L_2} = 0$

$$L_3 = \{ w_{i+1} i^x b w_l l^y b \mid l > j, w_{i+1} \neq w_l \}$$

Recordando la remarca precedente tenemos;

$$\text{pila}_{L_3} = \min\{|w_{i+1}|, |w_l|\}$$

$$L_4 = \{ w_{i+1} i^x b l^y b \mid y > 0 \}$$

Es un lenguaje racional, luego $\text{pila}_{L_4} = 0$

Tratemos el caso (b).

$$L_5 = \{ p^x b w b \mid w \neq 0^{x+1} \}$$

Evidentemente $\text{pila}_{L_5} = \min\{x, |w|\}$

Vamos ahora a generalizar el lenguaje de Godstine.

Lema 1.6 El lenguaje siguiente:

$$\text{Gold}(q) = \{ 0 b w_2 b w_3 b \dots w_{j-1} b w_j b \dots w_r b \mid r \geq 2, w_i \in B_q, \\ i = 2, \dots, r, \exists j, 2 \leq j \leq r, w_j \neq \text{suc}_q(w_{j-1}) \}$$

Es un lenguaje a contador y a un pico, que satisface:

$$p_{\text{Gold}(q)}(n) = O\left(n^{\frac{1}{q}}\right)$$

Prueba Evidentemente $\text{Gold}(q)$ es a contador y a un pico, dado que admite la factorización.

$$\text{Gold}(q) = \text{Ob}(B_q \setminus 1) b (B_q b)^* \cup \text{Ob}(B_q b)^* \text{Trans}_q (B_q b)^*$$

A fin de obtener un límite superior a la altura de pila, vamos a considerar la primera aparición de j que cumple $w_j \neq \text{suc}(w_{j-1})$; cuando leemos una palabra w de $\text{Gold}(q)$ de izquierda a derecha, tenemos:

$$w = \text{Ob} \text{suc}_q(0) b \text{suc}_q^2(0) b \dots \text{suc}_q^{j-1}(0) b w_j b \dots w_r b \text{ con}$$

$$w_j \neq \text{suc}_q^{j-1}(0)$$

El testar que $\text{suc}_q^{j-1}(0) b w_j b \in \text{Trans}_q$ puede hacerse con una pila:

$$p(w) \leq \min(|\text{suc}_q^{j-1}(0)|, |w_j|) = O(|\text{suc}_q^{j-1}(0)|) = O(\sqrt[j]{j})$$

por otra parte $w = m_{q, j-1} w_j b \dots w_r b$ con lo que

$$|w| \geq |m_{q, j-1}| = \Theta(j \sqrt[j]{j}) \text{ con lo que}$$

$$p(w) = O(|w|^{\frac{1}{q}})$$

Nos queda por demostrar que dichos lenguajes no son racionales, este es el objetivo del lema siguiente.

Lema 1.7 Los lenguajes $\text{Gold}(q)$ no son racionales.

Prueba Si $\text{Gold}(q)$ fuera racional, también lo sería el lenguaje:

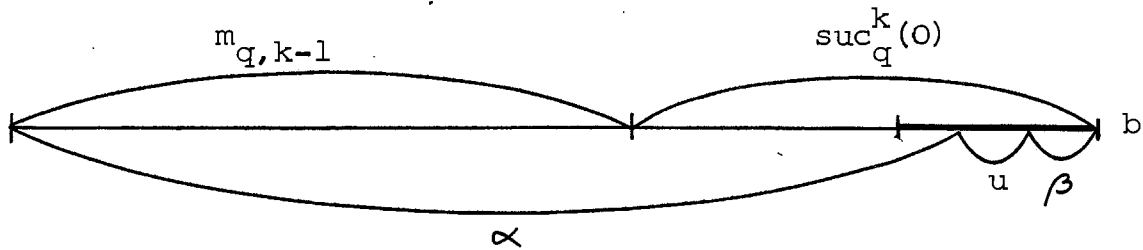
$$L = (B_q b)^2 (B_q b)^* \setminus \text{Gold}(q) = m_{q, j} \mid j \geq 2$$

dado que la diferencia de dos lenguajes racionales es racional. Su-

pongamos L racional, sea k la constante del lema de Ogden, consideremos la palabra $m_{q,k}$ en la que hemos marcado las últimas letras.

Tenemos:

$$m_{q,k} = m_{q,k-1} \text{ suc}_q^k(0)b$$



Dado que la función sucesor es única, tenemos que

$$m' = \alpha \beta \notin \{ m_{q,j} \mid j \geq 2 \}$$

con lo que llegamos a una contradicción.

II. Imbricación de lenguajes de Goldstine

Dicho método se halla descrito en [5], vamos a exponer intuitivamente su fundamento.

Con menores variaciones, podemos definir el lenguaje de Goldstine, a partir de la palabra infinita

321 211 2111

Supongamos que queremos contar los distintos bloques 21^x , $x > 0$ con la ayuda de otro lenguaje de Goldstine; tendremos

21 contado por 210

211 210 100

2111 210 100 1000

con lo que obtendremos la palabra

3210 210100 2101001000

dicha palabra puede obtenerse iterando el morfismo

$$h_3 = \langle 3 \rightarrow 3210 ; 2 \rightarrow 210 ; 1 \rightarrow 10 ; 0 \rightarrow 0 \rangle$$

tenemos

$$h_3(3) = 3210$$

$$h_3^2(3) = h_3(h_3(3)) = 3210 210 100$$

$$h_3^3(3) = h_3(h_3^2(3)) = 3210 210 100 210 100 10000$$

podemos decir que

$$\lim_{n \rightarrow \infty} h_3^n(3) = 3210 210100 \dots$$

Definamos formalmente dicho proceso. Sean

$$Z_0 = 0^*$$

$$Z_1 = 10^*$$

$$Z_2 = 2(10^*)^*$$

$$Z_3 = 3(2(10^*)^*)^*$$

es decir $Z_0 = 0^*$ y $Z_{q+1} = (q+1)Z_q^*$ $q \geq 0$

vamos a definir formalmente el morfismo h_q

Definición 1.6 Para todo i , $0 \leq i \leq q$ tomamos

$$h_q(i) = i(i-1)(i-2) \dots 10$$

En los párrafos siguientes notaremos $h_q^j(q) \equiv w_j^{(q)}$. Tenemos el siguiente lema de factorización:

Lema [5] 1.8. Se cumple:

$$w_i^{(q+1)} = (q+1)w_1^{(q)}w_2^{(q)} \dots w_i^{(q)}$$

Dicho lema nos permite estimar la longitud de $w_j^{(q)}$ en función de q y j , tenemos:

Lema 1.9 Los $w_i^{(q)}$ definidos precedentemente, satisfacen

$$|w_i^{(q)}| = \Theta(i^q)$$

Prueba Razonemos por inducción sobre q . Cuando $q = 1$, tenemos

$$w_i^{(1)} = 10^i \text{ con lo que } |w_i^{(1)}| = \Theta(i)$$

Supongamos que es cierto hasta q y demostrémoslo para $q + 1$.

Empleando el lema precedente, tenemos:

$$w_i^{(q+1)} = (q+1)w_1^{(q)} \dots w_i^{(q)}$$

con lo que

$$|w_i^{(q+1)}| = 1 + \sum_{j=1}^i |w_j^{(q)}| = 1 + \sum_{j=1}^i \Theta(j^q) = \Theta(j^{q+1})$$

Podemos ahora generalizar el lenguaje de Goldstine con:

Definición [5] 1.7 Para $q \geq 1$, definimos el lenguaje de Goldstine $q + 1$ imbricado por:

$$G_{q+1} = \{ (q+1)u_1u_2 \dots u_j \dots u_r \mid r \geq 1, u_i \in Z_q, i = 1, \dots, r, \\ \exists 1 \leq j \leq r \text{ que satisface } u_j \neq h_q^j(q) \}$$

A fin de encontrar un límite superior a la altura de pila, vamos a analizar las propiedades de factorización de las palabras de G_{q+1} .

Lema 1.10 Sea $w \in G_{q+1}$, w se factoriza de modo único en

(a) $w = (q+1)u_1u_2 \dots u_r$, $u_i \in Z_q$ y $u_1 \neq w_1^{(q)}$

(b) Existe $j \geq 1$ que satisface:

$$w = w_j^{(q+1)} u_{j+1} \dots u_r, u_i \in Z_q \text{ y } u_{j+1} \neq w_{j+1}^{(q)}$$

Prueba Recordemos que $w \in Z_{q+1}$, es decir tiene una estructura en bloques de Z_q . Basta, entonces, considerar el mayor número de bloques que coinciden con $w_r^{(q+1)}$ comenzando por la izquierda. ■

Vamos a encontrar un límite superior a los cálculos de G_{q+1} .

Lema 1.11 Sea $w \in G_{q+1}$, existe un autómata a pila que satisface

$$p(w) = 0(j) \text{ en que } w = w_j^{(q+1)} u_{j+1} \dots u_r \text{ con } u_{j+1} \neq w_{j+1}^{(q)}.$$

Prueba Razonemos por inducción sobre q .

Cuando $q = 1$, G_2 es básicamente el lenguaje de Goldstine y el lema precedente expresa la parte en que w coincide con la palabra infinita, siendo j el número de bloques.

Tomemos $(q + 1)$, u_{j+1} se factoriza:

$$u_{j+1} = qv_1v_2 \cdots v_j \cdots v_r, v_j \in Z_{q-1}$$

Hay dos casos $u_{j+1} \neq w_{j+1}$.

(1) $r \neq j + 1$. Podemos testarlo empilando las j letras q en $w_j^{(q+1)}$, es decir, contando el número de bloques en $w_j^{(q+1)}$ y contando el número de bloque v_i en u_{j+1} ; es decir, depilando al llegar a u_{j+1} cada vez que encontramos una letra $(q - 1)$. Dicho caso aparece cuando la mala transición tiene lugar en el nivel de imbricación más externo.

(2) $r = j + 1$, en este caso podemos reescribir la desigualdad $u_{j+1} \neq w_{j+1}^{(q)}$ diciendo que $u_{j+1} \notin G_q$. Es decir, la mala transición aparece en las capas más profundas de G_{q+1} .

En este caso u_{j+1} , se factoriza en :

$$u_{j+1} = w_{\ell}^{(q)} v_{\ell+1} \cdots v_{j+1}, v_{\ell+1} \neq w_{\ell+1}^{(q-1)}$$

Existe por hipótesis un autómata en G_q que satisface:

$$p(u_{j+1}) = O(\ell) = O(j + 1) = O(j)$$

La transformación de dicho autómata en un autómata de G_{q+1} , no presenta problemas, con lo que

$$p(w) = p(u_{j+1}) = O(j)$$

Notemos que las construcciones precedentes pueden realizarse con

autómatas a contador y a un pico.

Considerando conjuntamente los lemas precedentes, llegamos a:

Lema 1.12 Tenemos que $p_{G_q}(n) = O(\sqrt[q]{n})$

Prueba Sea $w \in G_q$, w satisface

$$w = w_j^{(q)} u_{j+1} \dots u_r$$

con $p(w) = O(j)$ y $|w| > |w_j^{(q)}| = \Theta(j^q)$, entonces

$$p(w) = O(|w|^{\frac{1}{q}})$$

Lenguajes p-Log Vamos a estudiar la existencia de lenguajes algebraicos no racionales cuya altura de pila es una función logarítmica de la longitud. Formalmente:

Definición 1.8 Diremos que un lenguaje L es p-Log ssi existe un autómata a pila a que lo acepta con

$$p_a(n) = O(\log n)$$

A fin de construir un lenguaje logarítmico, vamos a estudiar

$$m_b = 1b \text{ suc}_b(1)b \text{ suc}_b^2(1)b \dots \text{ suc}_b^j(1)b \dots$$

en la que suc_b es la función sucesor en el sistema binario. Sea $\text{Bin} = 1(0 + 1)^*$ tenemos:

Definición 1.9 Sea suc_b la aplicación definida por

$$\text{suc}_b : \text{Bin} \longrightarrow \text{Bin}$$

por $\text{suc}_b(w 01^x) = w 10^x$, $w \in \text{Bin}$, $x \geq 0$

$$\text{suc}_b(1^x) = 10^x, \quad x > 0$$

El crecimiento de m_b con respecto a suc_b viene expresado en el lema siguiente:

Lema 1.13 La palabra

$$m_{b,j} = 1b \text{suc}_b(1)b \text{suc}_b^2(1)b \dots \text{suc}_b^j(1)b$$

satisface:

$$|m_{b,j}| = \Theta(j \log j) \text{ y } |\text{suc}_b^j(1)| = \Theta(\log j)$$

Prueba Es bien conocido que

$$|\text{suc}_b^j(1)| = |\langle \text{expresión en binario de } j + 1 \rangle| = \Theta(\log j)$$

por otra parte

$$|m_{b,j}| = \Theta\left(j + \sum_{i=1}^j \log i\right) = \Theta(j \log j)$$

recordemos que

$$\int \log x \, dx = x \log x$$



Vamos seguidamente a generalizar el concepto de malatransición:

Lema 1.14 Sean w_1, w_2 pertenecientes a Bin , tenemos que

$w_2 \neq \text{suc}_b(w_1)$ en los casos:

(a) Cuando $w_1 = w 0 1^x$, $w \in \text{Bin}$, $x \geq 0$, entonces :

$$w_2 = w' 1 0^y, w' \in \text{Bin} \cup \varepsilon \quad \text{y} \quad (w' \neq w) \vee (x \neq y)$$

o bien

$$w_2 = 1^z, z > 0$$

(b) Cuando $w_1 = 1^x$, $x > 0$ entonces

$$w_2 = w 1 0^y, w \in \text{Bin} \cup \varepsilon \quad \text{y} \quad (w \neq \varepsilon) \vee (x \neq y)$$

o bien

$$w_2 = 1^z, z > 0$$

Prueba Evidente a partir de la función suc_b .

Lema 1.15 Para el lenguaje:

$$\text{Trans}_b = \{w_1 b w_2 b \mid w_1, w_2 \in \text{Bin}, w_2 \neq \text{suc}_b(w_1)\}$$

existe un autómata a pila, a contador, y a un pico que satisface:

$$p(w_1 b w_2 b) = O(\min\{|w_1|, |w_2|\})$$

Prueba Análoga a la dada para suc_q .

Estamos ahora en situación de definir un nuevo lenguaje de Goldstine:

Lema 1.16 El lenguaje siguiente:

$$\text{Gold}_{\text{bin}} = \left\{ lbw_2^b \dots w_{j-1}^b w_j^b \dots w_r^b \mid r \geq 2, w_i \in \text{Bin}, \right. \\ \left. i = 1, \dots, r, \exists j, 2 \leq j \leq r \text{ con } w_j \neq \text{suc}_b(w_{j-1}) \right\}$$

Es a contador a un pico y satisface

$$p_{\text{Gold}_{\text{Bin}}}(n) = O(\log n)$$

Prueba Tenemos la factorización

$$\text{Gold}_{\text{Bin}} = lb(\text{Bin} \setminus 10)b(\text{Bin } b)^* \cup Ob(\text{Bin } b)^* \text{Trans}_b(\text{Bin } b)^*$$

Considerando la primera mala transición que aparece, tenemos

$$w = m_{b, j-1} w_j^b \dots w_r$$

y $\text{suc}_b^{j-1}(1) b w_j^b \in \text{Trans}_b$ con lo que

$$p(w) = O(|\text{suc}_b^{j-1}(1)|) = O(\log j)$$

$$|w| = \Theta(j \log j)$$

con lo que $p(w) = O(\log |w|)$

La prueba que Gold_{Bin} no es racional no presenta problemas. ■

V.2 Operaciones entre lenguajes y cadenas de full-A.F.L.

Esta segunda sección se halla dividida en tres subsecciones. En la primera subsección, se estudia el comportamiento de la p -medida con respecto a las operaciones unión, producto, estrella, transducción racional y substitución sintáctica.

Como consecuencia de ello, obtenemos que $p\text{-}\overline{\text{Sub}}$, $p\text{-}\overline{\text{Root}(q)}$ para q fijo y $p\text{-}\overline{\text{Log}}$ son full-A.F.L. cerrados por substitución.

A fin de estudiar las inclusiones estrictas de los full-A.F.L. antes definidos, debemos estudiar los límites inferiores de la p -medida, y para ello debemos comenzar por estudiar los lemas de iteración en términos de un autómata a pila.

La segunda subsección cumple este objetivo. Comenzamos por ver que cuando la pila oscila en demasía, aparece un factor iterante, y cuando la pila aumenta demasiado, aparece un par iterante.

Definimos seguidamente lo que es un scan, y vemos cual es la longitud máxima de un scan en el que no aparecen elementos iterantes. Vemos, entonces, que si en un cálculo no aparecen elementos iterantes, todos sus scan son dotados y que la pila es una función lineal de la longitud. Podemos, entonces, demostrar que tanto una variación del lenguaje de Goldstine como del lenguaje de Patterson, necesitan estrictamente la raíz cuadrada para ser reconocidos por un autómata a pila.

El tercer párrafo demuestra que todos los A.F.L. definidos anteriormente, están incluidos estrictamente unos en otros.

Para ello, comenzamos definiendo el formato de Patterson de orden q ,

el bloque de Patterson de orden q y anchura n , y el casi bloque de Patterson de orden q y anchura n .

Con estos elementos definimos una variación del lenguaje de Patterson que llamamos "Patterson-menor de orden q " que precisa estrictamente la raíz q -ésima para ser reconocido, lo que nos da las distintas inclusiones estrictas; hemos demostrado por lo tanto:

$$p\text{-}\overline{\text{Log}} \subsetneq \dots \subsetneq p\text{-}\overline{\text{Root}(q)} \subsetneq p\text{-}\overline{\text{Root}(q-1)} \subsetneq \dots \subsetneq p\text{-}\overline{\text{Root}(2)}$$

V.2.1 Operaciones entre lenguajes

Vamos a estudiar seguidamente el comportamiento del índice inicial respecto a las principales operaciones entre lenguajes.

Comencemos por estudiar las operaciones racionales.

Teorema 2.1 Sean L_1 y L_2 dos lenguajes algebraicos, tenemos:

- (1) $p_{L_1+L_2}(n) \leq \max \{ p_{L_1}(n), p_{L_2}(n) \}$
- (2) $p_{L_1 \cdot L_2}(n) \leq \max \{ p_{L_1}(n), p_{L_2}(n) \} + 1$
- (3) $p_{L_1^*}(n) \leq p_{L_1}(n) + 1$

Prueba Sea $a_1 = \langle X_1, Z_1, Q_1, q_1, z_1, T_1 \rangle$ un autómata a pila que cumple $\text{Null}(a_1) = L_1$ y $p_{a_1}(n) = p_{L_1}(n)$.

Sea $a_2 = \langle X_2, Z_2, Q_2, q_2, z_2, T_2 \rangle$ uno análogo para L_2 , con $Z_1 \cap Z_2 = \emptyset$ y $Q_1 \cap Q_2 = \emptyset$.

(1) Para construir un autómata para L_1, L_2 basta considerar la reunión de los dos autómatas, es decir $a_3 = \langle X_3, Z_3, Q_3, q_3, z_3, T_3 \rangle$ con $X_3 = X_1 + X_2$, sea $z_3 \notin Z_1 + Z_2$ entonces $Z_3 = Z_1 + Z_2 + Z_3$, sea $q_3 \notin Q_1 + Q_2$ entonces $Q_3 = Q_1 + Q_2 + q_3$, las transiciones se-

rán:

$$T_3 = T_1 + T_2 + (q_3, \varepsilon, z_3) \vdash (q_1, z_1) + (q_2, z_2)$$

Evidentemente $p_{a_3}(n) = \max \{ p_{L_1}(n), p_{L_2}(n) \}$, con lo que

$$p_{L_1+L_2}(n) \leq p_{a_3}(n)$$

(2) Vamos a tratar el caso $\varepsilon \notin L_2$. Cuando $\varepsilon \in L_2$ hay variaciones menores. Dado que los autómatas reconocen por pila vacía, debemos añadir un nuevo fondo de pila a a_1 .

Sea $a_3 = \langle X_3, Z_3, Q_3, q_3, z_3, T_3 \rangle$ en que X_3, Z_3, Q_3 tienen la misma significación que en el párrafo precedente y

$$T_3 = T_1 + T_2 + \left\{ (q_3, \varepsilon, z_3) \vdash (q_1, z_2 z_1) \right\} \cup \left\{ (q, \varepsilon, z_2) \vdash (q_2, \varepsilon, z_2) \mid q \in Q_1 \right\}$$

Notemos que hemos añadido z_2 como nuevo fondo de pila en cualquier cálculo en L_1 . Tras leer una palabra de L_1 , quedaremos en un estado $q \in Q_1$ y con Z_2 en el fondo de pila.

La transición $(q, \varepsilon, z_2) \vdash (q_2, \varepsilon, z_2)$, nos permite reinicializar el cálculo con L_2 .

Evidentemente:

$$p_{a_3}(n) \leq \max \{ p_{a_1}(n), p_{a_2}(n) \} + 1$$

(3) Vamos a construir un autómata para L_1^* . Sea

$$a_3 = \langle X_1, Z_3, Q_3, q_3, z_3, T_3 \rangle \text{ en que } Z_3 = Z_1 + Z_2$$

$Q_3 = Q_1 + q_3$, y T_3 está definida por T_1 al que añadimos las transiciones:

$$(q_3, \varepsilon, z_3) \vdash (q_3, \varepsilon), \text{ a fin de garantizar que } a_3 \text{ acepta } \varepsilon.$$

$(q_3, \varepsilon, z_3) \vdash (q_1, z_3 z_1)$, a fin de poder leer L_1

$(q, \varepsilon, z_3) \vdash (q_1, z_3 z_1)$, $q \in Q$, a fin de poder relanzar el autó-
mata tras una lectura de una palabra de
 L_1 .

$(q, \varepsilon, z_3) \vdash (q, \varepsilon)$ a fin de poder vaciar la pila.

Evidentemente $p_{a_3}(n) \leq p_{a_1}(n) + 1$



Vamos a estudiar seguidamente las operaciones de cono racional.

A este fin vamos a recordar el lema fundamental demostrado en el capítulo precedente.

Lema 2.1 Sea L un lenguaje algebraico y ψ un morfismo alfabético existe una constante k que depende de L , que satisface:

$$n \psi(L) \subset \psi(knL)$$

Dicho lema es básico en la prueba de:

Lema 2.2 Sea L un lenguaje algebraico, ψ un morfismo alfabético y R un lenguaje racional, tenemos:

(1) Existe una constante k que depende de L que satisface

$$p_{\psi(L)}(n) \leq p_L(kn)$$

$$(2) p_{\psi^{-1}(L)}(n) \leq p_L(n) + 1$$

$$(3) p_{L \cap R}(n) \leq p_L(n)$$

Prueba Sea $\mathcal{A} = \langle X, Z, Q, q_0, z_0, T \rangle$ un autómata para L con $p_{\mathcal{A}}(n) = p_L(n)$, sea $\psi : X^* \rightarrow Y^*$.

(1) Definimos $\psi(\mathcal{A}) = \langle Y, Z, Q, q_0, z_0, \psi(T) \rangle$ en que si $(q, u, z) \vdash (q', \alpha)$ está en T , ponemos $(q, \psi(u), z) \vdash (q', \alpha)$ en $\psi(T)$.

Evidentemente $\psi(\mathcal{A})$ reconoce $\psi(L)$. Sea $w \in \psi(L)$, de acuerdo con el lema fundamental, existe $w' \in \text{kn}L$ que satisface $\psi(w') = w$. Por construcción w y w' tienen la misma evolución de pila, con lo que

$$p_{\psi(\mathcal{A})}(w) \leq p_{\mathcal{A}}(w') = p(\text{kn})$$

(2) Sea $\psi : X^* \rightarrow Y^*$ definimos $X_{\varepsilon} = \{x \in X \mid \psi(x) = \varepsilon\}$

Sea $\mathcal{A} = \langle Y, Z, Q, q_0, z_0, T \rangle$ con $\text{Null}(\mathcal{A}) = L$.

Vamos a construir $\psi^{-1}(\mathcal{A}) = \langle X, Z_1, Q_1, q_1, z_1, T_1 \rangle$

Añadiremos en principio un nuevo fondo de pila z_1 .

$$(q_1, \varepsilon, z_1) \vdash (q_0, z_1 z_0)$$

Transformamos las reglas de T por morfismo: Si $(q, u, z) \vdash (q', \alpha)$ está en T añadimos:

$$(q, \varepsilon, z) \vdash (q', \alpha) \text{ si } u = \varepsilon$$

$$(q, v, z) \vdash (q', \alpha) \text{ si } u \neq \varepsilon, v \in X, \psi(v) = u$$

Añadimos bucles en las letras que se borran por ψ , para todo $q \in Q_1, z \in Z_1, x \in X_{\varepsilon}$ añadimos

$$(q, x, z) \vdash (q, z)$$

Notemos que dado que hemos añadido z_1 en la pila, podemos hacer X_{ϵ}^* antes de comenzar a "leer" cualquier letra y después de leer todas las letras.

Evidentemente:

$$p_{\psi^{-1}(a)}^{(n)} \leq p_a^{(n)} + 1$$

(3) Sea $a_1 = \langle X, Z, Q_1, q_1, z, T_1 \rangle$ un autómata para L y sea

$a_2 = \langle X, Q_2, q_2, T_2 \rangle$ un autómata para R.

Formemos $a_1 \cap a_2 = \langle X, Z, Q_1 \times Q_2, q_1 \times q_2, T_3 \rangle$ clásicamente por:

Si $(q, u, z) \xrightarrow{\alpha} (q', \alpha) \in T_1$ y $(\bar{q}, u, \bar{q}') \in T_2$ entonces

$([q, \bar{q}], u, z) \xrightarrow{\alpha} ([q', \bar{q}'], \alpha)$ pertenece a T_3 .

Evidentemente

$$p_{a_1 \cap a_2}^{(n)} \leq p_{a_1}^{(n)}$$

El lema precedente nos permite enunciar el comportamiento del índice de pila en las transducciones racionales; tenemos:

Teorema 2.2 Sea L un lenguaje algebraico y τ una transducción racional, existe un entero k que satisface

$$p_{\tau(L)}^{(n)} \leq p_L(kn)$$

Prueba Según el teorema de Nivat sabemos que toda transducción puede factorizarse como:

$$\tau(w) = \psi(\varphi^{-1}(w) \cap R)$$

en que ψ y φ son alfabéticos y R es racional. Tenemos que:

$$\begin{aligned} p_{\tau(L)}(n) &= p_{\psi(\varphi^{-1}(L) \cap R)}(n) \leq p_{\varphi^{-1}(L) \cap R}(kn) \\ &\leq p_{\varphi^{-1}(L)}(kn) \leq p_L(kn) \end{aligned}$$



Vamos seguidamente a estudiar el comportamiento del índice de pila en la substitución sintáctica.

Recordemos que

$$L_1 \uparrow L_2 = \{a_1 m_1 a_2 m_2 \dots a_r m_r \mid a_1 a_1 \dots a_r \in L_1 \text{ y } m_i \in L_2\}$$

Teorema 2.3 Sean L_1, L_2 lenguajes algebraicos, tenemos:

$$p_{L_1 \uparrow L_2}(n) \leq p_{L_1}(n) + p_{L_2}(n) + 1$$

Prueba Sea $\mathcal{A} = \langle X, Z, Q, q_0, z_0, T \rangle$ un autómata para L_1 y

$\bar{\mathcal{A}} = \langle \bar{X}, \bar{Z}, \bar{Q}, \bar{q}_0, \bar{z}_0, \bar{T} \rangle$ uno para L_2 .

Suponemos $Z \cap \bar{Z} = \emptyset$, $Q \cap \bar{Q} = \emptyset$

Una palabra w de $L_1 \uparrow L_2$ se factoriza no forzosamente de modo único en

$$w = a_1 m_1 a_2 m_2 \dots a_r m_r$$

Con lo que tras la lectura de una letra a_i , debemos leer una palabra m_i de L_2 . Notemos que si $\varepsilon \notin L_2$ y $X \cap \bar{X} = \emptyset$ no pueden existir dos letras de X seguidas. Para ello vamos a separar las transiciones de T en tres tipos.

(1) ϵ -transiciones en las que aún no se ha leído una letra de X , marcaremos estos estados sin sombrero

$$(q, \epsilon, z) \longmapsto (q', \alpha)$$

(2) Lectura de una letra de X , pasaremos a un estado con sombrero

$$(q, x, z) \longmapsto (\hat{q}, \alpha)$$

(3) Lectura de ϵ , con sombrero

$$(\hat{q}, \epsilon, z) \longmapsto (\hat{q}, \alpha)$$

Notemos que hemos modificado en el autómata \mathcal{A} los conjuntos Q y T .

Sea $\mathcal{A}_{\hat{q}}$ el autómata \mathcal{A} en el que memorizamos \hat{q} . $\bar{T}_{\hat{q}}$ será de la forma:

$$(\bar{q}_{\hat{q}}, u, \bar{z}) \longmapsto (\bar{q}'_{\hat{q}}, \bar{\alpha})$$

Cuando estamos en un estado \hat{q} de \mathcal{A} debemos leer una palabra m , en este caso pasamos el control a $\mathcal{A}_{\hat{q}}$ con la transición.

$$(\hat{q}, \epsilon, z) \longmapsto (\bar{q}_0, \hat{q}, z\bar{z}_0)$$

Tras leer la palabra m , dado que $Z \cap \bar{Z} = \emptyset$ volveremos a encontrar z arriba de la pila, y estaremos en un estado $\bar{q}_{\hat{q}}$.

Pasamos nuevamente control al autómata \mathcal{A} con la transición

$$(\bar{q}_{\hat{q}}, \epsilon, z) \longmapsto (q, \epsilon, z)$$

en la que q es el estado q pero sin sombrero (podemos comenzar a leer ϵ -transiciones hasta encontrar otra letra a).

Hay un pequeño problema cuando hayamos leído la última letra a_r de w . En este caso la pila estará vacía y no podremos leer m_r . Para evitarlo empilamos un símbolo suplementario Z_1 al principio; cambiando el estado inicial q_1 , tenemos

$$(q_1, \varepsilon, z_1) \vdash (q_0, z_1 z_0)$$

Si queremos que el autómata reconozca con pila vacía, deberemos depilar Z_1 con una ε -transición.

Llamemos $a_1 \uparrow a_2$ el autómata que acabamos de esquematizar, tenemos

$$p_{a_1 \uparrow a_2}^{(n)} \leq p_{a_1}^{(n)} + p_{a_2}^{(n)} + 1$$

Podemos ahora estudiar cuáles son las propiedades de cierre de las familias $p\text{-}\overline{\text{Sub}}$, $p\text{-}\overline{\text{Root}(q)}$ y $p\text{-}\overline{\text{Log}}$, definidas precedentemente.

Tenemos el siguiente teorema que liga dichas familias a la teoría clásica de A.F.L.

Teorema 2.4 Las familias siguientes son full-A.F.L. cerrados por substitución:

- (1) $p\text{-}\overline{\text{Sub}}$
- (2) $p\text{-}\overline{\text{Root}(q)}$, para todo q
- (3) $p\text{-}\overline{\text{Log}}$

Prueba Basta ver que cada una de las clases de funciones que sirven de base a la definición de dichas familias, son estables en las operaciones de: max, producto por una constante, suma, adición de la unidad. Recordemos que estas son las operaciones que aparecen en las operaciones de full-A.F.L., cerrado por substitución.

Por construcción tenemos la cadena de inclusiones

$$\text{Rat} \not\subseteq \overline{\text{p-Log}} \subseteq \dots \subseteq \overline{\text{p-Root}(p+1)} \subseteq \overline{\text{p-Root}(p)} \subseteq \dots \subseteq \overline{\text{p-Root}(2)} \subseteq \overline{\text{p-Sub}} \not\subseteq \overline{\text{p-Root}(1)} = \text{Alg}$$

Vamos seguidamente a demostrar que la mayoría de dichas inclusiones son estrictas, cosa que nos permitirá definir una jerarquía decreciente de full-A.F.L. distinta.

V.2.2 Lemas de iteración y límites inferiores

A fin de demostrar algunos límites inferiores para el índice de pila, comenzamos por demostrar algunos lemas de iteración de los autómatas a pila.

A fin de obtener escrituras mas simples para las pruebas, adoptaremos las notaciones de Valiant [52] .

Notaremos las transiciones $(q, x, z) \xrightarrow{\quad} (q', w)$ por $(q, z) \xrightarrow{x} (q', w)$ llamaremos al par $(q, z) \in Q \times Z$ y al par $c: (q, w) \in Q \times Z^*$ configuración.

Tenemos, entonces, que una derivación o cálculo γ es una sucesión de transiciones a través de configuraciones.

Llamaremos longitud del cálculo γ , $|\gamma|$, el número de transiciones realizadas por dicho cálculo. Si durante γ se lee una palabra f , tenemos $|\gamma| \geq |f|$ debido a la existencia de ϵ -transiciones.

Notaremos $|Q| = q$ y $|Z| = t$.

Diremos que c es una configuración staking de Υ si todas las configuraciones que la siguen tienen un peso (altura de pila) $> |c|$.

Diremos que c es una configuración popping si todas las configuraciones que la preceden tienen peso $> |c|$.

Comenzamos seguidamente con el primer lema de iteración:

Lema [52] 2.3 Sea $\mathcal{A} = \langle X, Z, Q, q_-, z_0, T \rangle$ un autómata a pila con $|Q| = q$ y $|Z| = t$, consideremos un cálculo δ entre dos configuraciones c_0 y c_1 ; sea c_m una configuración de altura máxima aparecida en Υ , tenemos:

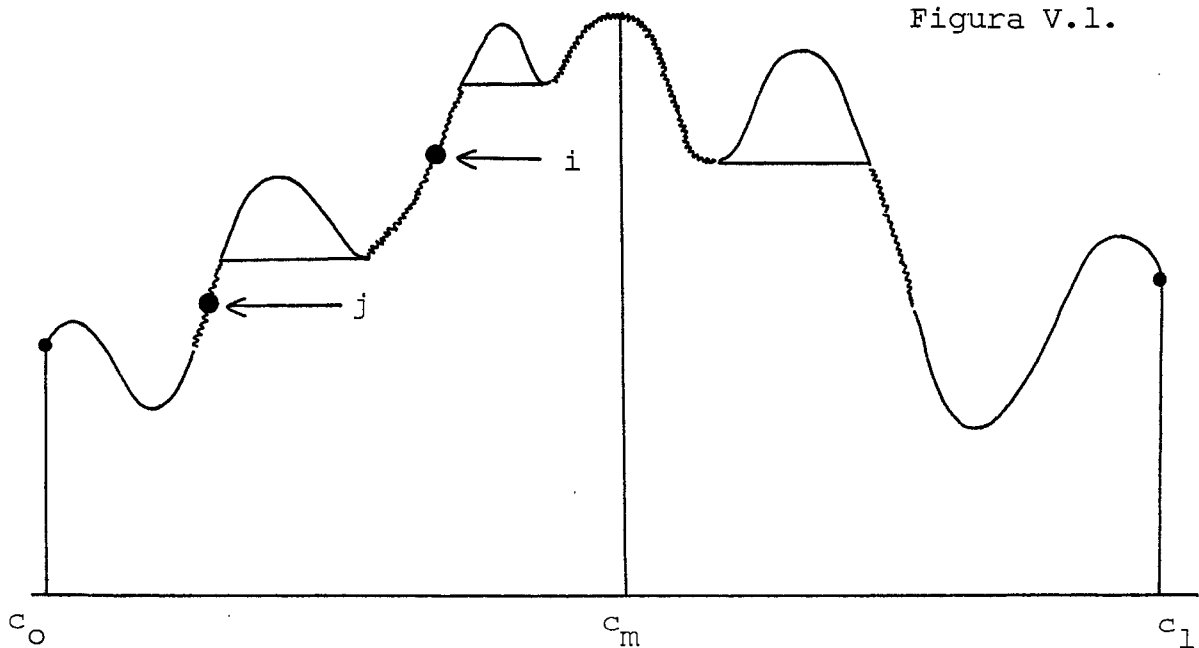
(a) Si $|c_0|$ y $|c_1|$ son menores que m y $|c_m| > n$ con $n - m > q^2 \cdot t$ existen u, v en T^+ que satisfacen $\delta = u\beta v\Upsilon$, siendo

$\alpha u^n \beta v^n \Upsilon$, $n \geq 0$ un cálculo entre c_0 y c_1 .

(b) Si $t > 1$ y $|\delta| > q \cdot (t + 1)^{|c_m|}$ ó $t = 1$ y $|\delta| > q^{|c_m|}$ existe $u \in T^+$ que satisface $\delta = \alpha u \beta$, siendo $\alpha u^n \beta$, $n \geq 0$ un cálculo entre c_0 y c_1 .

Prueba

(a) El cálculo δ puede esquematizarse en la figura V.1



Consideremos el conjunto $(Q \times Z) \times Q$, formado de elementos $([q, z], q')$, tiene como máximo cardinal q^2 . t. Dado que $n - m > q^2$. t. deben existir dos enteros i, j que satisfacen

. Los modos de la configuración staking entre c_0 y c_m de δ coinciden en las alturas $i + 1$ y $j + 1$.

. Los estados de la configuración popping entre c_m y c_1 de δ coinciden en las alturas i y j .

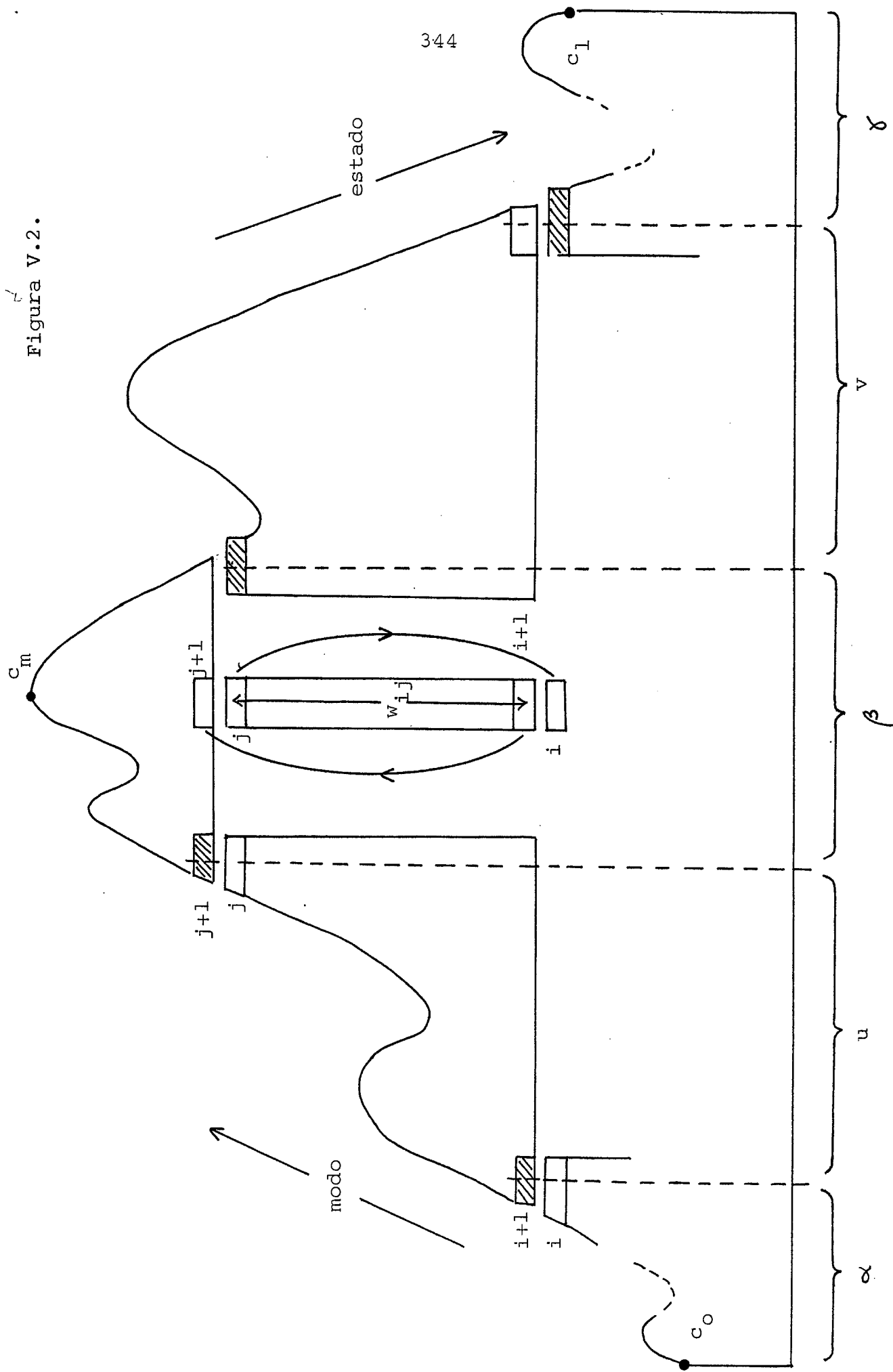
Notemos que podemos considerar que en $([q, z], q')$ tomamos el modo $[q, z]$ a una altura $i + 1$ y q' a la altura $j + 1$.

Dicha factorización se halla esquematizada en la figura V.2, en la que se muestra la factorización de δ en $\alpha u \beta v \gamma$. La coincidencia de modos en los niveles $i + 1$ y $j + 1$, nos permite empilar el segmento u un número $n > 0$ de veces reiteradamente, en cada caso aumentaremos la pila de un segmento w_{ij} .

La coincidencia de los estados en los niveles j e i , nos permite depilar el segmento w_{ij}^n iterando el cálculo v , n veces. Dichos procesos se hallan esquematizados en las figuras V.3 y V.4.

Notemos que por construcción $|u|, |v| \geq 1$, contienen la posición $i + 1$.

Figura V.2.



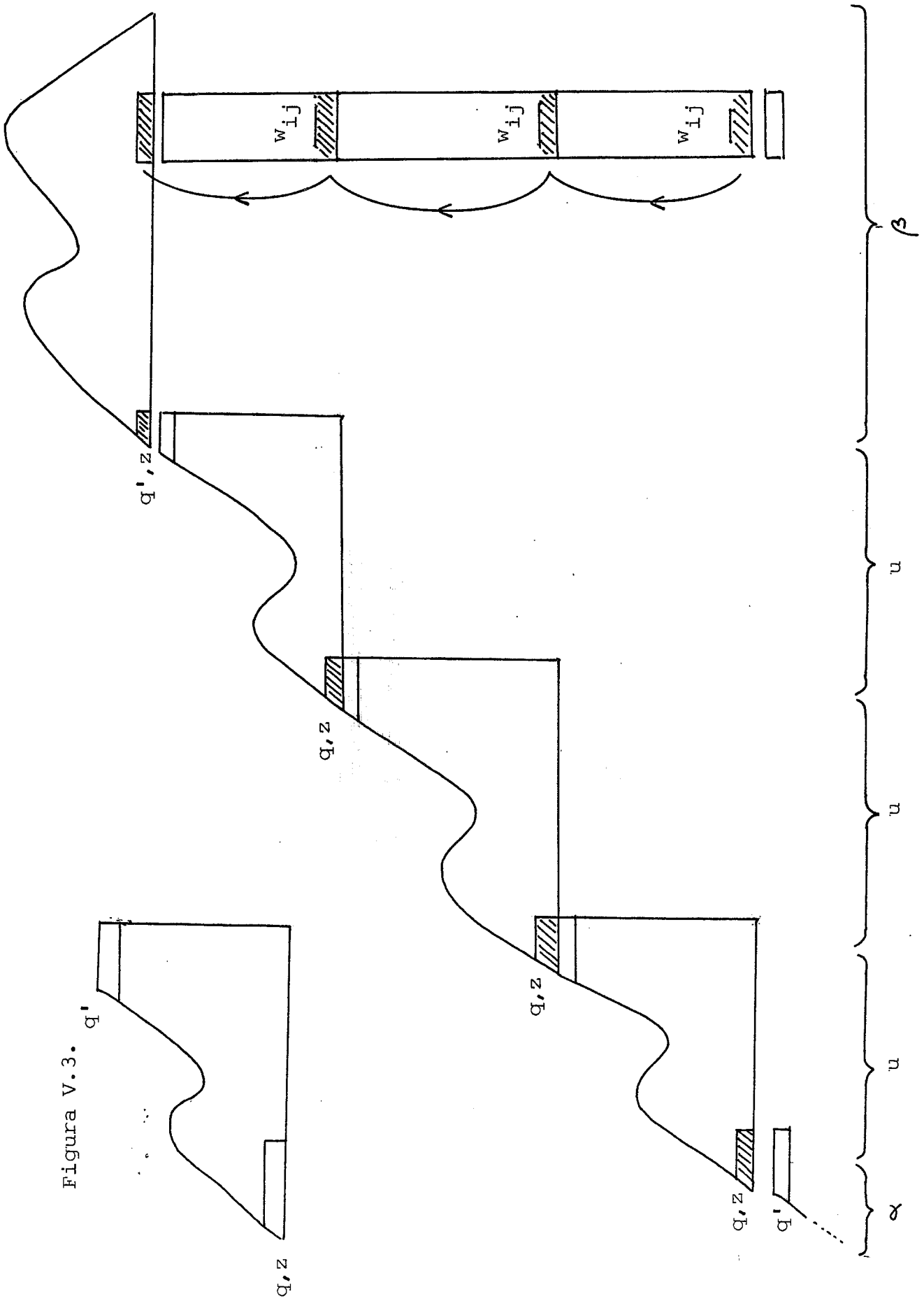
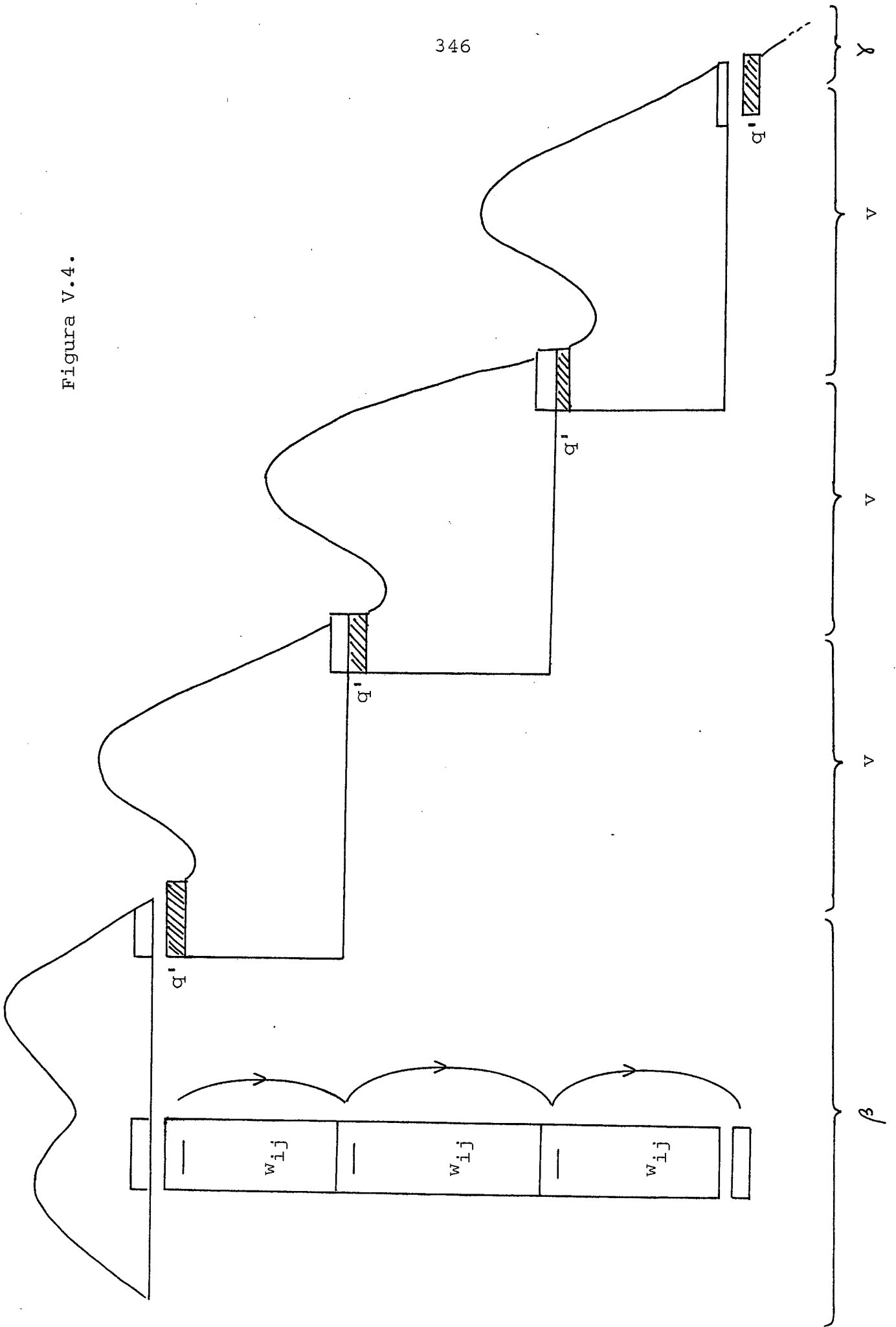


Figura V.3. q'

Figura V.4.



(b) Tratemos el caso $t > 1$. Vamos a demostrar que cuando

$|\delta| > q \cdot (t + 1)^{|c_m|}$ existen en δ dos configuraciones que se repiten.

Sea $a \notin Z$ un símbolo auxiliar. Consideremos δ esquematizado en la figura V.5.

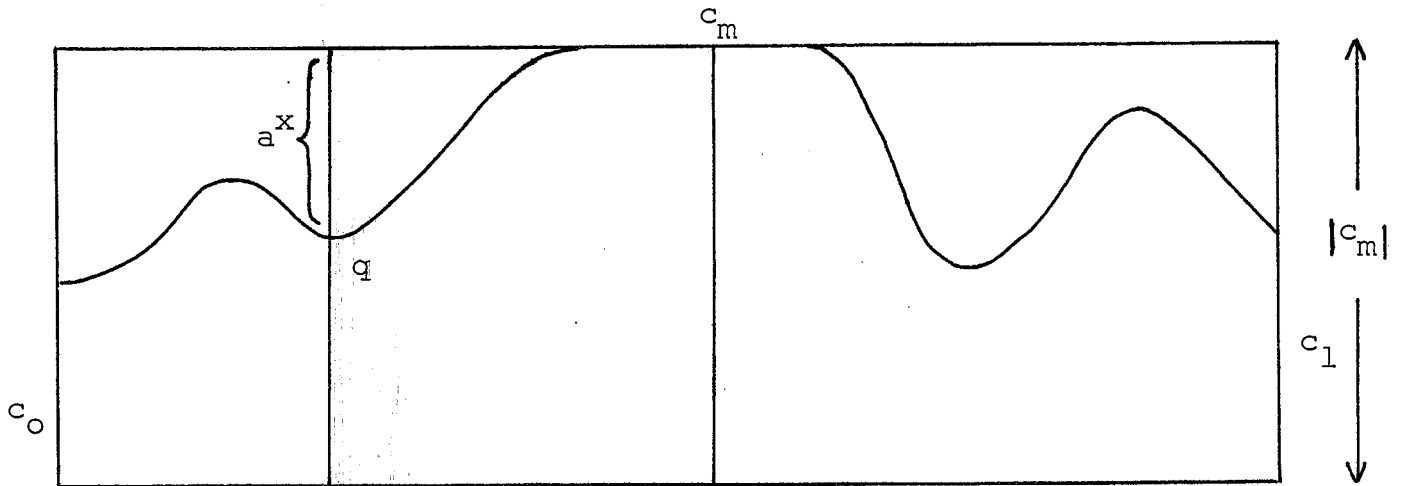


Figura V.5.

en la que hemos completado cada configuración con una sucesión de a hasta llegar a $|c_m|$. El número máximo de elementos que de este modo podemos obtener es el cardinal del conjunto $Q \times (Z + a)^{|c_m|}$, es decir $q(t + 1)^{|c_m|}$.

Si $|\delta| > q(t + 1)^{|c_m|}$ deben aparecer al menos dos veces la misma configuración, es decir, tenemos la factorización esquematizada en la figura V.6.

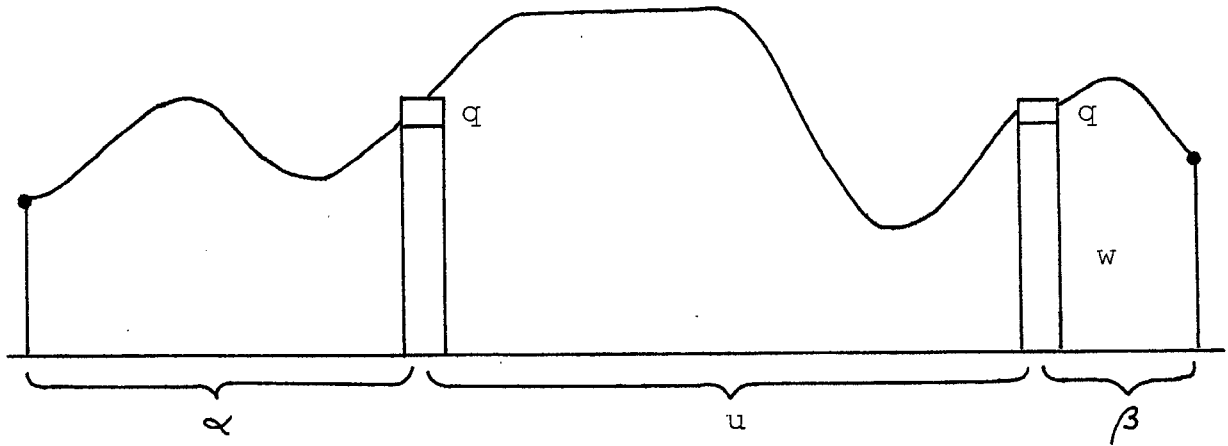


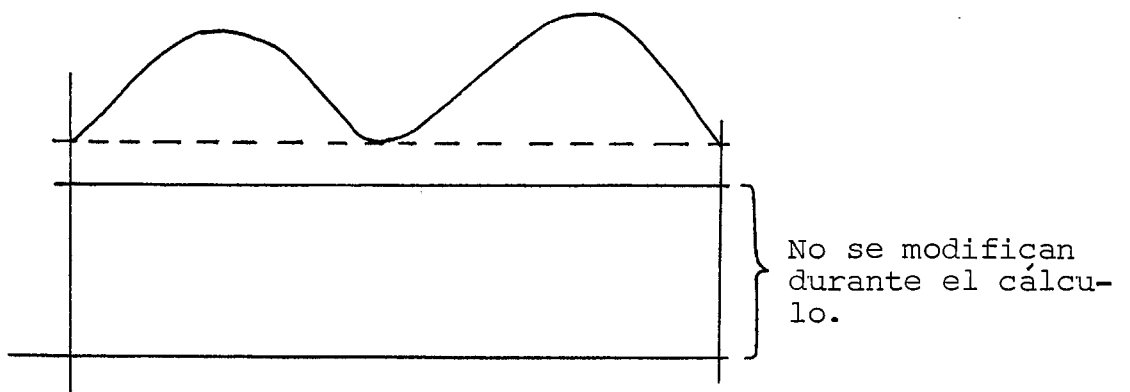
Figura V.6.

Tenemos, entonces, que $\alpha u^n \beta$ es un cálculo válido entre c_0 y c_1 para todo n .



Recordemos la definición de scan [27]. Intuitivamente un scan es una derivación en que los símbolos de pila que están por debajo del símbolo superior, al comienzo del cálculo, no intervienen en la definición. La figura V.7 esquematiza un scan:

Figura V.7.



formalmente:

Definición [27] 2.1 Sean c_0, c_1 dos configuraciones con $|c_0| = |c_1|$, diremos que un cálculo Υ entre c_0 y c_1 realiza un scan si la pila durante todo el cálculo Υ contiene al menos $|c_0|$ símbolos.

El lema siguiente nos muestra que todo scan demasiado largo o demasiado alto, contiene forzosamente una iteración.

Lema 2.4 Sea Υ un scan entre c_0 y c_1 si

$$(t > 1) \text{ y } (|\Upsilon| > q \cdot (t + 1)q^{2 \cdot t}) \text{ ó } (t = 1) \text{ y } (|\Upsilon| > q^{3 \cdot t})$$

existe un scan mas corto entre c_0 c_1 .

Prueba Trataremos el caso $t > 1$. Sea c_m una de las configuraciones cuya altura de pila es máxima en Υ . De acuerdo el lema precedente si $|\Upsilon| > q \cdot (t + 1) \frac{|c_m| - |c_0|}{q}$ el cálculo Υ debe poseer un factor iterante. Si $|c_m| - |c_0| > q^2 \cdot t$ debe existir en Υ un par iterante con lo que si:

$$|\Upsilon| > q \cdot (t + 1)q^{2 \cdot t}$$

podemos encontrar una derivación más corta debido ya sea a la aparición de un par o un factor iterante en Υ .

■

Los lemas precedentes, tienen por objeto ayudarnos a mostrar que si en un cálculo Υ depilamos una configuración c_0 de manera que no aparezcan iteraciones, debe cumplirse:

$$|c_0| = \Theta(|Y|)$$

formalmente:

Lema 2.5 Sea $c_0 = (q, w)$, $c_1 = (q', \epsilon)$ y Y un cálculo sin elementos iterantes entre c_0 y c_1 , existe una constante k que depende del autómata que satisface:

$$|Y| \geq |c_0| \geq \frac{1}{k} |Y| - 1$$

Prueba Consideremos el cálculo Y entre c_0 y c_1 . Dicho cálculo se halla esquematizado en la figura siguiente.

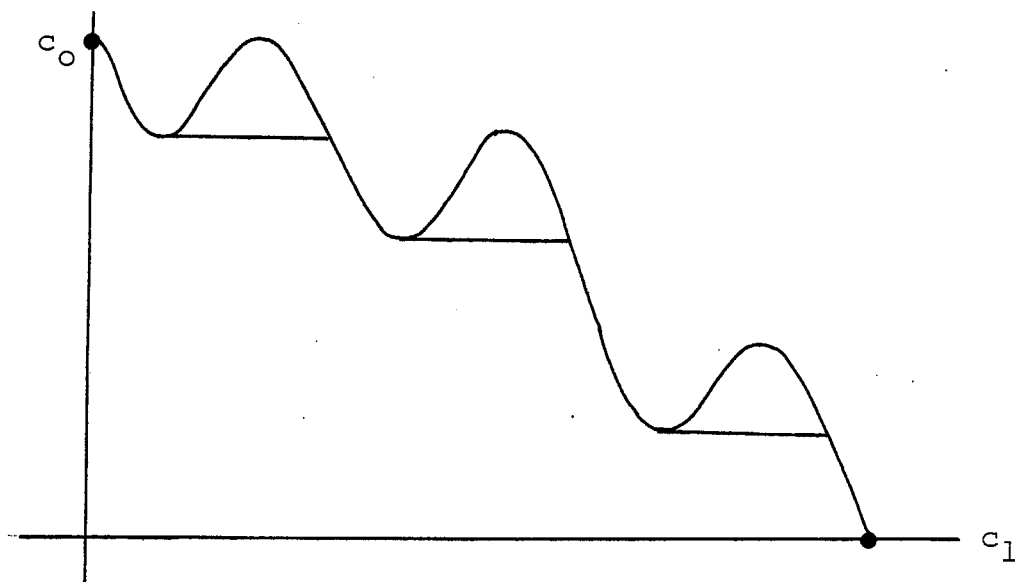


figura en la que hemos puesto de manifiesto los scans. Dado que no contiene elementos iterantes (pares o factores iterantes), el lema precedente nos muestra que existe k que limita la longitud de cada scan.

El modo más lento de descender en la pila es cuando Y se factoriza.

$$Y = s_1 d_1 s_2 d_2 s_3 d_3 \dots s_r d_r$$

en que a cada paso se produce un scan de longitud maximal y seguidamente depilamos un elemento. Tenemos entonces $r = |w|$.

Dado que $|s_i| \leq k$, obtenemos $|Y| \leq (r + 1)k = (|w| + 1)k$, o sea

$$\frac{1}{k}|Y| - 1 \leq |w| = |c_0|$$

El modo más rápido de depilar, aparece cuando no hay ningún scan y depilamos un símbolo a cada paso, es decir

$$Y = d_1 d_2 \dots d_r$$

con $r = |w|$, es decir $|Y| \geq |w| = |c_0|$. Finalmente obtenemos

$$|Y| \geq |c_0| \geq \frac{1}{k}|Y| - 1$$



Los lemas de iteración que acabamos de obtener, nos permiten estudiar límites inferiores para el reconocimiento de lenguajes. Comenzamos mostrando la técnica de cálculo detalladamente en los dos lemas siguientes.

Lema 2.6 El lenguaje siguiente satisface:

$\text{Gold}_< = \{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_r} b \mid r \geq 1, \exists j, 1 \leq j \leq r \text{ que } i_j < j \}$

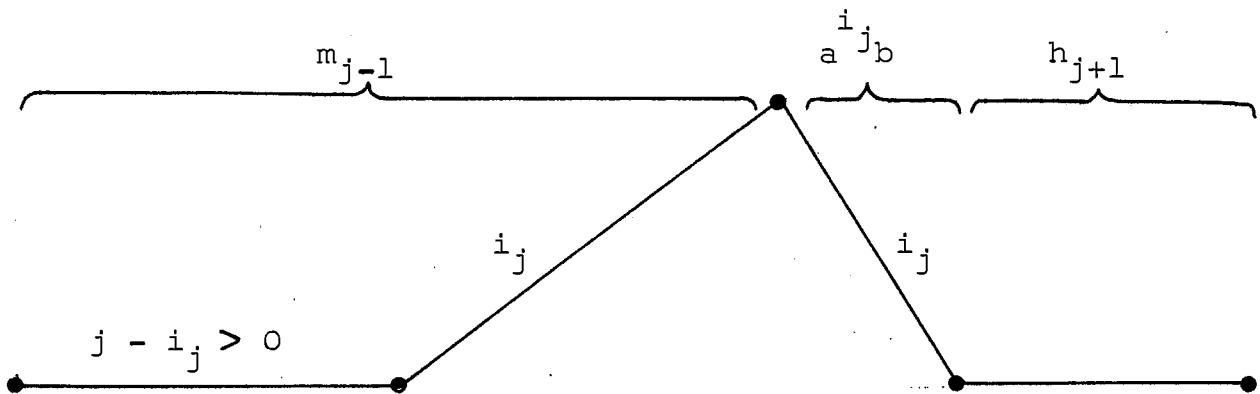
$$(1) p_{\text{Gold}_<}(n) = O\left(n^{\frac{1}{2}}\right)$$

(2) Sea $n \in \left\{ \frac{1}{2}x(x+1) + 2x + 1 \mid x \in N_+ \right\}$ se satisface

$$p_{\text{Gold}_<}(n) = \Omega\left(n^{\frac{1}{2}}\right)$$

Prueba

(1) Para demostrar esta primera parte, basta exhibir un autómata a pila que acepte el lenguaje en $O(n^{1/2})$ de la longitud de la palabra. Consideremos el autómata no determinista que verifica la existencia de un mal bloque. Dicho autómata empila las letras b (cuenta el número de bloques), de manera no determinista, elige un bloque de letras a y depila, verificando que hay menos letras a que bloques empilados. Guardando las notaciones definidas precedentemente, dicho cálculo puede esquematizarse en:



Notemos que en este cálculo c tenemos $p_c(w) = \Theta(i_j)$.

Consideremos que testamos la primera mala transición, es decir, el primer bloque (leyendo de izquierda a derecha) que satisface $i_j < j$, tenemos que m_{j-1} se factoriza:

$$m_{j-1} = a^{i_1} b a^{i_2} b \dots a^{i_r} b \dots a^{i_{j-1}} b \text{ con } i_r \geq r, 1 \leq r \leq i_{j-1}$$

Es decir, podemos reescribir m_{j-1} , de la forma :

$m_{j-1} = a^{1+x_1} b a^{2+x_2} b a^{3+x_3} b \dots a^{r+x_r} b \dots a^{(j-1)+x_{j-1}} b$ con las $x_r \geq 0$.

Tenemos, entonces, que $|m_{j-1}| \geq |aba^2b \dots a^{j-1}b| = \Theta(j^2)$, en consecuencia

$$p(w) = O(|w|^{\frac{1}{2}}).$$

(2) Para demostrar la segunda igualdad, hay que demostrar que cualquier autómatata a pila contiene para las longitudes

$$\left\{ \frac{1}{2} x(x+1) + 2x + 1 \mid x \in \mathbb{N}_+ \right\}$$

palabras cuyo cálculo necesita como mínimo una raíz cuadrada.

Consideremos el conjunto de palabras siguiente:

$$w_j = aba^2ba^3b \dots a^jba^jb, \quad j > 0$$

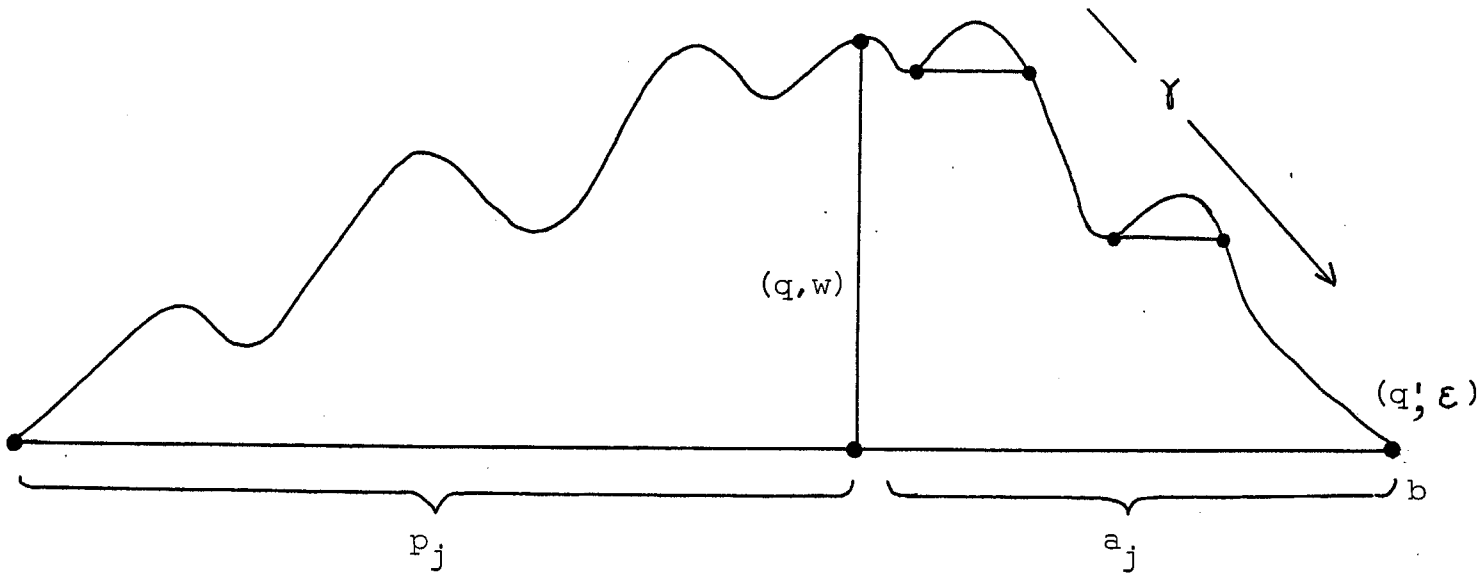
palabras en las que el mal bloque aparece en último lugar, y que en longitud crecen lo más suavemente posible.

Notemos $|w_j| = \frac{1}{2} j(j+1) + 2j + 1$

Llamemos $p_j = aba^2b \dots a^jb$, entonces $w_j = p_j a^j b$.

Sea \mathcal{A} un autómatata a pila cualquiera que reconozca $\text{Gold}_<$.

Consideremos en dicho autómatata un cálculo para w_j , más específicamente consideremos la altura de la pila tras la lectura de p_j , esquemáticamente tenemos:

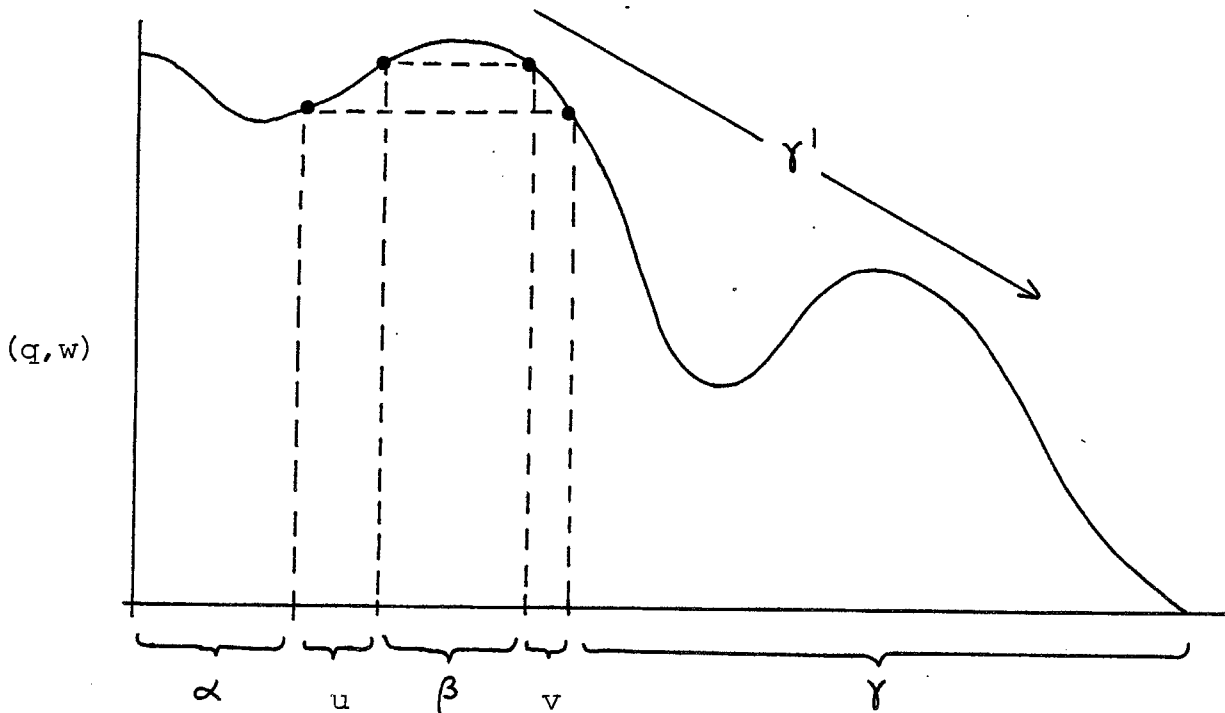


Consideremos los elementos iterantes que aparecen en el cálculo γ de $a^j b$ que va de (q, w) a (q', ϵ) .

Hay dos casos:

(1) Un scan cualquiera contiene un elemento iterante generado enteramente por ϵ -transiciones. En este caso dado que la iteración no toca al número de letras a , podemos quitarlo sin modificar el resultado, obteniendo otro cálculo γ' .

(2) Existe en γ' un scan que contiene un elemento iterante que contiene alguna letra a del bloque a^j . Supongamos que sea un par. Esquematizamos la situación en la figura siguiente:



con $|uv| \neq \varepsilon$.

Tenemos que las palabras $p_j \propto u^n \beta v^n \gamma$ admiten un cálculo en \mathcal{A} con lo que en particular la palabra:

$$p_j \propto u^2 \beta v^2 \gamma \in \text{Gold}_{\mathcal{A}}$$

pero como $uv \in a^+$, supongamos $uv = a^{1+\lambda}$, $\lambda \geq 0$, con lo que

$$p_j \propto u^2 \beta v^2 \gamma = aba^2b \dots a^j b a^{j+1+\lambda} b \in \text{Gold}_{\mathcal{A}}$$

llegamos, pues, a una contradicción y en dicho cálculo no pueden existir elementos iterantes del tipo (2). Hemos obtenido, pues, un cálculo γ' sin elementos iterantes. Aplicando el lema 2.5 tenemos:

$$|\gamma'| \geq |w| \geq \frac{1}{k} |\gamma'| - 1$$

es decir $|w| = \Theta(|\gamma'|)$. Como $|\gamma'| \geq |a^j b| > j$

Llegamos $|w| = \Omega(j)$. Es decir en el autómata \mathcal{A}

tenemos $p(w_j) = \Omega(j)$

dado que $|w_j| = \Theta(j^2)$ llegamos al resultado. ■

Vamos a dar un segundo ejemplo que constituye una variación del lenguaje de Patterson.

Lema 2.7 El lenguaje siguiente satisface:

$\text{Pat}_{<} = \{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_r} b \mid r \geq 1, \exists j, 1 \leq j \leq r$
que $i_j < r \}$

$$(1) p_{\text{Pat}_{<}}(n) = O(n^{\frac{1}{2}})$$

$$(2) \text{ Sea } n \in \{ x^2 + x + 1 \mid x \geq N_+ \}$$

$$p_{\text{Pat}_{<}}(n) = \Omega(n^{\frac{1}{2}})$$

Prueba

(1) Como en el ejemplo precedente, vamos a mostrar un autómata a pila que acepta dicho lenguaje con pila $O(n^{1/2})$.

Dicho autómata va a testar la existencia de un bloque con $i_j < r$.

En particular, siempre podemos suponer, que de manera no determinista, testa la existencia del i_j más pequeño con $i_j < r$. Sea

$$x = \{ \min i_j \mid \text{es el exponente de un bloque} \}$$

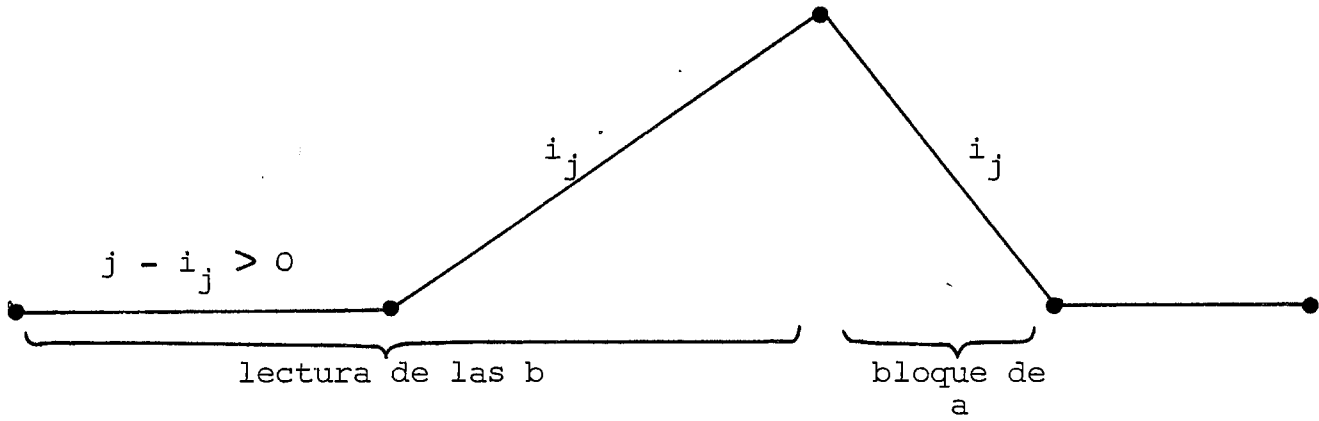
Respecto a la longitud de las palabras, tenemos:

$$|a^{i_1} b a^{i_2} b \dots a^{i_r} b| \geq |(a^x b)^r| = \Theta(rx)$$

Vamos a tratar separadamente dos casos:

Caso $i_j < j$ En este caso el funcionamiento del autómata a pila pue-

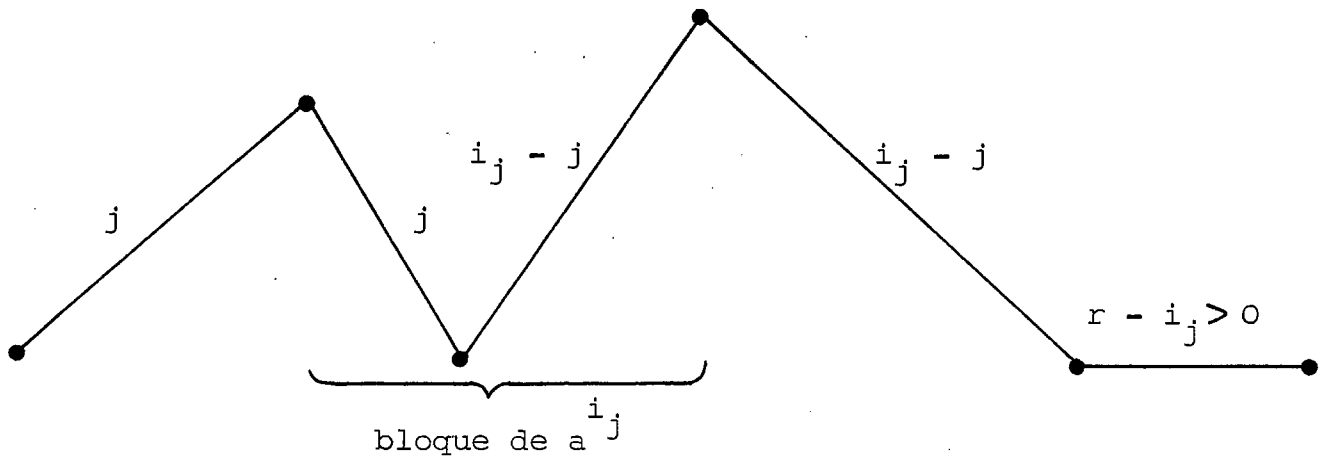
de esquematizarse



en este caso la pila = $\Theta(i_j)$. Como $i_j = x < j \leq r$, tenemos

$$|a a^{i_1} b a^{i_2} b \dots a^{i_r} b| > \Theta(rx) = \Theta(x^2) = \Theta(i_j^2)$$

Caso $i_j > j$ Tenemos que forzosamente $j < r$. Podemos esquematizar el cálculo en:



En este caso la pila = $\max \{j, i_j - j\} < i_j$. Como $r > x = i_j$, tenemos $|a^{i_1} b a^{i_2} b \dots a^{i_r} b| \geq \Theta(i_j^2)$

(2) Vamos a tratar seguidamente los límites inferiores. Para ello consideremos la sucesión de palabras

$$w_r = \underbrace{a^r b a^r b \dots a^r b}_{r-1} a^{r-1} b = (a^r b)^{r-1} a^{r-1} b$$

dichas palabras pertenecen a $\text{Pat}_{<}$. En particular:

$$|w_r| = |(a^r b)^{r-1} a^{r-1} b| = (r-1)(r+1) + r = r^2 + r - 1$$

Llamemos $p_r = (a^r b)^{r-1}$, tenemos $w_r = p_r a^{r-1} b$.

Consideremos un autómata a pila \mathcal{A} para $\text{Pat}_{<}$. Realizando el mismo razonamiento que en el párrafo precedente, vemos que en un cálculo para w_2 , tras la lectura de p_r , la altura de la pila debe servir para contar las $r-1$ ocurrencias de a ; es decir, el cálculo para a^{r-1} no puede contener elementos iterantes que no ocurran en ϵ -transiciones; con lo que llegamos

$$p(w_r) = \Omega(r)$$



Los dos últimos lemas nos muestran que, dado que la raíz $n^{\frac{1}{2}}$ ocurre en orden de magnitud una infinidad de veces, en cualquier autómata que reconozca $\text{Gold}_{<}$ y $\text{Pat}_{<}$, no podemos hallar para dichos lenguajes un autómata que trabaje con una pila

$$o(n^{\frac{1}{2}})$$

Tenemos, entonces que $n^{\frac{1}{2}}$ es una estimación precisa para los autó-

matas a pila que reconocen $\text{Gold}_{<}$ y $\text{Pat}_{<}$

Vamos a generalizar el lenguaje $\text{Pat}_{<}$ a fin de obtener una sucesión de lenguajes $\text{Pat}_{<}(q)$. Dichos lenguajes se reconocen en

$O(n^{\frac{1}{q}})$ y por otra parte $n^{\frac{1}{q}}$ ocurre una infinidad de veces; es decir, no se reconocen en $o(n^{\frac{1}{q}})$.

V.2.3 Cadenas de full- A.F.L.

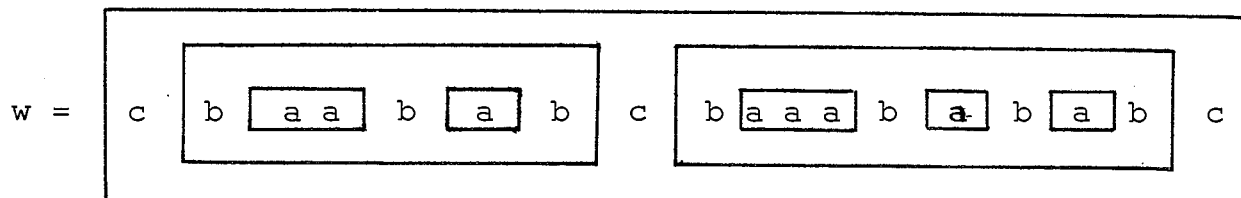
Este apartado tiene por objeto generalizar el lenguaje $\text{Pat}_{<}$.

Dado que $\text{Pat}_{<}$ es un lenguaje a estructura de bloques, $\text{Pat}_{<} \subset (a^*b)^*$ comenzaremos por definir un formato de palabra en que la estructura en bloques aparezca fácilmente.

Consideremos a modo de ejemplo el lenguaje:

$$c(b(a^+b)^+c)^+$$

Una palabra w de dicho lenguaje, tiene una estructura en bloques perfectamente clara, como vemos en el ejemplo siguiente:



Vemos, en particular, que nunca pueden haber dos letras b ó c seguidas, que cualquier bloque de letras a está rodeado de dos letras b , y que, cualquier bloque dominado por letras b está entre dos letras c . Tenemos en particular, que dos bloques distintos "no se tocan".

Podemos generalizar la construcción precedente en:

Definición 2.2 Sea $C = \{c_1, c_2, c_3, \dots, c_q, c_{q+1}, \dots\}$

definimos el formato de Patterson de orden q , P_q recursivamente por:

$$P_1 = c_1^+$$

$$P_{q+1} = c_{q+1} (P_q c_{q+1})^+ , \quad q \geq 1$$

Notemos que $P_3 = c_3 (c_2 (c_1^+ c_2)^+ c_3)^+$ es el lenguaje que hemos definido precedentemente. Podemos esquematizar P_q en

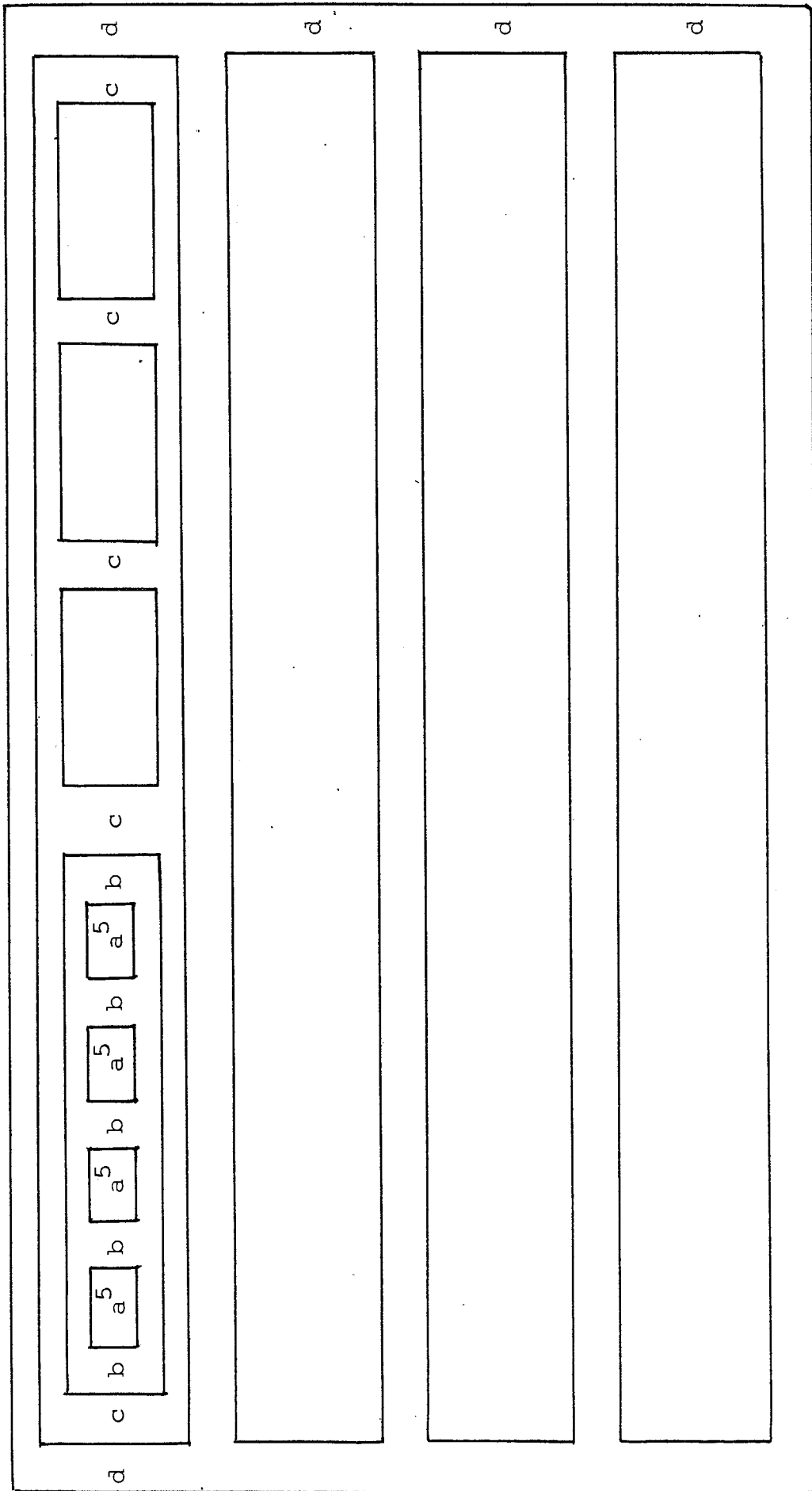
$$P_q = c_q (\dots c_4 (c_3 (c_2 (c_1^+ c_2)^+ c_3)^+ c_4)^+ \dots c_q)^+$$

Una palabra que resulta importante en el estudio de $\text{Pat} <$, es la palabra en la que todos los bloques son correctos a todos los niveles, es decir $(a^n b)^n$. Supongamos que queremos generalizar una palabra de este estilo a un alfabeto de cuatro letras a, b, c, d , obtendremos para $n = 5$ la figura V.8 en la que hemos esquematizado la estructura en bloques. La definición y lema siguientes muestran como podemos construir recursivamente estas palabras y medir su longitud.

Definición 2.3 Sea $C = \{c_1, c_2, \dots, c_q, c_{q+1}, \dots\}$, definimos el bloque de Patterson de orden q y anchura n , Pb_n^q , con $n \geq 2$ recursivamente por:

$$Pb_{n+1}^1 = c_1^{n+1}$$

$$Pb_{n+1}^{q+1} = c_{q+1} (Pb_{n+1}^q c_{q+1})^n , \quad q \geq 1$$



$$Sb_5^4 =$$

Figura V.8.

$$Pb_5^4 = d(c(b(a^5b)^4c)^4d)^4$$

Veamos explícitamente cual es la forma de Pb_{n+1}^q , tenemos

Lema 2.8 El bloque de Patterson de orden q y anchura $n + 1$, satisface

$$Pb_{n+1}^q = c_q (\dots c_4 (c_3 (c_2 (c_1^{n+1} c_2)^n c_3)^n c_4)^n \dots c_q)^n$$

$$\left| Pb_{n+1}^q \right| = (n + 1) \sum_{x=0}^{q-1} n^x = \Theta(n^q)$$

Prueba La primera afirmación es evidente a partir de la definición recursiva. Demostremos la segunda igualdad razonando por inducción sobre q . Cuando $q = 2$

$$Pb_{n+1}^2 = c_2 \underbrace{c_1^{n+1} c_2 c_1^{n+1} c_2 \dots c_2 c_1^{n+1} c_2}_{n \text{ veces } c_1^{n+1}}$$

con lo que $\left| Pb_{n+1}^2 \right| = n \left| c_1^{n+1} \right| + (n + 1) |c_2| = (n + 1)(n + 1)$

Supongámoslo para q y demostrémoslo para $q + 1$. Sabemos por definición que $Pb_{n+1}^{q+1} = c_{q+1} (Pb_{n+1}^q c_{q+1})^n$ con lo que

$$\begin{aligned} \left| Pb_{n+1}^{q+1} \right| &= n \left| Pb_{n+1}^q \right| + (n + 1) |c_{q+1}| = n(n + 1) \sum_{x=0}^{q-1} n^x + n + 1 \\ &= (n + 1) \sum_{x=0}^q n^x \end{aligned}$$

Recordemos que el cálculo del límite inferior para $\text{Pat}_<$ se basa en la palabra $(a^r b)^r$, en la que quitamos la última letra a , es decir, $(a^r b)^{r-1} a^{r-1} b$. Supongamos que quitamos la última (más a la derecha) aparición de la letra a en la palabra.

$$d(c(b(a^5 b)^4 c)^4 d)^4$$

definida en la figura V.8., vemos por inspección que la palabra que obtenemos puede factorizarse en:

$$\boxed{d(c(b(a^5 b)^4 c^4)^4 d)^3} \cdot \boxed{c(b(a^5 b)^4 c)^3} \cdot \boxed{b(a^5 b)^3} \boxed{a^4} b c d$$

La definición siguiente tiene por objeto definir recursivamente dichas palabras.

Definición 2.4 Llamamos casi bloque de Patterson de orden q y anchura n , Cb_n^q , la palabra definida recursivamente por:

$$Cb_{n+1}^1 = c_1^n$$

$$Cb_{n+1}^{q+1} = c_{q+1} (Cb_{n+1}^q c_{q+1})^{n-1}$$

A partir de los casi bloques definimos las palabras de Patterson de orden q y anchura n , W_n^q recursivamente por:

$$W_{n+1}^1 = c_1^n$$

$$W_{n+1}^{q+1} = Cb_{n+1}^{q+1} W_{n+1}^q c_{q+1}, \quad q \geq 1$$

Construimos seguidamente los primeros bloques, casi bloques y pa-

labras de Patterson para $n = 5$.

Comencemos por los bloques de Patterson:

$$Pb_5^1 = c_1^5$$

$$Pb_5^2 = c_2 (Pb_5^1 c_2)^4 = c_2 (c_1^5 c_2)^4$$

$$Pb_5^3 = c_3 (Pb_5^2 c_3)^4 = c_3 (c_2 (c_1^5 c_2)^4 c_3)^4$$

$$Pb_5^4 = c_4 (Pb_5^3 c_4)^4 = c_4 (c_3 (c_2 (c_1^5 c_2)^4 c_3)^4 c_4)^4$$

$$Pb_5^5 = c_5 (Pb_5^4 c_5)^4 = c_5 (c_4 (c_3 (c_2 (c_1^5 c_2)^4 c_3)^4 c_4)^4 c_5)^4$$

Demos seguidamente los primeros casi bloques de Patterson:

$$Cb_5^1 = c_1^4$$

$$Cb_5^2 = c_2 (Pb_5^1 c_2)^3 = c_2 (c_1^5 c_2)^3$$

$$Cb_5^3 = c_3 (Pb_5^2 c_3)^3 = c_3 (c_2 (c_1^5 c_2)^4 c_3)^3$$

$$Cb_5^4 = c_4 (Pb_5^3 c_4)^3 = c_4 (c_3 (c_2 (c_1^5 c_2)^4 c_3)^4 c_4)^3$$

$$Cb_5^5 = c_5 (Pb_5^4 c_5)^3 = c_5 (c_4 (c_3 (c_2 (c_1^5 c_2)^4 c_3)^4 c_4)^4 c_5)^3$$

Damos finalmente las primeras palabras de Patterson:

$$W_5^1 = Cb_5^1 = c_1^4$$

$$W_5^2 = Cb_5^2 W_5^1 c_2 = Cb_5^2 Cb_5^1 c_2 = c_2 (c_1^5 c_2)^3 c_1^4 c_2$$

$$W_5^3 = Cb_5^3 W_5^2 c_3 = Cb_5^3 Cb_5^2 Cb_5^1 c_2 c_3 =$$

$$= c_3 (c_2 (c_1^5 c_2)^4 c_3)^3 c_2 (c_1^5 c_2)^3 c_1^4 c_2 c_3$$

$$W_5^4 = Cb_5^4 W_5^3 c_4 = Cb_5^4 Cb_5^3 Cb_5^2 Cb_5^1 c_2 c_3 c_4 =$$

$$= c_4 (c_3 (c_2 (c_1^5 c_2)^4 c_3)^4 c_4)^3 c_3 (c_2 (c_1^5 c_2)^4 c_3)^3 c_2 (c_1^5 c_2)^3 c_1^4 c_2 c_3 c_4$$

En el lema siguiente vemos la relación entre las palabras y los bloques de Patterson. Vemos que una palabra de Patterson es un Bloque de Patterson, al que le falta un c_1 ; tenemos formalmente:

Lema 2.9 Los bloques y las palabras de Patterson satisfacen:

$$W_{n+1}^{q+1} = Cb_{n+1}^{q+1} W_{n+1}^q c_{q+1}, \quad n > 1$$

$$Pb_{n+1}^{q+1} = Cb_{n+1}^{q+1} Pb_{n+1}^q c_{q+1}, \quad n > 1$$

o bien explícitamente:

$$W_n^q = Cb_n^q Cb_n^{q-1} Cb_n^{q-2} \dots Cb_n^3 Cb_n^2 Cb_n^1 c_2 c_3 \dots$$

$$\dots c_{q-2} c_{q-1} c_q$$

$$Pb_n^q = Cb_n^q Cb_n^{q-1} Cb_n^{q-2} \dots Cb_n^3 Cb_n^2 Cb_n^1 c_1 c_2 c_3 \dots$$

$$\dots c_{q-2} c_{q-1} c_q$$

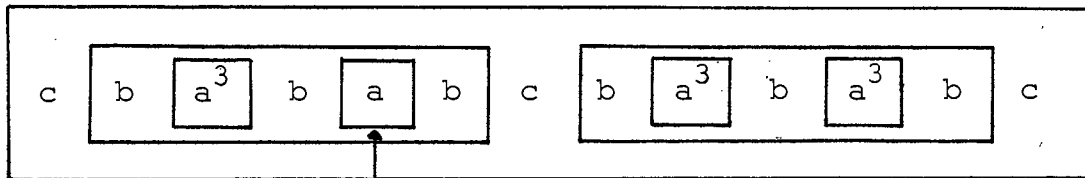
Prueba Recordemos que $Pb_{n+1}^{q+1} = c_{q+1} (Pb_{n+1}^q c_{q+1})^n =$

$$= c_{q+1} (Pb_{n+1}^q c_{q+1})^{n-1} Pb_{n+1}^q c_{q+1} = Cb_{n+1}^{q+1} Pb_{n+1}^q c_{q+1}$$

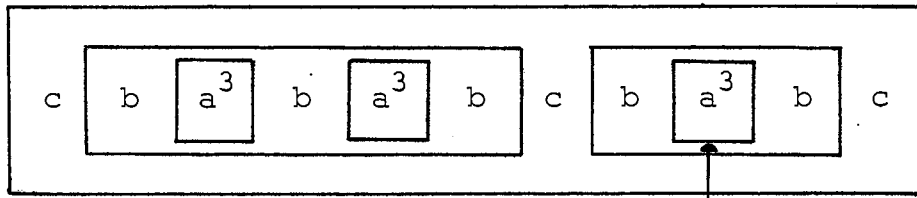
La segunda igualdad de Pb_n^q es debida a que $Pb_n^1 = Cb_n^1 c_1$.

Tras estas definiciones y lemas preliminares, podemos definir el lenguaje de Patterson de orden q , Pat_{\leq}^q . Intuitivamente, una palabra está en dicho lenguaje, si contiene un bloque tal que, el número de letras que lo separan en subbloques es menor que el número de letras que separan al bloque más exterior en subbloques.

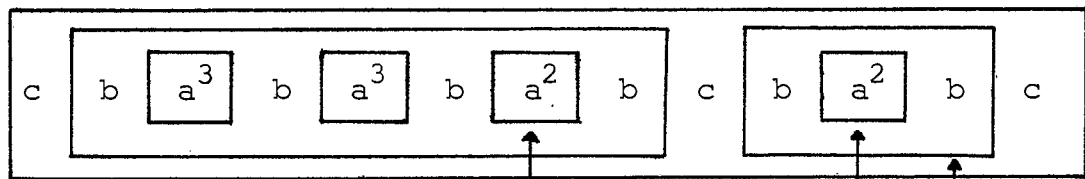
Podemos parafrasear esto diciendo que la palabra contiene un sub-bloque con menor número de letras que el correspondiente subbloque de Patterson. Efectivamente, los bloques de Patterson no pertenecen al lenguaje, pero los casi bloques, sí. Damos seguidamente varios ejemplos de este tipo de palabras



Vemos que en el lugar indicado faltan dos letras a.



Vemos que falta un bloque $a^3 b$



Pertenece por tres motivos.

Notemos que el conteo se hace siempre comparando el número de letras del subbloque con el número total de ocurrencias de c. Formalmente tenemos:

Definición 2.5 Para $q \geq 2$, definimos el lenguaje de Patterson-menor de orden q, $Pat_{<}^q$ por:

$$Pat_{<}^q = \left\{ w \in P_q \mid \exists j, 1 < j < q \text{ tal que } w \text{ se factoriza en } w = uc_jvc_jh \text{ con } v \in P_{j-1} \text{ y } |v|_{c_{j-1}} < |w|_{c_q} \right\}$$

El teorema siguiente nos muestra las propiedades del lenguaje $\text{Pat}_{<}^q$.

Teorema 2.5 El lenguaje $\text{Pat}_{<}^q$ es un lenguaje algebraico que satisface:

$$(a) \quad p_{\text{Pat}_{<}^q}(n) = O(n^{\frac{1}{q}})$$

(b) Sea $n \in \left\{ \frac{x+1}{x-1} (x^q - 1) - 1 \mid x > 2 \right\}$ tenemos

$$p_{\text{Pat}_{<}^q}(n) = \Omega(n^{\frac{1}{q}})$$

Prueba Consideremos un autómata a pila que va a testar la existencia de un mal bloque. Dicho autómata procede, fundamentalmente, como el autómata descrito para $\text{Pat}_{<}$ en la sección precedente.

Sea $w \in \text{Pat}_{<}^q$, $w = u c_j v c_j h$ con $|v|_{c_{j-1}} < |w|_{c_q}$.
 Nonos vamos a ocupar de demostrar que $w \in P_q$, dado que P_q es un lenguaje racional, y por tanto, puede verificarse sin memoria.

A fin de testar la desigualdad $|v|_{c_{j-1}} < |w|_{c_q}$, mientras leemos u , empilamos las ocurrencias de c_q ; cuando llegamos al bloque v depilamos a cada ocurrencia de c_{j-1} ; cuando llegamos al fondo de pila, si aún quedan c_{j-1} en la parte de v que quedan por leer, las empilamos; pasamos luego a h , depilamos, si los hay, los símbolos correspondientes a c_{j-1} , con símbolos c_q y testamos $|v|_{c_{j-1}} < |w|_{c_q}$.

Notemos que si el bloque v aparece pronto, respecto al número de letras c_q de u , dicho autómata tiene dos picos.

(a) A fin de hallar un límite superior, vamos a testar la existencia de un bloque v que contenga el menor número de separadores c_{j-1} respecto al número de c_q . Es decir, retraducimos la condición dada para $\text{Pat}_<$. Llamemos x un tal exponente. Por definición de x tenemos $|w| < |\text{Pb}_x^q| = \Theta(x^q)$ de acuerdo con los lemas precedentes. Por otra parte, la pila es como máximo x , con lo que $\text{pila}(w) = O(x) = O(|w|^{\frac{1}{q}})$

(b) Vamos a mostrar la última parte. Para ello vamos a aplicar el argumento dado para $\text{Gold}_<$ y $\text{Pat}_<$ a las palabras de Patterson w_n^q . Sabemos que

$$w_n^q = \text{Cb}_n^q \text{Cb}_n^{q-1} \dots \text{Cb}_n^2 c_1^n c_2 \dots c_{q-1} c_q$$

Consideremos la altura de la pila la lectura del prefijo

$\text{Cb}_n^q \text{Cb}_n^{q-1} \text{Cb}_n^2$. Cuando el autómata lee c_1^n , sabemos que no pueden aparecer elementos iterantes en el cálculo dado, que:

$$\text{Cb}_n^q \text{Cb}_n^{q-1} \dots \text{Cb}_n^2 c_1^n c_1^+ c_2 \dots c_{q-1} c_q$$

pertenecen al lenguaje. De acuerdo con los lemas de iteración vistos precedentemente, podemos concluir que $\text{pila}(w_n^q) = \Theta(n)$.

Nos queda ahora por estudiar la longitud de las palabras de Patterson. Hemos visto que una palabra de Patterson es un bloque de Patterson al que le falta una letra c_1 , con lo que $|w_n^q| = |\text{Pb}_n^q| - 1$.

Por otra parte hemos visto que $|\text{Pat}_n^q| = (n+1) \sum_{x=0}^{q-1} n^x =$

$= \frac{n+1}{n-1} (n^q - 1)$ con lo que $|W_n^q| = \frac{n+1}{n-1} (n^q - 1) - 1$, vemos que $\text{pila}(W_n^q) = \Theta(n^{\frac{1}{q}})$

Remarca 2.1 Notemos que al igual que hemos definido $\text{Pat}_{<}^q$ podemos definir Pat_{\neq}^q , formalmente:

$$\text{Pat}_{\neq}^q = \left\{ w \in P_q \mid \exists j, 1 < j < q \text{ tal que } w \text{ se factoriza en } \right. \\ \left. w = uc_jvc_jh \text{ con } v \in P_{j-1} \text{ y } |v|_{c_{j-1}} \neq |w|_{c_q} \right\}$$

Con las mismas técnicas que las empleadas en el párrafo precedente, podemos demostrar que Pat_{\neq}^q tiene un límite superior de pila en $n^{\frac{1}{q}}$, sin embargo, en este caso no sabemos hallar el límite inferior.

El teorema que acabamos de demostrar, nos permite ver que las inclusiones entre los distintos full-A.F.L., definidos en la sección precedente, es estricta. Tenemos:

Lema 2.9 Se satisface:

$$\text{Pat}_{<}^q \subset \overline{p\text{-Root}(q)} \setminus \overline{p\text{-Root}(q+1)}$$

$$\text{Pat}_{<}^q \subset \overline{p\text{-Root}(q)} \setminus \overline{p\text{-Log}}$$

Prueba Dado que existe una infinidad de valores n_i , para los que

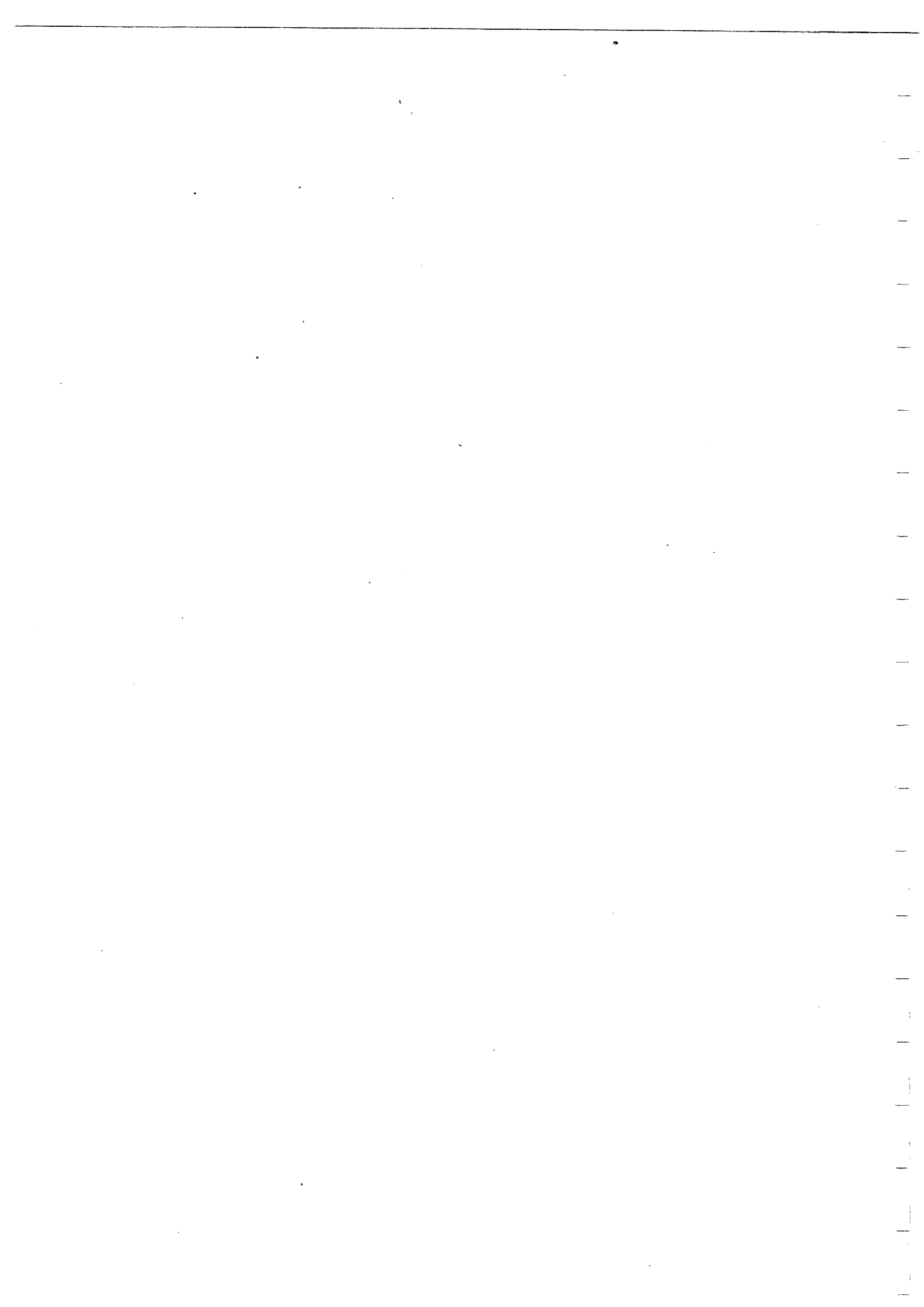
$p_{\text{Pat}^q} (n) = \Theta (n^{\frac{1}{q}})$ no es posible que

$$p_{\text{Pat}^q} (n) = o (n^{\frac{1}{q}})$$

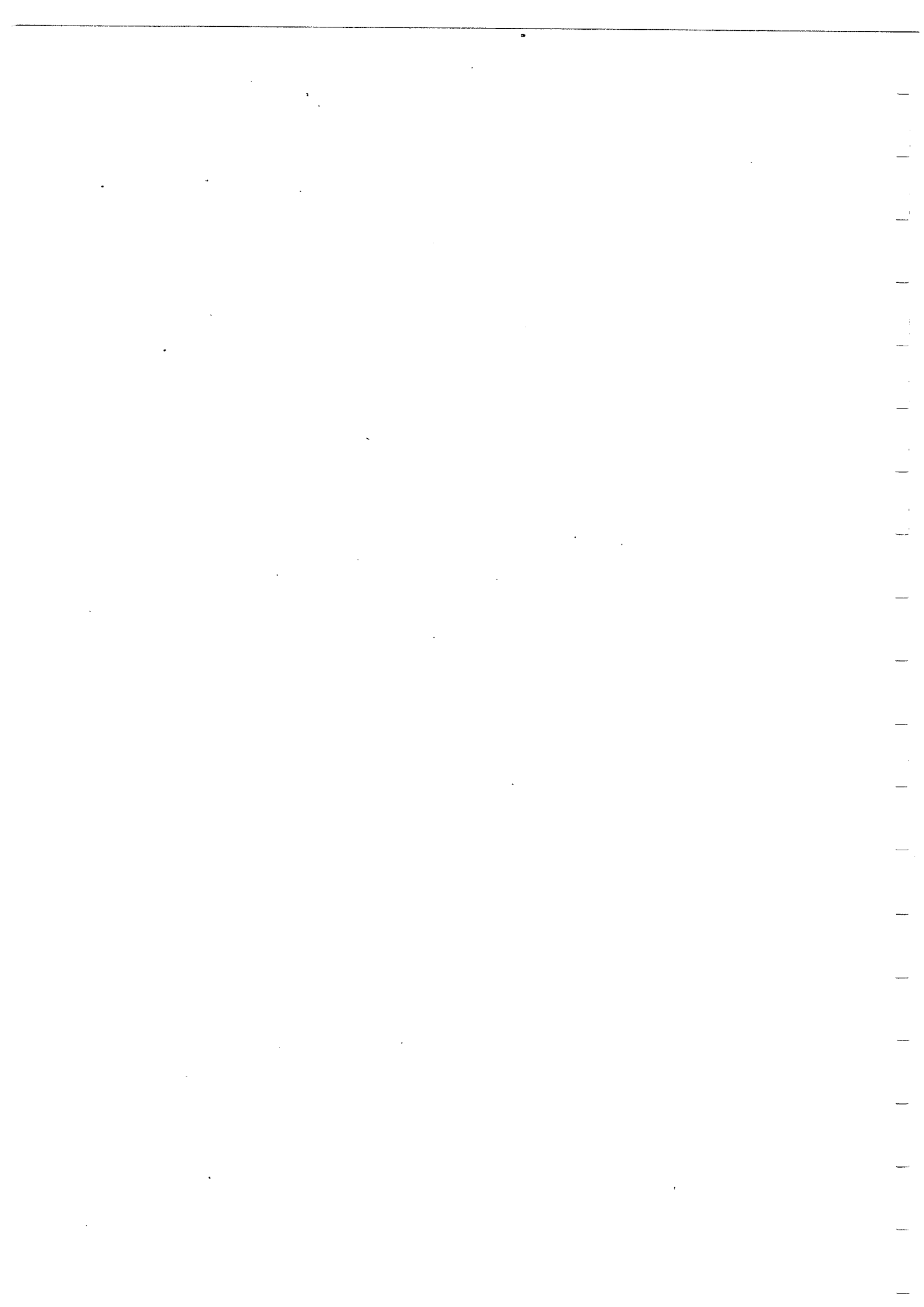
Tenemos, pues, el teorema:

Teorema 2.6 Los full-A.F.L. $p\text{-}\overline{\text{Root}(q)}$ y $p\text{-}\overline{\text{Log}}$ satisfacen

$p\text{-}\overline{\text{Root}(2)} \not\preceq p\text{-}\overline{\text{Root}(3)} \not\preceq \dots \not\preceq p\text{-}\overline{\text{Root}(q)} \not\preceq \dots \not\preceq p\text{-}\overline{\text{Log}}$



CAPITULO VI. COMPARACION ENTRE DISTINTAS MEDIDAS DE
COMPLEJIDAD.



En este último capítulo vamos a comparar entre sí las diferentes medidas de complejidad que hemos tratado anteriormente, es decir:

- . función de crecimiento
- . índice inicial
- . índice racional
- . a-medida
- . costo rígido
- . coste boleano

Para cada una de estas medidas, podemos definir la familia de los lenguajes polinomiales y la de los lenguajes exponenciales.

Diremos, en particular, que dos de estas medidas son incomparables, si los lenguajes polinomiales de una cortan los exponenciales de otra, y viceversa. En el teorema 1.1, condensamos las distintas informaciones respecto a la comparabilidad de dichas medidas.

En los capítulos II y V, hemos introducido dos otras medidas, para el caso de lenguajes algebraicos:

- . h-medida
- . p-medida

basadas en la altura del árbol de derivación y la altura de la pila de un autómata. En el teorema 2.1, damos una relación entre ambas medidas.

Para estas dos medidas tiene interés estudiar los lenguajes sublineales, el último lema de dicho capítulo prueba su existencia para la h-medida.

Recordemos rápidamente las distintas funciones de medida que hemos tratado hasta ahora para un lenguaje L .

La primera consiste en contar el número de palabras hasta una longitud dada, dicha medida ha sido introducida por Milnor [45] .

Definición 1.1 Sea L un lenguaje, su función de crecimiento g_L está dada por:

$$g_L(n) : n \mapsto \text{Card} \{ w \in L \mid |w| \leq n \}$$

Las dos medidas siguientes se fundan en la capacidad que tienen los autómatas finitos para describir distintas partes de L . La primera de ella, cuenta el número de estados de un autómata no determinista que reconoce $\{ w \in L \mid |w| \leq n \}$. Formalmente:

Definición 1.2 Sea L un lenguaje y $a_n, n \geq 1$ una sucesión de autómatas no deterministas que reconocen $nL = \{ w \in L \mid |w| \leq n \}$ con un número de estados mínimo.

Llamamos índice inicial μ_L a la función:

$$\mu_L(n) : n \mapsto \text{número de estados de } a_n.$$

La tercera medida se basa en la capacidad que tienen los autómatas finitos en describir mal un lenguaje. Dicha medida ha sido introducida por Boasson, Courcelle y Nivat en [8] :

Definición 1.3 Para un lenguaje L en X^* su índice racional ρ_L viene dado por:

$$\rho_L(n) = \max \{ \min \{ |w| \mid w \in L \cap K \} \mid L \cap K \neq \emptyset \text{ y } K \in \text{Rat}_n(X^*) \}$$

en que $\text{Rat}_n(X)$ es la familia de lenguajes reconocidos por un autómata no determinista con n estados como máximo.

Otra medida, introducida por Paredaens y Vyncke [47] consiste en

medir el comportamiento asintótico del autómata minimal que reconoce L (infinito si L no es racional).

Definición 1.4 Sea L un lenguaje de X^* y sea $\mathcal{A} = \langle X, Q, q_0, Q_+, \cdot \rangle$ el autómata determinista minimal que lo reconoce, llamaremos a-medida de L a la función:

$$a_L(n) = \text{Card } q_0 \cdot nX^*$$

Tras los autómatas finitos, podemos medir L con respecto a otros mecanismos. Uno de los métodos clásicos de medir un problema, consiste en considerar el costo de un programa rígido (straight-line) para resolverlo. Dicha idea ha sido aplicada a los lenguajes formales por Goodrich, Ladner y Fischer [31], formalmente:

Definición 1.5 Sea L un lenguaje de X^* , notemos $L_n = L \cap X^n$.

Sea β un programa rígido que calcula L_n a partir de $x \in X, \epsilon$, tomando como operaciones la unión y concatenación de lenguajes.

Llamaremos costo de β , $c(\beta)$ al número de operaciones de dicho programa. Llamamos costo rígido de L a la función:

$$uc_L(n) = \min \{ c(\beta) \mid \beta \text{ es un programa rígido que calcula } L_n \}$$

■

Cuando consideramos un lenguaje L en $\{0,1\}^n$, su función característica es una función booleana de n variables f_L . Tomemos como base (\wedge, \vee, \neg) podemos considerar un circuito que sintetiza, y definir su costo como su número de puertas.

Uno de los primeros en estudiar la relación entre L y f_L ha sido

Savage [50], definimos:

Definición 1.6 Sea L un lenguaje de $\{0,1\}^*$, notemos

$L_n = L \cap \{0,1\}^n$ y llamemos f_{L_n} a su función característica.

Sea $c(f_{L_n})$ el mínimo número de puertas necesarias para sintetizar f_{L_n} . Llamaremos coste booleano de L a la función:

$$b_L(n) = c(f_{L_n})$$

Para cada una de estas medidas podemos definir las familias de lenguajes polinomiales y exponenciales. Un primer intento de comparación entre las distintas medidas puede intentarse a través de dichas familias. Es lógico decir que dos medidas son muy distintas si ambas familias se entrecortan; formalmente:

Definición 1.7 Sean α, β dos medidas de complejidad.

. Diremos que α y β no son comparables, $\exists c$, si se cumple

$$\text{Pol}_\alpha \cap \overline{\text{Exp}_\beta} \neq \emptyset \text{ y } \overline{\text{Exp}_\alpha} \cap \text{Pol}_\beta \neq \emptyset$$

. Diremos que α y β son comparables, c , si se cumple:

$$\begin{aligned} \text{Pol}_\alpha \subset \text{Pol}_\beta \quad \text{ó} \quad \text{Pol}_\beta \subset \text{Pol}_\alpha \quad \text{ó} \\ \overline{\text{Exp}_\alpha} \subset \overline{\text{Exp}_\beta} \quad \text{ó} \quad \overline{\text{Exp}_\beta} \subset \overline{\text{Exp}_\alpha} \end{aligned}$$

Vamos a estudiar seguidamente la comparabilidad de las distintas medidas de complejidad que hasta ahora hemos encontrado.

Cuando nos restringimos a los lenguajes algebraicos, podemos definir dos nuevos tipos de medida, en función de gramáticas o autó-

matas a pila. La primera medida, introducida por Culik II y Maurer [15] se funda en la altura del árbol de derivación.

Definición 1.8 Diremos que un lenguaje L tiene h -medida $h_L(n)$, si existe una gramática G que engendra L , tal que toda palabra $\{ w \in L \mid |w| \leq n \}$ admite un árbol de derivación de altura acotada por $h_L(n)$.

La segunda medida de este tipo es muy parecida a la precedente, y se basa en los autómatas a pila.

Definición 1.9 Diremos que un lenguaje L tiene índice de pila $p_L(n)$ si existe un autómata a pila que lo engendra, tal que toda palabra de $\{ w \in L \mid |w| \leq n \}$ admite un cálculo, cuya altura de pila esta acotada por $p_L(n)$.

En estas dos últimas familias no tiene sentido definir las familias Pol y Exp. En vez de ellas, la familia Exp se halla substituida por los lenguajes estrictamente lineales, y la familia Pol por la familia de las raíces n -cismas.

Vamos seguidamente a comparar la función de crecimiento con las otras funciones de complejidad.

Lema 1.1 La función de crecimiento $g_L(n)$ y el índice racional ρ_L no son comparables.

Prueba

(a) Vamos a demostrar en primer lugar que $\text{Pol}_g \cap \overline{\text{Exp}}_\rho \neq \emptyset$.

Para ello consideremos el lenguaje:

$$L = \{ a^x b^{2^x} \mid x \geq 0 \}$$

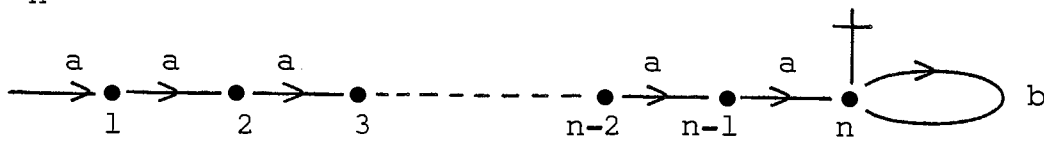
Vamos a estimar $g_L(n)$. Para ello estudiamos el cardinal de

$\{ a^x b^{2^x} \mid x + 2^x \leq n \}$. Para x grande $2^x \gg x$, con lo que podemos

substituir la condición precedente por $2^x \leq n$; llegamos a $x \leq \log n$. El cardinal del conjunto anterior puede, entonces, aproximarse por $\{a^x b^{2^x} \mid x \leq \log n\}$ y dicho conjunto tiene cardinal $\log n$. Llegamos a $g_L(n) = \Theta(\log n)$.

En consecuencia la función de crecimiento acotada por cualquier polinomio, con lo que $L \in \underline{\text{Pol}} \mu$.

Vamos ahora a estudiar el índice racional de L . Tomemos el autómata R_n .



Evidentemente $L(R_n) \cap L = a^{n-1} b^{2^{n-1}}$, con lo que $\rho_L(n) \geq n-1+2^{n-1}$.
 $\rho_L(n) = \Omega(2^n)$, con lo que $L \in \overline{\text{Exp}}$. Tenemos por lo tanto que $L \in \underline{\text{Pol}} \mu \cap \overline{\text{Exp}}$.

(b) Consideremos ahora el lenguaje $L' = \{w \tilde{w} \mid w \in \{a, b\}^*\}$

Hemos visto en el capítulo II que $\rho_{L'}(n) = \Theta(n^2)$. Tenemos por otra parte que $g_{L'}(n) = \sum_{0 \leq x \leq n/2} 2^x$, con lo que $L' \in \overline{\text{Exp}} \mu \cap \underline{\text{Pol}} \rho$.

Tenemos por lo tanto que las medidas g_L y ρ_L son incomparables.



Veamos seguidamente la relación entre la función de crecimiento $g_L(n)$ y el índice inicial $\mu_L(n)$.

Lema 1.2

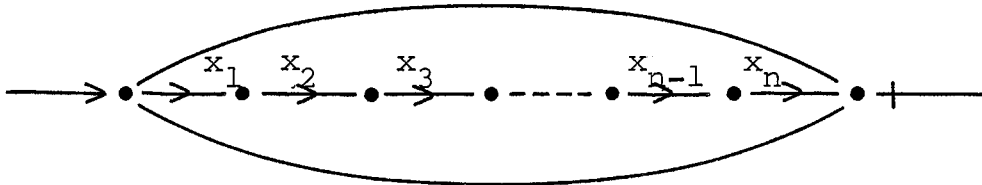
. Cuando un lenguaje tiene una función de crecimiento polinomial,

también lo es su índice inicial, formalmente $\underline{\text{Pol}}_g \subset \underline{\text{Pol}}_\mu$

. Cuando un lenguaje tiene un índice inicial exponencial, su función de crecimiento también lo es, formalmente $\overline{\text{Exp}}_\mu \subset \overline{\text{Exp}}_g$.

Prueba

. Consideremos el autómata siguiente:



con $x_1 x_2 x_3 \dots x_{n-1} x_n \in nL$. Dicho autómata tiene evidentemente un número de estados polinomial.

. El autómata que reconoce un mínimo de palabras con un máximo de estados para un lenguaje nL es de la forma dada en el dibujo precedente. El número de palabras admitirá como cota inferior $\frac{1}{n} \mu_L(n)$. Si $\mu_L(n) = 2^{\alpha n}$ entonces $\frac{1}{n} \mu_L(n) = \Omega(2^{\frac{\alpha}{2} n})$ con lo que $g_L(n)$ es aún exponencial.



Veamos que las inclusiones dadas precedentemente son estrictas.

Efectivamente $\underline{\text{Pol}}_g \not\subset \underline{\text{Pol}}_\mu$ y $\overline{\text{Exp}}_\mu \not\subset \overline{\text{Exp}}_g$, para ello consideremos $L = (a,b)^*$, se cumple $L \in \underline{\text{Pol}}_\mu \setminus \underline{\text{Pol}}_g$ y $L \in \overline{\text{Exp}}_g \setminus \overline{\text{Exp}}_\mu$.

Vamos seguidamente a comparar g_L con a_L , tenemos:

Lema 1.3 La función de crecimiento g_L y la a-medida a_L no son

comparables.

Prueba

(a) Consideremos el lenguaje, sea $X^* = \{a, b\}^*$

$$L = \{w \# \tilde{w} c^{2^{|w|}} \mid w \in \{a, b\}^*\}$$

Comencemos por estudiar su función de crecimiento; para ello vamos a estimar $L \cap X^n = \{w \# \tilde{w} c^{2^{|w|}} \mid 2|w| + 1 + 2^{|w|} = n\}$, cuando

n es grande $2|w| + 1 + 2^{|w|} \approx 2^{|w|} \approx n$, con lo que $|w| \approx \log_2 n$.

Podemos entonces poner $L \cap X^n \approx \{w \# \tilde{w} c^{2^{|w|}} \mid |w| = \log_2 n\}$.

Vamos ahora a contar cuantas palabras hay en $\{a, b\}^*$ de longitud $\log_2 n$, tenemos $\{a, b\}^{\log_2 n} = n$. Con lo que $\text{Card}(L \cap X^n) = \Theta(n)$;

llegamos a la conclusión que para n grandes $g_L(n) = \Theta(n^2)$.

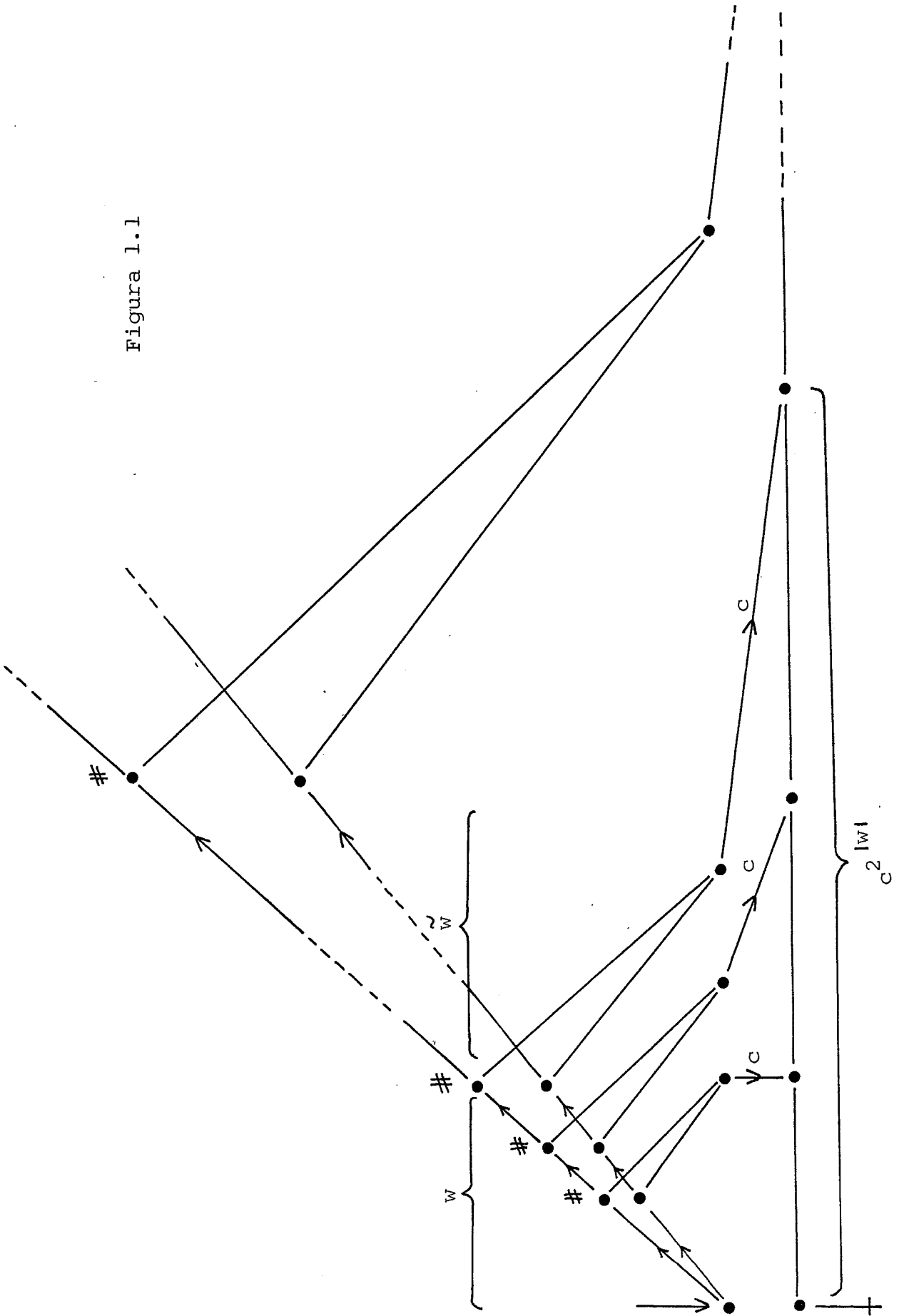
Vamos ahora a calcular la a -medida de L . El autómata minimal determinista que reconoce L puede esquematizarse en la figura 1.1.

Vemos que en este autómata, el comportamiento asintótico de $q_0 \cdot \{a, b, \#, c\}^n$ es análogo al comportamiento asintótico obtenido para el autómata minimal que reconoce $\{w \tilde{w} \mid w \in \{a, b\}^*\}$; el comportamiento de dicho autómata es del orden de $\Omega(2^n)$.

Hemos obtenido que $L \in \underline{\text{Pol}}_g \cap \overline{\text{Exp}}_a$.

(b) Consideremos $L' = \{a, b\}^*$. Evidentemente $g_{L'}(n) = 2^n$, y $a_{L'}(n) = 2$, con lo que $L' \in \overline{\text{Exp}}_g \cap \underline{\text{Pol}}_a$.

Figura 1.1



Vamos seguidamente a comparar la función de crecimiento con el costo rígido, tenemos:

Lema 1.4

- . Cuando un lenguaje tiene una función de crecimiento polinomial, su costo rígido también lo es; formalmente $\underline{\text{Pol}}_g \subset \underline{\text{Pol}}_{uc}$.
- . Cuando una función tiene un costo rígido exponencial, su función de crecimiento también lo es; formalmente $\overline{\text{Exp}}_{uc} \subset \overline{\text{Exp}}_g$.

Prueba

. Consideremos el programa rígido que construye las palabras una por una y letra a letra, y que va realizando la unión de cada nueva palabra construida. Evidentemente dicho programa realiza como máximo un número polinomial de operaciones.

. Sea L un lenguaje en X^* , sea $L_n = L \cap X^n$. El programa que con el máximo de operaciones realiza el mínimo de palabras, es el programa que acabamos de describir. Aún en este caso más desfavorable, el número de palabras generado es exponencial, dado que $g_L(n) = \Omega\left(\frac{1}{n} c_L^{uc}(n)\right)$.



Remarquemos que dado que $L = \{w \tilde{w} \mid w \in \{a,b\}^*\}$ satisface

$L \in \overline{\text{Exp}}_g$ y $L \in \underline{\text{Pol}}_{uc}$, tenemos que $\underline{\text{Pol}}_g \not\subset \underline{\text{Pol}}_{uc}$ y $\overline{\text{Exp}}_{uc} \not\subset \overline{\text{Exp}}_g$.

Vamos finalmente a comparar la función de crecimiento con el coste booleano, tenemos:

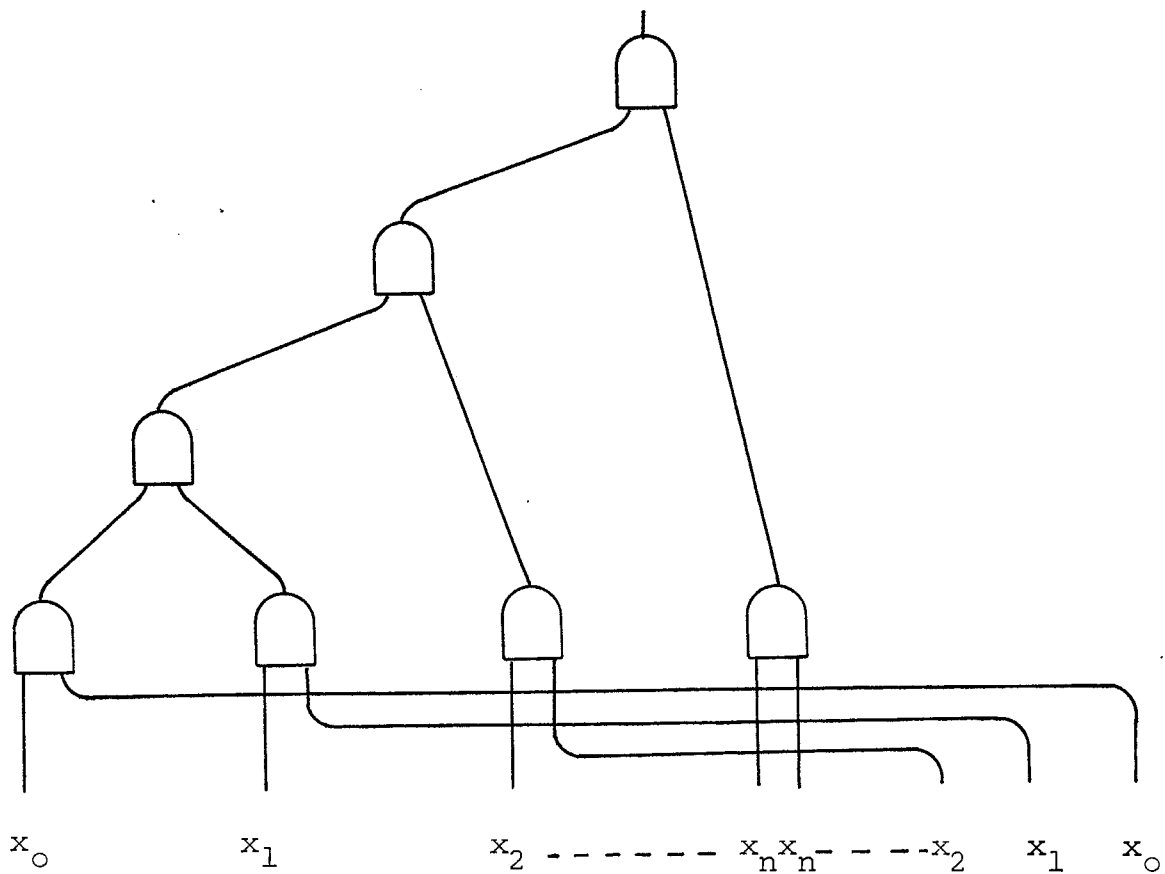
Lema 1.5 Los lenguajes contenidos en $\{0,1\}^*$ de función de creci-

miento polinomial, están estrictamente contenidos en los lenguajes de coste booleano polinomial.

Prueba

(a) Si $L \subseteq \{0,1\}^*$ crece de modo polinomial, lo mismo podemos decir de $L_n = L \cap \{0,1\}^n$. A cada palabra $x_0 x_1 x_2 \dots x_n$, $x_i \in \{0,1\}$ podemos asociarles un minterm y considerar la función booleana que corresponde a la suma de todos los minterms de L_n . Por construcción dicha función tiene un número de minterms polinomial.

(b) La inclusión es estricta, dado que $L = \{w \bar{w} \mid w \in \{0,1\}^*\}$, admite el circuito esquematizado en la figura siguiente.



Vamos seguidamente a comparar el índice inicial con las restantes funciones de complejidad. Comencemos con el índice racional; tenemos:

Lema 1.6 El índice inicial $\mu_L(n)$ y el índice racional $\rho_L(n)$ no son comparables.

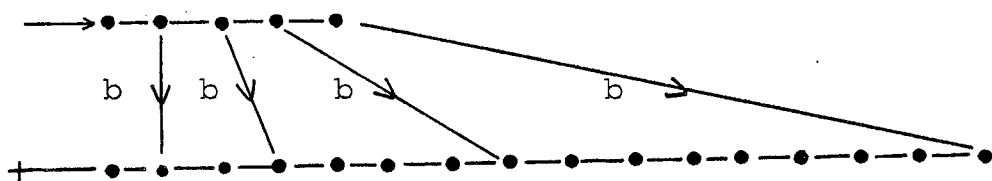
Prueba

(a) Tomemos en este caso el lenguaje:

$$L = \{ a^x b^{2^x} \mid x \geq 0 \}$$

que ya hemos encontrado al comparar g_L y ρ_L . Allí hemos visto que

$\rho_L(n) = \Omega(2^n)$. Por otra parte $\mu_L(n) = \Theta(n)$ dado que nL es aceptado por el autómata esquematizado en la figura siguiente:



tenemos, por tanto, que $L \in \overline{\text{Exp}}\rho \cap \underline{\text{Pol}}\mu$.

(b) Si consideramos el lenguaje $L' = \{ w \in \{a,b\}^* \mid w = \tilde{w} \}$, se cumple $\rho_{L'}(n) = \Theta(n^2)$ y $\mu_{L'}(n) = \Theta(2^{n/2})$, con lo que:

$$L' \in \underline{\text{Pol}}\rho \cap \overline{\text{Exp}}\mu$$



El ejemplo siguiente, debido a Deleage [17], nos muestra una cadena infinita de lenguajes de índice inicial y racional polinomial. Vemos que mientras el grado del polinomio del índice inicial, permanece estacionario, el grado del índice racional crece estrictamente.

Ejemplo 1.1 La sucesión de lenguajes

$$L_p = \{a_1^x a_2^x \dots a_p^x \mid x \in \mathbb{N}_+, 1 \leq i \leq p\}, p \geq 2$$

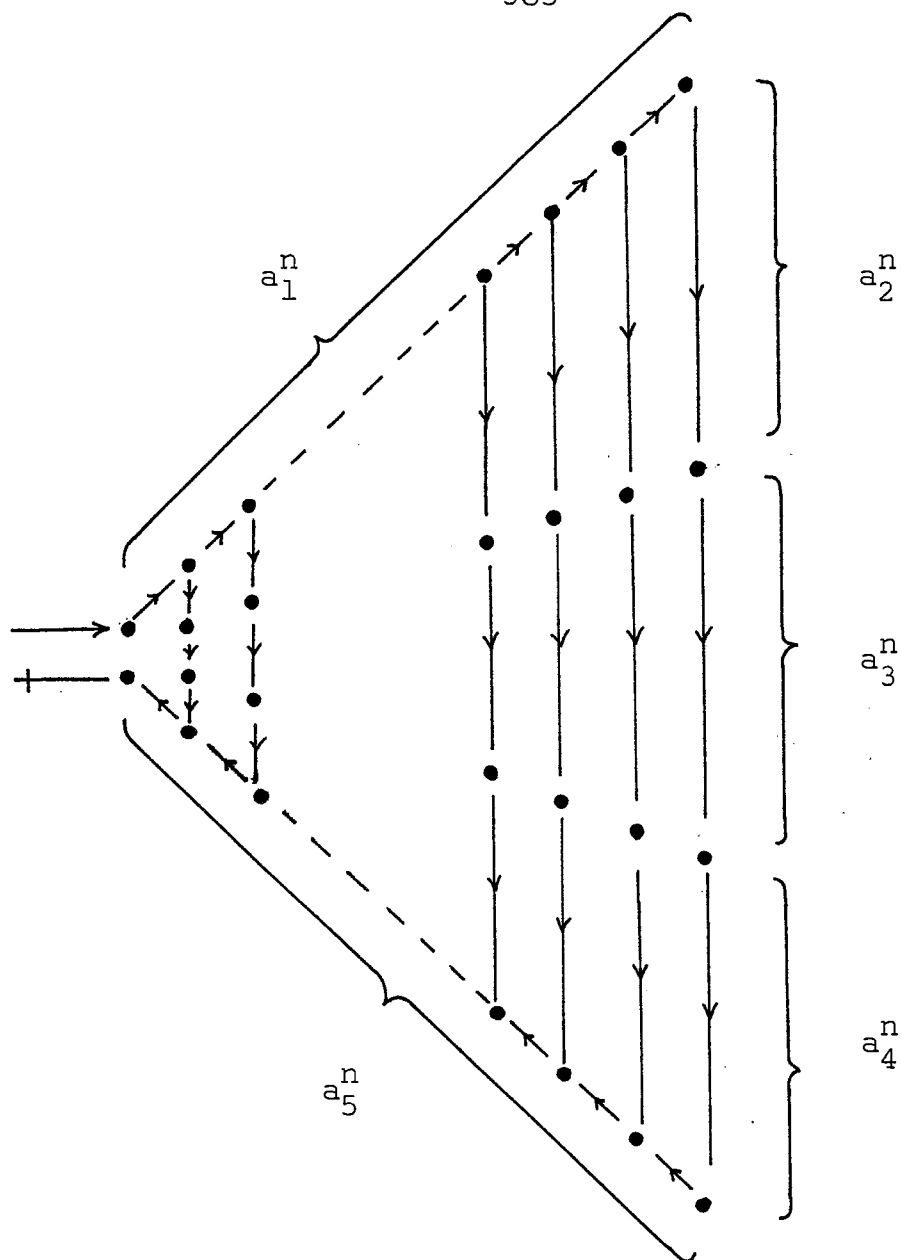
satisface $\mu_{L_p}(n) = \Theta(n^2)$ y $\rho_{L_p}(n) = \Theta(n^p)$

Prueba

(a) Comencemos por analizar el índice inicial. En el estudio de los segmentos iniciales, resulta cómodo limitarse a los múltiplos de p , es decir

$$pnL_p = \{a_1^x a_2^x \dots a_p^x \mid x \leq n\}$$

En la figura siguiente se halla esquematizado un autómata para $p = 5$.



Evidentemente dicho autómata satisface $\mu_{L_5}(n) = \Theta(n^2)$, dicha construcción es generalizable sin problemas, obteniendo entonces

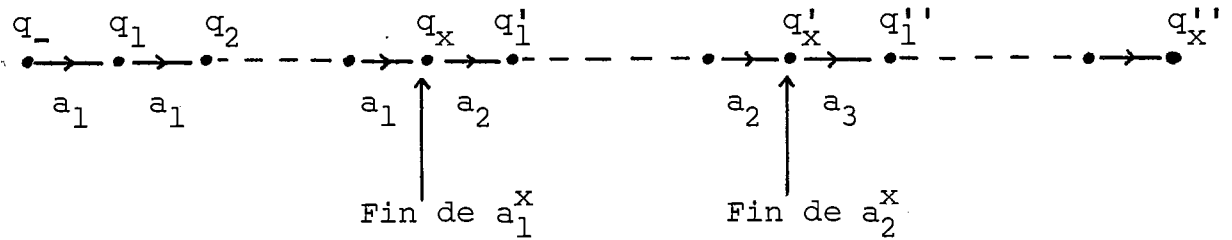
$$\mu_{L_p}(n) = \Theta(n^2).$$

(b) Vamos ahora a estudiar el índice racional de L_p . En un primer tiempo vamos a analizar el caso $p = 3$. Mostraremos luego, como pue-

den generalizarse las construcciones.

Comenzaremos por hallar un límite superior al índice racional de L_3 . Sea $K \in \text{Rat}_n \{a_1 \dots a_3\}^*$ con $L_3 \cap K \neq \emptyset$.

Vamos a especificar el estado en que queda el autómata K tras la lectura de cada letra. Sea $w = a_1^x a_2^x a_3^x \in K$, tenemos:

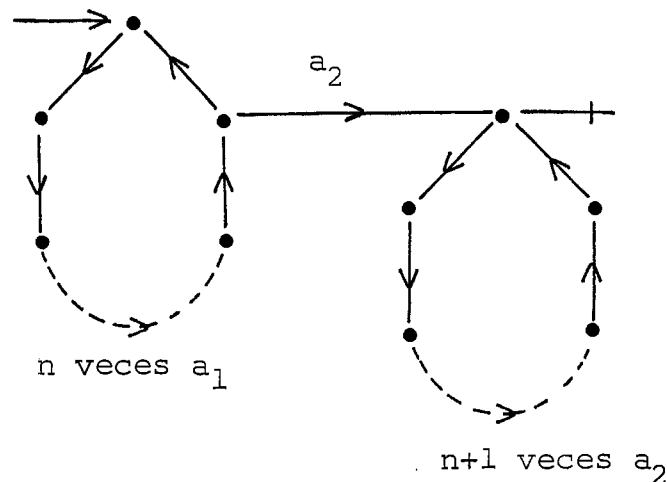


Consideremos ahora los triples (q_1, q'_1, q''_1) , (q_2, q'_2, q''_2) , ..., $(q_{x-1}, q'_{x-1}, q''_{x-1})$, hay como máximo n^3 triples distintos, con lo que aplicando el lema de Dirichlet, si hay más de n^3 triples distintos deben haber al menos dos de iguales. Dicha construcción nos da

$\rho_{L_3}(n) = O(n^3)$. Si en vez de considerar tripletes de estados, consideramos p -upletes, obtendremos $\rho_{L_p}(n) = O(n^p)$.

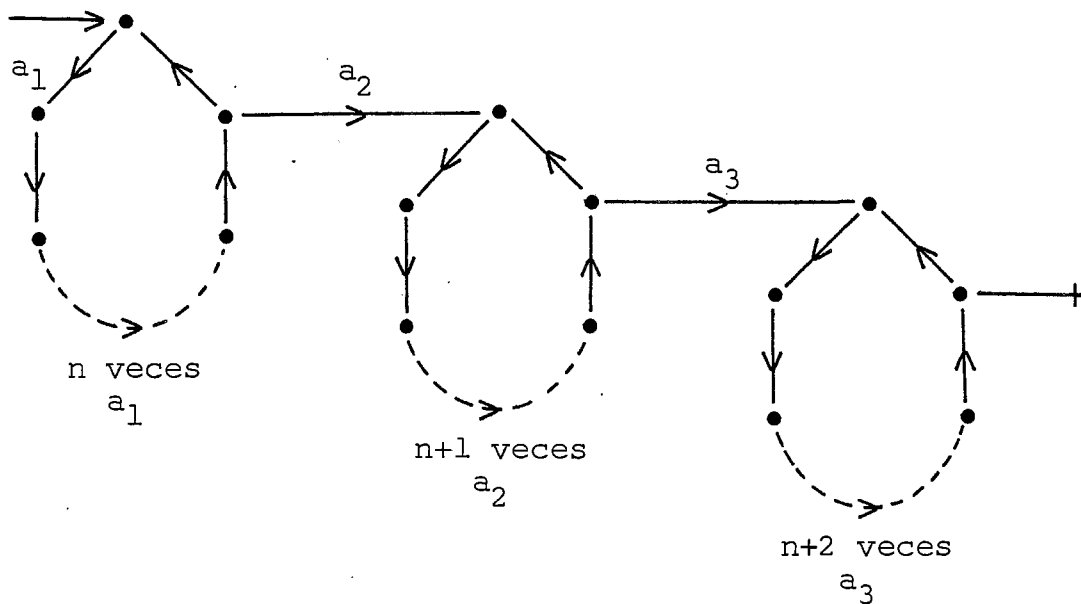
Vamos seguidamente a estudiar el límite inferior para ρ_{L_p} .

Para $p = 2$, tenemos que dos enteros consecutivos son primos entre sí. Consideremos, entonces, el autómata K_2 siguiente:



La palabra más corta de $L_2 \cap K_2$ es $a_1^{n(n+1)} a_2^{(n+1)n}$, con lo que
 $\rho_{L_2}(n) = \Omega(n^2)$.

Tratemos $p = 3$. Cuando n es impar, los números $n, n + 1$ y $n + 2$, son primos entre sí. Consideremos, entonces, la generalización del autómata precedentemente. Sea K_3 el siguiente autómata:



La palabra más corta de $L_3 \cap K_3$ es:

$a_1^{n(n+1)(n+2)} a_2^{n(n+1)(n+2)} a_3^{n(n+1)(n+2)}$
 con lo que $\rho_{L_3}(n) = \Omega(n^3)$.

Vamos a tratar seguidamente el caso k cualquiera. Para ello es necesario encontrar k enteros que dependan linealmente de n y primos dos a dos. Notemos:

$p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 5, \dots, p_k = \text{el } k-1 \text{ número}$

primo,...

Sea $\pi = p_1 p_2 \dots p_k$, definimos:

$$n_1 = n\pi + p_1, n_2 = n\pi + p_2, \dots, n_k = n\pi + p_k$$

Tenemos que dichos números son primos entre sí. La generalización del autómata definido precedentemente nos permite mostrar que

$$\rho_{L_k}(n) = \Omega(n^k).$$



Sabemos que $\text{Gre} \not\subseteq \text{Pol}$ [24], por otra parte en el capítulo IV, hemos visto que $\text{Fcl} \subseteq \text{Pol} \mu$. Considerando $\{w \in \{a,b\}^* \mid w = \tilde{w}\}$ vemos que $\text{Fcl} \not\subseteq \text{Gre}$ [6]. Estas consideraciones nos inducen a hacer la conjetura:

Conjetura 1.1 Todo lenguaje algebraico de índice inicial polinomial tiene también índice racional polinomial.

Vamos seguidamente a considerar conjuntamente índice inicial y a-medida.

Lema 1.7 El índice inicial $\mu_L(n)$ y la a-medida $a_L(n)$ satisfacen:

$$\mu_L(n) \leq (n+1) a_L(n)$$

Prueba Consideremos el autómata A_L minimal, infinito si L no es reconocible, que reconoce $L \subset X^*$. Sea \mathcal{H}_n el autómata con $(n+1)$ estados que reconoce nX^* . La intersección $A_L \cap \mathcal{H}_n = A_n$ reconoce nL . El número de estados de A_n , está acotado por $(n+1)a_L(n)$.



Remarca 1.1 El comportamiento obtenido en el lema precedente es óptimo. Consideremos D_1^* , tenemos que $a_{D_1^*}(n) = \Theta(n)$ y que

$$\mu_{D_1^*}(n) = \Theta(n^2).$$

A partir del lema precedente, obtenemos el corolario:

Corolario 1.1 Los lenguajes cuya a-medida es polinomial, están estrictamente incluidos en los lenguajes de índice inicial polinomial.

Prueba Cuando $a_L(n) = O(n^k)$ tenemos que $(n+1) \sum_{j=0}^n a_L(j) = O(n^{k+2})$, con lo que $\mu_L(n) = O(n^{k+2})$.

A fin de demostrar la inclusión estricta, consideremos el lenguaje:

$$L = \{ f \# g \mid f \neq \bar{g}, f, g \in \{a, b\}^n \}$$

Dicho lenguaje tiene un índice inicial acotado por n^3 y una a-medida en $2^{\frac{n}{2}}$.

Podemos también considerar el lenguaje:

$$L = \{ w \# \bar{w} c^{2^{|w|}} \mid w \in \{x_1, x_2, \dots, x_n\}^* \}$$

tiene un índice inicial lineal y una a-medida exponencial.

Vamos a estudiar seguidamente la relación entre el índice inicial y la uc-medida, tenemos:

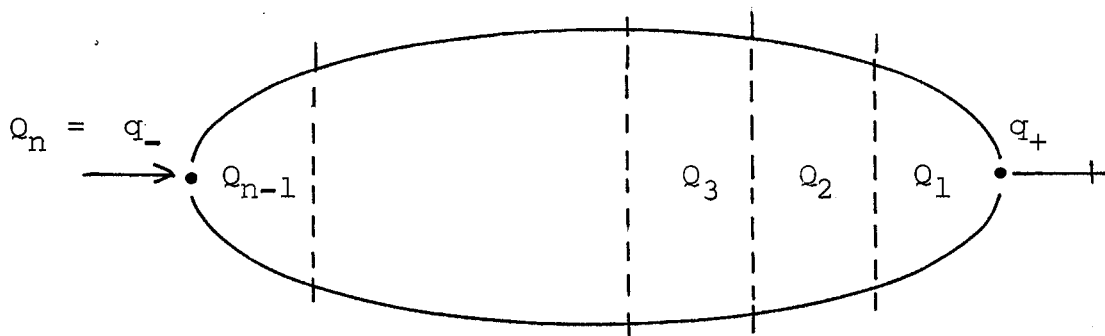
Lema 1.8 Sea $L \subset X^*$, existe un k tal que el índice inicial y la

uc-medida satisfacen:

$$uc_L(n) < k \cdot \mu_L^2(n)$$

Prueba Sea \mathcal{A} un autómata pequeño para nL .

A partir de dicho autómata podemos extraer un autómata \mathcal{A}' que acepta solamente $L_n = nL \cap X^n$. Notemos que \mathcal{A} no tiene bucles. Podemos, entonces, considerar \mathcal{A} como un grafo orientado que va de izquierda a derecha, tomando el estado inicial más a la izquierda (si hay más de uno pueden confundirse) y el estado final más a la derecha obtenemos \mathcal{A}' . El autómata \mathcal{A}' puede esquematizarse en:



En el que Q_1 contiene todos los estados a "distancia 1" de q_+ (de cualquier estado de Q_1 , leyendo "una" letra accedemos a q_+). Q_2 contiene todos los estados a distancia 2, y así sucesivamente. A partir de \mathcal{A}' podemos construir el programa rígido siguiente:

principio

$q_- \leftarrow \epsilon$

para $q \neq q_-$ hacer $q \leftarrow \emptyset$

para $i = 1$ a n hacer

para $q \in Q_i$ hacer

$$q \leftarrow \sum (x_1 + x_2 + \dots + x_r) \cdot q'$$

$$(q', xq) \in T$$

fin

Vamos a estimar seguidamente el número de operaciones de dicho programa. Supongamos $q \in Q_i$ en $q \leftarrow \sum (x_1 + \dots + x_r) \cdot q'$ hay como máximo $(\|X\| - 1) \cdot \|Q_{i-1}\|$ adiciones y como máximo $\|Q_{i-1}\|$ multiplicaciones. Sobre cada $q \in Q_i$, se realizan como máximo $\|X\| \cdot \|Q_{i-1}\|$ operaciones. Dado que hay como máximo $\|Q_i\|$ elementos q , el último bucle realiza como máximo $\|Q_i\| \cdot \|X\| \cdot \|Q_{i-1}\|$ operaciones.

El número total de operaciones es como máximo:

$$\|X\| \sum_{1 \leq i \leq n} \|Q_i\| \cdot \|Q_{i-1}\|$$

dado que $\sum \|Q_i\| \cdot \|Q_{i-1}\| \leq (\sum \|Q_i\|) (\sum \|Q_{i-1}\|) = \mu_L^2(n)$ obtenemos el resultado.

Dicho lema admite como corolario que:

Corolario 1.2 Todo lenguaje de índice inicial polinomial, admite un programa uc-rígido también polinomial. Formalmente $\text{Pol}_\mu \subset \text{Pol}_{uc}$. Dicha inclusión es estricta.

Prueba Hemos visto que $\{w w \mid w \in \{a, b\}^*\}$ es un lenguaje de índice inicial polinomial, pero su uc-medida no lo es.

Vamos, finalmente, a comparar el índice inicial y el coste booleano.

Tenemos:

Lema 1.9 Sea $L \subset X^* = \{0,1\}^*$, existe k tal que el índice inicial y el coste booleano satisfacen:

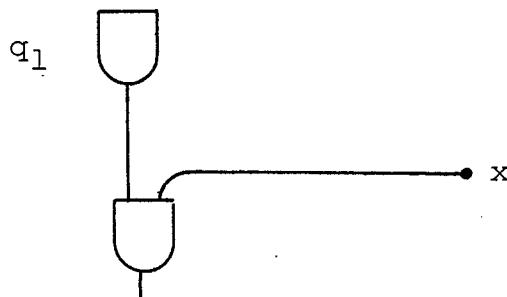
$$b_L(n) \leq k \cdot \mu_L^2(n)$$

Prueba Comencemos por extraer a partir de un autómata pequeño \mathcal{A} que reconoce nL , un autómata \mathcal{A}' que reconozca $L_n = nL \cap X^n$. En el lema precedente, hemos visto que esto puede hacerse sin aumentar el número de estados.

La idea de la demostración, consiste en asociar a cada estado, una puerta lógica que será \wedge ó \vee según el número de arcos que llegen a dicho estado.

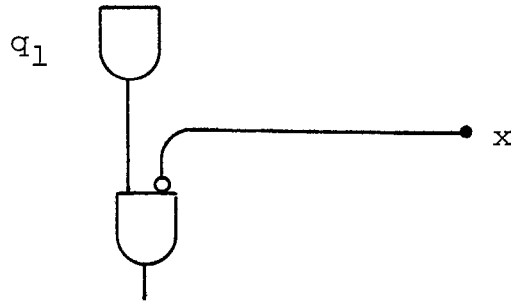
Asociamos a q_1 una puerta \wedge , cuyas dos entradas están siempre a 1. (De hecho podemos eliminar dicha puerta si queremos).

Sea T la función de transición de \mathcal{A}' . Si $(q_1, x, q_2) \in T$ con $x = 1$, añadimos a partir de la puerta \wedge que corresponde a q_1 , el circuito:



Si al estado q_2 no llega ningún otro arco, la nueva puerta \wedge es la puerta que corresponde a q_2 .

Si tenemos el arco $(q_1, x, q_2) \in T$ con $x = 0$ añadimos:

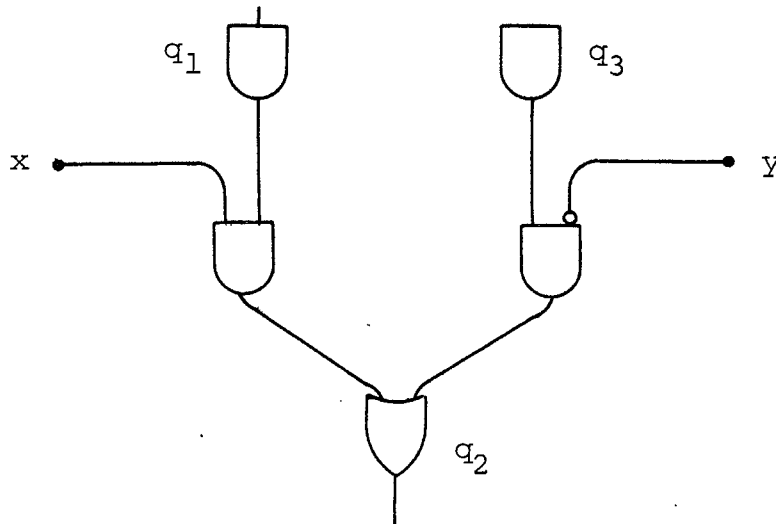


Si a q_2 no llega ningún otro arco, la puerta \wedge que acabamos de añadir es la puerta que corresponde a q_2 .

Supongamos que a q_2 llegan varios arcos. Tomemos como ejemplo

(q_1, x, q_2) y (q_3, y, q_2) con $x = 1, y = 0$.

Realizamos entonces las construcciones dadas precedentemente, pero reunimos las 2 puertas \wedge con una puerta \vee , y dicha puerta \vee será la que corresponde a q_2 , esquemáticamente tenemos:



En el caso general reunimos los distintos arcos que llegan a un estado, utilizando puertas \vee . Con este procedimiento obtenemos una

sola salida. La puerta que nos da dicha salida, es a la que damos el nombre del estado.

Vamos a estimar, seguidamente, el número de puertas lógicas en función del índice inicial.

Estimemos el número de puertas \wedge . Dos estados distintos tienen como máximo $\|X\|$ letras que los conectan. Dado que hay como máximo $\mu_L(n)$, a un estado llegan como máximo $\|X\| \mu_L(n)$ arcos, con lo que el primer tipo de construcción nos da $\|X\| \mu_L(n)$ puertas \wedge como máximo. Si todas las variables son 0, debemos añadir

$\|X\| \mu_L(n)$ puertas \neg . Reunir estos arcos con puertas \wedge puede costar, como máximo, $\|X\| \mu_L(n)$. Con lo que cada estado puede costarnos como máximo $3 \|X\| \mu_L(n)$.

Dado que hay $\mu_L(n)$ estados distintos, el número total de puertas será siempre menor a $3 \|X\| \mu_L^2(n)$.



Ejemplo 1.2 Vamos a tratar el caso de los palíndromes:

$$PAL_2 = \{ w \tilde{w} \mid w \in \{0,1\}^* \}$$

Damos en la figura siguiente un autómata pequeño para $PAL_2 \cap \{0,1\}^n$ con $n = 6$.

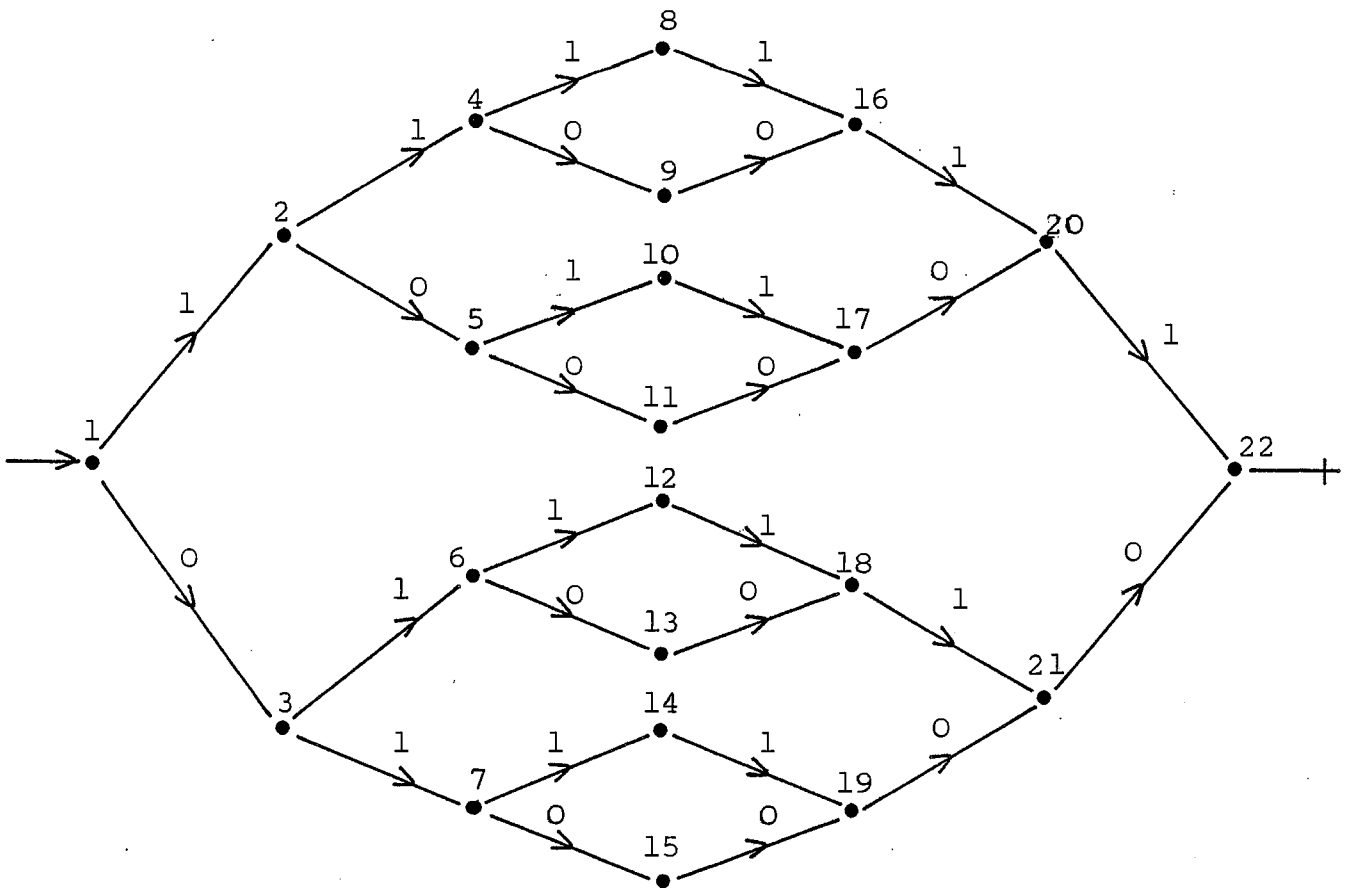
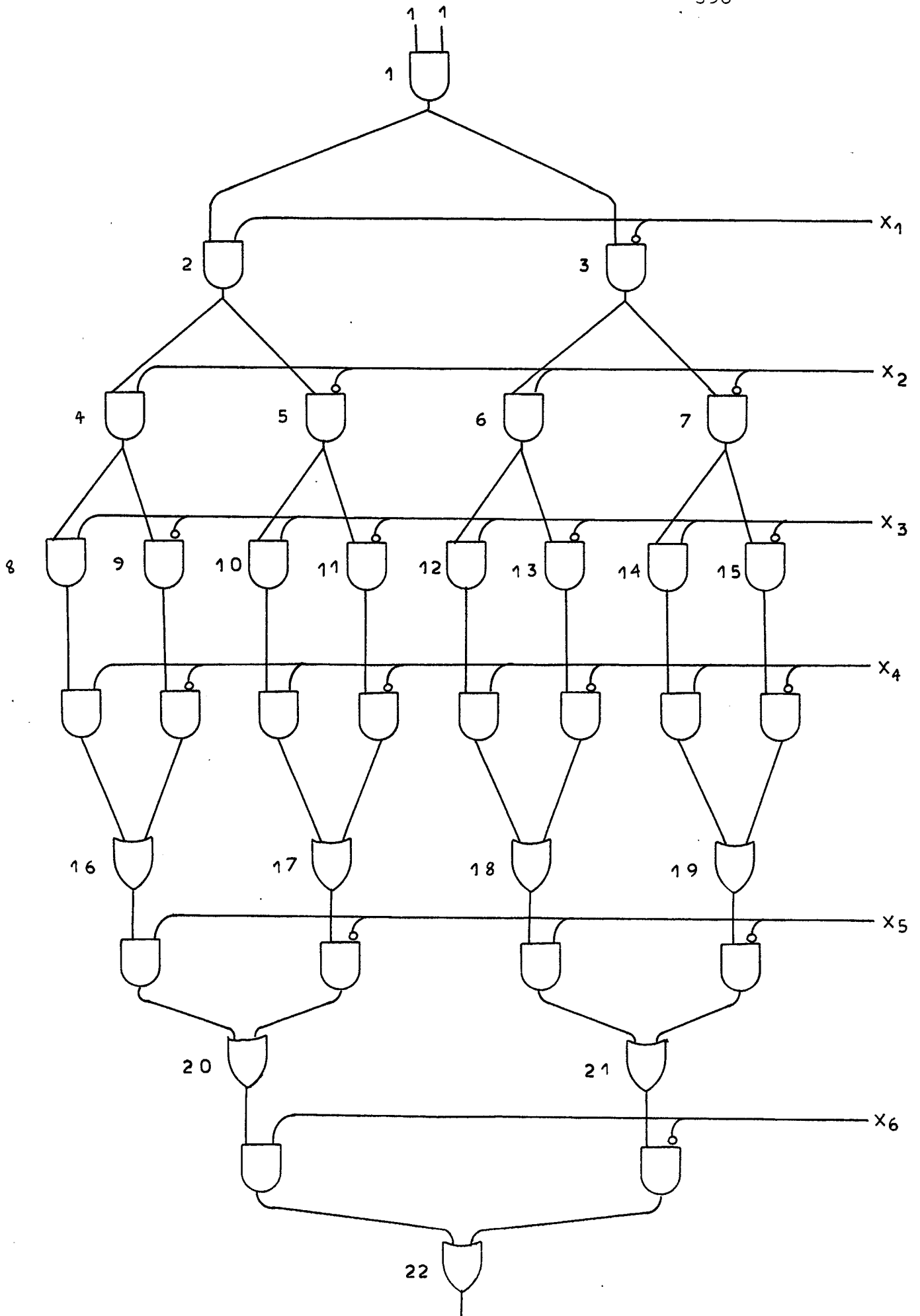


Figura 1.2.

Utilizamos dicho autómata para construir un circuito boleano que reconozca la función característica, según el lema indicado precedentemente, figura 1.3.

Vemos que dicho autómata simula el autómata dado precedentemente, y, en consecuencia, tiene un número exponencial de estados. Notemos que en el capítulo 2 tiene una uc-medida lineal. La construcción del lema precedente no da, al menos en este caso, una buena aproximación.



El lema y el ejemplo precedente nos conducen a:

Corolario 1.3. Todo lenguaje sobre $\{0,1\}^*$ con índice inicial, tiene una función característica sintetizable, con un número polinomial de puertas. Formalmente $\underline{\text{Pol}}_\mu \not\subseteq \underline{\text{Pol}}_b$.

Vamos seguidamente a estudiar el índice racional en relación a las otras medidas aún no estudiadas.

Comencemos por la a-medida.

Lema 1.10. El índice racional ρ_L y la a-medida a_L no son comparables.

Prueba

(a) Si consideramos el lenguaje:

$$L = \{ w \# \bar{w} \mid w \in \{a,b\}^* \}$$

tenemos que $L \in \underline{\text{Pol}}_\rho \cap \overline{\text{Exp}}_a$

(b) Si consideramos el lenguaje:

$$L = \{ 0^x 1^{2^x} \mid x \geq 0 \}$$

tenemos que $L \in \overline{\text{Exp}}_\rho \cap \underline{\text{Pol}}_a$

Vamos a comparar el índice racional con la uc-medida.

Lema 1.11 El índice racional ρ_L y la uc-medida uc_L no son comparables.

Prueba

(a) Consideremos el lenguaje:

$$\text{COPY} = \{w w \mid w \in \{0,1\}^*\}$$

dicho lenguaje satisface $\text{COPY} = \underline{\text{Pol}}_\rho \cap \overline{\text{Exp}}_{uc}$

(b) Consideremos el lenguaje:

$$D_2^* = \langle S \rightarrow aS\bar{a}S + bS\bar{b}S + \epsilon \rangle$$

dicho lenguaje satisface:

$$D_2^* \in \overline{\text{Exp}}_\rho \cap \underline{\text{Pol}}_{uc}$$

Notemos que el lenguaje $L = a^*$ satisface:

$$L \in \underline{\text{Log}}_{uc} \cap \underline{\text{Pol}}_\rho$$

En particular la cadena de lenguajes:

$$L_k = \{a_1^n a_2^n \dots a_k^n \mid n \geq 1\}$$

de índice racional polinomial estrictamente creciente, colapsa en lenguajes cuyo programa rígido uc es logarítmico.

Respecto al índice racional y al coste booleano, podemos decir que

$$L = \{0^x 1^{2^x} \mid x \geq 1\}$$

Satisface $L \in \overline{\text{Exp}}_\rho \cap \underline{\text{Pol}}_b$.

Vamos seguidamente a analizar la a-medida con relación a las dos restantes medidas de complejidad. Tenemos:

Lema 1.12 La a-medida, satisface con relación a la uc-medida y a la b-medida:

$$\begin{aligned}uc_L(n) &\leq k_1(n+1)^2 \cdot a_L^2(n) \\ b_L(n) &\leq k_2(n+1)^2 \cdot a_L^2(n)\end{aligned}$$

Prueba Si hacemos la intersección del autómata minimal que reconoce L con el autómata que reconoce las palabras de longitud n de L , obtenemos un autómata con $(n+1) \cdot a_L(n)$ estados como máximo. Aplicando las construcciones dadas para la uc -medida y la b -medida, obtenemos el resultado. ■

Considerando $COPY_2$ y el lema precedente, tenemos:

Corolario 1.4 Los lenguajes cuya a -medida es el polinomial, tienen uc -medida y b -medida (cuando ésta está definida) polinomial, y la inclusión es estricta. ■

Damos finalmente un teorema debido a Goodrich, Ladner y Fischer, que nos relaciona la uci -medida y la b -medida; tenemos:

Lema [31] 1.13 Sea $L \subset \{0,1\}^*$, existe una constante k que cumple:

$$uci_L(n) \leq k \cdot (b_L(n) + n)$$

Prueba Vamos a construir un programa uci , que simula el circuito boleano $\beta = (\beta_1, \beta_2, \dots, \beta_k)$, $k = \text{longitud}(\beta)$.

principio

para $i = 1$ a n hacer

$$x_i \leftarrow \{0,1\}^{i-1} . 1 . \{0,1\}^{n-i-1} ;$$

$$\bar{x}_i \leftarrow \{0,1\}^{i-1} . 0 . \{0,1\}^{n-i-1} ;$$

para $j = 1$ a longitud (β) hacer

principio

$$\underline{\text{si}} \beta_j = x_i \underline{\text{entonces}} \hat{\beta}_j \leftarrow x_i ;$$

$$\underline{\text{si}} \beta_j = \neg x_i \underline{\text{entonces}} \hat{\beta}_j \leftarrow \bar{x}_i ;$$

$$\underline{\text{si}} \beta_j = (\vee ; \beta_p ; \beta_q) \underline{\text{entonces}} \hat{\beta}_j \leftarrow \hat{\beta}_p + \hat{\beta}_q ;$$

$$\underline{\text{si}} \beta_j = (\wedge ; \beta_p ; \beta_q) \underline{\text{entonces}} \hat{\beta}_j \leftarrow \hat{\beta}_p \cap \hat{\beta}_q ;$$

fin

Obtenemos así un programa uci $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_k)$ que simula β .

El cuadro siguiente nos da la relación entre las distintas familias definidas con la ayuda de las medidas de complejidad, tenemos:

Teorema 1.1 La relación entre las distintas medidas de complejidad viene dada por:

μ_L	ρ_L	a_L	uc_L	b_L , lenguajes en $\{0,1\}^*$	g_L
$\mu_L(n) \leq (n+1)g_L(n)$ $\overline{Pol}_g \not\subseteq \overline{Pol}\mu$ $\overline{Exp}\mu \not\subseteq \overline{Exp}g$	no comparable	no comparable	$\overline{Pol}_g \not\subseteq \overline{Pol}_{uc}$ $\overline{Exp}_{uc} \not\subseteq \overline{Exp}_g$	$\overline{Pol}_g \not\subseteq \overline{Pol}_b$	g_L
$\mu_L(n) \leq (n+1)a_L(n)$ $\overline{Pol}_a \not\subseteq \overline{Pol}\mu$	no comparable	$\mu_L(n) \leq (n+1)a_L(n)$ $\overline{Pol}_a \not\subseteq \overline{Pol}\mu$	$uc_L(n) \leq k \cdot \mu_L^2(n)$ $\overline{Pol}\mu \not\subseteq \overline{Pol}_{uc}$	$b_L(n) \leq k \cdot \mu_L^2(n)$ $\overline{Pol}\mu \not\subseteq \overline{Pol}_b$	μ_L
		no comparable	no comparable	$\overline{Exp}\rho \cap \overline{Pol}_b \neq \emptyset$	ρ_L
			$uc_L(n) \leq k(n+1)^2 a_L^2(n)$ $\overline{Pol}_a \not\subseteq \overline{Pol}_{uc}$	$b_L(n) \leq k(n+1)^2 a_L^2(n)$ $\overline{Pol}_a \not\subseteq \overline{Pol}_b$	a_L
				$\overline{Exp}_{uc} \cap \overline{Pol}_b \neq \emptyset$	uc_L



Vamos a estudiar finalmente la relación entre la altura de la pila y la altura del árbol de derivación en el caso de los lenguajes algebraicos, es decir, la relación entre la h-medida y la p-medida; tenemos:

Teorema 2.1 Sea L un lenguaje algebraico, existe $k > 0$ que cumple:

$$p_L(n) \leq k \cdot h_L(n)$$

Prueba Sea $G = \langle X, V, P, S \rangle$ una gramática para L que realiza $h_L(n)$. Vamos a construir a partir de dicha gramática un autómata a pila y estudiar su comportamiento.

Sea $A = \langle X, Z, Q, z_0, q_0, \delta \rangle$ el autómata a pila asociado canónicamente a la gramática por:

$$Z = X + V ; Q = \{q\} ; z_0 = S ; q_0 = q$$

las reglas están dadas por:

$$(q, \epsilon, v) \vdash (q, \alpha) \text{ si } v \rightarrow \tilde{\alpha} \text{ es una regla de } P$$

$$(q, x, x) \vdash (q, \epsilon) \text{ si } x \in X$$

Dicho autómata recorre el árbol de derivación "top-down". Se cumple que $L = \text{Null}(A)$.

Indiquemos esquemáticamente en la figura 1.4, cuál es la situación en un momento dado del cálculo tras haber leído el prefijo w, la configuración a la que hemos llegado es:

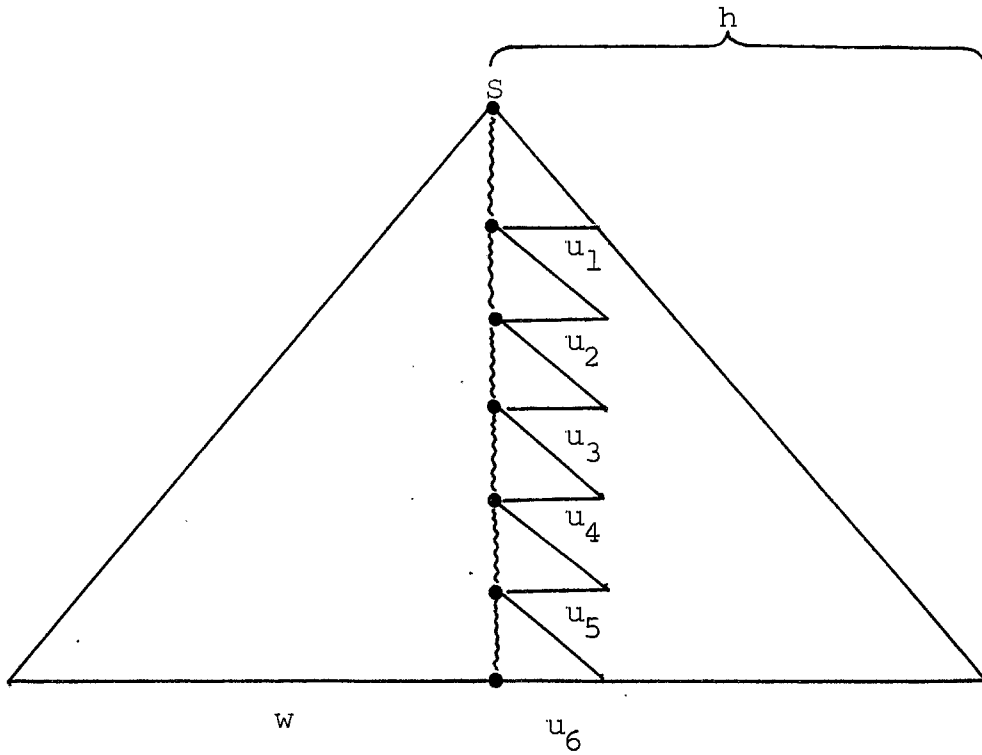


Figura 1.4.

$(q, wh, s) \xrightarrow{*} (q, h, \tilde{u}_1 \tilde{u}_2 \tilde{u}_3 \tilde{u}_4 \tilde{u}_5 \tilde{u}_6)$. La pila más alta se producirá cuando aparecen en el árbol caminos de la forma figura 1.5., en que básicamente se emplea toda la parte derecha de la regla.

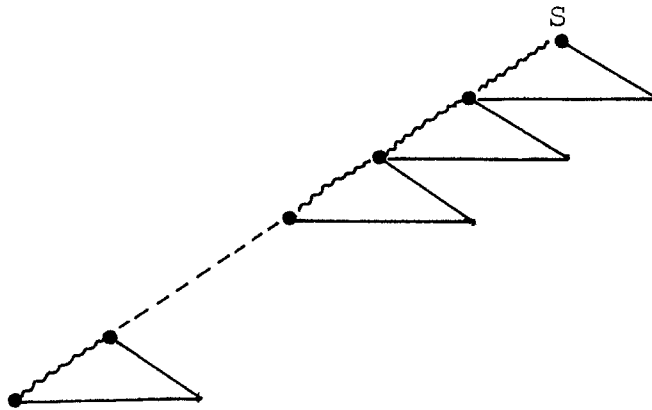


Figura 1.5.

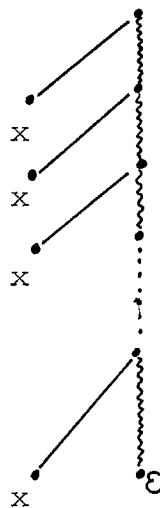
En este caso tendremos que la altura de la pila viene acotada por k veces la longitud del camino, siendo k

$$k = \max \{ |\alpha| \mid v \rightarrow \alpha \in P \}$$

Si queremos acotar la altura del árbol de derivación en función de la altura de pila, el procedimiento sería construir una gramática que realice un árbol de derivación cuya altura esté acotada por una función de p_L . Dicho procedimiento no funciona, como vemos seguidamente.

Sea R un lenguaje racional. Cualquier autómata $\mathcal{A} = \langle X, Q, q_0, Q_+, \cdot \rangle$ que lo acepte, tiene una altura de pila constantemente igual a 1, con lo que $p_R(n) = 1$.

Si consideramos la gramática lineal derecha asociada a dicho autómata, genera árboles tipo:



con lo que la gramática tiene una altura lineal en función de la palabra. Vemos pues que hemos obtenido el caso más desfavorable posible. En el capítulo II, hemos visto que $h_R(n) = \log_2 n$.

Este ejemplo tan simple, nos muestra que el procedimiento dado por Ginsburg, que permite construir a partir de un autómata a pila una gramática; no es de utilidad aquí, es un problema abierto hallar una cota de $h_L(n)$ en función de $p_L(n)$.

En el capítulo II, hemos suscitado el problema de la existencia de lenguajes L , tales que $h_L(n)$ sea una función sublineal de la altura. El lema siguiente, da una respuesta afirmativa a este problema.

Lema 1.14. El lenguaje de Goldstine:

$$\text{Gold}_{\neq} = \{ a^{i_1} b a^{i_2} b \dots a^{i_j} b \dots a^{i_k} b \mid k > 0, \exists j \text{ con } i_j \neq j \}$$

satisface $h_{\text{Gold}_{\neq}}(n) = O(n^{\frac{1}{2}})$.

Prueba En el capítulo V, hemos visto que $p_{\text{Gold}_{\neq}}(n) = O(n^{\frac{1}{2}})$; veamos que el mismo tipo de análisis allí empleado nos permite encontrar una gramática de altura sublineal.

Fundamentalmente, el lenguaje de Goldstine, puede factorizarse

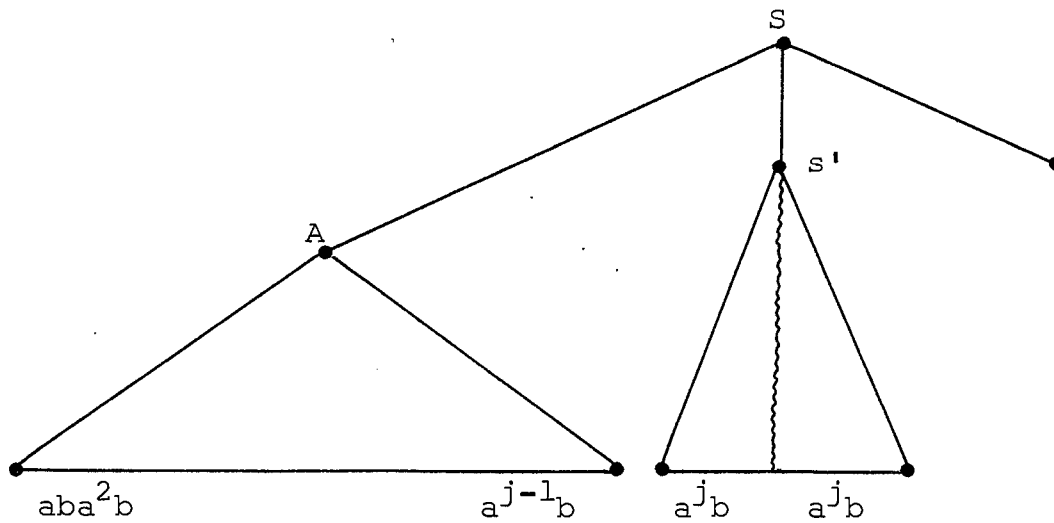
$$(a^*b)^* \{ a^i b a^j b \mid i + 1 \neq j \} (a^*b)^*$$

Es decir, la gramática contendrá un axioma $S \rightarrow AS'B$ con A y B engendrando el racional $(a^*b)^*$ y S' engendrando el lenguaje lineal $\{ a^i b a^j b \mid i + 1 \neq j \}$.

Consideremos la palabra:

$$w = ab a^2 b a^3 b \dots a^{j-1} b a^j b a^j b$$

dicha palabra admite un árbol de derivación esquematizado en:



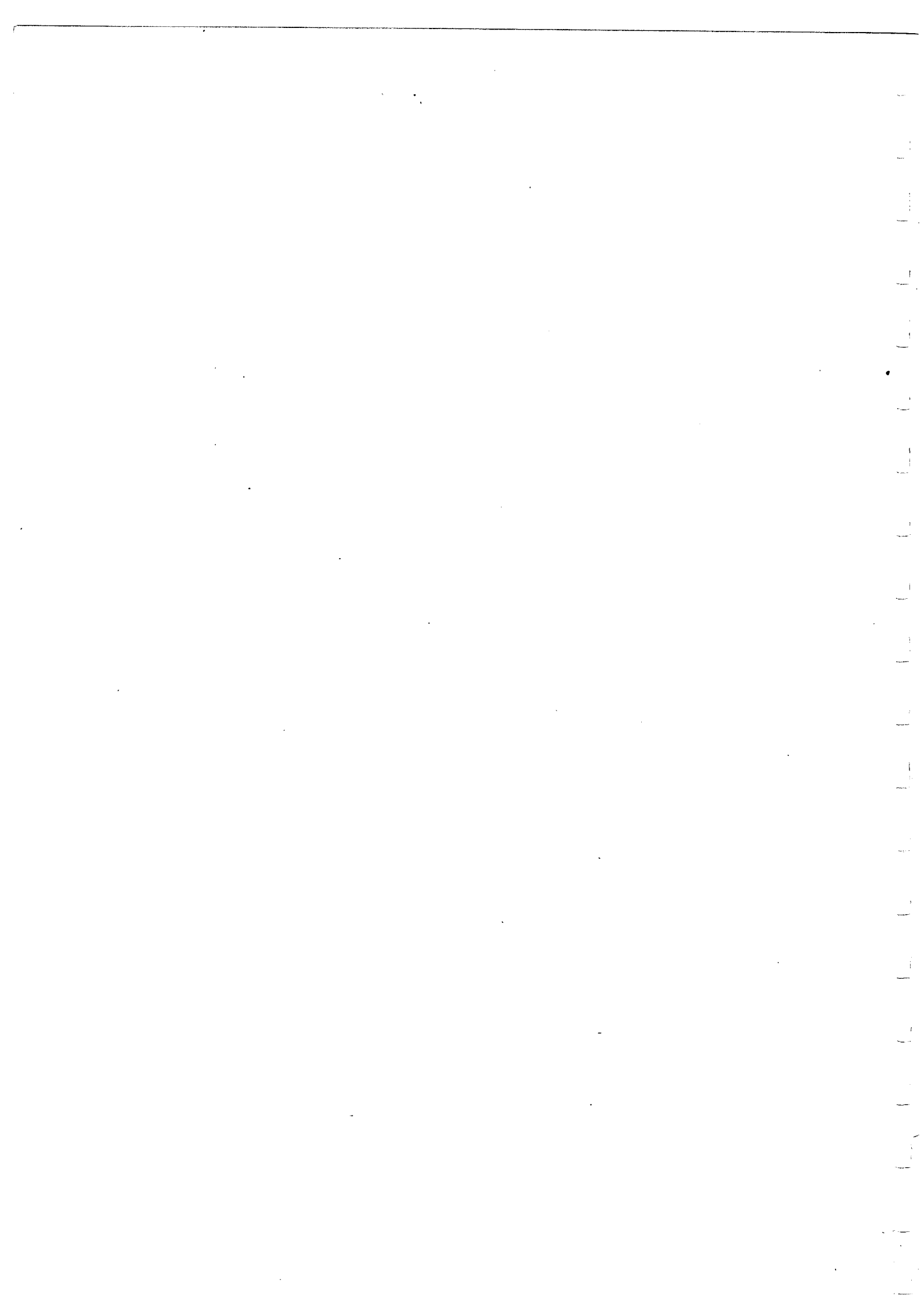
El árbol de raíz A engendra $ab a^2b \dots a^{j-1}b$, cuya longitud es $\Theta(j^2)$ de modo racional, por lo tanto esto puede ser hecho con una altura $h_A = \log(\Theta(j^2)) = \Theta(\log j)$.

El árbol de raíz S' engendra $a^j b a^j b$ con una gramática lineal, con lo que $h_{S'} = \Theta(j)$.

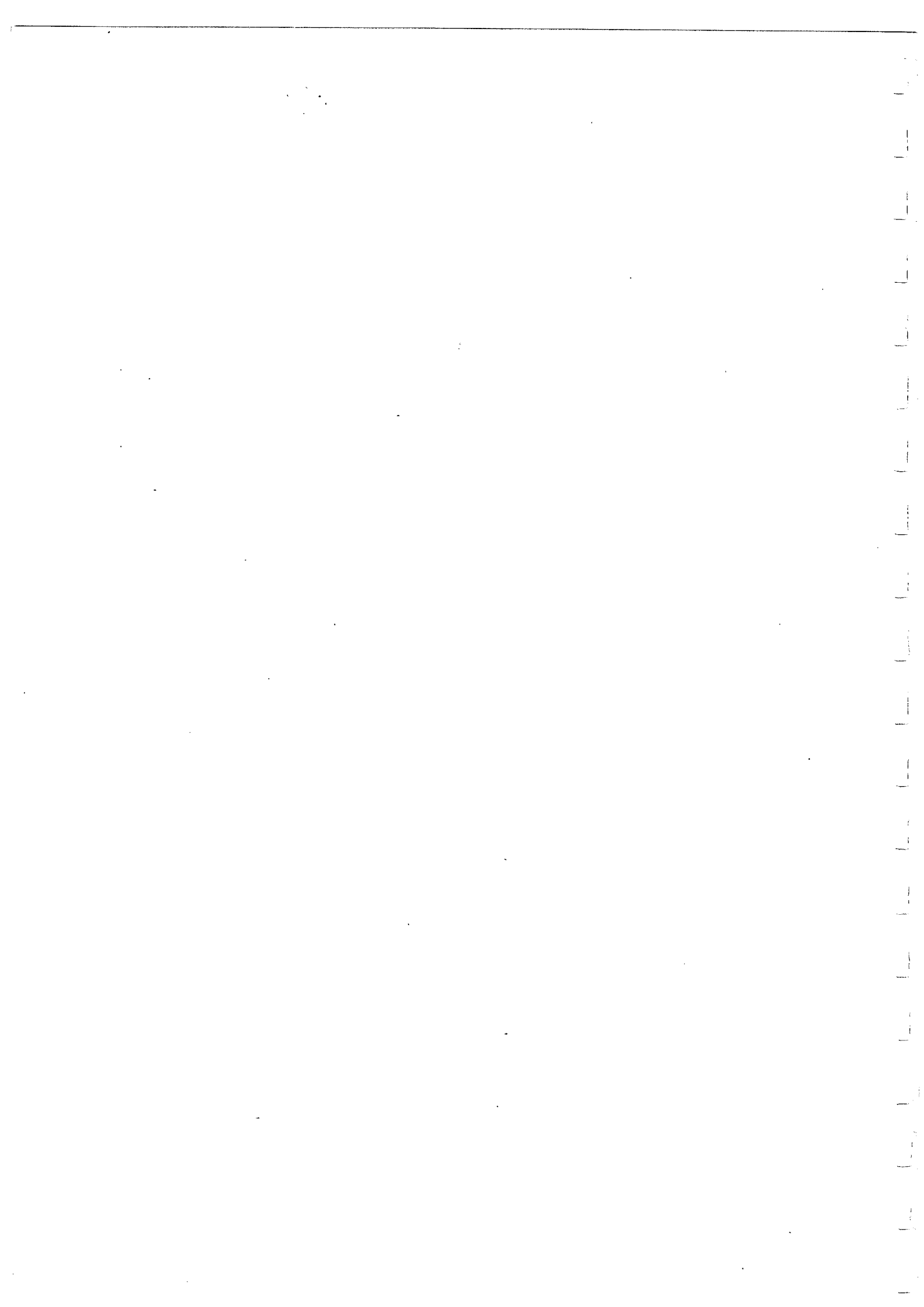
Tenemos entonces que $h_S = \max\{h_A, h_{S'}\} + 1 = \Theta(j)$, dado que

$w = j^2$, $h_S = \Theta(|w|^{\frac{1}{2}})$ y en consecuencia:

$$h_{\text{Gold}}(n) = O(n^{\frac{1}{2}}).$$



BIBLIOGRAFIA



- [1] Alt,H. [1979] "Lower bounds on space complexity for Context-Free languages".Acta Informática 12,33-61.
- [2] Autebert,J. [1978] "Cylindres de langages algebriques".Thèse d' Etat,Paris 7.
- [3] Autebert,J. [1983] .Comunicación personal.
- [4] Autebert,J; Beauquier,J;Boasson,L. [1981] ."Notes de cours de DEA" en Actes de l' Ecole de Printemps,editor Blab.M,documento LITP 82-14.
- [5] Autebert,J; Beauquier,J; Boasson,L; Latteux,M. [1980] "Very small families of algebraic non rational languages" en Formal Language Theory,editor Book,R,Academic Press.
- [6] Berstel,J. [1979] ."Transductions and Context-free languages",Teubner,Stuttgart.
- [7] Berstel,J. [1983] ."CANA,notes de cours",manuscrito no publicado,Paris 6.
- [8] Boasson,L; Courcelle,B; Nivat,M. [1981] ."The rational index a complexity measure for languages".SIAM Journal on Computing 10,2,284-296.
- [9] Boasson,L; Nivat,M. [1980] "Adherences of languages".JCSS, 20,3,285-309.
- [10] Boasson,L; Nivat,M. [1981] "Centres of languages",G.I.5. Lec.Not.Comp.Sc. 104 pág.245-251.
- [11] Book,R. [1982] "Intersection of CFL's and related structures" en Actes de l'Ecole de Printemps,editor Blab.M,publicacion LITP n° 82-14
- [12] Bucher,W; Culik,K; Maurer,H; Wotschke,D. [1981] ."Concise

- description of finite languages".Theor.Comput.Sci.14,3 pág. 211-347.
- [13] Cook,S. [1971] . "The complexity of theorem-proving procedures".3 rd. ACM-STOC,New York 151-158.
- [14] Crespi-Reghezzi,S; Mandrioli,D. [1977] . "Petri nets and Szi-lard languages".Inf and control,33 pág. 177-192.
- [15] Culik,K; Maurer,H. "On the derivation of trees".Publicación interna.
- [16] Chomsky,N. [1956] . "Three models for the description of lan-gage".IRE Transductions on Inf.Theor.pág. 113-124.
- [17] Deleage,J. [1982] . "Memoire de DEA",Paris 7,manuscrito no pu-blicado.
- [18] Eilenberg,S. [1974] . "Automata, languages and machines",Vol A, Academic Press,New-York,London.
- [19] Elgolt,C; Mezei,J. [1965] . "On relations defined by genera-lized finite automata" IBM J. of Res. and Dev. 88-101.
- [20] Finkel,A. [1983] .Comunicación personal.
- [21] Fischer,P; Meyer,A; Rosenberg,A. [1968] "Counter machines and counter languages",Math.System.Theory 2.3 pág. 265-283.
- [22] Fischer,P; Pippenger,M [1974] . "M.J.Fischer Lectures on Net-work Complexity".Universität Frankfurt,preprint.
- [23] Gabarro,J. [1981] "Index rationnel,centre et langages alge-briques".Thèse de 3 cycle,Paris 6,también documento LITP n° 81-54.
- [24] Gabarro,J. [1982] . "Une application des notions de centre et index rationnel à certains langages algebriques".RAIRO Inf.

Theor.16,4 pág. 317-329.

- [25] Gabarro,J. [1983] . "Initial index a new complexity function for languages".10 ICALP,Barcelona 83.Lec.Not.Comp.Sci.
- [26] Gabarro,J. [1983] . "Small p.d.a' s" documento LITP,pendiente de publicación.
- [27] Geller,M; Hunt,H; Szymansky,G; Ullman,D. [1977] . "Economy of description by parsers DPDA'S and PDA'S.Theor.Comp.Sci.4 pág. 143-153.
- [28] Ginsburg,S. [1966] . "The mathematical Theory of Context-free languages".Mc Graw Hill,New York.
- [29] Ginsburg,S; Greibach,S. [1969] . "Abstrac families of languages" en Abstract families of languages.Mem.of the Amer.Math. Soc.87 pág. 1-32.
- [30] Ginsburg,S; Spanier,H. [1964] . "Bounded Algol-like languages". Trans of Amer.Math.Soc.113 pág. 333-368.
- [31] Goodrich,G; Ladner,R; Fischer,M. [1977] . "Straight-line programs to compute finite languages".A conference on Theor.Comp. Sci.Waterloo,Canadá.
- [32] Greibach,S. [1975] . "One-counter languages and the IRS condition".J.of Comp. and Syst.Sci.10 pág. 237-247.
- [33] Greibach,S. [1975] . "Erasable Context-free languages".Inf. and Control,29 pág. 301-326.
- [34] Greibach,S. [1976] . "Remarks on the complexity of non deterministic counter languages".Theor.Comput.Sci.1 pág. 269-288.
- [35] Greibach,S. [1978] . "Remarks on blind and partially blind one way multicounter machines".Theor.Comp.Sci.7 pág. 311-324.

- [36] Hack, M. [1975] . "Petri nets languages". Computation Structures Group Memo 124, Project MAC, MIT Cambridge, Man.
- [37] Harrison, M. [1965] . "Introduction to switching and automata theory". Mc. Graw-Hill.
- [38] Harrison, M. [1978] . "Introduction to formal language theory". Addison-Wesley.
- [39] Jantzen, M. [1979] . "On the hierarchy of Petri net languages". RAIRO Inf. Theor. 13, 1 pág. 19-30.
- [40] Jordan, C. [1960] . "Calculus of finite differences". Chelsea Publishing Company, New York.
- [41] Karp, M; Lipton, J. [1980] . "Some connections between non uniform and uniform complexity classes". ACM-STOC pág. 302-309.
- [42] Latteux, M. [1980] . "Langages a un compteur". Publicación de la Universidad de Lille I n° I.T-25-80.
- [43] Luca, A; Restivo, A; Salemi, S. [1982] . "On the centers of a language". Comunicación interna.
- [44] Maurer, H; Nivat, M. [1980] . "Rational bijection of rational sets". Acta Inf. 13 pág. 365-378.
- [45] Milnor, J. [1968] . "A note on curvature and fundamental group". J. Diff. Geom. 2 pág. 1-7.
- [46] Nivat, M. [1968] . "Transductions des langages de Chomsky". Ann. Inst. Fourier 18 pág. 339-456.
- [47] Paredaens, J; Vyncke, R. [1977] . "A class of measures on formal languages". Acta Inf. 9 pág. 73-86.
- [48] Peterson, J. [1976] . "Computation sequence sets". J. Comp. and Syst. 13, 1, pág. 1-24.

- [49] Peterson, J. [1981] . "Petri net theory and the modeling of systems". Prentice Hall.
- [50] Savage, J. [1972] . "Computational Work and time on finite machines". J.A.C.M. 19, 4, pág. 660-674.
- [51] Steyaert, J. [1980] . Comunicació personal.
- [52] Valiant, L. [1976] . "A note on the succincstness of description of deterministic languages". Inf. and Control, 32 pág. 139-145.



