UNIVERSIDAD NACIONAL DE COLOMBIA

TEOREMAS DE PUNTO FIJO PARA LA SOLUCIÓN DE ECUACIONES SOBRE LENGUAJES

JOSÉ LUIS RAMÍREZ RAMÍREZ

Código: 830290

Universidad Nacional de Colombia

Facultad de Ciencias

Departamento de Matemáticas

Bogotá, Febrero de 2011

TEOREMAS DE PUNTO FIJO PARA LA SOLUCIÓN DE ECUACIONES SOBRE LENGUAJES

JOSÉ LUIS RAMÍREZ RAMÍREZ

Código: 830290

Tesis presentada como requisito parcial para optar al título de:

MAGISTER EN CIENCIAS MATEMÁTICAS

Director:

Ph.D., RODRIGO DE CASTRO KORGI

Universidad Nacional de Colombia

Facultad de Ciencias

Departamento de Matemáticas

Bogotá, Febrero de 2011

Resumen

En este trabajo se aborda el estudio de teoremas de punto fijo sobre retículos completos con el fin de ser aplicados a una clase de funciones entre lenguajes, llamadas funciones polinomiales. Estas funciones permiten caracterizar los lenguajes regulares y los independientes de contexto como una componente del menor punto fijo de una determinada función polinomial. Además, permiten solucionar algunas ecuaciones sobre lenguajes, en particular se demuestra el lema de Arden y una generalización de éste, lo cual permitirá caracterizar algunos lenguajes lineales.

Palabras Claves
Ecuaciones sobre lenguajes, funciones polinomiales sobre lenguajes, teoremas de punto fijo, retículos completos, retículos booleanos, lema de Arden, teorema de Ginsburg-Rice.

Abstract

In this work we study fixed-point theorem on complete lattices to be applied to a class of functions between languages, called polynomial functions. These functions allow characterization of regular languages and context-free languages as one component of the minor fixed point of a given polynomial function. Also, would solve some languages equations, in particular proves the Arden's lemma and a generalization of it, which will characterize some linear languages.

Keywords
Languages equations, languages polynomial functions, fixed-point theorems, complete lattices, boolean lattices, Arden's lemma, Ginsburg-Rice theorem.

Índice general

Introducción vi

1. Conceptos y Resultados Preliminares 1
- 1.1. Lenguajes . . . 1
- 1.2. Autómatas Finitos Deterministas AFD . . . 3
- 1.3. Gramáticas Regulares . . . 4
- 1.4. Lenguajes Lineales . . . 6
  - 1.4.1. Gramáticas Lineales . . . 6

2. Teoremas de Punto Fijo en Retículos Completos 12
- 2.1. Retículos y Conjuntos Ordenados . . . 12
- 2.2. Teoremas de Punto Fijo . . . 13
- 2.3. Retículos Booleanos . . . 15
  - 2.3.1. Retículos como Estructuras Algebraicas . . . 15

3. Funciones Polinomiales sobre Lenguajes Formales 20
- 3.1. Notación, Definiciones y Resultados Básicos . . . 20
- 3.2. Puntos Fijos de Funciones Polinomiales . . . 23

4. Funciones Polinomiales Lineales 26
- 4.1. Funciones Polinomiales Lineales . . . 26
- 4.2. Lema de Arden . . . 27
- 4.3. Matrices y Lenguajes . . . 30
- 4.4. Puntos Fijos y Lenguajes Regulares . . . 31
- 4.5. Lenguajes Simétricos . . . 34
  - 4.5.1. Inserción Simétrica . . . 34
  - 4.5.2. Generalización Lema de Arden . . . 39

4.5.3. Lenguajes Simétricos y Lineales 42

5. Puntos Fijos y Lenguajes Independientes de Contexto 49

5.1. Funciones Polinomiales Finitas 49

5.2. Teorema de Ginsburg-Rice 51

5.3. Familias de Ecuaciones sobre Lenguajes 55

Introducción

Un resultado bien conocido en la Teoría de la Computación es el Teorema de Kleene, el cual afirma que un lenguaje es regular si y sólo si es aceptado por un autómata finito. Una forma de probar una de las direcciones del teorema es asociando un sistema de ecuaciones al autómata; dicho sistema se reduce a una sola ecuación, cuya solución es precisamente el lenguaje aceptado por el autómata, que es a su vez una expresión regular. Para solucionar el sistema de ecuaciones se aplica reiteradamente el llamado Lema de Arden.

Esta forma de solucionar sistemas de ecuaciones conlleva a que algunos autores caractericen los lenguajes regulares como una componente del menor punto fijo de una determinada función polinomial a izquierda (o a derecha).

Dicha forma de abordar el Teorema de Kleene puede extenderse a los Lenguajes Independientes de Contexto (LIC); en particular los LICs coinciden con la clase de lenguajes que son componentes del menor punto fijo de una determinada función; esto se basa en un resultado conocido como el Teorema de Ginsburg-Rice. Este teorema aunque es clásico, no es muy conocido en la literatura y pocas veces se menciona en los cursos de Teoría de la Computación [5].

En este trabajo se estudia una clase general de funciones entre lenguajes, llamadas polinomiales y sus puntos fijos. Estos últimos se abordan como casos particulares de teoremas de punto fijo sobre retículos completos. Algunas subfamilias de estas funciones permiten caracterizar los lenguajes regulares y los independientes de contexto como puntos fijos de funciones polinomiales. Esta forma de abordar el estudio como un problema de retículos no es muy conocida; lo más cercano es el trabajo de Kupka [6], el cual prueba el Lema de Arden como un caso particular de un teorema de punto fijo.

Además, el estudio de los puntos fijos de estas funciones se puede aplicar a la solución de

ecuaciones particulares sobre lenguajes, como por ejemplo

$$X = A_1XB_1 \cup A_2XB_2 \cup \ldots \cup A_nXB_n \cup C \tag{1}$$

donde A_i, B_i, C $(i = 1, ..., n)$ son lenguajes sobre un alfabeto Σ. Esta última ecuación generaliza el Lema de Arden y provee una forma de caracterizar una subfamilia de lenguajes lineales.

Las ecuaciones sobre lenguajes son una línea actual de investigación, la cual ha sido liderada en los últimos años por autores como Alexander Okhotin y Michal Kunc, los cuales retomaron esta manera de relacionar los lenguajes con sistemas de ecuaciones y funciones polinomiales, y han estudiado distintas clases de ecuaciones sobre lenguajes; en particular han clasificado las ecuaciones en siete familias, una de estas familias es precisamente la que se estudia en el Teorema Ginsburg-Rice.

Este documento está dividido en cinco capítulos. El primer capítulo presenta los conceptos y resultados básicos de teoría de la computación, como alfabeto y lenguaje, lenguajes regulares, lineales e independientes de contexto, autómatas finitos, gramáticas y el teorema de Kleene, que serán utilizados en los siguientes capítulos.

El segundo capítulo introduce algunas definiciones básicas y resultados sobre retículos completos; en particular, se presentan varios teoremas de punto fijo sobre conjuntos ordenados, los cuales serán aplicados a funciones entre lenguajes. Entre los más importantes aparecen el teorema 2.3.10 y el corolario 2.3.11, ya que dan condiciones necesarias y suficientes para que cierto tipo de operador entre retículos booleanos tengan un único punto fijo.

El tercer capítulo define las funciones polinomiales entre lenguajes y se prueban algunas propiedades, en particular se demuestra que toda función polinomial tiene al menos un punto fijo. También se muestra un ejemplo de una función polinomial que tiene infinitos puntos fijos.

El cuarto capítulo introduce una clase especial de función polinomial, llamada función polinomial lineal. Un caso particular de esta clase de funciones es la que aparece en el Lema de Arden; este último se demuestra aplicando los teoremas de punto fijo y lo utilizamos para caracterizar los lenguajes regulares, como el menor punto fijo de una determinada función polinomial. Además, se consideran algunas extensiones del Lema de Arden. El capítulo finaliza con la introducción de una nueva operación entre lenguajes llamada inserción simétrica, que se utiliza para generalizar el lema de Arden y para definir una nueva clase de lenguajes, llamados simétricos, los cuales son una subfamilia de los lenguajes lineales.

El quinto capítulo presenta una familia de funciones polinomiales llamadas funciones polinomiales finitas, las cuales se utilizan para caracterizar los lenguajes independientes de contexto como una componente de un punto fijo de una función polinomial finita. En particular, se demuestra el teorema de Ginsburg-Rice, el cual relaciona precisamente el menor punto fijo de una función polinomial finita con el lenguaje generado por las gramáticas asociadas a la función. El capítulo finaliza con un marco general del estudio actual de las ecuaciones sobre lenguajes.

Capítulo 1

Conceptos y Resultados Preliminares

En este capítulo presentamos algunos conceptos y resultados básicos de teoría de la computación con los que debe contar un potencial lector de este escrito.

1.1. Lenguajes

Un **alfabeto** es un conjunto finito no vacío cuyos elementos se llaman **símbolos**. Denotamos un alfabeto arbitrario con la letra Σ. Una **cadena** sobre un alfabeto Σ es una secuencia finita de elementos de Σ, se denotan por las letras $u, v, w, \ldots$. Se supone la existencia de una única cadena que no tiene símbolos, la cual se denomina **cadena vacía** y se denota con λ.

El conjunto de todas las cadenas sobre un alfabeto Σ, incluyendo la cadena vacía, se denota por Σ^*. Dado un alfabeto Σ y dos cadenas $u, v \in \Sigma^*$, la **concatenación** de u y v, es la cadena que se forma al escribir los símbolos de u y a continuación los símbolos de v. Se denota por uv. Recursivamente se define de la siguiente manera. Si $u, v \in \Sigma^*, a \in \Sigma$, entonces

1. $u\lambda = \lambda u = u$
2. $u(va) = (uv)a$

La **longitud** de una cadena $u \in \Sigma^*$, denotada con $|u|$, se define como el número de símbolos de u. Recursivamente se define de la siguiente manera.

$$|u| = \begin{cases} 0, & \text{si } u = \lambda \\ |v| + 1, & \text{si } u = wa \end{cases}$$

para todo $a \in \Sigma$ y $v \in \Sigma^*$.

Un **lenguaje** L sobre un alfabeto Σ es un subconjunto de Σ^*. Todo lenguaje L satisface $\varnothing \subseteq L \subseteq \Sigma^*$ y puede ser finito o infinito. Los lenguajes se denotan con letras mayúsculas $A, B, C, \ldots L, M, N, \ldots$. Puesto que los lenguajes sobre Σ son subconjuntos de Σ^*, las operaciones usuales entre conjuntos son también operaciones válidas entre lenguajes. Así, si A y B son lenguajes sobre Σ, entonces los siguientes también son lenguajes sobre Σ:

$A \cup B$	Unión
$A \cap B$	Intersección
$A - B$	Diferencia
$\bar{A} = \Sigma^* - A$	Complemento

La **concatenación** de dos lenguajes A y B sobre Σ, notada AB, se define como

$$AB = \{uv : u \in A, v \in B\}$$

Dado un lenguaje A sobre Σ ($A \subseteq \Sigma^*$) y un número natural n, se define A^n de la siguiente forma

$$A^0 = \{\lambda\},$$
$$A^n = A^{n-1}A = \{u_1 \cdots u_n : u_i \in A, \quad \text{para todo } i, \quad 1 \leq i \leq n\}$$

La **estrella de Kleene** o simplemente **estrella** de un lenguaje A, $A \subseteq \Sigma^*$, es la unión de todas las potencias de A y se denota por A^*.

$$A^* = \bigcup_{i \geq 0} A^i = A^0 \cup A^1 \cup A^2 \cup \cdots \cup A^n \cup \cdots$$

De manera similar se define la **clausura positiva** de un lenguaje A, denotada por A^+.

$$A^+ = \bigcup_{i \geq 1} A^i = A^1 \cup A^2 \cup \cdots \cup A^n \cup \cdots$$

Existe una clase especial de lenguajes, conocidos como **lenguajes regulares**. Los lenguajes regulares sobre un alfabeto dado Σ son todos los lenguajes que se pueden formar a partir de los lenguajes básicos $\varnothing, \{\lambda\}, \{a\}$, $a \in \Sigma$, por medio de las operaciones de unión, concatenación y estrella de Kleene.

A continuación presentamos una definición recursiva de los lenguajes regulares. Sea Σ un alfabeto:

1. $\varnothing, \{\lambda\}, \{a\}$, para todo $a \in \Sigma$, son lenguajes regulares sobre Σ. Estos se denominan lenguajes regulares básicos.

2. Si A y B son lenguajes regulares sobre Σ, también lo son

$A \cup B$	Unión
AB	Concatenación
A^*	estrella de Kleene

Ejemplo 1.1.1. Sea $\Sigma = \{0, 1\}$. Los siguientes son lenguajes regulares sobre Σ:
(i) El lenguaje A de todas las cadenas que tienen exactamente un 0: $A = \{1\}^* \{0\} \{1\}^*$
(ii) El lenguaje B de todas las cadenas que tienen un número impar de símbolos:

$$B = [(\{0\} \cup \{1\}) (\{0\} \cup \{1\})]^* (\{0\} \cup \{1\})$$

Para simplificar la descripción de los lenguajes regulares se definen las **expresiones regulares**, cuya definición recursiva sobre un alfabeto Σ dado es:

1. Expresiones regulares básicas:
   $\varnothing$ es una expresión regular que representa al lenguaje $\varnothing$.
   λ es una expresión regular que representa al lenguaje $\{\lambda\}$.
   a es una expresión regular que representa al lenguaje $\{a\}$, para todo $a \in \Sigma$.

2. Si R y S son expresiones regulares sobre Σ, también lo son

$$(R \cup S)$$
$$(R)(S)$$
$$(R)^*$$

   $(R \cup S)$ representa la unión de los lenguajes representados por R, S; $(R)(S)$ representa su concatenación, y $(R)^*$ representa la clausura de Kleene del lenguaje representado por R. Los paréntesis (y) son símbolos de agrupación y se pueden omitir si no hay peligro de ambigüedad.

Ejemplo 1.1.2. Los dos lenguajes del ejemplo 1.1.1 se pueden representar con expresiones regulares de la siguiente manera:
(i) El lenguaje A de todas las cadenas que tienen exactamente un 0: $A = 1^*01^*$.
(ii) El lenguaje B de todas las cadenas que tienen un número impar de símbolos:

$$B = [(0 \cup 1)(0 \cup 1)]^* (0 \cup 1)$$

1.2. Autómatas Finitos Deterministas AFD

Los **autómatas finitos** son máquinas abstractas que procesan cadenas de entrada, las cuales son aceptadas o rechazadas. El autómata actúa leyendo los símbolos escritos sobre una cinta semi-infinita, dividida en celdas o casillas, sobre la cual se escribe una cinta de entrada u, un

símbolo de entrada por casilla. El autómata posee una unidad de control (también llamada cabeza lectora) que tiene un número finito de configuraciones internas, llamadas estados del autómata. Entre los estados del autómata se destacan el estado inicial y los estados finales o de aceptación. Formalmente,

Definición 1.2.1. Un **autómata finito determinista** (AFD) M está definido por cinco parámetros $M = (\Sigma, Q, q_0, F, \delta)$, donde:

- Σ es un alfabeto llamado alfabeto de cinta. Todas las cadenas que procesa M pertenecen a Σ^*.
- $Q = \{q_0, q_1, \ldots, q_n\}$ es un conjunto de estados internos del autómata.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es un conjunto de estados finales o de aceptación. $F \neq \varnothing$.
- La función de transición del autómata

$$\begin{aligned} \delta : Q \times \Sigma &\longrightarrow Q \\ (q, a) &\longmapsto \delta(q, a) \end{aligned}$$

Además de los AFD existen otro tipo de máquinas que resultan ser equivalentes a éstos, conocidos como los autómatas finitos no deterministas (AFN) y los autómatas finitos con transiciones λ (AFN-λ).

Existe una forma de relacionar los lenguajes regulares y los autómatas finitos, este resultado se conoce como el teorema de Kleene.

Teorema 1.2.2 (Kleene)**.** *Un lenguaje es regular si y sólo si es aceptado por un autómata finito (AFD o AFN o AFN-λ) .*

Demostración. Ver [2]. □

1.3. Gramáticas Regulares

Los autómatas son dispositivos que procesan cadenas de entrada, sin embargo, existen otros tipos de mecanismos que generan cadenas a partir de un símbolo inicial, las **gramáticas generativas**. A continuación presentaremos un tipo de gramática generativa, denominada gramática independiente del contexto.

Definición 1.3.1. Una **gramática independiente del contexto** (GIC), es una cuádrupla, $G = (V, \Sigma, S, P)$, donde:

- V es un alfabeto, cuyos elementos se llaman variables o símbolos no terminales.
- Σ es un alfabeto, cuyos elementos se llaman símbolos terminales. Σ y V son disyuntos.
- $S \in V$ es una variable especial, llamada símbolo inicial de la gramática.
- Un conjunto finito $P \subseteq V \times (V \cup \Sigma)^*$ de producciones o reglas de re-escritura. Una producción $(A, w) \in P$ de G se denota por $A \to w$ y se lee "A produce w"; su significado es: la variable A se puede reemplazar (sobre-escribir) por la cadena w. En la producción $A \to w$, A se denomina la cabeza y w el cuerpo de la producción.

Las variables se denotan con letras mayúsculas $A, B, C, \ldots$ Los elementos de Σ o símbolos terminales se denotan con letras minúsculas $a, b, c, \ldots$. Si $u, v \in (V \cup \Sigma)^*$ y $A \to w$ es una producción, de dice que uwv se **deriva directamente** de uAv, lo cual se denota por $uAv \Longrightarrow uwv$. Si se quiere hacer referencia a la gramática G, se escribe $uAv \overset{G}{\Longrightarrow} uwv$.

Si $u_1, u_2, \ldots, u_n$ son cadenas en $(V \cup \Sigma)^*$ y hay una sucesión de derivaciones directas

$$u_1 \overset{G}{\Longrightarrow} u_2, \quad u_2 \overset{G}{\Longrightarrow} u_3, \ \ldots \ , u_{n-1} \overset{G}{\Longrightarrow} u_n$$

se dice que u_n se deriva de u_1 y se escribe $u_1 \overset{*}{\Longrightarrow} u_n$. La anterior sucesión de derivaciones directas se representa como

$$u_1 \Longrightarrow u_2 \Longrightarrow u_3 \Longrightarrow \cdots \Longrightarrow u_{n-1} \Longrightarrow u_n$$

y se dice que es una **derivación** o una **generación** de u_n a partir de u_1. Para toda cadena w se asume que $w \overset{*}{\Longrightarrow} w$; por lo tanto, $u \overset{*}{\Longrightarrow} v$ significa que v se obtiene de u utilizando cero, uno o más producciones de la gramática. Análogamente, $u \overset{+}{\Longrightarrow} v$ significa que v se obtiene de u utilizando uno o más producciones de la gramática.

El lenguaje generado por una gramática G se denota por $L(G)$ y se define como

$$L(G) := \left\{ w \in \Sigma^* : S \overset{+}{\Longrightarrow} w \right\}.$$

Un lenguaje L sobre un alfabeto Σ se dice que es un **lenguaje independiente del contexto** (LIC) si existe una GIC G tal que $L(G) = L$.

Existe un tipo de particular de GIC, llamadas **gramáticas regulares**, las cuales son importantes ya que un lenguaje es regular si y solo si es generado por una gramática regular.

Definición 1.3.2. Una GIC $G = (V, \Sigma, S, P)$ se dice **regular por la derecha** si todas las

producciones son de la forma

$$A \to vB,$$
$$A \to v$$

donde $A, B \in V$ y $v \in \Sigma^*$. Una GIC $G = (V, \Sigma, S, P)$ se dice **regular por la izquierda** si todas las producciones son de la forma

$$A \to Bv,$$
$$A \to v$$

donde $A, B \in V$ y $v \in \Sigma^*$.
Una gramática es **regular** si es una gramática regular por la derecha o regular por la izquierda, [4] .

Teorema 1.3.3. *Un lenguaje es regular si y sólo si es generado por una gramática regular.*

Demostración. Ver [4]. □

Los lenguajes regulares en general se pueden caracterizar a partir de las expresiones regulares, para lo cual se utilizan las expresiones regulares básicas y las operaciones unión, concatenación y estrella de Kleene. También, se pueden caracterizar a partir de autómatas finitos (teorema 1.2.2) o de gramáticas regulares (teorema 1.3.3).

1.4. Lenguajes Lineales

En esta sección introducimos un caso particular de LIC, conocidos como lenguajes lineales, los cuales contienen a los lenguajes regulares. Existen dos tipos de caracterización para estos lenguajes, la primera a partir de un tipo especial de autómata de pila, conocida como Autómatas de Giro y la segunda a partir de las Gramáticas Lineales. Sin embargo, para lenguajes regulares existe una caracterización algebraica a partir de expresiones regulares, lo cual no existe para los lenguajes lineales. Una caracterización de esta índole se llevará a cabo en capítulo 4 para una subfamilia de lenguajes lineales.

1.4.1. Gramáticas Lineales

Existe un tipo particular de GIC conocidas como gramáticas lineales, similares a las gramáticas regulares.

Definición 1.4.1. Una GIC $G = (V, \Sigma, S, P)$ se dice **Gramática Lineal** si todas las producciones son de la forma

$$A \to vBu,$$
$$A \to v$$

donde $A, B \in V$ y $v, u \in \Sigma^*$.

Ejemplo 1.4.2. La siguiente es una gramática lineal que acepta el lenguaje de los palíndromos de longitud par.

$$G : \Big\{ S \to aSa \mid bSb \mid \lambda$$

Definición 1.4.3. Una GIC $G = (V, \Sigma, S, P)$ está en **Forma Normal Lineal** (FNL) si todas las producciones son de la forma

$$A \to a,$$
$$A \to aB,$$
$$A \to Bb$$

donde $A, B \in V$ y $a \in (\Sigma \cup \{\lambda\})$.

Teorema 1.4.4. *Sea L un lenguaje sobre un alfabeto Σ. Existe una gramática lineal G tal que $L(G) = L$ si y solo si existe una gramática G' en Forma Normal Lineal tal que $L(G') = L$. Es decir, las gramáticas lineales y las gramáticas en Forma Normal Lineal, generan los mismos lenguajes.*

Demostración. Una gramática en Forma Normal Lineal es claramente una gramática lineal. Recíprocamente, en una gramática lineal $G = (V, \Sigma, S, P)$, una producción de la forma

$$A \to a_1 a_2 \dots a_n B b_1 b_2 \dots b_m$$

donde los a_i, $b_j \in \Sigma$, n, $m \geq 1$, $B \in V$, se puede simular con producciones de la forma $A \to aB$, $A \to Ba$ y $A \to a$. En efecto se introducen $n + m - 2$ variables nuevas

$A_1, A_2, ..., A_{n-1}, B_1, B_2, \ldots, B_{m-1}$ cuyas únicas producciones son:

$$\begin{aligned} A &\to a_1 A_1 \\ A_1 &\to a_1 A_2 \\ &\vdots \\ A_{n-1} &\to a_n B_1 \\ B_1 &\to B_2 b_m \\ B_2 &\to B_3 b_{m-1} \\ &\vdots \\ B_{m-1} &\to B b_1 \end{aligned}$$

De esta manera se puede construir una gramática en Forma Lineal Normal equivalente a G. □

Definición 1.4.5. Un lenguaje L sobre un alfabeto Σ se dice que es un **lenguaje lineal** si existe una Gramática Lineal G tal que $L(G) = L$.

Teorema 1.4.6. *Todo lenguaje Regular es Lineal.*

Demostración. Dado un lenguaje regular A, entonces por el teorema 1.3.3 éste es generado por una gramática regular, la cual es claramente lineal. □

El recíproco de este teorema no es cierto, por ejemplo el lenguaje $L = \{a^n b^n : n \geq 0\}$ es generado por la gramática lineal

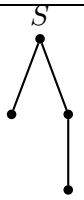
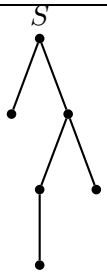
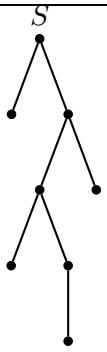
$$G: \begin{cases} S & \to aA \mid \lambda \\ A & \to Sb \end{cases}$$

sin embargo, este no es regular (aplicación directa del Lema de Bombeo para lenguajes regulares).

Ahora bien, no todo lenguaje lineal es independiente de contexto. Para ello probaremos un Lema de Bombeo para lenguajes lineales, en el que se supondrá que la gramática lineal está en FNL, sin variables inútiles, sin producciones λ (excepto posiblemente $S \to \lambda$), sin producciones unitarias, es decir de la forma $A \to B$, donde A y B son variables y con variable inicial no recursiva, es claro que al suponer la gramática en dicha forma esta sigue siendo lineal y aceptando el mismo lenguaje, ver lema 1.4.9.

Teorema 1.4.7. *Sea $G = (V, \Sigma, S, P)$ una gramática lineal en FNL y $w \in \Sigma^*$. Si la longitud de la trayectoria más larga en un árbol de derivación de $S \overset{*}{\Longrightarrow} w$ tiene k (o menos) nodos, entonces $|w| \leq k - 1$, $k \geq 2$.*

Demostración. La siguiente tabla muestra las relaciones obtenidas entre k = número de nodos de la trayectoria más larga de $S \overset{*}{\Longrightarrow} w$ y la longitud de w, en los casos $k = 2, 3, 4, 5$. En la tabla se muestra los casos extremos, es decir, los árboles con el mayor número posible de nodos.

k	Árbol de derivación	Longitud de w
$k = 2$		$\|w\| \leq 1 = 2 - 1 = k - 1$
$k = 3$		$\|w\| \leq 2 = 3 - 1 = k - 1$
$k = 4$		$\|w\| \leq 3 = 4 - 1 = k - 1$
$k = 5$		$\|w\| \leq 4 = 5 - 1 = k - 1$

Es claro que en cada paso solo puede aparecer un terminal, entonces si la trayectoria más larga tiene k nodos, como el primer nodo corresponde a la variable S, entonces la longitud de w es a lo más $k - 1$. □

Corolario 1.4.8. *Sea $G = (V, \Sigma, S, P)$ una gramática lineal en FNL y $w \in \Sigma^*$.*

1. *Si la longitud de la trayectoria más larga en un árbol de derivación de $S \overset{*}{\Longrightarrow} w$ tiene $k + 1$ (o menos) nodos, entonces $|w| \leq k$, $k \geq 1$.*

2. *Si* $|w| > k$ $(k \geq 1)$ *entonces la longitud de la trayectoria más larga en un árbol de derivación de* $S \overset{*}{\Longrightarrow} w$ *tiene más de* $k+1$ *nodos.*

Demostración. 1. Se sigue inmediatamente del teorema 1.4.7.

2. Es la afirmación contra-recíproca de la parte 1.

□

Lema 1.4.9 (Lema Bombeo para Lenguajes Lineales)**.** *Dado un lenguaje lineal* L*, existe una constante* n *(llamada constante de bombeo) tal que para toda* $z \in L$ *con* $|z| > n$*, existe una descomposición en la forma* $z = uvwxy$ *donde*

1. $|uvxy| \leq n$.
2. $|vx| \geq 1$.
3. $uv^i wx^i y \in L$ *para todo* $i \geq 0$.

Demostración. Sea $G = (V, \Sigma, S, P)$ una gramática lineal en FNL, con variable inicial no recursiva, tal que $L(G) = L$. Sea $k = |V|$ = número de variables de G y $n = k$. Sea $z \in L$ con $|z| > n = k$, por la parte (2) del Corolario 1.4.8, la trayectoria más larga en el árbol de una derivación $S \overset{*}{\Longrightarrow} z$ tiene más de $k+1$ nodos. Consideremos los primeros $k+1$ nodos de tal trayectoria (siguiendo el orden que va desde la raíz S hasta las hojas del árbol). El primer nodo de esa trayectoria es S y los restantes k nodos son variables (distintas de S). Como hay sólo k variables en la gramática, entonces hay por lo menos una variable $A \neq S$ repetida en la trayectoria. Por lo tanto, existen cadenas $u, v, w, x, y \in \Sigma^*$ tales que

$$S \overset{*}{\Longrightarrow} uAy, \qquad A \overset{*}{\Longrightarrow} vAx, \qquad A \overset{*}{\Longrightarrow} w$$

Así que

$$S \overset{*}{\Longrightarrow} uAy \overset{*}{\Longrightarrow} uvAxy \overset{*}{\Longrightarrow} uvwxy = z$$

Puesto que la distancia de S a la segunda A (desde la raíz) es una trayectoria de longitud a lo más $k+1$ nodos, entonces $|uvxy| \leq k+1 = n$, ya que en cada derivación hay un terminal. Además:

$$S \overset{*}{\Longrightarrow} uAy \overset{*}{\Longrightarrow} uvAxy \overset{*}{\Longrightarrow} uv^i Ax^i y \overset{*}{\Longrightarrow} uv^i wx^i y \qquad \text{para todo} \quad i \geq 0.$$

Finalmente, la derivación $A \overset{*}{\Longrightarrow} vAx$ se puede escribir como

$$A \Longrightarrow aB \overset{*}{\Longrightarrow} vAx \quad o \quad A \Longrightarrow Ba \overset{*}{\Longrightarrow} vAx$$

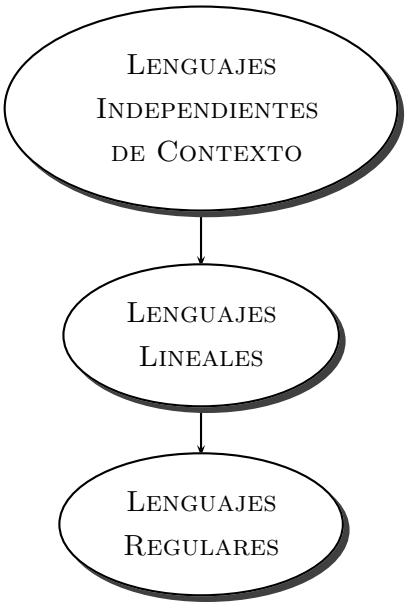
utilizando una producción de la forma $A \to aB$ o $A \to Ba$ como primer paso. Se deduce que u y x no pueden ser ambas λ ya que se tendría $aB \overset{*}{\Longrightarrow} A$ lo cual no es posible porque no hay producciones unitarias. □

Ejemplo 1.4.10. La siguiente gramática genera el conjunto de palabras sobre el alfabeto $\{a, b\}$ que tienen tantas $a's$ como $b's$.

$$G : \Big\{ S \to SS \mid aSb \mid bSa \mid \lambda$$

Es decir $L(G) = \left\{ w \in \{a, b\}^* : |w|_a = |w|_b \right\}$, luego L es un LIC. Ahora bien L no es lineal, ya que si lo fuera, existiría una constante de bombeo n para L. Sea $z = a^n b^{2n} b^n \in L$. Entonces por el lema de bombeo z se puede descomponer como $z = uvwxy$, con $|uvxy| \leq n$ y $|vx| \geq 1$. Por lo tanto u, v, x y y constan únicamente de $a's$. Entonces $uv^2wx^2y = a^{n+l} b^{2n} a^{n+k}$, con $l \geq 1$ o $k \geq 1$, resultando que $uv^2wx^2y \notin L$, lo cual contradice el Lema 1.4.9.

Resumiendo estos resultados se tiene la siguiente jerarquía:



Capítulo 2

Teoremas de Punto Fijo en Retículos Completos

En este capítulo presentamos algunos resultados sobre puntos fijos en retículos completos, en particular dos teoremas de puntos fijo que garantizan la existencia de puntos fijos de funciones definidas sobre retículos booleanos. Estos resultados se aplicarán en el siguiente capítulo para funciones entre lenguajes.

2.1. Retículos y Conjuntos Ordenados

A continuación presentamos algunas definiciones y resultados [1] necesarios para el desarrollo del capítulo.

Definición 2.1.1. Un **orden parcial** sobre un conjunto no vacío P es una relación binaria $\leq$ sobre P que es reflexiva, antisimétrica y transitiva, es decir, para todo $x, y, z \in P$:

1. Reflexiva: $x \leq x$.
2. Antisimétrica: $x \leq y,\ y \leq x \Rightarrow x = y$.
3. Transitiva: $x \leq y,\ y \leq z \Rightarrow x \leq z$.

La pareja $(P, \leq)$ se denomina un **conjunto parcialmente ordenado**.

Definición 2.1.2. Sea $(P, \leq)$ un conjunto parcialmente ordenado y sea S un subconjunto de P. Un elemento $s \in P$ se llama **cota superior** de S si $x \leq s$ para todo $x \in S$. Un elemento $m \in P$ se llama **cota inferior** de S si $m \leq x$ para todo $x \in S$.

Definición 2.1.3. Sea $(P, \leq)$ un conjunto parcialmente ordenado y sea S un subconjunto de P. Llamaremos **supremo** de S ($\sup S$) a la mínima de las cotas superiores de S, cuando

exista. Análogmente, llamaremos **ínfimo** de S ($\text{ínf}\, S$) a la máxima de las cotas inferiores de S, cuando exista.

Siendo el máximo y el mínimo únicos cuando existen, entonces $\sup S$ e $\text{ínf}\, S$ son únicos cuando existen.

Adoptaremos la siguiente notación, $x \bigvee y$ en lugar de $\sup\{x, y\}$ cuando este exista, $x \bigwedge y$ en lugar de $\text{ínf}\{x, y\}$ cuando este exista. Similarmente, se escribirá $\bigvee S$ e $\bigwedge S$ en lugar de $\sup S$ e $\text{ínf}\, S$ en caso que existan.

Definición 2.1.4. Sea $(P, \leq)$ un conjunto parcialmente ordenado.

1. Si $x \bigvee y$ e $x \bigwedge y$ existen para todo $x, y \in P$ entonces $(P, \leq)$ es un **retículo**.
2. Si $\bigvee S$ e $\bigwedge S$ existen para todo $S \subseteq P$ entonces $(P, \leq)$ es un **retículo completo**.

Ejemplo 2.1.5. (i) Un conjunto totalmente ordenado es un retículo, pero no necesariamente un retículo completo. Por ejemplo, los enteros $\mathbb{Z}$ respecto al orden usual es un retículo, pero no es completo.
(ii) Si S es un conjunto no vacío, entonces $(\wp(S), \subseteq)$ es un retículo completo, donde el sup coincide con la unión y el ínf con la intersección. En particular $(\wp(\Sigma^*), \subseteq)$, donde el conjunto Σ^* es el de todas las cadenas sobre un alfabeto Σ, incluyendo la cadena vacía, es un retículo completo bajo la unión e intersección. Además, el elemento máximo es Σ^* y el elemento mínimo es $\varnothing$.

Definición 2.1.6. Sean P y Q conjuntos ordenados y una función $T : P \to Q$. T es una **función monótona o que preserva el orden**, si para todo $x, y \in P$ con $x \leq y$ se tiene que $T(x) \leq T(y)$.

Si T preserva el orden entonces T^n también lo hace. Recordemos que T^0 es la función identidad y $T^n = T \circ T^{n-1}$ para $n \geq 1$.

2.2. Teoremas de Punto Fijo

Los siguientes dos teoremas garantizan la existencia de puntos fijos de funciones definidas sobre retículos completos, uno de ellos en un resultados clásico, conocido como el Teorema de Punto Fijo de Knaster-Tarski.

Teorema 2.2.1 (Knaster-Tarski)**.** *Sean $(P, \leq)$ un retículo completo, $T : P \to P$ una aplicación monótona y* $\text{Fix} = \{x \in P : T(x) = x\}$ *el conjunto de puntos fijos. Entonces se tiene lo siguiente:*

(i) Fix *es no vacío.*

(ii) Fix *es un retículo completo.*

(iii) $\bigvee \{x \in P : x \leq T(x)\} = \bigvee \text{Fix} = p^* \quad y \quad p^* \in \text{Fix}.$

(iv) $\bigwedge \{x \in P : T(x) \leq x\} = \bigwedge \text{Fix} = p_* \quad y \quad p_* \in \text{Fix}.$

Por lo tanto, p^* es el mayor punto fijo de T y p_* es el menor punto fijo de T.

Demostración. Ver [1] o [10]. □

Definición 2.2.2. Un subconjunto no vacío D de un conjunto parcialmente ordenado P es **dirigido** si todo par de elementos de D está acotado superiormente en D.

Es fácil ver que todo conjunto directo P tiene la propiedad que todo subconjunto finito está acotado superiormente en P.

Definición 2.2.3. Un conjunto ordenado P es **completo** o un CPO si

(i) P tiene un elemento mínimo $\perp$.

(ii) $\bigvee D$ existe para todo subconjunto dirigido D de P.

En caso de que solo se cumpla la segunda condición, entonces se dice que P es un DCPO.

Es claro que todo retículo completo es un CPO.

Definición 2.2.4. Sean P y Q DCPO y $T : P \to Q$ una función. Se dice que T es **continua** si para todo subconjunto dirigido D en P, el subconjunto $T(D)$ de Q también es dirigido y

$$T\left(\bigvee (D)\right) = \bigvee \left(T(D)\right)$$

Si T es continua entonces preserva el orden.

Se puede demostrar que una función continua en este sentido es continua topológicamente cuando P y Q están dotados de la topología de Scott, [1].

El siguiente teorema es importante ya que no solo garantiza la existencia de puntos fijos, sino que también exhibe explícitamente el menor punto fijo.

Teorema 2.2.5. *Sea P un CPO y $T : P \to P$ una función continua, entonces T tiene un punto fijo mínimo dado por*

$$\alpha := \bigvee_{n \geq 0} T^n(\perp)$$

Demostración. Veamos primero que α está bien definido. Puesto que $\perp \leq T(\perp)$, entonces aplicando la función T^n se tiene que $T^n(\perp) \leq T^{n+1}(\perp)$ para todo n. Por lo tanto se tiene la cadena

$$\perp \leq T(\perp) \leq \cdots \leq T^n(\perp) \leq T^{n+1}(\perp) \leq \cdots$$

en P, la cual es un conjunto dirigido. Puesto que P es un DCPO entonces $\alpha := \bigvee_{n\geq 0} T^n(\perp)$ existe. Veamos ahora que α es un punto fijo y que es el menor:

$$\begin{aligned} T(\alpha) &= T\left(\bigvee_{n\geq 0} T^n(\perp)\right) \\ &= \bigvee_{n\geq 0} \left(T\left(T^n(\perp)\right)\right) \quad \text{Por ser } T \text{ continua} \\ &= \bigvee_{n\geq 1} T^n(\perp) \\ &= \bigvee_{n\geq 0} T^n(\perp) = \alpha \quad \text{ya que } \perp \leq T^n(\perp) \text{ para todo } n. \end{aligned}$$

Sea β un punto fijo de T, entonces por inducción se tiene que $T^n(\beta) = \beta$ para todo n. Puesto que $\perp \leq \beta$, entonces $T^n(\perp) \leq T^n(\beta) = \beta$. Así por definición de α se concluye que $\alpha \leq \beta$. □

2.3. Retículos Booleanos

Los retículos pueden definirse de dos formas: la primera basada en la existencia de una relación de orden que satisface ciertas propiedades y la segunda basada en la existencia de una operación binaria que satisface ciertas propiedades algebraicas. En esta sección abordaremos la segunda forma y demostraremos que son equivalentes, luego definiremos los retículos booleanos y probaremos un teorema de punto fijo sobre dichos retículos, el cual se aplicará en capítulos posteriores para la solución de algunas ecuaciones sobre lenguajes.

2.3.1. Retículos como Estructuras Algebraicas

Dado un retículo $(P, \leq)$, se pueden definir dos operaciones $\vee$ y $\wedge$ tales que:

$$\begin{aligned} \vee &: P \times P \longrightarrow P \\ &\quad (a, b) \longmapsto a \vee b := \sup\{a, b\} \\ \wedge &: P \times P \longrightarrow P \\ &\quad (a, b) \longmapsto a \wedge b := \text{ínf}\{a, b\} \end{aligned}$$

A continuación veremos que la estructura algebraica $(P, \vee, \wedge)$ se relaciona estrechamente con $(P, \leq)$.

Lema 2.3.1. *Sea $(P, \leq)$ un retículo y sean $a, b \in P$. Entonces las siguientes afirmaciones son equivalentes*

1. $a \leq b$;
2. $a \vee b = b$;
3. $a \wedge b = a$.

Demostración. Es inmediato que (1) implica (2) y (3). Recíprocamente, si $a \vee b = b$ entonces b es una cota superior de $\{a, b\}$, luego $a \leq b$. Similarmente se prueba que (3) implica (1). □

Teorema 2.3.2. *Sea $(P, \leq)$ un retículo. Entonces $\vee$ y $\wedge$ satisfacen las siguientes propiedades para todo $a, b, c \in P$:*
(L1) (Asociativa)
$(a \vee b) \vee c = a \vee (b \vee c)$
$(a \wedge b) \wedge c = a \wedge (b \wedge c)$
(L2) (Conmutativa)
$a \vee b = b \vee a$
$a \wedge b = b \wedge a$
(L3) (Idempotencia)
$a \vee a = a$
$a \wedge a = a$
(L4) (Absorción)
$a \vee (a \wedge b) = a$
$a \wedge (a \vee b) = a$

Demostración. (L1) Se tiene ya que el conjunto de las cotas superiores de $\{a \vee b, c\}$ y $\{a, b \vee c\}$ coincide con el de las cotas superiores de $\{a, b, c\}$. (L2) Se tiene ya que el sup no depende del orden en que estén enlistados los elementos. (L3) Puesto que $\leq$ es reflexiva, entonces $a \leq a$ luego por el lema 2.3.1 $a \vee a = a$. (L4) Es consecuencia directa del lema 2.3.1.

□

Teorema 2.3.3. *Sea $(P, \vee, \wedge)$ un conjunto no vacío dotado con dos operaciones binarias que satisfacen (L1-L4), entonces*

1. *Para todo $a, b \in P$, se tiene que $a \vee b = b$ si y sólo si $a \wedge b = a$.*
2. *Sobre P se define $\leq$ por $a \leq b$ si $a \vee b = b$. Entonces $\leq$ es una relación de orden.*
3. *$(P, \leq)$, con $\leq$ definida como en (2), es un retículo completo en el cual las operaciones originales corresponden con las operaciones inducidas, es decir, para todo $a, b \in P$, $a \vee b = \sup\{a, b\}$ y $a \wedge b = \text{ínf}\{a, b\}$.*

Demostración. 1. Si $a \vee b = b$. Entonces

$$\begin{aligned} a =& a \wedge (a \vee b) \text{ Por L4} \\ =& a \wedge b \text{ Hipótesis} \end{aligned}$$

Recíprocamente asumamos que $a \wedge b = a$. Entonces

$$\begin{aligned} b =& b \vee (b \wedge a) \text{ Por L4} \\ =& b \vee (a \wedge b) \text{ Por L2} \\ =& b \vee a \text{ Hipótesis} \\ =& a \vee b \text{ Por L2} \end{aligned}$$

2. $\leq$ es reflexiva por (L3), antisimétrica por (L2) y transitiva por (L1).

3. $a \vee b$ pertenece al conjunto de las cotas superiores de $\{a, b\}$ ya que $a \vee (a \vee b) = a \vee b$ luego $a \leq (a \vee b)$, similarmente se prueba que $b \leq (a \vee b)$. Si d es una cota superior de $\{a, b\}$ entonces $(a \vee b) \vee d = a \vee (b \vee d) = a \vee d = d$, luego $(a \vee b) \leq d$. Análogamente para $\wedge$.

□

Definición 2.3.4. Sea $(P, \vee, \wedge)$ un retículo. Decimos que L tienen un **uno** si existe un elemento $1 \in P$ tal que $a = a \wedge 1$ para todo $a \in P$. Análogamente, L tiene un **cero** si existe un elemento $0 \in P$ tal que $a = a \vee 0$ para todo $a \in P$. Si 0 y 1 existen decimos que $(P, \vee, \wedge)$ es **acotado**.

Es claro que $(P, \vee, \wedge)$ tiene un 1 (0) si y sólo si $(P, \leq)$ tiene máximo $\top$ (mínimo $\bot$).

Definición 2.3.5. Un retículo P es un **retículo distributivo** si satisface las propiedades distributivas

1. $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$

2. $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

para todo $a, b, c \in P$.

Definición 2.3.6. Sea L un retículo acotado. Un **complemento** de $a \in P$ es un elemento $b \in P$ que cumple $a \wedge b = 0$ y $a \vee b = 1$. Se dice que a y b son complementarios.

Definición 2.3.7. (i) Un retículo acotado P para el cual todo elemento tiene un complemento se denomina **retículo complementado**.
(ii) Un retículo complementado y distributivo P se denomina **retículo booleano**.

Ejemplo 2.3.8. Si S es un conjunto no vacío, entonces el conjunto de partes $\wp(S)$ es un retículo booleano bajo la unión e intersección. En particular el conjunto $\wp(\Sigma^*)$, donde el conjunto Σ^* es el de todas las cadenas sobre un alfabeto Σ, incluyendo la cadena vacía, es un retículo booleano bajo la unión e intersección.

Definición 2.3.9. Sea P un retículo booleano. Un operador $G : P \to P$ **preserva sups contables** si $G(\bigvee_{n\geq 0} X_n) = \bigvee_{n\geq 0} G(X_n)$, para cualquier familia contable $\{X_n\}_{n\geq 0}$ de subconjuntos de P.

Nótese que si un operador G preserva sups entonces es monótono.

El siguiente teorema es mucho más fuerte que los dos teoremas de punto fijo vistos hasta el momento, aunque requiere de hipótesis más fuertes, ya que garantiza la existencia de puntos fijos, muestra todos ellos y da condiciones necesarias y suficientes para que haya un único punto fijo, [3]. Esto último se aplicará en la solución de algunas ecuaciones sobre lenguajes.

Teorema 2.3.10. *Sea P un retículo booleano, K un subconjunto (fijo) de P y $G : P \to P$ un operador que preserva uniones contables. Si $T : P \to P$ está definido por $T(X) = G(X) \vee K$, entonces*

(i) *$X_0 := \bigvee_{n\geq 0} G^n(K)$ es un punto fijo de T. Además, X_0 es el más pequeño punto fijo; es decir, si X es un punto fijo de T entonces $X_0 \leq X$ (se entiende que $G^0(K) = K$).*

(ii) *Si $\mathrm{Fix}(T)$ denota la colección de todos los puntos fijos de T entonces*

$$\mathrm{Fix}(T) = \Big\{ \bigvee_{n\geq 0} G^n(K \vee Y) : Y \leq G(Y) \Big\}.$$

(iii) *X_0 es el único punto fijo de T si y sólo si G es no-extensiva, i.e., $Y \leq G(Y)$ implica $Y = 0$.*

Demostración. (i)

$$\begin{aligned} T(X_0) &= T\Big(\bigvee_{n\geq 0} G^n(K) \Big) \\ &= G\Big(\bigvee_{n\geq 0} G^n(K) \Big) \vee K \\ &= \bigvee_{n\geq 0} G^{n+1}(K) \vee K = X_0. \end{aligned}$$

Si X es un punto fijo de T entonces

$$X = T(X) = G(X) \vee K$$

por lo tanto, $K \leq X$. También se tiene que $G(K) \leq X$, en efecto aplicando G a la igualdad anterior se obtiene que

$$G(X) = G(G(X) \vee K) = G(G(X)) \vee G(K)$$

entonces $G(K) \leq G(X) \leq X$; a partir de esta contenencia y usando inducción sobre n, se puede demostrar que $G^n(K) \leq X$ para todo $n \geq 1$. Así, $X_0 \leq X$.

(ii) Es fácil ver que para cualquier Y con $Y \leq G(Y)$, el conjunto $\bigvee_{n\geq 0} G^n(K \vee Y)$ es un punto fijo de T. Recíprocamente, sea X un punto fijo de T. Entonces, por (i), $X \leq X_0$ y X puede ser escrito como $X = X_0 \vee Y$ donde $Y \wedge X_0 = 0$. Por consiguiente

$$\begin{aligned} X_0 \vee Y = T(X_0 \vee Y) &= G(X_0 \vee Y) \vee K = G(X_0) \vee G(Y) \vee K \\ &= F(X_0) \vee G(Y) = X_0 \vee G(Y). \end{aligned} \tag{2.1}$$

Interceptando ambos lados de (2.1) con Y se concluye que $Y = Y \wedge G(Y)$, de donde $Y \leq G(Y)$. Además, usando (2.1) e inducción matemática se pude mostrar que $X_0 \vee G^n(Y) = X_0 \vee G^{n+1}(Y)$ para todo $n \geq 0$. Así que

$$X = X_0 \vee Y = X_0 \vee \bigvee_{n\geq 0} G^n(Y) = \bigvee_{n\geq 0} G^n(K \vee Y)$$

lo que demuestra (ii).

(iii) Se deduce inmediatamente de (ii).

□

Corolario 2.3.11. *Sea S un conjunto no vacío, K un subconjunto (fijo) de S y $G : \wp(S) \to \wp(S)$ un operador que preserva uniones contables. Si $T : \wp(S) \to \wp(S)$ está definido por $T(X) = G(X) \cup K$, entonces*

(i) *$X_0 := \bigcup_{n\geq 0} G^n(K)$ es un punto fijo de T. Además, X_0 es el más pequeño punto fijo; es decir, si X es un punto fijo de T entonces $X_0 \subseteq X$ (se entiende que $G^0(K) = K$).*

(ii) *Si $\mathrm{Fix}(T)$ denota la colección de todos los puntos fijos de T entonces*

$$\mathrm{Fix}(T) = \Big\{ \bigcup_{n\geq 0} G^n(K \cup Y) : Y \subseteq G(Y) \Big\}.$$

(iii) *X_0 es el único punto fijo de T si y sólo si G es no-extensiva, i.e., $Y \subseteq G(Y)$ implica $Y = \varnothing$.*

Nótese que si $G(\varnothing) = \varnothing$, entonces la parte (i) del corolario 2.3.11 es consecuencia del Teorema 2.2.5.

Capítulo 3

Funciones Polinomiales sobre Lenguajes Formales

En este capítulo se introduce una clase especial de función entre lenguajes, denominada función polinomial y se estudian sus puntos fijos. Esto nos permitirá solucionar sistemas de ecuaciones sobre lenguajes, y a su vez caracterizar los lenguajes regulares y los independientes de contexto como puntos fijos de funciones polinomiales.

3.1. Notación, Definiciones y Resultados Básicos

Sea Σ un alfabeto dado, entonces $(\wp(\Sigma^*))^n$ denotará el conjunto de las n-uplas donde cada componente es un lenguaje sobre Σ, es decir

$$(\wp(\Sigma^*))^n := \{(L_1, L_2, \ldots, L_n) : L_i \subseteq \Sigma^*, 1 \leq i \leq n\}.$$

Si $(L_1, L_2, \ldots L_n) \in (\wp(\Sigma^*))^n$, entonces la n-upla la escribiremos como $\vec{L}$. Si $L = (L_1, L_2, \ldots L_n)$ y $K = (K_1, K_2, \ldots K_n) \in (\wp(\Sigma^*))^n$, entonces diremos que $\vec{L} \subseteq \vec{K}$ si $L_i \subseteq K_i$, para $1 \leq i \leq n$. Notaremos la n-upla $(\varnothing, \varnothing, \ldots, \varnothing)$ como $\vec{\varnothing}$.

Si f es una función n-aria $f : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ usaremos la notación $f\left(\vec{L}\right)$ para $f((L_1, L_2, \ldots L_n))$. Estas funciones se pueden combinar de una manera natural a partir de la unión y la concatenación.

Definición 3.1.1. Sean $f, g : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$, entonces las funciones unión $f \cup g$ y

producto fg, se definen como

$$\begin{aligned} f \cup g : (\wp(\Sigma^*))^n &\longrightarrow \wp(\Sigma^*) \\ (f \cup g)(\vec{X}) &= f(\vec{X}) \cup g(\vec{X}) \end{aligned}$$

$$\begin{aligned} fg : (\wp(\Sigma^*))^n &\longrightarrow \wp(\Sigma^*) \\ (fg)(\vec{X}) &= f(\vec{X})\, g(\vec{X}) \end{aligned}$$

para todo $\vec{X} \in \wp(\Sigma^*)^n$.

A una sucesión finita de funciones n-arias la notaremos por $(f_1, f_2, \ldots, f_m)$, donde $f_i : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ para $1 \leq i \leq m$ y $n \geq 1$. A continuación extenderemos la definición de unión y producto para sucesiones finitas de funciones.

Definición 3.1.2. Sea $(f_1, f_2, \ldots, f_m)$ una sucesión finita de funciones n-arias, entonces la unión $\bigcup_{1 \leq i \leq m} f_i$ y el producto $\prod_{1 \leq i \leq m} f_i$ se definen como

$$\begin{aligned} \left(\bigcup_{1 \leq i \leq m} f_i \right)(\vec{X}) &:= \bigcup_{1 \leq i \leq m} f_i(\vec{X}) \\ \left(\prod_{1 \leq i \leq m} f_i \right)(\vec{X}) &:= \prod_{1 \leq i \leq m} f_i(\vec{X}) \end{aligned}$$

para todo $\vec{X} \in (\wp(\Sigma^*))^n$.

Definición 3.1.3. Sea $(f_1, f_2, \ldots f_m)$ una sucesión finita de funciones n-arias, entonces la función $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^m$, se define como

$$\vec{f}(\vec{X}) = \big(f_1(\vec{X}), f_2(\vec{X}), \ldots, f_m(\vec{X})\big)$$

para todo $\vec{X} \in (\wp(\Sigma^*))^n$. Diremos que $f_1, f_2, \ldots, f_m$ son las componentes de $\vec{f}$.

A continuación definimos la función constante y la función proyección, ya que a partir de éstas se definen recursivamente las funciones polinomiales.

Definición 3.1.4. Sea $L \subseteq \Sigma^*$ un lenguaje sobre Σ, la **función constante** $C_L : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ se define como

$$C_L(\vec{X}) = C_L(X_1, X_2, \ldots, X_n) = L$$

para todo $X_1, X_2, \ldots, X_n \in \wp(\Sigma^*)$ y $n \geq 1$.

La **función proyección** $P_i^n : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ se define como

$$P_i^n(\vec{X}) = P_i^n(X_1, X_2, \ldots, X_n) = X_i$$

para todo $X_1, X_2, \ldots, X_n \in \wp(\Sigma^*)$ y $1 \leq i \leq n$.

Definición 3.1.5. Una función n-aria f es una **función polinomial** si:

1. f es una función constante o una proyección n-aria.
2. Si f, g son funciones polinomiales, entonces también lo son $f \cup g$ y fg.

Ejemplo 3.1.6. Las siguientes son funciones polinomiales.
(i) $f(X) = AX \cup B$. Es claro que $f(X) = AX \cup B = C_A(X)P_1^1(X) \cup C_B(X)$

(ii) $f(X) = XX \cup \{aa\}$. En este caso $f(X) = XX \cup \{aa\} = P_1^1(X)P_1^1(X) \cup C_{\{aa\}}(X)$.

(iii) $f(X_1, X_2) = aX_1X_2 \cup X_1\{a, b\}X_2 \cup \{a\}^*$. En este último ejemplo se tiene que

$$\begin{aligned} f(X_1, X_2) =& C_{\{a\}}(X_1X_2)P_1^2(X_1X_2)P_2^2(X_1X_2) \cup P_1^2(X_1X_2)C_{\{a,b\}}(X_1X_2)P_2^2(X_1X_2) \\ & \cup C_{\{a\}^*}(X_1X_2) \end{aligned}$$

Definición 3.1.7. Sea $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^m$, donde $\vec{f} = (f_1, f_2, \ldots, f_m)$. Diremos que $\vec{f}$ es una función polinomial si cada componente f_i es polinomial.

Definición 3.1.8. Sea $f : \wp(\Sigma^*) \longrightarrow \wp(\Sigma^*)$, entonces la k-ésima iteración de f es la función $f^k : \wp(\Sigma^*) \longrightarrow \wp(\Sigma^*)$ definida recursivamente por

$$\begin{aligned} f^0 &= Id \\ f^{k+1} &= f \circ f^k \end{aligned}$$

para todo $k \geq 0$. $f \circ f^k$ es la composición usual de funciones.

Ejemplo 3.1.9. Sea $f(X) = AX \cup B$ la función polinomial del ejemplo 3.1.6, entonces las primeras iteraciones son:

$$\begin{aligned} f^2(X) &= A(AX \cup B) \cup B = A^2X \cup AB \cup B \\ f^3(X) &= A^3X \cup A^2B \cup AB \cup B \end{aligned}$$

se verifica por inducción sobre k que

$$f^k(X) = A^kX \cup A^{k-1}B \cup \ldots \cup AB \cup B$$

para toda $X \in \wp(\Sigma^*)$ y $k \geq 1$.

Definición 3.1.10. Sean $f : (\wp(\Sigma^*))^m \longrightarrow \wp(\Sigma^*)$ y $g_i : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ para $1 \leq i \leq m$, se define la función $f(g_1, \ldots, g_m) : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))$ como

$$f(g_1, \ldots, g_m)(\vec{X}) = f(g_1(\vec{X}), \ldots, g_m(\vec{X}))$$

para todo $\vec{X} \in (\wp(\Sigma^*))^n$.

Es claro que si $f, g : \wp(\Sigma^*) \longrightarrow \wp(\Sigma^*)$ son funciones constantes o proyecciones, entonces también lo es $f(g(\vec{L}))$ y por inducción se prueba que si $f : (\wp(\Sigma^*))^m \longrightarrow \wp(\Sigma^*)$ y $g_i : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ para $1 \leq i \leq m$, son polinomiales entonces también lo es $f(g_1, \ldots, g_m)$. Esto se resume en el siguiente teorema.

Teorema 3.1.11. *Si f es una función polinomial n-aria y $g_i : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ para $1 \leq i \leq m$ son polinomiales, entonces $f(g_1, \ldots, g_m)$ es polinomial.*

Corolario 3.1.12. *Si $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^n$ es polinomial, entonces la k-ésima iteración de $\vec{f}$ es polinomial.*

Demostración. Sea $\vec{f} = (f_1, \ldots f_m)$, como $\vec{f}$ es polinomial, entonces f_i $(1 \leq i \leq n)$ es polinomial. Como $\vec{f}^k = \vec{f}\vec{f}^{k-1}$ y por inducción $\vec{f}^{k-1}$ es polinomial, entonces por el teorema anterior $\vec{f}^k$ es polinomial. □

3.2. Puntos Fijos de Funciones Polinomiales

En esta sección probaremos que toda función polinomial tiene al menos un punto fijo y mostraremos cómo encontrar el menor de ellos; este resultado será una aplicación directa de los teoremas de punto fijo sobre retículos completos. Para tal fin probaremos los siguientes dos lemas relacionados con monotonía y la continuidad de funciones polinomiales.

Lema 3.2.1. *Si $\vec{K} \subseteq \vec{L}$ para $\vec{L}, \vec{K} \in (\wp(\Sigma^*))^n$ entonces $f(\vec{K}) \subseteq f(\vec{L})$ para toda función polinomial f.*

Demostración. El argumento es por inducción sobre la definición de función polinomial. Si f es constante o una proyección, se tiene claramente la propiedad. Supongamos que $f, g : (\wp(\Sigma^*))^n \longrightarrow \wp(\Sigma^*)$ verifican la propiedad, si $\vec{K} \subseteq \vec{L}$ entonces $f(\vec{K}) \subseteq f(\vec{L})$ y $g(\vec{K}) \subseteq g(\vec{L})$, luego $f(\vec{K}) \cup f(\vec{K}) \subseteq f(\vec{L}) \cup g(\vec{K})$ y $f(\vec{K})f(\vec{K}) \subseteq f(\vec{L})g(\vec{L})$, así la unión y el producto son monótonas. □

Definición 3.2.2. Sean $L_1, \ldots, L_n, \ldots$ n-uplas de lenguajes, $L_i \in (\wp(\Sigma^*))^n$, $i \geq 1$, donde $L_i = (L_{i1}, \ldots, L_{in})$. Se define $\bigcup\{L_i : i \geq 1\}$ como la n-upla $\vec{L} = (L_1, \ldots, L_n)$, donde

$$L_s = \bigcup\{L_{is} : i \geq 1\} \quad 1 \leq s \leq n.$$

Lema 3.2.3. *Sea f una función polinomial, entonces para toda sucesión creciente de n-uplas $\vec{L}_1, \ldots, \vec{L}_m, \ldots$ ($\vec{L}_i \subseteq \vec{L}_j$, para $i \leq j$) se tiene que*

$$f\left(\bigcup_{i\geq 1} \vec{L}_i\right) = \bigcup_{i\geq 1} f\left(\vec{L}_i\right).$$

Demostración. Puesto que $\vec{L}_j \subseteq \bigcup_{i\geq 1} \vec{L}_i$ entonces por el lema 3.2.1 se tiene que $f\left(\vec{L}_j\right) \subseteq f\left(\bigcup_{i\geq 1} \vec{L}_i\right)$ para todo $j \geq 1$, entonces $\bigcup_{i\geq 1} f\left(\vec{L}_i\right) \subseteq f\left(\bigcup_{i\geq 1} \vec{L}_i\right)$.
La implicación recíproca se prueba por inducción sobre la definición de función polinomial. Si f es constante o una proyección se tiene claramente la propiedad. Supongamos que la propiedad es válida para f y g, entonces

$$\begin{aligned}(f \cup g)\left(\bigcup_{i\geq 1} L_i\right) &= f\left(\bigcup_{i\geq 1} L_i\right) \cup g\left(\bigcup_{i\geq 1} L_i\right) \\ &\subseteq \bigcup_{i\geq 1} f(L_i) \cup \bigcup_{i\geq 1} g(L_i) \\ &= \bigcup_{i\geq 1} (f \cup g)(L_i)\end{aligned}$$

además

$$\begin{aligned}(fg)\left(\bigcup_{i\geq 1} L_i\right) &= f\left(\bigcup_{i\geq 1} L_i\right) g\left(\bigcup_{i\geq 1} L_i\right) \\ &\subseteq \bigcup_{i\geq 1} f(L_i) \bigcup_{i\geq 1} g(L_i)\end{aligned}$$

Sea $w \in (fg)\left(\bigcup_{i\geq 1} L_i\right)$ entonces por lo anterior $w \in \bigcup_{i\geq 1} f(L_i) \bigcup_{i\geq 1} g(L_i)$ es decir $w = uv$, donde $u \in \bigcup_{i\geq 1} f(L_i)$ y $v \in \bigcup_{i\geq 1} g(L_i)$, por lo tanto $u \in f(L_j)$ y $v \in f(L_k)$ para algún $j, k \geq 1$. Sea $m = \max\{j, k\}$ entonces $uv \in f(L_m)\, g(L_m) = (fg)(L_m)$, es decir $w = uv \in \bigcup_{i\geq 1} (fg)(L_i)$. □

Corolario 3.2.4. *Sea $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^n$ una función polinomial, entonces $\vec{f}$ es monótona y continua.*

Definición 3.2.5. Una n-upla $\vec{L} \in (\wp(\Sigma^*))^n$ es un punto fjo de $\vec{f}$ si $\vec{L} = \vec{f}\left(\vec{L}\right)$.

Teorema 3.2.6. *Sean $f_1, \ldots f_n$ funciones polinomiales sobre $\wp(\Sigma^*)$, entonces $\vec{f} = (f_1, \ldots f_n)$ tiene al menos un punto fijo. Además, la n-upla $\alpha = \bigcup_{i\geq 0} \vec{f}^{\,i}(\vec{\varnothing})$ es el menor punto fijo.*

Demostración. Por el corolario 3.2.4 se tiene que $\vec{f}$ es continua sobre $\wp(\Sigma^*)$, el cual es un CPO, entonces $\wp(\Sigma^*)^n$ también lo es, luego por el teorema 2.2.5 se tiene que α es el menor punto fijo de $\vec{f}$. □

Ejemplo 3.2.7. Al aplicar el teorema anterior a la función polinomial $f(X) = AX \cup B$ introducida en el ejemplo 3.1.6, se obtiene que su menor punto fijo es:

$$\bigcup_{i\geq 0} f^i(\varnothing) = B \cup AB \cup A^2B \cup \cdots = A^*B.$$

El teorema 3.2.6 es de gran importancia, ya que nos asegura la existencia de un punto fijo para toda función polinomial y cómo encontrar el menor punto fijo, sin embargo, la unicidad que es importante para la solución de ecuaciones sobre lenguajes no se tiene en general, como lo muestra el siguiente ejemplo adaptado de [7].

Ejemplo 3.2.8. $f(X) = XX \cup \{aa\}$ tiene infinitos puntos fijos. En particular el lenguaje $L_k = L_0 \cup L_{0k}$, es un punto fijo para todo $k \geq 0$, donde

$$L_0 = \left\{a^{2n} : n \geq 0\right\}$$
$$L_{0k} = \left\{a^{2n+k} : n \geq 0\right\}$$

En efecto, para todo $k \geq 0$ se tiene que

$$\begin{aligned}
L_kL_k &= (L_0 \cup L_{0k})(L_0 \cup L_{0k}) \\
&= L_0L_0 \cup L_0L_{0k} \cup L_{0k}L_0 \cup L_{0k}L_{0k} \\
&= L_0L_0 \cup L_0L_{0k} \cup L_{0k}L_0 \cup L_{0k}L_{0k} \\
&= \left\{a^{2n} : n \geq 0\right\} \cup \left\{a^{2n}a^{2m+k} : n, m \geq 0\right\} \cup \left\{a^{2n+k}a^{2m} : n, m \geq 0\right\} \cup \left\{a^{2n+k}a^{2m+k} : n, m \geq 0\right\} \\
&= \left\{a^{2n} : n \geq 0\right\} \cup \left\{a^{2s+k} : s \geq 0\right\} \cup \left\{a^{2s+k} : s \geq 0\right\} \cup \left\{a^{2s+2k} : s \geq 0\right\} \\
&= \left\{a^{2n} : n \geq 0\right\} \cup \left\{a^{2s+k} : s \geq 0\right\} \\
&= L_0 \cup L_{0k}
\end{aligned}$$

Entonces $L_kL_k \cup \{aa\} = L_0 \cup L_{0k} \cup \{aa\} = L_0 \cup L_{0k} = L_kL_k$, para todo $k \geq 0$.

Capítulo 4

Funciones Polinomiales Lineales

En el capítulo anterior abarcamos una clase de función entre lenguajes formales denominada funciones polinomiales y probamos (teorema 3.2.6) que toda función polinomial tiene al menos un punto fijo y cómo encontrar el menor de ellos; además, mostramos que una función polinomial puede tener infinitos puntos fijos. En este capítulo se introduce un caso particular de función polinomial, llamada Función Polinomial Lineal, la cual tiene un único punto fijo y son las que permiten caracterizar los lenguajes regulares. Además, se demostrará el Lema de Arden y una generalización de éste; en particular se solucionarán ecuaciones sobre lenguajes, como por ejemplo

$$X = AXB \cup C$$
$$X = A_1XB_1 \cup A_2XB_2 \cup \ldots \cup A_nXB_n \cup C$$

donde A, A_i, B, B_i, C $(i = 1, ..., n)$ son lenguajes sobre un alfabeto Σ. Para solucionar esta última ecuación se introduce una nueva operación entre lenguajes llamada inserción simétrica. Esta nueva operación también permitirá definir una nueva clase de lenguajes, que se definen de una manera similar a los lenguajes regulares. Estos resultarán ser una subfamilia de los lenguajes lineales.

4.1. Funciones Polinomiales Lineales

Las funciones polinomiales lineales son una clase particular de funciones polinomiales, que poseen la propiedad de tener un único punto fijo, además, permiten caracterizar los lenguajes regulares como una componente del menor punto fijo de una determinada función polinomial lineal.

Definición 4.1.1. Sea f una función polinomial n-aria $f : (\wp(A^*))^n \longrightarrow \wp(A^*)$, diremos

que f es una **función polinomial lineal en n variables** si es de la forma

$$f(X_1, \ldots, X_n) = A_1 X_1 B_1 \cup \cdots \cup A_n X_n B_n \cup C$$

para todo $X_1, \ldots, X_n \in \wp(A^*)$, donde $A_1, A_2, ..., A_n, B_1, B_2, ..., B_n, C$ son lenguajes sobre el alfabeto A, llamados los coeficientes de f.

Definición 4.1.2. Sea f una función polinomial $f : \wp(A^*) \longrightarrow \wp(A^*)$, diremos que f **es lineal** si es de la forma

$$f(X) = A_1 X B_1 \cup \cdots \cup A_n X B_n \cup C$$

para todo $X \in \wp(A^*)$, donde $A_1, A_2, ..., A_n, B_1, B_2, ..., B_n, C$ son lenguajes sobre el alfabeto A, llamados los coeficientes de f.
En particular, f **es una función lineal a derecha** si es de la forma

$$f(X) = A_1 X \cup \cdots \cup A_n X \cup C$$

para todo $X \in \wp(A^*)$, donde $A_1, A_2, ..., A_n, C$ son lenguajes sobre el alfabeto Σ. Análogamente, f **es una función lineal a izquierda** si es de la forma

$$f(X) = X A_1 \cup \cdots \cup X A_n \cup C$$

para todo $X \in \wp(A^*)$, donde $A_1, A_2, ..., A_n, C$ son lenguajes sobre el alfabeto Σ.

Las funciones lineales son importantes ya que se les puede calcular con cierta facilidad el menor punto fijo y así solucionar algunas ecuaciones sobre lenguajes.

4.2. Lema de Arden

A continuación utilizaremos los teoremas de punto fijo para demostrar el Lema de Arden, el cual garantiza no solo la existencia, sino que también la unicidad de la solución de la ecuación $X = AX \cup B$. Luego aplicaremos el teorema de punto fijo 3.2.6 para una caracterización de los lenguajes regulares y se utilizará el Lema de Arden para encontrar el menor punto fijo.

Lema 4.2.1 (Lema de Arden)**.** *Si A y B son lenguajes sobre un alfabeto Σ y $\lambda \notin A$, entonces la función lineal $T(X) = AX \cup B$ tiene un único punto fijo dado por $X = A^*B$. En otras palabras, la ecuación $AX \cup B = X$ tiene una única solución dada por $X = A^*B$.*

Demostración. La idea es utilizar el teorema de punto fijo 2.3.11, para ello definimos $T(X) =$

$G(X) \cup B$, donde $G(X) = AX$. Probaremos que $G(X)$ preserva uniones contables, en efecto:

$$G\left(\bigcup_{n\geq 0} L_n\right) = A\bigcup_{n\geq 0} L_n = \bigcup_{n\geq 0} AL_n = \bigcup_{n\geq 0} G(L_n)$$

donde $\{L_n\}_{n\geq 0}$ es una familia contable de lenguajes sobre el alfabeto Σ. Además, $G(X)$ es no extensiva, si $L \subseteq G(L) = AL$ y se tuviera que $L \neq \varnothing$, existiría una cadena $u \in L$ de longitud mínima. Entonces $u = vw$, con $v \in A$, $w \in L$. Como $\lambda \notin A$, $v \neq \lambda$; por consiguiente $|w| < |u|$. Esta contradicción muestra que necesariamente $L = \varnothing$, luego G es no extensiva. Así por el corolario 2.3.11, T tiene un único punto fijo dado por

$$X_0 = \bigcup_{n\geq 0} G^n(B) = \bigcup_{n\geq 0} (A^n B) = A^* B$$

tal como se quería. □

Una prueba de este lema utilizando otro teorema de punto fijo, se encuentra en [6].

Corolario 4.2.2. *Si A y B son lenguajes sobre un alfabeto Σ y $\lambda \notin A$, entonces la ecuación $X = XA \cup B$ tiene una única solución dada por $X = BA^*$*

Teorema 4.2.3. *Si A y B son lenguajes sobre un alfabeto Σ y $\lambda \in A$, entonces Z es una solución de la ecuación $X = AX \cup B$ si y sólo si $Z = A^*(B \cup D)$ para algún $D \subseteq \Sigma^*$.*

Demostración. Resulta de aplicar la segunda parte del corolario 2.3.11. □

A continuación presentamos una primera generalización del Lema de Arden.

Lema 4.2.4. *Si A, B y C son lenguajes sobre un alfabeto Σ y $\lambda \notin (A \cup B)$, entonces la ecuación $X = AXB \cup C$ tiene una única solución dada por*

$$X = \bigcup_{n=0}^{\infty} (A^n C B^n)$$

Demostración. Sea $T(X) = G(X) \cup C$, donde $G(X) = AXB$, entonces $G(X)$ preserva uniones contables, en efecto:

$$G\left(\bigcup_{n\geq 0} L_n\right) = A\bigcup_{n\geq 0} L_n B = \bigcup_{n\geq 0} AL_n B = \bigcup_{n\geq 0} G(L_n)$$

donde $\{L_n\}_{n\geq 0}$ es una familia contable de lenguajes sobre el alfabeto Σ. Además, $G(X)$ es no extensiva, si $L \subseteq G(L) = ALB$ y se tuviera que $L \neq \varnothing$, existiría una cadena $u \in L$ de longitud mínima. Entonces $u = vws$, con $v \in A$, $w \in L$, $s \in B$. Como $\lambda \notin (A \cup B)$, entonces $v \neq \lambda$ o $s \neq \lambda$; por consiguiente $|w| < |u|$. Esta contradicción muestra que necesariamente $L = \varnothing$, luego G es

no extensiva. Así por el corolario 2.3.11, T tiene un único punto fijo dado por

$$X_0 = \bigcup_{n\geq 0} G^n(C) = \bigcup_{n\geq 0} (A^n C B^n)$$

□

Lema 4.2.5. *Si A, B y C son lenguajes sobre un alfabeto Σ y $\lambda \in (A \cap B)$, entonces Z es una solución de la ecuación $X = AXB \cup C$ si y sólo si*

$$Z = \bigcup_{n=0}^{\infty} (A^n (C \cup D) B^n)$$

Demostración. Resulta de aplicar la segunda parte del corolario 2.3.11. □

Ejemplo 4.2.6. (i) La ecuación $X = aX \cup b^*ab \cup bX \cup a^*$ se puede escribir de la forma $X = (a \cup b)X \cup (b^*ab \cup a^*)$. Por el Lema de Arden la ecuación tiene solución única $X = (a \cup b)^*(b^*ab \cup a^*)$.

(ii) Sean A, B lenguajes sobre el alfabeto $\{a, b\}^*$ tales que

$$A = aA \cup bB \cup \lambda$$
$$B = bB \cup \lambda$$

entonces aplicando el Lema de Arden a la segunda ecuación, se obtiene que $B = b^*\lambda = b^*$. Nuevamente aplicando el Lema de Arden a la primera ecuación, resulta que $A = a^*(bB \cup \lambda) = a^*(bb^* \cup \lambda) = a^*(b^+ \cup \lambda) = a^*b^*$.

(iii) Sea $X = AXB \cup \{\lambda\}$, donde $A = \{a^n b^n : n \geq 1\}$ y $B = \{b^n a^n : n \geq 1\}$, entonces por el lema 4.2.4 la ecuación tiene una única solución

$$X = \bigcup_{i=0}^{\infty} A^i \{\lambda\} B^i = \bigcup_{i=0}^{\infty} A^i B^i$$

donde $A^i = \left\{(a^n b^n)^i : n \geq 1\right\}$ y $B^i = \left\{(b^n a^n)^i : n \geq 1\right\}$.

(iv) Sea la siguiente gramática lineal

$$\begin{cases} S \to 01 \mid 0A \\ A \to S1 \end{cases}$$

Entonces $S = 01 \cup 0S1$, así $S = \bigcup_{i=0}^{\infty} \left(0^i (01) 1^i\right)$, por consiguiente $L(G) = \left\{0^i 1^i : i \geq 1\right\}$.

4.3. Matrices y Lenguajes

A continuación introducimos una notación matricial sobre lenguajes, que es análoga a la utilizada en el álgebra lineal, con el fin de facilitar la escritura.

Definición 4.3.1. Una matriz de lenguajes M de tamaño $m \times n$ sobre el alfabeto Σ es una función $M : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \wp(\Sigma^*)$. Si $m = n$ decimos que la matriz M es cuadrada.

Los valores de la función $M_{ij} = M(i, j)$ se dispondrán en un arreglo rectangular de m filas y n columnas. Las matrices sobre lenguajes se combinarán de dos maneras, unión y producto, las cuales se definen análogamente a la suma y producto de matrices.

Definición 4.3.2. Sean M y N matrices $m \times n$, entonces la unión $M \cup N$ es una matriz S de tamaño $m \times n$, cuyo elemento ij es $S_{ij} = M_{ij} \cup N_{ij}$.

Ejemplo 4.3.3. Sean las matrices M y N sobre el alfabeto $\Sigma = \{a, b\}$, tales que

$$M = \begin{pmatrix} a & a^*b & \lambda \\ a^+ & b^*a & b \end{pmatrix} \quad N = \begin{pmatrix} ba & b^* & a \\ \lambda & \lambda & a^*b^* \end{pmatrix} \quad \text{entonces} \quad M \cup N = \begin{pmatrix} a \cup ba & a^*b \cup b^* & a \\ a^* & b^*a & b \cup a^*b^* \end{pmatrix}$$

Definición 4.3.4. Sean M una matriz $m \times p$ y N una matriz $p \times n$, entonces el producto MN es una matriz S de tamaño $m \times n$, cuyo elemento ij es $S_{ij} = \bigcup \{M_{ik}N_{kj} : 1 \leq k \leq p\}$.

Ejemplo 4.3.5. Sean las matrices M y N sobre el alfabeto $\Sigma = \{a, b\}$, tales que

$$M = \begin{pmatrix} a^* & \varnothing & \lambda \\ a & b & \lambda \end{pmatrix} \quad N = \begin{pmatrix} a \\ b \\ ab \end{pmatrix} \quad \text{entonces} \quad MN = \begin{pmatrix} a^+ \cup ab \\ a^2 \cup b^2 \cup ab \end{pmatrix}$$

Es claro que la matriz $n \times n$

$$I_n = \begin{pmatrix} \lambda & \varnothing & \cdots & \varnothing \\ \varnothing & \lambda & \cdots & \varnothing \\ \vdots & \vdots & \cdots & \vdots \\ \varnothing & \varnothing & \cdots & \lambda \end{pmatrix}$$

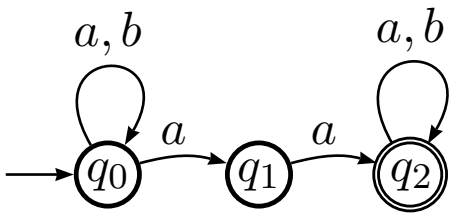
es el elemento identidad respecto al producto, además, si M es una matriz cuadrada, entonces $M^* = \bigcup \{M^i : i \in \mathbb{N}\}$.

Definición 4.3.6. Sea $M = (\Sigma, Q, q_0, F, \Delta)$ un AFN, donde $|Q| = n$. La matriz asociada al autómata M, es una matriz cuadrada $n \times n$ sobre el lenguaje Σ, denotada L_M, definida como

$$(L_M)_{ij} := \left\{ \bigcup_{a \in \Sigma} a : q_j \in \Delta(q_i, a) \right\}$$

para $1 \leq i, j \leq n$.

Ejemplo 4.3.7. El siguiente autómata no determinista M



tiene la siguiente matriz asociada

$$L_M = \begin{pmatrix} a \cup b & a & \varnothing \\ \varnothing & \varnothing & a \\ \varnothing & \varnothing & a \cup b \end{pmatrix}$$

Teorema 4.3.8. *Sea $M = (\Sigma, Q, q_0, F, \Delta)$ un AFN, donde $|Q| = n$ y sea L_M su matriz asociada. Entonces $w \in \left(L_M^{(k)}\right)_{ij}$ si y sólo si $|w| = k$ y $\Delta^*(q_i, w) = q_j$ para $1 \leq i, j \leq n$.*

Demostración. La demostración es por inducción sobre $k \geq 0$. Para $k = 0$ tenemos que $L_M^{(0)} = I_n$. Si $i = j$, entonces $w \in \left(L_M^{(0)}\right)_{ij}$ si y sólo si $w = \lambda$; en cuyo caso $\Delta^*(q_i, \lambda) = q_i = q_j$. Si $i \neq j$, $\left(L_M^{(0)}\right)_{ij} = \varnothing$, se tiene la equivalencia trivialmente.
Supongamos que la propiedad se tiene para $k > 1$, entonces:

$$\begin{aligned}
& w \in \Sigma^*, |w| = k+1 \quad \text{y} \quad \Delta^*(q_i, w) = q_j. \\
& \Leftrightarrow w = ua, \quad \text{con} \quad |u| = k, \Delta^*(q_i, u) = q_s \quad \text{y} \quad \Delta(q_s, a) = q_j \quad \text{para algún} \quad s, 1 \leq s \leq n. \\
& \Leftrightarrow w = ua, \quad \text{con} \quad |u| = k, u \in \left(L_M^{(k)}\right)_{is} \quad \text{y} \quad a \in (L_M)_{sj} \quad \text{para algún} \quad s, 1 \leq s \leq n. \\
& \Leftrightarrow w \in \left(L_M^{(k)}\right)_{is} (L_M)_{sj} \quad \text{para} \quad s, 1 \leq s \leq n. \\
& \Leftrightarrow w \in \left(L_M^{(k+1)}\right)_{ij}.
\end{aligned}$$

quedando demostrado. □

4.4. Puntos Fijos y Lenguajes Regulares

A continuación probaremos que todo lenguaje regular es una componente del menor punto fijo de una función cuyas componentes son funciones polinomiales lineales de n variables.

Teorema 4.4.1. *Sea $M = (\Sigma, Q, q_0, F, \Delta)$ un AFN, donde $|Q| = n$ y sea L_M su matriz asociada. Sea $\vec{Q}_M : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^n$ una función n-aria definida como $\vec{Q}_M(\vec{X}) = L_M\vec{X} \cup$*

$\vec{K}_M$, *donde* $\vec{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}$, $\vec{K}_M = \begin{pmatrix} K_0 \\ \vdots \\ K_{n-1} \end{pmatrix}$ *tal que* $X_1, \ldots, X_n \in \wp(\Sigma^*)$ *y* $K_i = \begin{cases} \lambda, & si \quad q_i \in F \\ \varnothing, & si \quad q_i \notin F \end{cases}$

Entonces el menor punto fijo de $\vec{Q}_M$ *es la n-upla* $\vec{A} = \begin{pmatrix} A_0 \\ \vdots \\ A_{n-1} \end{pmatrix}$ *donde* $A_i = \{w \in \Sigma^* : \Delta(q_i, w) \cap F \neq \varnothing\}$

Demostración. $\vec{Q}_M$ se puede escribir como $\vec{Q}_M = (f_1, f_2, \ldots, f_n)$ con $f_i(\vec{X}) = (L_M)_{i1} X_1 \cup (L_M)_{i2} X_2 \cup \cdots \cup (L_M)_{in} X_n \cup K_i$, es claro que f_i es una función polinomial lineal de n variables, luego por la definición 3.1.7, $\vec{Q}_M$ es polinomial. Así por el teorema 3.2.6, $\vec{Q}_M$ tiene un punto fijo, donde el menor esta dado por

$$\alpha = \bigcup_{n \geq 0} \vec{Q}_M^n (\vec{\varnothing})$$

Por inducción sobre n se verifica que

$$\vec{Q}_M^n (\vec{\varnothing}) = (L_M)^n \vec{K}_M \cup (L_M)^{n-1} \vec{K}_M \cup \cdots \cup (L_M) \vec{K}_M \cup \vec{K}_M$$

por lo tanto $\alpha = L_M^* \vec{K}_M \in (\wp(\Sigma^*))^n$, falta verificar que la componente $i - esima$ de α corresponde a A_i. En efecto

$$\begin{aligned}
\left(L_M^* \vec{K}_M\right)_i &= \bigcup_{j=1}^{n} (L_M^*)_{ij} K_j \\
&= \bigcup_{j=1}^{n} \left(\bigcup_{n \geq 0} L_M^{(n)} \right)_{ij} K_j \\
&= \bigcup_{n \geq 0} \bigcup_{j=1}^{n} \left(L_M^{(n)} \right)_{ij} K_j \\
&= \bigcup_{n \geq 0} \bigcup_{j=1}^{n} \{w \in \Sigma^* : \Delta^*(q_i, w) = q_j, |w| = n\} K_j \quad \text{por el teorema 4.3.8} \\
&= \bigcup_{n \geq 0} \bigcup_{j=1}^{n} \{w \in \Sigma^* : \Delta^*(q_i, w) = q_j, |w| = n, q_j \in F\} \quad \text{por def. de } \vec{K}_M, \text{ para } 0 \leq i \leq n-1.
\end{aligned}$$

así $\left(L_M^* \vec{K}_M\right)_i = \{w \in \Sigma^* : \Delta^*(q_i, w) \in F\} = A_i$ □

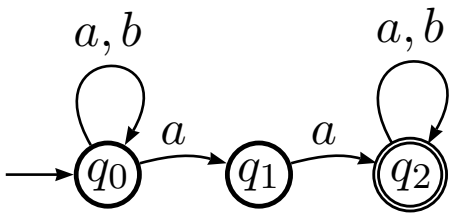
Corolario 4.4.2. *Todo lenguaje regular es una componente del menor punto fijo de una función* ($\vec{Q}_M$)*, en particular la componente* A_0.

Demostración. Sea L un lenguaje regular entonces existe un AFN M tal que $L(M) = L$. Sea

$\vec{Q}_M$ la función asociada a M, entonces por el teorema 4.4.1, su menor punto fijo es $\vec{A} = \begin{pmatrix} A_0 \\ \vdots \\ A_{n-1} \end{pmatrix}$ donde $A_i = \{w \in \Sigma^* : \Delta(q_i, w) \cap F \neq \varnothing\}$, es claro que $A_0 = L$. □

Para poder encontrar el menor punto fijo, se puede utilizar sucesivas veces el lema de Arden, como muestra el siguiente ejemplo.

Ejemplo 4.4.3. Sea el siguiente autómata no determinista M considerado en el ejemplo 4.3.7.



Se tiene que

$$\vec{Q}_M(\vec{X}) = L_M\vec{X} \cup \vec{K}_M = \begin{pmatrix} a \cup b & a & \varnothing \\ \varnothing & \varnothing & a \\ \varnothing & \varnothing & a \cup b \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} \cup \begin{pmatrix} \varnothing \\ \varnothing \\ \lambda \end{pmatrix}$$

$\vec{Q}_M$ se puede escribir como

$$\vec{Q}_M \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} (a \cup b)X_1 \cup aX_2 \\ aX_3 \\ (a \cup b)X_3 \cup \lambda \end{pmatrix}$$

Sean $A_i = \{w \in \Sigma^* : \Delta(q_i, w) \cap F \neq \varnothing\}\ (0 \leq i \leq 2)$ las componentes del menor punto fijo de $\vec{Q}_M$, encontrar estas es equivalente a solucionar el sistema de ecuaciones

$$\begin{cases} (1) \quad X_1 = (a \cup b)X_1 \cup aX_2 \\ (2) \quad X_2 = aX_3 \\ (3) \quad X_3 = (a \cup b)X_3 \cup \lambda \end{cases}$$

Aplicando el lema de Arden a la ecuación (3):
(4) $X_3 = (a \cup b)^*$
Reemplazando (4) en (2):
(5) $X_2 = a(a \cup b)^*$
Reemplazando (5) en (1):

(6) $X_1 = (a \cup b)X_1 \cup a^2(a \cup b)^*$

Aplicando el lema de Arden a la ecuación (6), concluimos que:

(7) $X_1 = (a \cup b)^* a^2 (a \cup b)^*$

Esto implica que el menor punto fijo de $\vec{Q}_M$ es

$$\begin{pmatrix} (a \cup b)^* a^2 (a \cup b)^* \\ a(a \cup b)^* \\ (a \cup b)^* \end{pmatrix}$$

y que el lenguaje regular $(a \cup b)^* a^2 (a \cup b)^*$ es el lenguaje aceptado por el autómata M y dicho lenguaje se puede ver como una componente del menor punto fijo de la función $\vec{Q}_M$.

4.5. Lenguajes Simétricos

En esta sección se generaliza el Lema de Arden y se utiliza para caracterizar una subfamilia de los lenguajes lineales. Para esto se presenta una operación denominada inserción simétrica [3].

4.5.1. Inserción Simétrica

La inserción simétrica es una nueva operación que simplifica la escritura a la hora de solucionar ecuaciones lineales más complejas que las trabajadas hasta el momento, además, las operaciones básicas (unión, concatenación y estrella de Kleene) no son suficientes para caracterizar lenguajes lineales, ya que no todo lenguaje lineal es regular.

Definición 4.5.1. Dados los lenguajes $A_1, A_2, ..., A_k, B_1, B_2, ..., B_k, C$ sobre un alfabeto Σ, la inserción simétrica de C en $\{(A_i, B_i)\}_{i=1,...,k}$, notada

$$\left\langle A_1, A_2, ..., A_k \overset{\downarrow}{}_C B_1, B_2, ..., B_k \right\rangle$$

se define como

$$\left\langle A_1, A_2, ..., A_k \overset{\downarrow}{}_C B_1, B_2, ..., B_k \right\rangle = \bigcup_{m \geq 1} H_m$$

donde

$$\begin{cases} H_1 = \bigcup\limits_{\substack{i=1,...k \\ n_i \geq 0}} A_i^{n_i} C B_i^{n_i}, \\ H_{m+1} = \bigcup\limits_{\substack{i=1,...k \\ n_i \geq 0}} A_i^{n_i} H_m B_i^{n_i} \end{cases}$$

Es claro que $H_1 \subseteq H_2 \subseteq H_3 \subseteq \cdots \subseteq H_n \subseteq \cdots$, luego $\bigcup_{m\geq 1} H_m = \lim_{m\to\infty} H_m$.

Ejemplo 4.5.2. (i) El siguiente ejemplo muestra un lenguaje no regular, que se puede escribir en términos de la inserción simétrica. Dado el alfabeto $\Sigma = \{a, b\}$ y los lenguajes $A_1 = \{a\}, B_1 = \{b\}$ y $C = \{\lambda\}$, entonces

$$\begin{aligned} H_1 &= \{\lambda\} \cup \{ab\} \cup \left\{a^2b^2\right\} \cup \left\{a^3b^3\right\} \cup \cdots = H_2 = H_3 = \cdots \\ \text{luego} \quad \left\langle \{a\} \overset{|}{}_{\{\lambda\}} \{b\} \right\rangle &= \bigcup_{i\geq 0} \left(\{a\}^i \{\lambda\} \{b\}^i \right) = \{\lambda\} \cup \{ab\} \cup \left\{a^2b^2\right\} \cup \left\{a^3b^3\right\} \cup \cdots \\ &= \left\{a^n b^n : n \geq 0\right\} \end{aligned}$$

(ii) Dado el alfabeto $\Sigma = \{a, b, c\}$ y los lenguajes $A_1 = \{a\}, B_1 = \{b\}$ y $C = \{c\}$, entonces

$$\begin{aligned} H_1 &= \{c\} \cup \{acb\} \cup \left\{a^2cb^2\right\} \cup \left\{a^3cb^3\right\} \cup \cdots = H_2 = H_3 = \cdots \\ \text{luego} \quad \left\langle \{a\} \overset{|}{}_{\{c\}} \{b\} \right\rangle &= \bigcup_{i\geq 0} \left(\{a\}^i \{c\} \{b\}^i \right) = \{c\} \cup \{acb\} \cup \left\{a^2cb^2\right\} \cup \left\{a^3cb^3\right\} \cup \cdots \\ &= \left\{a^n cb^n : n \geq 0\right\} \end{aligned}$$

(iii) Dado el alfabeto $\Sigma = \{a, b\}$ y los lenguajes $A_1 = A_2 = \{a\}, B_1 = B_2 = \{b\}$ y $C = \{\lambda\}$, entonces

$$\begin{aligned} H_1 &= \{\lambda\} \cup \{aa\} \cup \{aaaa\} \cup \{aaaaaa\} \cup \cdots \cup \{bb\} \cup \{bbbb\} \cup \{bbbbbb\} \cup \cdots \\ H_2 &= H_1 \cup \{baab\} \cup \{baaaab\} \cup \{baaaaaab\} \cup \cdots \cup \{bbaabb\} \cup \{bbaaaabb\} \cup \{bbaaaaaabb\} \cup \cdots \\ &\quad \cup \{abba\} \cup \{abbbba\} \cup \{abbbbbba\} \cup \{aabbaa\} \cup \{aabbbbaa\} \cup \{aabbbbbbaa\} \cup \cdots \\ &\vdots \\ \text{luego} \quad \left\langle \{a\}, \{b\} \overset{|}{}_{\{\lambda\}} \{a\}, \{b\} \right\rangle &= \left\{ww^R : w \in \{a, b\}\right\} \end{aligned}$$

Para facilitar la escritura se usará la notación vectorial $\left\langle \vec{A} \overset{|}{}_{C} \vec{B} \right\rangle$ para representar la inserción simétrica $\left\langle A_1, A_2, ..., A_k \overset{|}{}_{C} B_1, B_2, ..., B_k \right\rangle$, el lenguaje $\{\lambda\}$ se representará por $\boldsymbol{\lambda}$ y en algunas ocasiones se usará $\left\langle \vec{A} | \vec{B} \right\rangle \lhd C$, para representar $\left\langle A \overset{|}{}_{C} B \right\rangle$, esto en el caso que el lenguaje C sea muy extenso.

El siguiente lema será de gran ayuda para simplificar algunas cuentas.

Lema 4.5.3. *Dados los lenguajes $A_1, A_2, ..., A_k, B_1, B_2, ..., B_k, C$ sobre un alfabeto Σ, enton-*

ces

$$\bigcup_{m=1}^{n} H_m = \bigcup_{\substack{r=1,\cdots,n,\ n_{i_j}\geq 0 \\ 1\leq i_1,i_2,\ldots,i_r\leq k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_r}^{n_{i_r}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_r}^{n_{i_r}}$$

Demostración. Lo probaremos por inducción sobre n. En efecto para $n = 1$

$$\begin{aligned} H_1 &= \bigcup_{\substack{i=1,\ldots k \\ n_i\geq 0}} A_i^{n_i} C B_i^{n_i}, \\ &= \bigcup_{n_1\geq 0} A_1^{n_1} C B_1^{n_1} \cup \bigcup_{n_2\geq 0} A_2^{n_2} C B_2^{n_2} \cup \cdots \cup \bigcup_{n_k\geq 0} A_k^{n_k} C B_k^{n_k} \\ &= \bigcup_{\substack{r=1,\ n_{i_j}\geq 0 \\ 1\leq i_1\leq k}} A_{i_r}^{n_{i_r}} C B_{i_r}^{n_{i_r}} \end{aligned}$$

Para el paso inductivo

$$\begin{aligned} \bigcup_{m=1}^{n+1} H_m &= H_{n+1} \\ &= \bigcup_{\substack{i=1,\ldots k \\ n_i\geq 0}} A_i^{n_i} H_n B_i^{n_i}, \\ &= \bigcup_{\substack{i=1,\ldots k \\ n_i\geq 0}} A_i^{n_i} \left(\bigcup_{\substack{r=1,\cdots,n,\ n_{i_j}\geq 0 \\ 1\leq i_1,i_2,\ldots,i_r\leq k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_r}^{n_{i_r}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_r}^{n_{i_r}} \right) B_i^{n_i} \\ &= \bigcup_{\substack{r=1,\cdots,n+1,\ n_{i_j}\geq 0 \\ 1\leq i_1,i_2,\ldots,i_r\leq k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_r}^{n_{i_r}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_r}^{n_{i_r}} \end{aligned}$$

□

Corolario 4.5.4. *Dados los lenguajes $A_1, A_2, ..., A_k, B_1, B_2, ..., B_k, C$ sobre un alfabeto Σ, entonces*

$$\left\langle A_1, A_2, ..., A_k \overset{\downarrow}{}_C B_1, B_2, ..., B_k \right\rangle = \bigcup_{\substack{r\geq 1,\ n_{i_j}\geq 0 \\ 1\leq i_1,i_2,\ldots,i_r\leq k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_r}^{n_{i_r}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_r}^{n_{i_r}}$$

Las siguiente proposición reúne una serie de propiedades básicas de la inserción simétrica.

Proposición 4.5.5. *Sean $C, A, B, A_1, A_2, ..., A_k, B_1, B_2, ..., B_k$ lenguajes sobre Σ. Entonces*

1. $\left\langle \vec{\boldsymbol{\lambda}} \overset{|}{_C} \vec{\boldsymbol{\lambda}} \right\rangle = C$
2. $\left\langle A \overset{|}{_{\boldsymbol{\lambda}}} \boldsymbol{\lambda} \right\rangle = A^*.$
   $\left\langle \boldsymbol{\lambda} \overset{|}{_{\boldsymbol{\lambda}}} B \right\rangle = B^*.$
3. $\left\langle A \overset{|}{_C} \boldsymbol{\lambda} \right\rangle = A^*C.$
   $\left\langle \boldsymbol{\lambda} \overset{|}{_C} B \right\rangle = CB^*.$
4. $\left\langle A \overset{|}{_C} B \right\rangle = \bigcup_{n\geq 0} A^n C B^n.$
5. $\left\langle A, \boldsymbol{\lambda} \overset{|}{_C} \boldsymbol{\lambda}, B \right\rangle = A^*CB^*.$
6. $\left\langle \vec{A} \overset{|}{_C} \boldsymbol{\lambda} \right\rangle = \left(A_1^* A_2^* \dots A_k^*\right)^* C = \left(A_1 \cup A_2 \cup \cdots \cup A_k\right)^* C.$
   $\left\langle \boldsymbol{\lambda} \overset{|}{_C} \vec{B} \right\rangle = C\left(B_1^* B_2^* \dots B_k^*\right)^* = C\left(B_1 \cup B_2 \cup \cdots \cup B_k\right)^*.$
7. $\left\langle \vec{A} \overset{|}{_C} \vec{B} \right\rangle = \left\langle A_{\pi(1)}, A_{\pi(2)}, \dots, A_{\pi(k)} \overset{|}{_C} B_{\pi(1)}, B_{\pi(2)}, \dots, B_{\pi(k)} \right\rangle$, *para toda permutación π de $\{1, 2, \dots, k\}$.*
8. *La inserción simétrica es distributiva con respecto a uniones arbitrarias, es decir, para una familia de lenguajes $\{C_i\}_{i\in I}$ sobre Σ*

$$\left\langle \vec{A} | \vec{B} \right\rangle \lhd \bigcup_{i\in I} C_i = \bigcup_{i\in I} \left\langle \vec{A} | \vec{B} \right\rangle \lhd C_i$$

9. $\left\langle A_1, A_2, ..., A_{i-1}, \boldsymbol{\lambda}, A_i, \dots, A_k \overset{|}{_C} B_1, B_2, ..., B_{i-1}, \boldsymbol{\lambda}, B_i, \dots, B_k \right\rangle$
   $= \left\langle A_1, A_2, ..., A_{i-1}, A_i, \dots, A_k \overset{|}{_C} B_1, B_2, ..., B_{i-1}, B_i, \dots, B_k \right\rangle = \left\langle \vec{A} \overset{|}{_C} \vec{B} \right\rangle$

Demostración. 1. Probaremos que $H_i = C$ para todo entero positivo i. En efecto $H_1 = \bigcup_{\substack{i=1,...k \\ n_i\geq 0}} \boldsymbol{\lambda}^{n_i} C \boldsymbol{\lambda}^{n_i} = C$. Para el paso inductivo, $H_{m+1} = \bigcup_{\substack{i=1,...k \\ n_i\geq 0}} \boldsymbol{\lambda}_i^{n_i} H_m \boldsymbol{\lambda}^{n_i} = \bigcup_{\substack{i=1,...k \\ n_i\geq 0}} \boldsymbol{\lambda}_i^{n_i} C \boldsymbol{\lambda}^{n_i} = C$, por tanto $\left\langle \vec{\boldsymbol{\lambda}} \overset{|}{_C} \vec{\boldsymbol{\lambda}} \right\rangle = C$.

2. Probaremos que $H_i = A^*$ para todo entero positivo i. En efecto $H_1 = \bigcup_{i\geq 0} A^i \boldsymbol{\lambda} \boldsymbol{\lambda}^i = \bigcup_{i\geq 0} A^i = A^*$. Para el paso inductivo, $H_{m+1} = \bigcup_{i\geq 0} A^i H_m \boldsymbol{\lambda}^i = \bigcup_{i\geq 0} A^i A^* = A^*$, por lo tanto $\left\langle A \overset{|}{_{\boldsymbol{\lambda}}} \boldsymbol{\lambda} \right\rangle = A^*$. La segunda parte es análoga.

3. Probaremos que $H_i = A^*C$ para todo entero positivo i. En efecto $H_1 = \bigcup_{i\geq 0} A^i C\boldsymbol{\lambda}^i = \bigcup_{i\geq 0} A^i C = A^*C$. Para el paso inductivo, $H_{m+1} = \bigcup_{i\geq 0} A^i H_m \boldsymbol{\lambda}^i = \bigcup_{i\geq 0} A^i A^*C = A^*C$, por lo tanto $\left\langle A \overset{|}{}_C \boldsymbol{\lambda} \right\rangle = A^*C$. La segunda parte es análoga.

4. Probaremos que $H_i = \bigcup_{i\geq 0} A^i C B^i$ para todo entero positivo i. En efecto $H_1 = \bigcup_{i\geq 0} A^i C B^i$. Para el paso inductivo, $H_{m+1} = \bigcup_{i\geq 0} A^i H_m B^i = \bigcup_{i\geq 0} A^i \left(\bigcup_{j\geq 0} A^j C B^j \right) B^i = \bigcup_{i\geq 0} A^i C B^i$, por lo tanto $\left\langle A \overset{|}{}_C B \right\rangle = \bigcup_{n\geq 0} A^n C B^n$.

5. Sean $A_1 = A, A_2 = \boldsymbol{\lambda} = B_1, B_2 = B$, entonces $H_1 = A^*C \cup CB^*$. Además $H_i = A^*CB^*$ para todo entero mayor o igual a 2. Por lo tanto $\left\langle A, \boldsymbol{\lambda} \overset{|}{}_C \boldsymbol{\lambda}, B \right\rangle = A^*C \cup CB^* \cup A^*CB^* = A^*CB^*$.

6. Probaremos que $H_i = (A_1 \cup A_2 \cup \cdots \cup A_k)^i C$ para todo entero positivo i. En efecto

$$\begin{aligned} H_1 &= \bigcup_{\substack{i=1,\ldots k \\ n_i \geq 0}} A_i^{n_i} C \boldsymbol{\lambda}^{n_i} \\ &= \bigcup_{\substack{i=1,\ldots k \\ n_i \geq 0}} A_i^{n_i} C \\ &= (A_1^* \cup A_2^* \cup \cdots \cup A_k^*)\, C \end{aligned}$$

Para el paso inductivo,

$$\begin{aligned} H_{m+1} &= \bigcup_{\substack{i=1,\ldots k \\ n_i \geq 0}} A_i^{n_i} H_m \boldsymbol{\lambda}^{n_i} \\ &= \bigcup_{\substack{i=1,\ldots k \\ n_i \geq 0}} A_i^{n_i} H_m \\ &= \bigcup_{\substack{i=1,\ldots k \\ n_i \geq 0}} A_i^{n_i} (A_1^* \cup A_2^* \cup \cdots \cup A_k^*)^m C \\ &= (A_1^* \cup A_2^* \cup \cdots \cup A_k^*)^{m+1} C \end{aligned}$$

luego

$$\begin{aligned}\left\langle \vec{A} \underset{C}{|} \boldsymbol{\lambda} \right\rangle &= \bigcup_{m\geq 1} H_m \\ &= \bigcup_{m\geq 1} \left(A_1^* \cup A_2^* \cup \cdots \cup A_k^*\right)^m C \\ &= \left(A_1^* \cup A_2^* \cup \cdots \cup A_k^*\right)^* C \\ &= \left((A_1^*)^* \cup (A_2^*)^* \cup \cdots \cup (A_k^*)^*\right) C \\ &= \left(A_1^* A_2^* \ldots A_k^*\right)^* C \\ &= \left(A_1^* \cup A_2^* \cup \cdots \cup A_k^*\right) C\end{aligned}$$

7. Directamente de la definición.

8. Se tiene ya que la concatenación es distributiva con respecto a uniones arbitrarias.

9. Directamente de la definición.

□

Es claro que todo lenguaje regular se puede escribir en términos de inserción simétrica. El recíproco no es cierto, como se mostró en el ejemplo 4.5.2.

Con esta nueva operación, el lema 4.2.4 se puede escribir de una manera.

Teorema 4.5.6. *Si A, B y C son lenguajes sobre un alfabeto Σ y $\lambda \notin (A \cup B)$, entonces la ecuación $X = AXB \cup C$ tiene una única solución dada por*

$$X = \left\langle A \underset{C}{|} B \right\rangle$$

4.5.2. Generalización Lema de Arden

En esta sección se solucionarán varios tipos de ecuaciones lineal sobre lenguajes, en particular la ecuación

$$X = A_1 X B_1 \cup A_2 X B_2 \cup \cdots \cup A_k X B_k \cup C$$

que es una generalización del Lema de Arden.

Teorema 4.5.7 (Generalización Lema de Arden)**.** *Sean $A_1, A_2, ..., A_k, B_1, B_2, ..., B_k, C$ lenguajes sobre el alfabeto Σ tales que $\lambda \notin (A_i \cup B_i)$ para $i = 1, \ldots k$, entonces*

$$X = A_1 X B_1 \cup A_2 X B_2 \cup \cdots \cup A_k X B_k \cup C$$

tiene una única solución dada por

$$X = \left\langle A_1, A_2, ..., A_k \overset{|}{_C} B_1, B_2, ..., B_k \right\rangle = \left\langle \vec{A} \overset{|}{_C} \vec{B} \right\rangle$$

Demostración. Sea $T(X) = \bigcup_{i=1}^{k} A_i X B_i \cup C$, entonces T es una función polinomial. T se puede escribir como $T(X) = G(X) \cup C$ donde $G(X) = \bigcup_{i=1}^{k} A_i X B_i$. Veamos que G preserva uniones contables.

$$G\left(\bigcup_{n\geq 0} L_n\right) = \bigcup_{i=1}^{k}\left(A_i\left(\bigcup_{n\geq 0} L_n\right)B_i\right) = \bigcup_{i=1}^{k} A_1 L_0 B_i \cup \bigcup_{i=1}^{k} A_i L_1 B_i \cup \cdots \cup \bigcup_{i=1}^{k} A_1 L_n B_i \cup \cdots$$
$$= G(L_0) \cup G(L_1) \cup \cdots G(L_n) \cup \cdots = \bigcup_{n\geq 0} G(L_n)$$

donde $\{L_n\}_{n\geq 0}$ es una familia contable de lenguajes sobre el alfabeto Σ. Además, $G(X)$ es no extensiva, si $L \subseteq G(L) = \bigcup_{i=1}^{k} A_i L B_i$ y se tuviera que $L \neq \varnothing$, existiría una cadena $u \in L$ de longitud mínima. Entonces $u = vws$, con $v \in A_i$, $w \in L$, $s \in B_i$ para algún i $(1 \leq i \leq k)$. Como $\lambda \notin (A_i \cup B_i)$ para $i = 1, \ldots k$, entonces $v \neq \lambda$ o $s \neq \lambda$; por consiguiente $|w| < |u|$. Esta contradicción muestra que necesariamente $L = \varnothing$, luego G es no extensiva. Así por el corolario 2.3.11, T tiene un único punto fijo dado por

$$X_0 = \bigcup_{n\geq 0} G^n(C)$$

Ahora bien, las primeras iteraciones de G son

$$\begin{aligned}
G^0(C) &= C \\
G^1(C) &= \bigcup_{i_1=1}^{k} A_{i_1} C B_{i_1} \\
G^2(C) &= \bigcup_{i_2=1}^{k} \bigcup_{i_1=1}^{k} A_{i_2} A_{i_1} C B_{i_1} B_{i_2} \\
G^3(C) &= \bigcup_{i_3=1}^{k} \bigcup_{i_2=1}^{k} \bigcup_{i_1=1}^{k} A_{i_3} A_{i_2} A_{i_1} C B_{i_1} B_{i_2} B_{i_3} \\
&\vdots
\end{aligned}$$

Por inducción sobre n se verifica que

$$G^n(C) = \bigcup_{i_n=1}^{k} \bigcup_{i_{n-1}=1}^{k} \cdots \bigcup_{i_1=1}^{k} A_{i_n} A_{i_{n-1}} \cdots A_{i_1} C B_{i_1} \cdots B_{i_{n-1}} B_{i_n} \quad \forall n \geq 0$$

Además:

$$G^0(C) \cup G^1(C) = \bigcup_{\substack{0\le n_{i_j}\le 1,\\ 1\le i_1\le k}} A_{i_1}^{n_{i_1}} C B_{i_1}^{n_{i_1}}$$

$$G^0(C) \cup G^1(C) \cup G^2(C) = \bigcup_{\substack{0\le n_{i_j}\le 2,\ n_{i_1}+n_{i_2}\le 2\\ 1\le i_1,i_2\le k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}}$$

$$G^0(C) \cup G^1(C) \cup G^2(C) \cup G^3(C) = \bigcup_{\substack{0\le n_{i_j}\le 3,\ n_{i_1}+n_{i_2}+n_{i_3}\le 3\\ 1\le i_1,i_2,i_3\le k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} A_{i_3}^{n_{i_3}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} B_{i_3}^{n_{i_3}}$$

$$\vdots$$

Por inducción sobre n se verifica que

$$\bigcup_{s=0}^{n} G^s(C) = \bigcup_{\substack{0\le n_{i_j}\le n,\ n_{i_1}+\cdots+n_{i_n}\le n\\ 1\le i_1,\ldots,i_n\le k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_n}^{n_{i_n}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_n}^{n_{i_n}}$$

Por lo tanto

$$\begin{aligned} X_0 &= \bigcup_{n\ge 0} G^n(C) \\ &= \bigcup_{\substack{r\ge 1,\ 0\le n_{i_j}\\ 1\le i_1,\ldots,i_r\le k}} A_{i_1}^{n_{i_1}} A_{i_2}^{n_{i_2}} \cdots A_{i_r}^{n_{i_r}} C B_{i_1}^{n_{i_1}} B_{i_2}^{n_{i_2}} \cdots B_{i_r}^{n_{i_r}} \\ &= \left\langle A_1, A_2, ..., A_k \overset{|}{}_C B_1, B_2, ..., B_k \right\rangle \end{aligned}$$

quedando demostrado el teorema. □

En el siguiente corolario reunimos las soluciones de algunas ecuaciones particulares.

Corolario 4.5.8. *Sean $A_1, A_2, ..., A_k, B_1, B_2, ..., B_k, C$ lenguajes sobre el alfabeto Σ, entonces*

- *La ecuación $X = A_1X \cup A_2X \cup \cdots \cup A_kX \cup C$ donde $\lambda \notin A_i$ $(1 \le i \le k)$, tiene una única solución dada por*

  $$X = \left\langle A_1, A_2, ..., A_k \overset{|}{}_C \boldsymbol{\lambda}, \boldsymbol{\lambda}, ..., \boldsymbol{\lambda} \right\rangle = \left\langle \vec{A} \overset{|}{}_C \vec{\boldsymbol{\lambda}} \right\rangle = (A_1^* A_2^* \ldots A_k^*)^* C = (A_1 \cup A_2 \cup \cdots \cup A_k)^* C.$$

- *La ecuación $X = A_1X \cup XB_1 \cup C$ donde $\lambda \notin A_1, B_1$, tiene una única solución dada por*

  $$X = \left\langle A_1, \boldsymbol{\lambda} \overset{|}{}_C \boldsymbol{\lambda}, B_1 \right\rangle = A_1^* C B_1^*$$

- *La ecuación $X = A_1XB_1 \cup A_2X \cup XB_2 \cup C$ donde $\lambda \notin A_1 \cup B_1, \lambda \notin A_2, B_2$, tiene una*

única solución dada por

$$X = \left\langle A_1, A_2, \boldsymbol{\lambda} \overset{|}{_C} B_1, \boldsymbol{\lambda}, B_2 \right\rangle$$

A continuación solucionaremos algunas ecuaciones y sistemas de ecuaciones particulares.

Ejemplo 4.5.9. (i) La ecuación $X = aX \cup Xb \cup c$ tiene solución única $X = a^*cb^*$.

(ii) Sea la siguiente gramática lineal

$$\begin{cases} S \to aSb \mid aAb \\ A \to aAa \mid bAb \mid \lambda \end{cases}$$

Entonces $A = aAa \cup bAb \cup \lambda$, así $A = \left\langle \{a\}, \{b\} \overset{|}{_{\{\lambda\}}} \{a\}, \{b\} \right\rangle = \left\{ ww^R : w \in \{a,b\} \right\}$. Además, $S = aSb \cup aAb = aSb \cup C = \left\langle \{a\} \overset{|}{C} \{b\} \right\rangle$, donde $C = \left\{ aww^Rb : w \in \{a,b\} \right\}$, luego $S = \{a^ncb^n : c \in C, n \geq 0\}$ por consiguiente $L(G) = \left\{ a^nww^Rb^n : w \in \{a,b\}, n \geq 1 \right\}$.

4.5.3. Lenguajes Simétricos y Lineales

A continuación vamos a definir un nueva clase de lenguaje llamada Lenguajes Simétricos. Estos se definen recursivamente de forma similar a los lenguajes regulares, además, resultan ser una subfamilia de los lenguajes lineales. Para poder pasar de un lenguaje simétrico a una gramática lineal se utilizará la generalización del Lema de Arden.

Definición 4.5.10. Sea Σ un alfabeto:

1. $\varnothing$ y $\{v\}$, para toda $v \in \Sigma^*$ son lenguajes simétricos sobre Σ. Estos se denominan lenguajes simétricos básicos.

2. Si A y B son lenguajes simétricos sobre Σ, también lo son:

$A \cup B$	Unión
$\{x\} A \{y\} \quad \forall x, y \in \Sigma^*$	Concatenación Restringida
$\left\langle \{x_1\}, \{x_2\}, \ldots, \{x_n\} \overset{|}{_A} \{y_1\}, \{y_2\}, \ldots, \{y_n\} \right\rangle$	Inserción simétrica básica

Ejemplo 4.5.11. Sea $\Sigma = \{a,b\}$. Los siguientes son lenguajes simétricos sobre Σ:

1. $\left\{ a^ib^j : i < j \right\} = \left\langle \{a\} \left\langle \boldsymbol{\lambda} \overset{|}{_{\{b\}}} \{b\} \right\rangle \{b\} \right\rangle$

2. $\left\{ a^ib^j : i < 2j \right\} = \left\langle \{aa\} \left\langle \boldsymbol{\lambda} \overset{|}{_{\{ab\}}} \{b\} \right\rangle \{b\} \right\rangle \cup \left\langle \{aa\} \left\langle \boldsymbol{\lambda} \overset{|}{_{\{b\}}} \{b\} \right\rangle \{b\} \right\rangle$

3. $\left\{ww^R : w \in \Sigma^*\right\} = \langle \{a\}, \{b\} \,|\, \{a\}, \{b\} \rangle \lhd (\boldsymbol{\lambda})$

Sea $\Sigma = \{a, b, c\}$. Las siguientes son lenguajes simétricos sobre Σ:

4. $\left\{a^i b^i c^j : i \geq 0, j \geq 0\right\} = \left\langle \boldsymbol{\lambda} \left\langle \begin{smallmatrix} | \\ \{a\} \overset{|}{\boldsymbol{\lambda}} \{b\} \end{smallmatrix} \right\rangle \{c\} \right\rangle$

5. $\left\{a^i b^j c^k : i + j = k\right\} = \left\langle \{a\} \left\langle \begin{smallmatrix} | \\ \{b\} \overset{|}{\boldsymbol{\lambda}} \{c\} \end{smallmatrix} \right\rangle \{c\} \right\rangle$

Es claro que todo lenguaje finito es lineal.

Con el propósito de simplificar la descripción de los lenguajes simétricos se introduce una nueva notación.
Sea Σ un alfabeto:

1. $\varnothing$ representa la expresión simétrica básica $\varnothing$.
2. v respresenta la expresión simétrica básica $\{v\}$, $v \in \Sigma^*$.
3. Si α y β representan expresiones simétricas básicas sobre Σ, también lo son:

$\alpha \cup \beta$	Unión
$x \alpha y \quad \forall x, y \in \Sigma^*$	Concatenación Restringida
$\left\langle x_1, x_2, \ldots, x_n \overset{|}{\alpha} y_1, y_2, \ldots, y_n \right\rangle = \left\langle \vec{x} \overset{|}{\alpha} \vec{y} \right\rangle$	Inserción simétrica básica

Los anteriores ejemplos se pueden reescribir como:

Ejemplo 4.5.12. Sea $\Sigma = \{a, b\}$. Las siguientes son expresiones simétricas básicas sobre Σ:

1. $\left\{a^i b^j : i < j\right\} = \left\langle a \left\langle \begin{smallmatrix} | \\ \boldsymbol{\lambda} \overset{|}{b} b \end{smallmatrix} \right\rangle b \right\rangle$

2. $\left\{a^i b^j : i < 2j\right\} = \left\langle aa \left\langle \begin{smallmatrix} | \\ \boldsymbol{\lambda} \overset{|}{ab} b \end{smallmatrix} \right\rangle b \right\rangle \bigcup \left\langle aa \left\langle \begin{smallmatrix} | \\ \boldsymbol{\lambda} \overset{|}{b} b \end{smallmatrix} \right\rangle b \right\rangle$

3. $\left\{ww^R : w \in \Sigma^*\right\} = \langle a, b | a, b \rangle \lhd \boldsymbol{\lambda}$

Sea $\Sigma = \{a, b, c\}$. Las siguientes son expresiones simétricas básicas sobre Σ:

4. $\left\{a^i b^i c^j : i \geq 0, j \geq 0\right\} = \left\langle \boldsymbol{\lambda} \left\langle \begin{smallmatrix} | \\ a \overset{|}{\boldsymbol{\lambda}} b \end{smallmatrix} \right\rangle c \right\rangle$

5. $\left\{a^i b^j c^k : i + j = k\right\} = \left\langle a \left\langle \begin{smallmatrix} | \\ b \overset{|}{\boldsymbol{\lambda}} c \end{smallmatrix} \right\rangle c \right\rangle$

El siguiente teorema muestra como una expresión simétrica se puede convertir en una gramática lineal.

Teorema 4.5.13. *Todo lenguaje simétrico puede ser generado por una gramática lineal*

Demostración. Sea $L(D)$ un lenguaje lineal representado por la expresión simétrica D. Puesto que la definición de las expresiones simétricas se hace recursivamente, la demostración se lleva a cabo razonando por inducción sobre D. Para las expresiones simétricas básicas, podemos construir fácilmente una gramática lineal que lo genere. Así, la gramática $S \to S$ acepta el lenguaje $\varnothing$. Las gramáticas $S \to \lambda$ y $S \to u$ generan los lenguajes representados por las expresiones simétricas $D = \lambda$ o $D = u$.

Paso inductivo: supongamos que para las expresiones simétricas D_1 y D_2 existen gramáticas lineales $G_1 = (V_1, \Sigma, S_1, P_1)$ y $G_2 = (V_2, \Sigma, S_2, P_2)$ tales que $L(G_1) = D_1$ y $L(G_2) = D_2$, sin pérdida de generalidad, podemos suponer que G_1 y G_2 no tienen variables en común (en caso contrario, simplemente cambiamos los nombres de las variables). Para construir una gramática lineal G que genere $D_1 \cup D_2$ introducimos una variable nueva S, la variable inicial de G, junto con las producciones $S \to S_1$ y $S \to S_2$. Las producciones de G_1 y G_2 se mantienen. Concretamente

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}).$$

Esquemáticamente, G tiene el siguiente aspecto

$$\begin{array}{ll} S \to S_1 | S_2 & \\ \left.\begin{array}{ll} S_1 \to & \cdots \\ \vdots & \vdots \end{array}\right\} & \text{producciones de } G_1 \\ \left.\begin{array}{ll} S_2 \to & \cdots \\ \vdots & \vdots \end{array}\right\} & \text{producciones de } G_2 \end{array}$$

claramente $L(G) = D_1 \cup D_2$.
Una gramática lineal que genere xDy con $x, y \in \Sigma^*$ se construye añadiendo la producción $S \to xS_1y$. Es decir,

$$G = (V_1 \cup \{S\}, \Sigma, S, P_1 \cup \{S \to xS_1y\}).$$

Esquemáticamente, G es la gramática

$$\begin{array}{rl} S \to & xS_1y \\ S_1 \to & \cdots \\ \vdots & \vdots \end{array} \Bigg\} \text{ producciones de } G_1$$

claramente $L(G) = xD_1y$.
Para generar $\left\langle x_1, x_2, \ldots, x_n \overset{|}{}_{D_1} y_1, y_2, \ldots, y_n \right\rangle$ se define G como

$$G = (V_1 \cup \{S\}, \Sigma, S, P_1 \cup \{S \to x_1Sy_1, S \to x_2Sy_2, \cdots, S \to x_nSy_n, S \to S_1, \}).$$

Esquemáticamente, G es la gramática

$$\begin{array}{rl} & S \to x_1Sy_1 \mid x_2Sy_2 \mid \cdots \mid x_nSy_n \mid S_1 \\ S_1 \to & \cdots \\ \vdots & \vdots \end{array} \Bigg\} \text{ producciones de } G_1$$

claramente $L(G) = \left\langle x_1, x_2, \ldots, x_n \overset{|}{}_{D_1} y_1, y_2, \ldots, y_n \right\rangle$ por el teorema 4.5.7. □

Corolario 4.5.14. *Todo lenguaje simétrico es lineal*

Ejemplo 4.5.15. 1. La gramática

$$G_1 : \begin{cases} S \to aSb \mid A \\ A \to Ab \mid b \end{cases}$$

cumple que $L(G_1) = \left\{a^i b^j : i < j\right\} = \left\langle a \overset{|}{}_{\left\langle \lambda \overset{|}{}_{b} b \right\rangle} b \right\rangle$.

2. La gramática

$$G_2 : \begin{cases} S \to A \mid C \\ A \to aaAb \mid B \\ B \to Bb \mid ab \\ C \to aaCb \mid D \\ D \to Db \mid b \end{cases}$$

cumple que $L(G_2) = \left\{a^i b^j : i < 2j\right\} = \left\langle aa \overset{|}{}_{\left\langle \lambda \overset{|}{}_{ab} b \right\rangle} b \right\rangle \cup \left\langle aa \overset{|}{}_{\left\langle \lambda \overset{|}{}_{b} b \right\rangle} b \right\rangle$

3. La gramática

$$G_3 : \Big\{ S \to aSa \mid bSb \mid \lambda$$

cumple que $L(G_3) = \left\{ ww^R : w \in \Sigma^* \right\} = \langle a, b | a, b \rangle \lhd \boldsymbol{\lambda}$

4. La gramática

$$G_4 : \begin{cases} S \to Sc \mid A \\ A \to aAb \mid \lambda \end{cases}$$

cumple que $L(G_4) = \left\{ a^i b^i c^j : i \geq 0, j \geq 0 \right\} = \left\langle \boldsymbol{\lambda} \begin{array}{|c} | \\ \left\langle a \atop \boldsymbol{\lambda} b \right\rangle \end{array} c \right\rangle$

5. La gramática

$$G_5 : \begin{cases} S \to aSc \mid A \\ A \to bAc \mid \lambda \end{cases}$$

cumple que $L(G_5) = \left\{ a^i b^j c^k : i + j = k \right\} = \left\langle a \begin{array}{|c} | \\ \left\langle b \atop \boldsymbol{\lambda} c \right\rangle \end{array} c \right\rangle$

El recíproco de este teorema es cierto para una subfamilia de lenguajes lineales.

Definición 4.5.16. Una variable A es recursiva si existe una variable B tal que

$$A \overset{+}{\Longrightarrow} uBv \quad \text{y} \quad B \overset{+}{\Longrightarrow} xAy$$

con $u, v, x, y \in (V \cup \Sigma)^*$

Teorema 4.5.17. *Un lenguaje es simétrico si y sólo si es lineal sin variables recursivas.*

Si el lenguaje es simétrico entonces es claramente lineal. Recíprocamente si un lenguaje es lineal sin variables recursivas, entonces existe una gramática lineal $G = (V, \Sigma, S, P)$ que lo genera. Una gramática tiene una única variable inicial pero cambiando dicha variable surgen nuevas gramáticas. Para cada $S_i \in V$, sea G_i la gramática que coincide con G pero con variable inicial S_i; concretamente, $G_i = (V, \Sigma, S_i, P)$. Al lenguaje generado por G_i lo denotaremos por A_i; es decir, $L(G_i) = A_i$. En particular, $A_0 = L(G)$. Puesto que los símbolos no terminales no se han alterado, se tiene que $A_i = \left\{ w \in \Sigma^* : S_i \overset{+}{\Longrightarrow} w \right\}$.
Cada A_i se puede escribir como

$$A_i = \begin{cases} \bigcup \{ uA_j v : u, v \in \Sigma^* \} \text{ Si } S_i \Longrightarrow uS_j v \\ \bigcup \{ u : u \in \Sigma^* \} \text{ Si } S_i \Longrightarrow u \end{cases} \tag{4.1}$$

Si $V = \{S_0, \ldots, S_n\}$, las igualdades de la forma (4.1) dan lugar a un sistema de $n+1$ ecuaciones con $n+1$ incógnitas (los A_i):

$$\begin{cases} A_0 & = C_{01}A_0D_{01} \cup C_{02}A_1D_{02} \cup \cdots \cup C_{0n}A_nD_{0n} \cup F_0 \\ A_1 & = C_{11}A_0D_{11} \cup C_{12}A_1D_{12} \cup \cdots \cup C_{1n}A_nD_{1n} \cup F_1 \\ & \vdots \\ A_n & = C_{n1}A_0D_{n1} \cup C_{n2}A_1D_{n2} \cup \cdots \cup C_{nn}A_nD_{nn} \cup F_n \end{cases}$$

donde cada coeficiente C_{ij}, D_{ij} o es $\varnothing$ o es un símbolo de Σ^*. El término F_i o es $\varnothing$ o es la unión de símbolos de Σ^*.

Puesto que la gramática no tiene variables recursivas, entonces se descartan ecuaciones de la forma $X = \left\langle \vec{A} \underset{X}{\downarrow} \vec{B} \right\rangle$. Por lo tanto utilizando sucesivamente la generalización del Lema de Arden (Teorema 4.5.7) se soluciona el sistema de ecuaciones; comenzando con la última ecuación, se escribe A_n en términos de los demás A_i, para lo cual se usa el teorema 4.5.7 si es necesario. Este valor de A_n se reemplaza en las demás ecuaciones y el sistema se reduce a n ecuaciones con n incógnitas. Similarmente, A_{n-1} se escribe en términos de los demás A_i, usando el teorema 4.5.7 si es necesario, y tal valor se reemplaza en las ecuaciones restantes. Prosiguiendo de esta manera, el sistema original se reduce a una sola ecuación cuya incógnita es precisamente A_0. Esta ecuación se soluciona recurriendo una vez más a la generalización del lema de Arden. Puesto que los coeficientes $(C)_{ij}, (D)_{ij}, (F)_i$ diferentes de $\varnothing$ son símbolos de Σ, se obtiene una expresión simétrica D tal que $L(G) = A_0 = D$.

Ejemplo 4.5.18. Sea la gramática La gramática

$$G: \begin{cases} S \to aSb \mid bS \mid A \mid B \mid a \\ A \to aA \mid Ab \mid C \mid ab \\ B \to bB \mid aB \mid C \\ C \to aaCbb \mid \lambda \end{cases}$$

El sistema de ecuaciones asociado con la gramática G es:

$$\begin{cases} (1)\ S = aSb \cup bS \cup A \cup B \cup a \\ (2)\ A = aA \cup Ab \cup C \cup ab \\ (3)\ B = bB \cup aB \cup C \\ (4)\ C = aaCbb \cup \lambda \end{cases}$$

Aplicando la generalización del lema de Arden en (4):

(5) $C = \left\langle aa \overset{|}{_\lambda} bb \right\rangle$

Reemplazando (5) en (3):

(6) $B = bB \cup aB \cup \left\langle aa \overset{|}{_\lambda} bb \right\rangle$

Aplicando la generalización del lema de Arden en (6):

(7) $B = \langle b, a \mid \lambda, \lambda \rangle \lhd \left\langle aa \overset{|}{_\lambda} bb \right\rangle$

Reemplazando (5) en (2):

(8) $A = aA \cup Ab \cup \left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup ab$

Aplicando la generalización del lema de Arden en (8):

(9) $A = \langle a, \lambda \mid \lambda, b \rangle \lhd \left(\left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup ab \right)$

Reemplazando (7) y (9) en (1):

$$(10) \quad S = aSb \cup bS \cup \langle a, \lambda \mid \lambda, b \rangle \lhd \left(\left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup ab \right) \cup \langle b, a \mid \lambda, \lambda \rangle \lhd \left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup a$$

Aplicando la generalización del lema de Arden en (10) concluimos que:

$$S = \langle a, b \mid b, \lambda \rangle \lhd \left(\langle a, \lambda \mid \lambda, b \rangle \lhd \left(\left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup ab \right) \cup \langle b, a \mid \lambda, \lambda \rangle \lhd \left\langle aa \overset{|}{_\lambda} bb \right\rangle \cup a \right)$$

Capítulo 5

Puntos Fijos y Lenguajes Independientes de Contexto

En este capítulo introducimos una familia de funciones polinomiales llamadas funciones polinomiales finitas. Estas son importantes ya que un lenguaje es independiente de contexto si y sólo si es una componente de un punto fijo de una función polinomial finita. Una de las implicaciones de este resultado es conocido como el teorema de Ginsburg-Rice, el cual aunque es un resultado clásico, no es muy citado en libros y cursos de teoría de la computación [5].

5.1. Funciones Polinomiales Finitas

La única diferencia entre las funciones polinomiales y las polinomiales finitas es que las funciones constantes se definen sobre lenguajes finitos.

Definición 5.1.1. La clase de funciones polinomiales finitas (f.p.f) n-arias, consiste de las siguientes funciones:
(i) Toda función constante n-aria C_L, donde L es un lenguaje finito.
(ii) Toda proyección n-aria.
(iii) Si f y g son f.p.f. n-arias, entonces $f \cup g$ y fg también lo son.

Definición 5.1.2. Sea Σ un alfabeto y sea $\Sigma_X = \{X_1, \ldots, X_n\}$ un conjunto tal que $\Sigma \cap \Sigma_X = \varnothing$, entonces

$$(\Sigma \cap \Sigma_X)_n := \left\{u : u \in (\Sigma \cap \Sigma_X)^*\right\}$$

A los elementos de este conjunto se le llaman términos n-arios (o simplemente términos) y los elementos de Σ_X variables.

Ejemplo 5.1.3. Sea $\Sigma = \{a, b, c\}$ y $\Sigma_X = \{X_1, X_2, X_3\}$, entonces los siguientes son términos de $(\Sigma \cap \Sigma_X)_3$.

$$u_1 = bcX_1X_2aaX_2, \quad u_2 = \lambda, \quad u_3 = aX_3, \quad u_4 = X_1$$

Definición 5.1.4. Sea u un término de $(\Sigma \cap \Sigma_X)_n$, la función generada por el término $u = u_1X_{i_1}u_2 \cdots u_kX_{i_k}u_{k+1}$, $f_u : (\wp\mathcal{O}(\Sigma^*))^n \longrightarrow \wp\mathcal{O}(\Sigma^*)$ está definida por

$$f_u(\vec{L}) := u_1L_{i_1}u_2 \cdots u_kL_{i_k}u_{k+1}$$

para todo $\vec{L} = (L_1, \ldots, L_n) \in (\wp\mathcal{O}(\Sigma^*))^n, 1 \leq i_j \leq n, 1 \leq j \leq k, k \geq 1$.

Por inducción sobre k, donde k es el número de variables que aparecen en la definición de f_u, de demuestra fácilmente que f es una f.p.f.

Ejemplo 5.1.5. Sea $\Sigma = \{a, b, c\}$ y $\Sigma_X = \{X_1, X_2, X_3\}$, entonces las siguientes son funciones generadas por los términos del ejemplo 5.1.3.

$$f_{u_1}(\vec{L}) = bcL_1L_2aaL_2, \quad f_{u_2}(\vec{L}) = \lambda, \quad f_{u_3}(\vec{L}) = aL_3, \quad f_{u_4}(\vec{L}) = L_1$$

donde $\vec{L} = (L_1, L_2, L_3)$.

Para las f.p.f que sean proyecciones, es decir que sean generadas por una sola variable, como por ejemplo $f_{u_4}(\vec{L}) = L_1$, utilizaremos la notación $f_{X_i}(\vec{L}) = L_i$.

Teorema 5.1.6. *Si $f_u : (\wp\mathcal{O}(\Sigma^*))^n \longrightarrow \wp\mathcal{O}(\Sigma^*)$ es una f.p.f entonces*

$$f(\vec{X}) = \bigcup_{i=1}^{m} g_i(\vec{X})$$

para todo $\vec{X} \in (\wp\mathcal{O}(\Sigma^*))^n$, donde $g_1, \ldots, g_m$ son funciones generadas por términos n-arios.

Demostración. La demostración es por inducción sobre la definición de f.p.f.. Si $f = C_L$, donde $L = \{w_1, \ldots, w_l\}$ entonces

$$C_L(\vec{X}) = \bigcup_{i=1}^{l} g_i(\vec{X})$$

donde $g_i(\vec{X}) = w_i$, para $1 \leq i \leq l$. Si $f = P_i^n$, entonces $f = f_{X_i}(\vec{X})$ para $1 \leq i \leq l$. Supongamos que la afirmación es cierta para las funciones f y f', entonces

$$f(\vec{X}) = \bigcup_{i=1}^{l} g_i(\vec{X})$$

$$f'(\vec{X}) = \bigcup_{i=1}^{l'} g'_i(\vec{X})$$

para todo $\vec{X} \in (\wp(\Sigma^*))^n$, donde $g_1, \ldots, g_l, g'_1, \ldots, g'_{l'}$, son funciones generadas por términos n-arios. Entonces

$$(f \cup f')(\vec{X}) = \bigcup_{i=1}^{l} g_i(\vec{X}) \cup \bigcup_{i=1}^{l'} g'_i(\vec{X})$$

lo que prueba que $f \cup f'$ tiene la forma establecida. Asimismo,

$$\begin{aligned}(ff')(\vec{X}) &= \bigcup_{i=1}^{l} g_i(\vec{X}) \bigcup_{j=1}^{l'} g'_j(\vec{X}) \\ &= \bigcup_{i=1}^{l} \bigcup_{j=1}^{l'} g_i(\vec{X}) g'_j(\vec{X})\end{aligned}$$

y puesto que $g_i(\vec{X})g'_j(\vec{X})$ es un término n-ario, para todo i, j, entonces queda probado el teorema. □

Definición 5.1.7. Sea $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^m$, donde $\vec{f} = (f_1, f_2, \ldots, f_m)$. Diremos que $\vec{f}$ es una función polinomial finita si cada componente f_i es polinomial finita.

5.2. Teorema de Ginsburg-Rice

En esta sección probaremos un resultado conocido como el teorema de Ginsburg-Rice, que caracteriza los LIC como el menor punto fijo de una f.p.f.

Definición 5.2.1. Sea $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^n$ una f.p.f, con $\vec{f} = (f_1, \ldots, f_n)$ y tal que $f_i(\vec{X}) = \bigcup_{j=1}^{l_i} g_{ij(\vec{X})}$ donde g_{ij} $(1 \leq i \leq n, 1 \leq j \leq l)$ son las respectivas funciones generadas por términos n-arios y l_i es el número de términos de f_i. Entonces las gramáticas asociadas con la f.p.f. $\vec{f}$ son las gramáticas $G_i = (\Sigma_X, \Sigma, X_j, P)$ donde el conjunto de producciones P consiste de todas las reglas de producción de la forma $X_j \to g_{ij}$ para $1 \leq i \leq n$ y $1 \leq j \leq l_i$.

Ejemplo 5.2.2. Sea $\vec{f} : (\wp(\Sigma^*))^4 \longrightarrow (\wp(\Sigma^*))^4$ una f.p.f, con $\vec{f} = (f_1, f_2, f_3, f_4)$, tales que

$$\begin{aligned}f_1(X_1, X_2, X_3, X_4) &= X_3X_2 \cup X_2X_4 \\ f_2(X_1, X_2, X_3, X_4) &= aX_2b \cup \lambda \\ f_1(X_1, X_2, X_3, X_4) &= aX_3 \cup a \\ f_1(X_1, X_2, X_3, X_4) &= bX_4 \cup b\end{aligned}$$

entonces las gramáticas asociadas con $\vec{f}$ son

$$G_1 = (\Sigma_X = \{X_1, X_2, X_3, X_4\}, \Sigma = \{a, b\}, X_1, P)$$
$$G_2 = (\Sigma_X = \{X_1, X_2, X_3, X_4\}, \Sigma = \{a, b\}, X_2, P)$$
$$G_3 = (\Sigma_X = \{X_1, X_2, X_3, X_4\}, \Sigma = \{a, b\}, X_3, P)$$
$$G_4 = (\Sigma_X = \{X_1, X_2, X_3, X_4\}, \Sigma = \{a, b\}, X_4, P)$$

donde
$P = \{X_1 \to X_3X_2,\ X_1 \to X_2X_4,\ X_2 \to aX_2b,\ X_2 \to \lambda,\ X_3 \to aX_3,\ X_3 \to a,\ X_4 \to bX_4,\ X_4 \to b\}$

Se puede probar que la gramática G_1

$$G_1 : \begin{cases} X_1 \to X_3X_2 \mid X_2X_4 \\ X_2 \to aX_2b \mid \lambda \\ X_3 \to aX_3 \mid a \\ X_4 \to bX_4 \mid b \end{cases}$$

cumple que $L(G_1) = \{a^m a^n : m \neq n\}$, el cual es un LIC, además, será una de las componentes del menor punto fijo de la función $\vec{f}$.

El teorema de Ginsburg-Rice relaciona precisamente el menor punto fijo de una f.p.f. con el lenguaje generado por las gramáticas asociadas a la función. La demostración que presentamos esta basada de [11].

Teorema 5.2.3 (Ginsburg-Rice)**.** *Sea* $\vec{f} : (\wp(\Sigma^*))^n \longrightarrow (\wp(\Sigma^*))^n$ *una f.p.f, entonces el menor punto fijo de* $\vec{f}$ *es una n-upla de LICs,* $\vec{L} = (L_1, \ldots, L_n)$ *donde* $L_j = L(G_j)$ *y* G_j *son las gramáticas asociadas a* $\vec{f}$*,* $1 \leq j \leq n$*.*

Demostración. Sea $\vec{F} = (F_1, \ldots, F_n)$ el menor punto fijo de $\vec{f}$; su existencia lo garantiza el teorema 3.2.6. Probaremos que $\vec{F} = \vec{L}$. Veamos que $\vec{F} \subseteq \vec{L}$, es decir que $F_i \subseteq L_i$ para todo i, $1 \leq i \leq n$. Por el teorema 3.2.6

$$F_i = \bigcup_{j \geq 0} f_i^j(\varnothing)$$

Es suficiente con probar que $f_i^j(\varnothing) \subseteq L(G_i)$ para todo $j \geq 0$, $i, 1 \leq i \leq n$. La prueba es por inducción sobre j. Para $j = 0$, es inmediato ya que $f_j^0(\varnothing) = \varnothing \subseteq L(G_i)$.
Sea $u \in \left(f_i^{j+1}(\varnothing)\right)$, esto significa que

$$u \in f_i\left(f_1^j(\varnothing), f_2^j(\varnothing), \ldots, f_n^j(\varnothing)\right)$$

Como f_i es igual a la unión de funciones generadas por términos n-arios (teorema 5.1.6), entonces existe un término

$$t_{ik} = w_1 X_{S_1} w_2 \cdots w_l X_{S_l} w_{l+1} \quad \text{donde } l_i \text{ es el número de términos de } f_i$$

tal que $w_1, \ldots, w_{l+1} \in \Sigma^*$, con $u = w_1 u_1 w_2 \cdots w_l u_l w_{l+1}$ y $u_r \in f^j_{S_s}(\varnothing)$, $1 \leq s \leq l$. Además, la regla de producción

$$X_i \longrightarrow w_1 X_{S_1} w_2 \cdots w_l X_{S_l} w_{l+1}$$

pertenece a P.
Por la hipótesis de inducción $u_s \in L(G_{S_s})$, así que existe una derivación

$$X_{S_r} \underset{G_{S_r}}{\overset{*}{\Longrightarrow}} u_r \quad (1 \leq r \leq l)$$

por lo tanto se tiene la siguiente derivación

$$\begin{aligned} X_i &\underset{G_i}{\overset{*}{\Longrightarrow}} w_1 X_{S_1} w_2 \cdots w_l X_{S_l} w_{l+1} \\ &\underset{G_i}{\overset{*}{\Longrightarrow}} w_1 u_1 w_2 \cdots w_l X_{S_l} w_{l+1} \\ &\vdots \\ &\underset{G_i}{\overset{*}{\Longrightarrow}} w_1 u_1 w_2 \cdots w_l u_l w_{l+1} = u \end{aligned}$$

luego $u \in L(G_i)$, para todo i, $(1 \leq i \leq n)$, es decir $\vec{F} \subseteq \vec{L}$.

Probemos ahora que $\vec{L} \subseteq \vec{F}$; es suficiente probar que si t y t' son términos n-arios, tales que $t \overset{*}{\Longrightarrow} t'$ entonces las funciones generadas por estos términos cumplen que $f_{t'}(\vec{F}) \subseteq f_t(\vec{F})$; en el caso que esto se cumpliera, si $u \in L(G_i)$ implica que $X_i \overset{*}{\Longrightarrow} u$, así que $u = f_u(\vec{F})$; como $f_u(\vec{F}) \subseteq f_{X_i}(\vec{F})$ y $f_{X_i}(\vec{F}) = F_i$, entonces $u \in F_i$.

Supongamos que $t \overset{n}{\Longrightarrow} t'$; la demostración es por inducción sobre n. Para $n = 0$ es trivial, ya que $t = t'$. Si $t \overset{n+1}{\Longrightarrow} t'$ y t'' es el paso n de la derivación entonces $t'' = t_0 X_i t_1$ y $t' = t_0 w t_1$, t_0, t_1 son términos n-arios, además la producción $X_i \to w$ fue aplicada en al menos un paso de la derivación. Por lo tanto

$$\begin{aligned} f_{t''}(\vec{F}) &= f_{t_0}(\vec{F}) f_{X_i}(\vec{F}) f_{t_1}(\vec{F}) \\ &= f_{t_0}(\vec{F}) F_i f_{t_1}(\vec{F}) \\ f_{t'}(\vec{F}) &= f_{t_0}(\vec{F}) f_w(\vec{F}) f_{t_1}(\vec{F}) \end{aligned}$$

Puesto que w es un término de la función polinomial f_i, entonces $f_w(\vec{F}) \subseteq f_i(\vec{F})$, pero por ser $\vec{F}$ un punto fijo de $\vec{f}$ entonces $f_i(\vec{F}) = F_i$, así $f_w(\vec{F}) \subseteq F_i$, por lo tanto $f_{t'}(\vec{F}) \subseteq f_{t''}(\vec{F})$. Por la hipótesis de inducción $f_{t''}(\vec{F}) \subseteq f_t(\vec{F})$, concluyendo que $f_{t'}(\vec{F}) \subseteq f_t(\vec{F})$, como se quería demostrar. □

Ahora vamos a realizar una construcción en la otra dirección, es decir, dada una GIC $G = (\Sigma_X, \sigma, S, P)$ le asociamos una f.p.f..

En la siguiente definición se asumirá que $\Sigma_X = \{X_1, \ldots, X_n\}$ y que las reglas de producción son de la forma $X_i \to t_{i1}, \ldots, X_i \to t_{il_i}$.

Definición 5.2.4. Sea $G = (\Sigma_X, \Sigma, S, P)$ una GIC, la f.p.f. asociada a G, $\vec{f}_G : (\wp(A^*))^n \longrightarrow (\wp(A^*))^n$ se define como $\vec{f}_G = (f_1, \ldots, f_n)$ donde

$$f_i(\vec{X}) = g_{t_{i1}}(\vec{X}) \cup \cdots \cup g_{t_{il_i}}(\vec{X})$$

para $1 \leq i \leq n$.

Teorema 5.2.5. *Todo LIC es una componente del menor punto fijo de una f.p.f.*

Demostración. Sea L un LIC y sea $G = (\Sigma_X, \Sigma, X_1, P)$ la GIC que lo genera ($L(G) = L$), donde $\Sigma_X = \{X_1, \ldots, X_n\}$. Sea $\vec{f}_G$ una f.p.f asociada a G; puesto que el símbolo inicial de G es X_1, entonces el teorema de Ginsburg-Rice implica que la primera componente del menor punto fijo de la función $\vec{f}_G$ coincide con el lenguaje generado por la gramática G_1; pero $G = G_0$, es decir que la primera componente del menor punto fijo de $\vec{f}_G$ es L. □

Corolario 5.2.6. *La clase de LICs coincide con la clase de lenguajes que son componentes del menor punto fijo de un f.p.f.*

Demostración. Es inmediata a partir de los teorema 5.2.3 y 5.2.5. □

Ejemplo 5.2.7. Consideremos la gramática G, definida por las siguientes reglas de producción

$$G : \begin{cases} S & \to aS_2 \mid bS_1 \\ S_1 & \to aS \mid bS_1S_1 \mid a \\ S_2 & \to bS \mid aS_2S_2 \mid b \end{cases}$$

entonces la función polinomial $\vec{f}_G : (\wp(\Sigma^*))^3 \longrightarrow (\wp(\Sigma^*))^3$ asociada a G esta dada por

$\vec{f}_G = (f_1, f_2, f_3)$ donde

$$f_1(\vec{L}) = aL_3 \cup bL_2$$
$$f_2(\vec{L}) = aL_1 \cup bL_2L_2 \cup a$$
$$f_3(\vec{L}) = bL_1 \cup aL_3L_3 \cup b$$

además las primeras aproximaciones del menor punto fijo de $\vec{f}$ son

$$\begin{aligned}
\vec{f}(\varnothing,\varnothing,\varnothing) &= (\varnothing, \{a\}, \{b\}) \\
\vec{f}^2(\varnothing,\varnothing,\varnothing) &= (\{ab, ba\}, \{a, baa\}, \{b, abb\}) \\
\vec{f}^3(\varnothing,\varnothing,\varnothing) &= (\{ab, ba, aabb, bbaa\}, \{a, aab, aba, baa, bbaaa, babaa, bbaabaa\}, \\
&\qquad \{b, bab, bba, abb, aabbb, ababb, aabbabb\}) \\
&\vdots
\end{aligned}$$

aunque no tenemos una forma de solucionar las ecuaciones que se generan para este ejemplo de verifica que el menor punto fijo de $\vec{f}$ es (L_1, L_2, L_3), donde $L_1 = \{w \in \{a,b\} : |w|_a = |w|_b\}$, $L_2 = \{w \in \{a,b\} : |w|_a > |w|_b\}$ y $L_3 = \{w \in \{a,b\} : |w|_a < |w|_b\}$.

5.3. Familias de Ecuaciones sobre Lenguajes

Las ecuaciones sobre lenguajes es una línea de investigación en Teoría de la Computación, cuyos primeros resultados son debidos a Ginsburg y Rice hacia el año 1962, quienes caracterizaron los LIC a partir de sistemas ecuaciones. Después aparecerían algunos resultados ocasionarles en esta área; sin embargo, es en los últimos años en los que se ha retomado el estudio en esta línea de trabajo, utilizando herramientas matemáticas más complejas. Por ejemplo, Kari (1994) estudió ecuaciones con operaciones no usuales, Kunc (2005) estudió la ecuación $XL = LX$, donde L es un lenguaje regular, pero sin duda uno de los autores que más trabajos ha desarrollado en esta área es Alexander Okhotin, junto con sus estudiantes, quienes han estudiado ecuaciones e inecuaciones con todas las operaciones booleanas [8].

La acumulación de resultados ha conllevado a que sea precisamente Okhotin el que proponga una clasificación de las ecuaciones en siete familias [9], que describiremos brevemente.

Consideremos el sistema de ecuaciones

$$\begin{cases} X_1 & = \varphi_1 (X_1, \ldots, X_n) \\ & \vdots \\ X_n & = \varphi_n (X_1, \ldots, X_n) \end{cases}$$

donde cada φ_i contiene variables, lenguajes formados por un sólo elemento, relacionados por concatenación y operaciones booleanas. En este capítulo se estudiaron sistemas cuya operación booleana es la intersección, además, cada φ_i es una función polinomial finita.

Dependiendo de las operaciones booleanas que se utilicen, se distinguen cada una de las siete familias, las cuales definen un lenguaje determinado. En la siguiente tabla se muestran cuatro familias con las operaciones booleanas utilizadas, los lenguajes que definen y los autores que han estudiado las ecuaciones que los definen; las otras tres familias son triviales, ya que no contienen operaciones booleanas.

Operación booleana	Lenguaje	Autores
Disyunción	LIC	Ginsburg y Rice (1962)
Disyunción y conjunción	Conjuntivos (Conjunctive)	Okhotin (2002)
Todas las operaciones booleanas	Recursivos	Okhotin (2003, 2006)
Negación	No tienen nombre	Okhotin y Yakimova (2006), Leiss (1994)

Bibliografía

[1] Davey, B., and Priestley, H. *Introduction to Lattices and Order*, segunda ed. Cambridge University Press, 2002.

[2] De Castro, R. *Teoría de la computación, lenguajes, autómatas, gramáticas*. Universidad Nacional de Colombia, 2004.

[3] De Castro, R. Un teorema de punto fijo para retículos booleanos completos y algunas aplicaciones. *preprint* (2004).

[4] Du, D., and Ko, K. *Problem solving in automata, languages, and comnplexity*. Jhon Wiley & Sons, 2001.

[5] Hesselink, W. Solutions of equations in languages. *Formal Aspects of Computing, Springer, London* (2009).

[6] Kupka, I. Unique fixpoints in complete lattices with applications to formal languages and semantics. In *Foundations of Computer Science: Potential - Theory - Cognition* (1997), pp. 107–115.

[7] Mandrioli, D., and Ghezzi, C. *Theoretical Foundations of Computer Science*. John Wiley & Sons, 1987.

[8] Michal, K. What do we know about language equations? In *Lecture Notes In Computer Science: Proceedings of the 11th international conference on Developments in language theory* (Berlin, Heidelberg, 2007), Springer-Verlag, pp. 23–27.

[9] Okhotin, A. Seven families of language equations. *AutoMathA 2007, Palermo, Italia* (2007).

[10] Roman, S. *Lattices and Ordered Sets*. Springer, 2008.

[11] Simovici, D., and Tenney, R. *Theory of Formal Languages with Applications.* World Scientific, 1999.