Notas de Curso

LENGUAJES Y OPERACIONES PARA LENGUAJES

TEORÍA

Alberto A. Herrera Becerra

Agosto 2022

Curso Universitario

ICAT - UNAM, Facultad de Ingeniería - UNAM
Financiamiento interno

LENGUAJES Y OPERACIONES PARA LENGUAJES: TEORÍA

Alberto A. Herrera Becerra

Grupo Académico de Modelado y Simulación de Procesos, Departamento de Instrumentación Científica e Industrial

RESUMEN

En este documento se presenta una introducción al tema de los lenguajes formales y sus operaciones como parte de un curso más amplio sobre la teoría de los lenguajes formales. En este documento, los lenguajes formales son entendidos como conjuntos de secuencias finitas de símbolos, las llamadas palabras. Esta es una concepción muy general de los lenguajes que incluye tanto a los lenguajes naturales (como el español) como a los lenguajes artificiales (como los lenguajes de programación). En este contexto, los lenguajes son analizados únicamente en términos de sus elementos constituyentes (las palabras), sin considerar su posible estructura interna (su gramática) o su significado. En el documento se destaca la importancia del problema de especificación para analizar lenguajes formales. Desafortunadamente, este enfoque a los lenguajes formales es tratado de manera poco rigurosa en muchos de los libros introductorios sobre la Teoría de Autómatas y Lenguajes Formales; en general, el tratamiento que se presenta es informal y un tanto superficial. Este documento se presenta como una ayuda al alumno (y al profesor) para subsanar tales deficiencias. El tema de los lenguajes formales se aborda desde el punto de vista de las matemáticas discretas, empleando nociones básicas de teoría de conjuntos y álgebra abstracta, temas que son conocidos para los estudiantes que ya han llevado un curso de estructuras discretas.

ÍNDICE

INTRODUCCIÓN

En un lenguaje natural occidental, como el español, típicamente se pueden distinguir tres entidades lingüísticas: las letras, las palabras y las oraciones. Existe un cierto paralelismo entre el hecho de que grupos de letras forman a las palabras y el hecho de que grupos de palabras forman a las oraciones. Pero no todas las colecciones de letras forman una palabra válida, ni todas las colecciones de palabras forman oraciones válidas. La analogía se puede continuar. Ciertos grupos de oraciones forman párrafos coherentes, ciertos párrafos forman textos coherentes, y así por el estilo.

Esta situación también existe en los lenguajes para computadoras. Ciertas cadenas de caracteres son palabras reconocibles (GOTO, END, etc.). Ciertas cadenas de palabras son comandos reconocibles. Ciertos conjuntos de comandos se vuelven un programa, con o sin datos.

Para construir una teoría general que unifique todos estos ejemplos es necesario adoptar una definición de una “estructura para el lenguaje más universal”, esto es, una estructura en la cual la decisión de si una cadena dada de unidades constituye una unidad válida más larga no dependa de adivinanzas o suposiciones intuitivas, sino que esté basada en reglas establecidas explícitamente.

Es muy difícil postular todas las reglas para, por ejemplo, el “inglés hablado”, debido a que muchas cadenas de palabras aparentemente incoherentes en realidad son usos entendibles para los hablantes del lenguaje. Esto se debe a los modismos, dialectos, idiomas, y la habilidad de los hablantes para interpretar metáforas poéticas y corregir errores (intencionales o no) en las oraciones que escuchan. Sin embargo, como un primer paso para definir una teoría general de los lenguajes abstractos es conveniente insistir en la necesidad de postular reglas precisas. Esto es muy significativo para los profesionales de la computación, ya que las computadoras “no son tolerantes” a los comandos de entrada imprecisos, como tal vez lo sea un hablante humano.

Así pues, en la *Teoría de los Lenguajes Formales*, el término “formal” se usa para hacer patente el hecho de que todas las reglas para describir a un lenguaje son postuladas explícitamente en términos de operaciones sobre ciertas cadenas de unidades básicas. No se aceptan usos inapropiados de las reglas y no es necesario recurrir a un conocimiento “profundo” de la naturaleza de las reglas. Las reglas se siguen al pie de la

letra, punto. Debido a esto, los lenguajes abstractos que se tratan en la teoría son únicamente conjuntos ordenados de símbolos escritos, y no son expresiones de ideas que provengan de la mente de los usuarios. En este modelo básico, un lenguaje no se utiliza para establecer comunicación entre humanos, sino es una especie de "juego" de símbolos con reglas formales. Así, el término "formal" se utiliza para señalar que hay una definición precisa, no ambigua, que siempre debe ser respetada. Interesan, ante todo, las consecuencias de las reglas. Debido a esto, las cadenas de unidades que forman a los lenguajes abstractos son interesantes por su estructura (su forma pues), pero no por su posible significado.

El estudio de los lenguajes formales comienza con la definición de las unidades básica de construcción, las palabras. Estos objetos se construyen de forma precisa a partir de unidades primitivas denominadas símbolos. Las reglas de construcción de las palabras y algunas de sus consecuencias fueron el tema tratado en el texto "Palabras y Operaciones para Palabras". En este texto se establecen las reglas para construir lenguajes a partir de palabras y algunas de las consecuencias de dichas reglas.

Se comienza con un breve repaso de algunos de los tópicos relevantes sobre las palabras.

1. CONCEPTOS INICIALES

Un *alfabeto* es un conjunto finito no vacío de símbolos, los cuales a veces serán denominados genéricamente *letras*, aunque no sean realmente letras de algún vocabulario. Esto es, si $V = \{0, 1, 2, \dots, 9\}$ es un alfabeto, entonces el símbolo (dígito) 5, por ejemplo, podrá a veces ser referido como una letra de V. Los símbolos de un alfabeto siempre son considerados como unidades indivisibles. En estas notas, se van a usar alfabetos pequeños, formados por unos cuantos símbolos, con caracteres muy simples: normalmente letras del vocabulario latino o dígitos; y esto es simplemente por comodidad.

Una *palabra* o *cadena* sobre un alfabeto V es una secuencia finita de cero o más letras de V, en las que una misma letra puede aparecer repetidas veces. En particular, la secuencia de cero letras será denominada la *palabra vacía* y siempre será denotada ε. De manera rigurosa, una palabra es una familia indexada de símbolos y debería ser representada en la forma general $(w_i)_{i \in I}$, donde el conjunto de índices I corresponde a un subconjunto ordenado de números naturales. Por ejemplo, las secuencias $(\,)$, (0),

$(1,0)$, $(0,1,1,0)$, $(1,0,1,1,0,0,1,1,1)$ son palabras sobre el alfabeto (binario) $V = \{0, 1\}$, donde la primera secuencia corresponde a la palabra vacía. Sin embargo, va a ser más conveniente representa esas palabras con la notación simplificada ε, 0, 10, 0110, 101100111, esto es, sin paréntesis y sin comas. Dicho sea de paso, esta notación simplificada explica la razón del uso del término cadena como un sinónimo de palabra. El conjunto de todas las palabras sobre V será denotado V^* (lo cual se lee "V estrella"), mientras que el conjunto de todas las palabras no vacías sobre V será denotado V^+ (lo cual se lee "V más").

El orden en el que aparecen las letras de una palabra es de suma importancia. Si u es una palabra no vacía de V^*, entonces u se expresa como $u = a_{i_1} a_{i_2} \dots a_{i_n}$, donde cada $a_i \in V$, y donde $u(k)$ denota a la k-ésima letra de u, esto es, $u(k) = a_{i_k}$. La primera letra de u es $u(1) = a_{i_1}$, la segunda letra es $u(2) = a_{i_2}$, y así por el estilo hasta la última letra $u(n) = a_{i_n}$. Esto permite remarcar el hecho de que las palabras, como han sido definidas, son objetos compuestos de naturaleza finita: siempre contiene una primera letra y una última letra, a excepción de la palabra vacía que no contiene letras. Nótese que para toda letra $u(i)$ de u existe una letra antecesora $u(i-1)$ y una letra sucesora $u(i+1)$, excepto en dos casos, la primera letra $u(1)$, que no tiene una antecesora, pero si una sucesora, y la última letra $u(n)$, que no tiene una sucesora, pero sí una antecesora. Un caso también excepcional lo constituyen las palabras unitarias, las que son de la forma $u = a_{i_1}$, pues en ellas su única letra es tanto la primera como la última letra, y esta letra no tiene ni antecesora ni sucesora. Una palabra unitaria suele escribirse simplemente $u = a_i$.

Aquí cabe hacer la siguiente aclaración. En el texto "Palabras y Operaciones para Palabras" una palabra como la u anterior hubiese sido descrita $u = a_{i_0} \dots a_{i_{n-1}}$, donde la primera letra es $u(0) = a_{i_0}$ y la última letra es $u(n-1) = a_{i_{n-1}}$. Esto es, la primera letra de u es la letra que se encuentra en la posición 0, la segunda letra es la que se encuentra en la posición 1, y así sucesivamente hasta la última letra, la que se encuentra en la posición $n-1$. Las dos formas de descripción son equivalentes, y aunque la que ahora se presenta es más informal, su uso es más común; y esta es la razón por la que se le va a utilizar.

Una característica importante de toda palabra es su *longitud*, o el número total de letras que la forman, incluyendo todas las posibles repeticiones. Así, para la palabra u dada en el párrafo anterior, su longitud es n, lo cual se denota $|u| = n$ o $long(u) = n$; esto quiere decir que u está formada por n letras en total, incluyendo repeticiones. Se debe entender que la palabra vacía ε es la única palabra que tiene longitud cero: $|\varepsilon| = 0$. Similarmente, para toda palabra u se define su *longitud-*a, denotada $|u|_a$ o $long_a(u)$, como el número de veces que aparece la letra a en u. Nótese que para ciertas palabras no vacías u puede suceder que $|u|_a = 0$ para ciertas letras a del su alfabeto; esto simplemente indica que la letra a no aparece en u.

Así pues, una palabra u sobre un alfabeto V es un objeto compuesto, está formada con símbolos de V, ordenado y en el que sus componentes pueden aparecer repetidas veces. Claramente, una palabra no es un objeto trivial.

2. LENGUAJES

Un *lenguaje* L sobre un alfabeto V es un conjunto de palabras sobre V, esto es, $L \subseteq V^*$. Así, el conjunto $\{1, 12, 123, 1234, 12345, 123456\}$ es un lenguaje sobre el alfabeto de los dígitos decimales. Es importante remarcar que un alfabeto V y un lenguaje L sobre V son dos objetos matemáticos diferentes a pesar de que ambos son conjuntos, el primero es un conjunto de símbolos, el segundo es un conjunto de palabras. La confusión puede surgir al considerar un alfabeto de n símbolos, digamos $V = \{a_1, a_2, \dots, a_n\}$, y el lenguaje sobre V de todas las palabras unitarias $L = \{a_1, a_2, \dots, a_n\}$; ciertamente los dos conjuntos se escriben de la misma manera, pero el contexto debe señalar inequívocamente la diferencia entre ellos.

Dado que un lenguaje L sobre un alfabeto V es un conjunto de palabras sobre V, entonces resulta que $\mathcal{P}(V^*)$, el conjunto potencia de V^*, es el conjunto de todos los lenguajes sobre V. En particular, en $\mathcal{P}(V^*)$ se encuentra el lenguaje formado por ninguna palabra –*el lenguaje vacío*. El lenguaje vacío se denota de la misma forma que el conjunto vacío, $\emptyset$. Éste no es el mismo lenguaje que el que consta únicamente de la palabra vacía $\{\varepsilon\}$, el *lenguaje de la palabra vacía*; claramente, el lenguaje vacío no tiene palabras, mientras que el lenguaje de la palabra vacía es un lenguaje unitario, está for-

mado por una única palabra. Asimismo, en $\mathcal{P}(V^*)$ también se encuentra el lenguaje universal V^*, el conjunto de todas las palabras sobre V.

Se denota $alph(\omega)$ al conjunto de símbolos que forman a la palabra ω. Esta definición implica que $alph(\varepsilon) = \emptyset$, $alph(abba) = \{a, b\}$ y $alph(2001) = \{0, 1, 2\}$. Se define ahora al conjunto de símbolos $alph(L) = \bigcup_{\omega \in L} alph(\omega)$ para todo lenguaje L sobre un alfabeto V. Dado que $L \subseteq V^*$, entonces $alph(L) \subseteq V$, pero la igualdad no necesariamente se presenta, por ejemplo, con $V = \{a, b\}$ y $L = \{a\}$ se tiene que $alph(L) = \{a\} \subseteq V$.

Si A y B son dos lenguajes sobre un alfabeto común V y si todas las palabras de A también son palabras de B, entonces se dice que A es un *sublenguaje* de B. Dado que esto corresponde exactamente con el concepto de subconjunto de la teoría de conjuntos, entonces se puede usar la expresión $A \subseteq B$ para denotar la relación de sublenguaje entre A y B. Por ejemplo, si $A = \{a, aa, aaa\}$ y $B = \{a, aa, aaa, aaaa, aaaaa\}$, entonces efectivamente se cumple que $A \subseteq B$.

Se dice que dos lenguajes A y B son *iguales* si contienen exactamente las mismas palabras, es decir, si son conjuntos iguales. Esto se denota $A = B$. El teorema siguiente muestra la relación que existe entre sublenguajes e igualdad.

Teorema 1
Sean A y B dos lenguajes sobre el alfabeto V. Entonces $A = B$ si, y sólo si, $A \subseteq B$ y $B \subseteq A$.

Prueba
Supóngase primero que $A = B$, esto es, se tiene que probar que $A \subseteq B$ y que $B \subseteq A$. Sea $\omega \in A$. Puesto que A y B tienen exactamente las mismas palabras, se tiene que $\omega \in B$, de lo que se deduce que $A \subseteq B$. Análogamente, si $\omega \in B$ y como A y B tienen las mismas palabras, se tiene que $\omega \in A$ y $B \subseteq A$.

Supóngase ahora que $A \subseteq B$ y que $B \subseteq A$. Esto significa que toda palabra de A está en B y viceversa. Por tanto, A y B tienen exactamente las mismas palabras, con lo que son iguales.

Esto es, el Teorema 1 proporciona una forma de determinar concluyentemente si dos lenguajes son iguales.

Los lenguajes pueden ser finitos o infinitos, y los primeros pueden ser pequeños, grandes o mucho muy grandes. En cualquiera de los casos, cada palabra del lenguaje

tiene una longitud finita. Cuando un lenguaje tiene un tamaño muy grande es difícil especificar que palabras le pertenecen. La especificación de las palabras de un lenguaje es uno de los temas principales de la Teoría de los Lenguajes Formales y, en consecuencia, se le dedica una amplia discusión.

3. ESPECIFICACIÓN DE LENGUAJES

La definición de un lenguaje como un conjunto de palabras sobre un alfabeto es extremadamente simple, pero no dice mucho acerca de qué palabras forman parte de determinados lenguajes; tales palabras pueden ser denominadas palabras *válidas*. La definición simplemente señala que, de alguna manera, se debe tener la capacidad de reconocer que palabras pertenecen un cierto lenguaje particular; dicho de otra manera, se debe ser capaz de distinguir a las palabras válidas de las que no lo son. Este es el problema de *especificación* de lenguajes.

Una manera de resolver el problema de especificación es señalando explícitamente a las palabras ω que constituyen a un determinado lenguaje L sobre un cierto alfabeto V. Esto es, proporcionando una lista con los elementos de L. Este fue el método empleado en los ejemplos de la sección anterior. Esta estrategia, sin embargo, sólo es útil en la práctica si el número de palabras válidas del lenguaje L es finito y pequeño. En este caso, la determinación de si una palabra arbitraria es válida o no se puede resolver por comparación directa de tal palabra con el listado de palabras válidas. Por ejemplo, si el lenguaje L está formado por exactamente n palabras, $L = \{\omega_1, \dots, \omega_n\}$, entonces se puede determinar si una palabra arbitraria δ pertenece o no a L comparando δ con cada una de las palabras válidas ω_i. Si se encuentra una coincidencia, entonces se sabe que $\delta \in L$; si no se encuentra coincidencia alguna, entonces se sabe que $\delta \notin L$.

Es importante hacer la aclaración siguiente. La expresión $L = \{\omega_1, \dots, \omega_n\}$ se puede interpretar de dos formas diferentes, pero equivalentes. Por un lado, la expresión significa que la colección de objetos $\omega_1, \dots, \omega_n$, y sólo de estos objetos, se ha denominado L. Por el otro lado, la expresión indica que el conjunto L y el conjunto de objetos $\omega_1, \dots, \omega_n$ efectivamente son iguales, esto es, contienen exactamente a los mismos elementos. En el texto se hará un uso indistinto de ambos significados.

Ahora bien, cuando L es un conjunto finito muy grande, cualquier listado de sus palabras válidas es igualmente grande y, la determinación por comparación directa de la validez de una palabra arbitraria se vuelve una tarea muy engorrosa. Y no se diga si el lenguaje L es infinito, ya que el listado, aunque factible en principio, ahora es ilimitado y la búsqueda de una determinada palabra válida se vuelve una tarea imposible de realizar en la práctica.

Así pues, se requieren otros métodos para resolver el problema de especificación de lenguajes. A continuación se presentan los primeros métodos de especificación de lenguajes que se usan en la teoría de los lenguajes formales. Otros métodos más precisos y rigurosos son los basados en sistemas generadores (gramáticas) y sistemas reconocedores (autómatas). Estos temas más avanzados serán tratados en documentos futuros.

3.1 Comprensión de conjuntos

Dado que un lenguaje es un conjunto, el primer método de especificación se basa precisamente en las técnicas empleadas en la teoría de conjuntos para construir conjuntos específicos.

Tal vez el método más empleado en la teoría de conjuntos para construir conjuntos particulares es hacer uso del cálculo de predicados de la lógica formal. Así, un conjunto particular se va a formar sólo con los objetos que satisfacen un predicado definitorio. Recuerde que los predicados son uno de los métodos empleados en la lógica formal para expresar propiedades de objetos. De esta manera, el uso de predicados para determinar los objetos que pertenecen a un determinado conjunto equivale a establecer las propiedades que debe satisfacer un objeto dado para que pueda ser considerado un elemento de un conjunto particular. Para los lenguajes, esto significa establecer las características distintivas de las palabras válidas de un lenguaje particular. Tales características deben ser propuestas en términos de las propiedades de las palabras, a saber: su longitud (o el número total de símbolos realmente empleados para construirla), el número de veces que aparecen los símbolos realmente usados (las longitudes-a, donde a es un símbolo del alfabeto empleado), y la posición en la que aparecen los símbolos realmente usados. Con base en esto, un lenguaje particular L sobre un alfabeto V se especifica de la manera siguiente:

$$L = \{\omega \in V^* \mid P(\omega)\}.$$

Esta expresión indica que el lenguaje denominado L está constituido por todas las palabras ω sobre el alfabeto V que poseen, o satisfacen, la propiedad $P(\omega)$ y sólo por esas palabras. Con esta forma de construcción, el problema de especificación se resuelve de la manera siguiente. Sea δ una palabra sobre V. Tal palabra pertenece al lenguaje L si y sólo si $P(\delta)$ es verdadera. Dicho de otra manera, δ pertenece a L si y sólo si δ posee la propiedad definitoria de L.

Con base en el método de comprensión, se definen algunos lenguajes sencillos pero muy importantes. Sea V un alfabeto, y sean los lenguajes sobre V siguientes:

$$V^i = \{\omega \in V^* \mid |\omega| = i\} \text{ para un } i \geq 0 \text{ dado.}$$

Este lenguaje también se puede especificar de la manera siguiente:

$$V^i = \{\text{palabras sobre } V \text{ cuya longitud es } i\}.$$

Nótese en particular que $V^0 = \{\varepsilon\}$, V^1 es el lenguaje de todas las palabras longitud uno, o palabras unitarias, V^2 es el lenguaje de todas las palabras de longitud dos sobre V, y así por el estilo. De la definición se deduce que $V^i \neq V^j$ si $i \neq j$. Recuerde que es importante no confundir V^1 con V, ya que el primero es un lenguaje (un conjunto de palabras unitarias), mientras que el segundo es un alfabeto (un conjunto de símbolos indivisibles). Se tiende a usar V en lugar de V^1, como una especie de simplificación, pero esto es un error que puede causar confusiones graves.

Considérese ahora el lenguaje L sobre el alfabeto $V = \{a, b\}$ siguiente:

$$L = \{a^i b^i \mid i \geq 1\}.$$

Claramente se está haciendo referencia al lenguaje de todas las palabras no vacías ω que tienen la misma cantidad de $a$es que de $b$es, esto es $|\omega|_a = |\omega|_b$, y en las que todas las $a$es aparecen antes que las $b$es. De esta manera, $aaabbb \in L$, pero $aabbb \notin L$. Similarmente, se tiene que $bbbaaa \notin L$, a pesar de que la palabra $bbbaaa$ tiene la misma cantidad de $a$es que de $b$es.

En el ejemplo anterior, la propiedad definitoria se expresó en términos de una fórmula, pero esto no tiene por qué ser siempre así. La propiedad definitoria típicamente se expresa en términos de un predicado general, o de una proposición individual, expresada textualmente. Por ejemplo, considérese ahora el siguiente lenguaje M sobre el mismo alfabeto $V = \{a, b\}$:

$$M = \{\omega \in V^* \mid \omega \text{ tiene una sola } a\}.$$

Con esta especificación, se puede aseverar que todas las palabras a, bab, $bbbba$,$bbabbb$, pertenecen a M, pero que las palabras ε, b, $abba$ no pertenecen a M.

Así pues, todo depende de poder establecer que la palabra que se está analizando realmente posea la propiedad definitoria. Y aquí se enfrentan dos problemas prácticos. Uno es que no siempre es fácil establecer la propiedad definitoria del lenguaje. El otro problema es que no necesariamente es una tarea sencilla mostrar que una palabra dada tiene la propiedad definitoria. A pesar de estas dificultades prácticas, la comprensión de conjuntos es uno de los métodos más empleados para especificar lenguajes.

Otro método de especificación de lenguajes se basa en las operaciones para lenguajes, las cuales se discuten a continuación.

4. OPERACIONES PARA LENGUAJES

En el estudio de las propiedades de los objetos matemáticos siempre es valioso contar con estrategias que permitan construir nuevos objetos a partir de objetos ya conocidos. Las operaciones, o transformaciones, cumplen este cometido. A continuación, se describen los dos principales conjuntos de operaciones que se han definido para los lenguajes.

4.1. Los lenguajes como conjuntos generales

Puesto que un lenguaje es una colección o conjunto de palabras, se pueden definir para los lenguajes las mismas operaciones booleanas que se definen para los conjuntos en general. De esta manera, si A y B son lenguajes sobre un alfabeto V, entonces la *unión* de A y B se denota por $A \cup B$ y es el lenguaje formado por todas las palabras que pertenecen al menos a uno de los dos lenguajes. Por lo tanto, se tiene que:

$$A \cup B = \{\omega \in V^* \mid \omega \in A \; o \; \omega \in B\}.$$

Por otro lado, la *intersección* de los lenguajes A y B, denotada $A \cap B$, es el lenguaje formado sólo por las palabras que pertenecen simultáneamente a los dos lenguajes. Así, se tiene que:

$$A \cap B = \{\omega \in V^* \mid \omega \in A \; y \; \omega \in B\}.$$

Luego se tiene la *diferencia* o *complemento relativo* de A y B, denotada $A - B$, que es el lenguaje formado por todas las palabras de A que no se encuentran en B. Esto es, se tiene que:

$$A - B = \{\omega \in V^* \mid \omega \in A \; y \; \omega \notin B\}.$$

Finalmente se tiene el *complemento* o la *diferencia absoluta* del lenguaje A, denotada A^c o $\bar{A}$, y que es el lenguaje de todas las palabras que no forman parte de A. Se tiene pues que:

$$A^c = \bar{A} = \{\omega \in V^* \mid \omega \notin A\}.$$

Se puede hacer notar que todas las operaciones para lenguajes que se han definido son cerradas en el conjunto $\mathcal{P}(V^*)$, formando entonces un álgebra booleana.

4.1.1. Especificación de lenguajes

Las operaciones para lenguajes se introducen para contar con estrategias que permitan obtener nuevos lenguajes a partir de lenguajes ya conocidos. Con base en esto, se pueden considerar dos métodos para especificar lenguajes empleando operaciones.

En el primer método se pueden tomar lenguajes ya especificados y combinarlos para obtener nuevos lenguajes cuya especificación se obtiene de la definición de las operaciones empleadas. Por ejemplo, considérense los lenguajes sobre el alfabeto $V = \{a, b, c\}$ siguientes:

$\{a^i c^j \mid i \geq 3, j \geq 0\}$,

el conjunto de todas palabras que comienzan con tres o más $a$es y terminan con cero o más $c$es; y

$\{a^i b^j \mid 1 \leq i < 3, j \geq 0\}$,

el conjunto de todas las palabras que comienzan con una o dos $a$es y terminan con cero o más $b$es.

Con estos conjuntos se puede construir el conjunto siguiente:

$$\{a^i c^j \mid i \geq 3, j \geq 0\} \cup \{a^i b^j \mid 1 \leq i < 3, j \geq 0\},$$

el cual es el conjunto de todas las palabras que comienzan con tres o más $a$es y terminan con cero o más $c$es o de todas las palabras que comienzan con una o dos $a$es y terminan con cero o más $b$es.

Se debe destacar que esto es algo más que un ejercicio de redacción de una propiedad, es una estrategia para determinar la pertenencia, o no, de una palabra dada al lenguaje bajo discusión. Así, es relativamente fácil y directo verificar que las palabras $aaaa$ y a pertenecen al último lenguaje, mientras que las palabras ac y $aaab$ no pertenecen. Cierto, la acción de reconocimiento corre a cargo del lector.

El segundo método es una variante de la idea anterior. Dados algunos lenguajes iniciales bien especificados, se les puede combinar mediante las operaciones para lenguajes para obtener nuevos lenguajes. Parece razonable comenzar con algunos de los lenguajes finitos pequeños sobre algún alfabeto V dado, debido a que todos los elementos de tales conjuntos pueden ser listados explícitamente. Posteriormente se pueden formar combinaciones finitas con estos lenguajes iniciales usando las operaciones para lenguajes. Pero para observar la potencialidad de este segundo método es conveniente contar con otras operaciones para lenguajes. Estas nuevas operaciones se discuten a continuación.

4.2. Los lenguajes como conjuntos de palabras

La idea de concatenación, potencia y reverso de palabras se puede extender a los lenguajes procediendo como sigue.

4.2.1. Concatenación

Dados dos lenguajes L_1 y L_2 sobre un alfabeto común V, su *concatenación* o *producto*, denotado $L_1 \cdot L_2$ o, simplemente, L_1L_2, es el lenguaje definido como sigue:

$$L_1 \cdot L_2 = L_1L_2 = \{\omega_1\omega_2 \in V^* \mid \omega_1 \in L_1 \, y \, \omega_2 \in L_2\}.$$

En palabras, el lenguaje $L_1 \cdot L_2$ está formado por todas las palabras que resultan de concatenar cada una de las palabras de L_1 con cada una de las palabras de L_2 en este orden. Recuerde que la concatenación de palabras es una operación ordenada y que, en lo general, no es conmutativa; esto es, en general se tiene que $\omega_1\omega_2 \neq \omega_2\omega_1$. Debido a esto, se tiene que la concatenación de lenguajes también es una operación ordenada que, en lo general, tampoco es conmutativa; esto es, en general $L_1 \cdot L_2 \neq L_2 \cdot L_1$.

A manera de ejemplo, considérese el alfabeto $V = \{x, y, z\}$ y sean los lenguajes finitos $L_1 = \{x, xy, z\}$ y $L_2 = \{\varepsilon, y\}$. Entonces se tiene que:

$L_1 \cdot L_2 = \{x, xy, xyy, z, zy\}$;

$L_2 \cdot L_1 = \{x, xy, z, yx, yxy, yz\}$.

Claramente se tiene que $L_1 \cdot L_2 \neq L_2 \cdot L_1$. Pero no sólo ocurre la desigualdad, además sucede que $L_1 \cdot L_2 \nsubseteq L_2 \cdot L_1$ y $L_2 \cdot L_1 \nsubseteq L_1 \cdot L_2$, aunque $(L_1 \cdot L_2) \cap (L_2 \cdot L_1) = \{x.xy, z\} \neq \emptyset$.

La operación de concatenación de lenguajes satisface un buen número de propiedades, de entre las cuales es conveniente señalar las siguientes. Sean L y L_i para $i = 1,2,3$ lenguajes sobre un alfabeto común V.

- La operación es asociativa.

$$(L_1 \cdot L_2) \cdot L_3 = L_1 \cdot (L_2 \cdot L_3).$$

- La operación tiene un elemento neutro o identidad.

$$L \cdot \{\varepsilon\} = \{\varepsilon\} \cdot L = L.$$

- La operación tiene un elemento anulador.

$$L \cdot \emptyset = \emptyset \cdot L = \emptyset.$$

- Se cumple que $\varepsilon \in L_1 \cdot L_2$ si y sólo si $\varepsilon \in L_1$ y $\varepsilon \in L_2$.

Si uno de los lenguajes involucrados en la concatenación es resultado de la unión o intersección de lenguajes, se tienen las propiedades siguientes.

- Se cumple la distribución respecto a la unión.

$$L_1 \cdot (L_2 \cup L_3) = L_1 \cdot L_2 \cup L_1 \cdot L_3.$$

Este resultado se puede probar como sigue. Sea $\omega \in L_1 \cdot (L_2 \cup L_3)$. Entonces $\omega = \gamma\delta$ para las palabras $\gamma \in L_1$ y $\delta \in L_2 \cup L_3$. De esta última inclusión se deduce que $\delta \in L_2$ o $\delta \in L_3$. Si sucede que $\delta \in L_2$, entonces $\gamma\delta \in L_1 \cdot L_2$, y por lo tanto $\gamma\delta \in L_1 \cdot L_2 \cup L_1 \cdot L_3$. Por otro lado, si sucede que $\delta \in L_3$, entonces $\gamma\delta \in L_1 \cdot L_3$, con lo que de nuevo se tiene que $\gamma\delta \in L_1 \cdot L_2 \cup L_1 \cdot L_3$. En ambos casos se obtiene que $L_1 \cdot (L_2 \cup L_3) \subseteq L_1 \cdot L_2 \cup L_1 \cdot L_3$.

Suponga ahora que $\omega \in L_1 \cdot L_2 \cup L_1 \cdot L_3$. Entonces $\omega \in L_1 \cdot L_2$ o $\omega \in L_1 \cdot L_3$. Si sucede la primera inclusión, $\omega \in L_1 \cdot L_2$, entonces $\omega = \gamma\delta$ para las palabras $\gamma \in L_1$ y $\delta \in L_2$. De esta última inclusión se deduce que $\delta \in L_2 \cup L_3$ y, por lo tanto, sucede que $\gamma\delta \in L_1 \cdot (L_2 \cup L_3)$. Por otro lado, si $\omega \in L_1 \cdot L_3$, entonces $\omega = \gamma\delta$ para las palabras $\gamma \in L_1$ y $\delta \in L_3$. De esta última inclusión se puede establecer que $\delta \in L_2 \cup L_3$, y por lo tanto se deduce que $\gamma\delta \in L_1 \cdot (L_2 \cup L_3)$. Luego $L_1 \cdot L_2 \cup L_1 \cdot L_3 \subseteq L_1 \cdot (L_2 \cup L_3)$. Empleando el Teorema 1 se obtiene que $L_1 \cdot (L_2 \cup L_3) = L_1 \cdot L_2 \cup L_1 \cdot L_3$.

- No se cumple una distribución completa respecto a la intersección.

$$L_1 \cdot (L_2 \cap L_3) \subseteq L_1 \cdot L_2 \cap L_1 \cdot L_3.$$

- La concatenación y la diferencia son no distribuibles de forma similar a como lo son la concatenación y la intersección. En general, se tiene que:

$$L_1 \cdot (L_2 - L_3) \neq L_1 \cdot L_2 - L_1 \cdot L_3.$$

Con base en los resultados anteriores, se tiene que en $\mathcal{P}(V^*)$, la familia de todos los lenguajes sobre el alfabeto V, la concatenación es una operación binaria cerrada sobre $\mathcal{P}(V^*)$. (El conjunto $\mathcal{P}(V^*)$ también es cerrado para las operaciones booleanas de conjuntos y para la operación reversa de lenguajes.) Ahora bien, como la concatenación es una operación asociativa, que además presenta una identidad, así como un anulador, y que se distribuye sobre la operación de unión de lenguajes, entonces el sistema matemático $(\mathcal{P}(V^*),\cdot,\cup)$ es un semianillo. (El sistema $(\mathcal{P}(V^*),\cdot)$ es un monoide, algo similar a lo que ocurre con $(V^*,\cdot)$, aunque las operaciones $\cdot$ aludidas no son las mismas.) Si bien el enfoque algebraico de los lenguajes ya no será discutido más profundamente en estas notas, al final de este texto se incluirá una breve sección en la que se describen algunas propiedades básicas de los sistemas algebraicos llamados semianillos.

4.2.2. Potencia

Ciertamente la definición de la concatenación de lenguajes parece señalar dos lenguajes diferentes, aunque nada impide que un lenguaje L se concatene consigo mismo, obteniéndose el lenguaje siguiente:

$$L \cdot L = \{\omega_1\omega_2 \in V^* \mid \omega_1, \omega_2 \in L\}.$$

Esto es, el lenguaje $L \cdot L$ está constituido por todas las palabras que se forman concatenando dos palabras de L, no necesariamente diferentes, en todos los órdenes posibles. Debido a esto, se acostumbra usar la designación L^2, lo cual se lee "L a la dos", para referirse a $L \cdot L$.

Considerando nuevamente los lenguajes $L_1 = \{x, xy, z\}$ y $L_2 = \{\varepsilon, y\}$, se obtienen los dos lenguajes siguientes:

${L_1}^2 = \{xx, xxy, xz, xyx, xyxy. xyz, zx, zxy, zz\}$.

Sean las palabras xy y z de L_1. Entonces las palabras $xy \cdot z = xyz$ y $z \cdot xy = zxy$ pertenecen a L_1. Las palabras xy y z se concatenaron en los dos órdenes posibles. De manera similar se construyen las restantes palabras de L_1.

$L_2{}^2 = \{\varepsilon, y, yy\}$.

Dado que ε, y son las únicas palabras de L_2, se deben considerar las concatenaciones $\varepsilon \cdot \varepsilon = \varepsilon$, $\varepsilon \cdot y = y \cdot \varepsilon = y$ y $y \cdot y = yy$ para formar las palabras de $L_2{}^2$.

En lo anterior cabe destacar el efecto que tiene la inclusión de la palabra vacía ε en un lenguaje al calcular concatenaciones.

Nada impide que un lenguaje L se concatene consigo mismo más de dos veces. Por decir, se puede considerar el lenguaje siguiente:

$$L \cdot L^2 = \{\omega\gamma \in V^* \mid \omega \in L \; y \; \gamma \in L^2\} = \{\omega_1\omega_2\omega_3 \in V^* \mid \omega_1, \omega_2, \omega_3 \in L\}.$$

La última igualdad resulta de considerar que toda palabra $\gamma \in L^2$ está formada por la concatenación de dos palabras de L. Con base en esta observación no resulta difícil mostrar que se cumple la igualdad $L \cdot L^2 = L^2 \cdot L$. Resulta entonces que el lenguaje $L \cdot L^2$ está constituido por todas las palabras que se forman al concatenar tres palabras de L, no necesariamente diferentes entre sí, en todos los órdenes posibles, y por esta razón dicho lenguaje es referido como L^3, "L a la tres".

Con base en lo que se ha obtenido, es factible definir el lenguaje L^n, para algún número natural n dado, como el lenguaje constituido por todas las palabras que se forman al concatenar n palabras de L, no necesariamente diferentes entre sí, en todos los órdenes posibles. Para hacer consistente la definición es necesario definir primero que $L^0 = \{\varepsilon\}$ y que $L^1 = L$. Así pues, se define la *potencia* n, para algún $n \geq 0$, de un lenguaje L como el lenguaje:

$$L^n = \begin{cases} \{\varepsilon\} & \text{si} \quad n = 0; \\ L \cdot L^{n-1} & \text{si} \quad n > 0. \end{cases}$$

Por consistencia, el lenguaje L^n se lee "L a la n".

La definición anterior establece que L^0 está formado por todas las palabras que se obtienen concatenando cero palabras de L, y también se ha señalado que la concatenación de cero palabras resuelta en la palabra vacía ε. Similarmente, L^1 está formado por todas las palabras que se obtienen concatenando una sola palabra de L, y que la concatenación de una sola palabra es la palabra misma. Esta interpretación es consistente con la operación señalada en la definición:

$$L^1 = L \cdot L^0 = L \cdot \{\varepsilon\} = L.$$

El resto es consistente con las concatenaciones repetidas que se habían introducido inicialmente:

$$L^2 = L \cdot L^1 = L \cdot L.$$

$$L^3 = L \cdot L^2 = L \cdot L \cdot L.$$

Y así sucesivamente.

La definición de la potencia origina algunos resultados curiosos. Por ejemplo, se tiene que $\emptyset^0 = \{\varepsilon\}^0 = \{\varepsilon\}$, pero $\emptyset^n$ es diferente de $\{\varepsilon\}^n$ para todo $n > 0$; un resultado posiblemente desconcertante pero correcto.

No es difícil mostrar que si $L_1 \subseteq L_2$, para lenguajes L_1 y L_2 sobre un mismo alfabeto V, entonces ${L_1}^n \subseteq {L_2}^n$, para todo $n \geq 0$.

Este resultado se prueba como sigue. Si $\omega \in {L_1}^n$, entonces $\omega = \omega_1\omega_2 \ldots \omega_n$ donde $\omega_i \in L_1$ para $i = 1, \ldots, n$. Pero como $L_1 \subseteq L_2$, se tiene entonces que $\omega_i \in L_2$ para todo $i = 1, \ldots, n$. De esta manera se tiene que $\omega = \omega_1\omega_2 \ldots \omega_n \in {L_2}^n$, con lo que ${L_1}^n \subseteq {L_2}^n$.

4.2.3. Reverso

También se puede desarrollar la idea de reverso, o imagen inversa o especular, de un lenguaje. El *reverso* de un lenguaje L sobre el alfabeto Σ es el conjunto:

$$L^R = \{\omega^R \in V^* \mid \omega \in L\}.$$

Esto es, el reverso de un lenguaje L es el lenguaje formado por todas las palabras reversas de L.

El reverso de un lenguaje también tiene propiedades interesantes, algunas de las cuales son las siguientes. Nuevamente, en lo que sigue L y L_i, para $i = 1, 2$, son lenguajes sobre un alfabeto común V.

Considérense primero algunos lenguajes muy simples.

- Si el alfabeto está formado por un solo símbolo, digamos $V = \{a\}$, entonces resulta que $L^R = L$ para cualquier lenguaje L sobre V.
- Para el lenguaje V^1, el lenguaje de todas las palabras unitarias sobre V, resulta que $(V^1)^R = V^1$.

Considérense ahora lenguajes más generales.

- Obsérvese que si se vuelve a realizar el reverso del reverso de todas las palabras de un lenguaje, entonces se obtiene, de nuevo, el lenguaje original. Por lo tanto, $(L^R)^R = L$.
- El reverso de la concatenación no sólo invierte las palabras concatenadas de los lenguajes sino que también cambia el orden de la concatenación de los lenguajes.

$$(L_1 \cdot L_2)^R = {L_2}^R \cdot {L_1}^R.$$

Este último resultado se puede probar como sigue. Sea $\omega \in (L_1 \cdot L_2)^R$. Entonces se tiene que $\omega^R \in L_1 \cdot L_2$, y se tiene que $\omega^R = \gamma\delta$ para las palabras $\gamma \in L_1$ y $\delta \in L_2$. Por lo tanto, $\omega = (\omega^R)^R = (\gamma\delta)^R = \delta^R\gamma^R$. Pero dado que $\delta \in L_2$, entonces $\delta^R \in {L_2}^R$, y dado que $\gamma \in L_1$, entonces $\gamma^R \in {L_1}^R$, y por lo tanto se obtiene que $\omega \in {L_2}^R \cdot {L_1}^R$. Esto prueba que $(L_1 \cdot L_2)^R \subseteq {L_2}^R \cdot {L_1}^R$. De manera conversa, si $\omega \in {L_2}^R \cdot {L_1}^R$, entonces $\omega = \gamma\delta$ para las palabras $\gamma \in {L_2}^R$ y $\delta \in {L_1}^R$. Pero entonces $\omega^R = \delta^R\gamma^R \in L_1 \cdot L_2$ y $\omega \in (L_1 \cdot L_2)^R$. Por eso ${L_2}^R \cdot {L_1}^R \subseteq (L_1 \cdot L_2)^R$, y por lo tanto $(L_1 \cdot L_2)^R = {L_2}^R \cdot {L_1}^R$.

A partir de este resultado, no resulta difícil mostrar que $(L^n)^R = (L^R)^n$, para $n \geq 0$.

4.3. Cerraduras

Si una operación (o un mapeo) sobre los miembros de un conjunto produce imágenes que también son miembros del mismo conjunto, entonces el conjunto se dice que es *cerrado* bajo esa operación, y esta propiedad se denomina la propiedad de *cerradura*. La definición de las operaciones binarias o n-arias implica que los conjuntos sobre los que están definidas tales operaciones están cerrados bajo estas operaciones. Esta propiedad distingue a las operaciones binarias o n-arias de otras funciones.

La operación binaria de concatenación sobre el conjunto $\mathcal{P}(V^*)$ permite definir un par de cerraduras muy importantes. La primera de ellas es la llamada *cerradura estrella* de Kleene, u *operación estrella* de Kleene. Antes de definir esta cerradura es conveniente introducir una definición preliminar.

Sea L un lenguaje sobre el alfabeto V. Se define el lenguaje siguiente:

$$L^{\leq n} = \bigcup_{i=0}^{n} L^i = L^0 \cup L^1 \cup \cdots \cup L^n.$$

Esto es, $L^{\leq n}$ es el lenguaje formado por todas las palabras que se obtienen concatenando no más de n palabras de L, no todas ellas diferentes entre sí, y en todos los órdenes posibles. Dicho lo anterior de otra manera, si ω es una palabra de $L^{\leq n}$, entonces dicha palabra es de la forma general $\omega = \omega_{i_1}\omega_{i_2}\cdots\omega_{i_m}$, para algún $m \leq n$, y donde ω_i es una palabra de L.

A partir de lo anterior, la cerradura estrella de Kleene se define como sigue. Para cualquier lenguaje L sobre un alfabeto V, su cerradura estrella, denotada L^* y que se lee "L estrella", es el lenguaje:

$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L^2 \cup \ldots$$

Literalmente, L^* es el lenguaje formado por todas las palabras de L^0 junto con todas las palabras de L^1, junto con todas las palabras de L^2, y así sucesivamente. Ahora bien, con base en la definición de la potencia n de un lenguaje, se tiene que L^* es el lenguaje formado por todas las palabras que se pueden obtener realizando cero o más concatenaciones de palabras de L en todos los órdenes posibles y donde se permite el uso de una misma palabra múltiples veces. Se debe entender que se realiza un número arbitrario, pero finito, de concatenaciones. Esto es, toda palabra ω de L^* es de la forma general $\omega = \omega_{i_1}\omega_{i_2}\cdots\omega_{i_n}$, para algún $n \geq 0$, y donde ω_i es una palabra de L.

La interpretación anterior también permite deducir que L^* es el superconjunto más pequeño de L que contiene a la palabra vacía ε y que es cerrado bajo la operación de concatenación. Esta última aseveración indica que si $\omega_1, \omega_2 \in L^*$, entonces $\omega_1\omega_2 \in L^*$.

Con base en la definición anterior se pueden establecer las igualdades siguientes:

$\emptyset^* = \bigcup_{i=0}^{\infty} \emptyset^i = \emptyset^0 \cup \emptyset^1 \cup \emptyset^2 \cup \ldots = \{\varepsilon\} \cup \emptyset \cup \emptyset \cup \ldots = \{\varepsilon\}$.

$\{\varepsilon\}^* = \bigcup_{i=0}^{\infty} \{\varepsilon\}^i = \{\varepsilon\}^0 \cup \{\varepsilon\}^1 \cup \{\varepsilon\}^2 \cup \ldots = \{\varepsilon\} \cup \{\varepsilon\} \cup \{\varepsilon\} \cup \ldots = \{\varepsilon\}$.

$(V^1)^* = \bigcup_{i=0}^{\infty} (V^1)^i = \bigcup_{i=0}^{\infty} V^i = V^*$ sobre cualquier alfabeto V.

La última igualdad hace consistente la notación empleada previamente para designar al lenguaje universal V^*.

$(V^2)^* = \bigcup_{i=0}^{\infty} (V^2)^i = \bigcup_{i=0}^{\infty} V^{2i} = \{\omega \in V^* \mid |\omega| = 2n \text{ para } n \geq 0\}$.

$(V^3)^* = \bigcup_{i=0}^{\infty}(V^3)^i = \bigcup_{i=0}^{\infty} V^{3i} = \{\omega \in V^* \mid |\omega| = 3n \text{ para } n \geq 0\}$.

$(V^4)^* = \bigcup_{i=0}^{\infty}(V^4)^i = \bigcup_{i=0}^{\infty} V^{4i} = \{\omega \in V^* \mid |\omega| = 4n \text{ para } n \geq 0\}$.

Nótese ahora lo siguiente: $(V^2)^* \cap (V^3)^* = \{\varepsilon\}$, $(V^4)^* \subset (V^2)^*$ y $(V^2)^* \cap (V^4)^* = (V^4)^*$.

También se define la *cerradura positiva* de Kleene, u *operación positiva*, u *operación más* de Kleene. Esta nueva cerradura se establece como sigue. Para cualquier lenguaje L sobre un alfabeto V, su cerradura positiva, denotada L^+ que se lee "L más", es el lenguaje siguiente:

$$L^+ = \bigcup_{i=1}^{\infty} L^i = L^1 \cup L^2 \ldots$$

Así, la cerradura positiva de un lenguaje L es el lenguaje que contiene a todas las palabras que se forman al concatenar una o más palabras de L, en todos los órdenes posibles y permitiendo que una misma palabra pueda ser usada múltiples veces. Nuevamente, se concatena un número arbitrario, pero finito, de palabras. Con base en esta nueva definición se tiene que $\emptyset^+ = \emptyset$, $\{\varepsilon\}^+ = \{\varepsilon\}$ y $(V^1)^+ = \bigcup_{i=1}^{\infty}(V^1)^i = \bigcup_{i=1}^{\infty} V^i = V^+$, el conjunto de todas las palabras no vacías sobre el alfabeto V.

De las dos definiciones anteriores se puede obtener de inmediato la igualdad siguiente:

$$L^+ = \begin{cases} L^* & \text{si} \quad \varepsilon \in L; \\ L^* - \{\varepsilon\} & \text{si} \quad \varepsilon \notin L. \end{cases}$$

Este resultado marca la importancia que tiene la inclusión de la palabra vacía en un lenguaje.

A continuación se describen algunas propiedades de las dos cerraduras. En lo que sigue, L y L_i, para $i = 1,2$, son lenguajes sobre un alfabeto común V.

- El primer resultado establece una relación de inclusión entre las dos cerraduras.

  $$L \subseteq L^+ \subseteq L^*.$$

  El resultado se deriva directamente de las definiciones; recuérdese que $L = L^1$.
- Es más, se tiene que $L^n \subseteq V^*$ para todo $n \geq 0$ y, por lo tanto, $L^* \subseteq V^*$ y $L^+ \subseteq V^+$. Observe también que dado que $L^n \subseteq L^*$ para todo n, entonces se tiene que $L^+ \subseteq L^*$.
- Existe una relación más entre las cerraduras, dada esta vez con base en la concatenación.

$$L^+ = L^*L = LL^*.$$

La prueba de este resultado es como sigue. Si $\omega \in L^+$, entonces $\omega \in \bigcup_{i=1}^{\infty} L^i$ y, por lo tanto, existe algún número natural $n > 0$ tal que $\omega \in L^n$. De esto se deduce que $\omega \in L^{n-1}L$ con $n - 1 \geq 0$ y con ello sucede que $\omega \in \bigcup_{i=0}^{\infty}(L^iL) = \left(\bigcup_{i=0}^{\infty} L^i\right)L = L^*L$. Así que $\omega \in L^*L$ y, en consecuencia, se tiene que $L^+ \subseteq L^*L$. Por otro lado, si $\omega \in L^*L$, entonces sucede que $\omega \in \left(\bigcup_{i=0}^{\infty} L^i\right)L = \bigcup_{i=0}^{\infty}(L^iL)$. Esto es, existe algún número natural $n \geq 0$ tal que $\omega \in L^nL$. De esto se tiene que $\omega \in L^{n+1}$ con $n + 1 > 0$. Así pues, $\omega \in L^+$ y con ello $L^*L \subseteq L^+$. De esta manera se tiene que $L^+ = L^*L$. La segunda igualdad se prueba empleando un razonamiento similar.

- Hay otra relación basada ahora en la inclusión de lenguajes.

  Si $L_1 \subseteq L_2$, entonces ${L_1}^* \subseteq {L_2}^*$. (Similarmente, ${L_1}^+ \subseteq {L_2}^+$).

Este resultado se deduce directamente del hecho de que ${L_1}^n \subseteq {L_2}^n$ para cualquier número natural n.

Es conveniente preguntarse cómo son los lenguajes que son cerraduras de cerraduras. Las respuestas son sorprendentemente sencillas.

- $(L^*)^* = L^*$ (y, similarmente, $(L^+)^+ = L^+$).

La prueba de estos resultados se obtiene como sigue. Primero, nótese que $L^* \subseteq (L^*)^*$ ya que $L^* = (L^*)^1$. Considérese ahora que $\omega \in (L^*)^*$. Entonces existe algún número natural n tal que $\omega = \omega_1 \ldots \omega_n$ con $\omega_i \in L^*$ para $1 \leq i \leq n$. Por lo tanto, para cada i anterior existe otro número natural m_i tal que $\omega_i \in L^{m_i}$. De esto se deduce que $\omega_i = \omega_{i_1} \ldots \omega_{i_{m_i}}$ con $\omega_{i_j} \in L$ para $1 \leq j \leq m_i$. De esta manera se tiene que $\omega \in L^{m_1+\cdots+m_n}$ y, en consecuencia, $\omega \in L^*$ y $(L^*)^* \subseteq L^*$.

El segundo resultado se prueba usando razonamientos similares a los anteriores. Supóngase pues que $\omega \in (L^+)^+$. Así pues, ya que $(L^+)^+ = \bigcup_{i=1}^{\infty}(L^+)^i$, se tiene que para algún número natural $n \geq 1$, $\omega \in (L^+)^n$ y, por lo tanto, $\omega = \omega_1 \ldots \omega_n$, donde cada $\omega_i \in L^+ = \bigcup_{j=1}^{\infty} L^j$. De esta última inclusión se deduce que existe algún número natural $m_i \geq 1$ para el cual sucede que $\omega_i \in L^{m_i}$. Por lo tanto, cada palabra ω_i es de la forma $\omega_i = \gamma_{i_1} \ldots \gamma_{i_{m_i}}$, donde cada $\gamma_{i_j} \in L$. De esta manera se tiene que $\omega = \left(\gamma_{1_1} \ldots \gamma_{1_{m_1}}\right) \ldots \left(\gamma_{n_1} \ldots \gamma_{n_{m_n}}\right)$. Pero ésta es justamente una cadena perteneciente al len-

guaje $L^{m_1+\cdots+m_n}$, y puesto que $m_i \geq 1$ para cada i, se tiene que $m_1 + \cdots + m_n \geq 1$. Así pues, resulta que $\omega \in L^+$ y, por lo tanto, $(L^+)^+ \subseteq L^+$. Por otro lado, puesto que $L^+ = (L^+)^1$, se tiene que $L^+ \subseteq (L^+)^+$. Combinando ambos resultados, se llega a que $(L^+)^+ = L^+$.

Otras cerraduras de cerraduras son las siguientes.

- $(L^+)^* = L^*$. (También se tiene que $(L^*)^+ = L^*$.)

A continuación se proporciona una prueba del primer resultado; el segundo resultado se prueba siguiendo un razonamiento similar. En primer término se tiene que $L \subseteq L^+$ y, en consecuencia, se tiene que $L^* \subseteq (L^+)^*$. Por otro lado, se tiene que $L^+ \subseteq L^*$ y de esto se deduce que $(L^+)^* \subseteq (L^*)^* = L^*$.

Estos últimos resultados se pueden interpretar como una indicación de que no se añaden nuevas palabras a los lenguajes L^* o L^+, aunque se vuelva a realizar sobre ellos cualquier tipo de cerradura. Esto desvela intuitivamente lo que significa el término *cerradura*.

A continuación se presentan un par de ejemplos peculiares sobre ciertos lenguajes y sus cerraduras.

Ejemplo 1

Considérese el alfabeto $V = \{0, 1, 2\}$ y defínase a L como el lenguaje formado por todas las palabras que no contienen al símbolo 2. De esta manera, ε, 0, 1 y 0110 son algunos ejemplos de palabras de L.

Ahora bien, obsérvese que al considerar algún número natural $k \geq 1$ y una palabra $\omega \in L^k$, entonces resulta que dicha palabra debe tener la forma $\omega = \omega_1 \ldots \omega_k$, donde las palabras $\omega_i \in L$ para $1 \leq i \leq k$. Puesto que ninguna de las palabras ω_i contienen al símbolo 2, entonces la palabra ω tampoco lo contiene. Por lo tanto, resulta que $\omega \in L$ y, en consecuencia, se tiene que $L^k \subseteq L$ para $k \geq 1$.

Por otro lado, cualquier palabra $\omega \in L$ se puede expresar como $\omega = \varepsilon^{k-1}\omega$ para algún número natural $k \geq 1$, con lo que se deduce que $\omega \in L^k$. De esto resulta que $L \subseteq L^k$ para $k \geq 1$ y, combinando los dos resultados obtenidos, se concluye que $L = L^k$ para todo $k \geq 1$. Además, dado que $L^0 = \{\varepsilon\} \subseteq L$, se deduce de inmediato que $L^* = L^0 \cup L^+ = L^0 \cup L = L$.

Se debe remarcar que el hecho de que los lenguajes L^* y L^+ coincidan en este caso depende totalmente del alfabeto sobre el que están definidos.

Ejemplo 2

Considérese el lenguaje $L = \{ab\}$ definido sobre el alfabeto $V = \{a, b\}$. Para este lenguaje se tiene que $L^+ = \{(ab)^i \mid i \geq 1\}$. Considérese ahora al lenguaje $(L^+)^2$, es cual está dado por $(L^+)^2 = \{(ab)^i \mid i \geq 2\}$. Nótese que $(L^+)^2 \subseteq L^+$ y que $(L^+)^2 \cap L^+ = (L^+)^2$. Considérese ahora al lenguaje $(L^+)^3$, es cual está dado por $(L^+)^3 = \{(ab)^i \mid i \geq 3\}$. Nótese ahora que $(L^+)^3 \subseteq (L^+)^2 \subseteq L^+$ y que $(L^+)^3 \cap (L^+)^2 = (L^+)^3 \cap L^+ = (L^+)^3$. Procediendo de manera similar a lo anterior, se puede definir cualquier lenguaje $(L^+)^k$ para cualquier número natural $k \geq 1$, para concluir finalmente que $(L^+)^+ = L^+$ como era de esperarse.

La situación cambia ligeramente si ahora se considera al lenguaje $M = \{\varepsilon, ab\}$ sobre el mismo alfabeto V. En este nuevo caso se tiene que $M^+ = \{(ab)^i \mid i \geq 0\}$. Pero ahora se tiene lo siguiente $(M^+)^2 = \{(ab)^i \mid i \geq 0\}$ y $(M^+)^3 = \{(ab)^i \mid i \geq 0\}$. Nótese que ahora sucede que $(M^+)^3 = (M^+)^2 = M^+$, y de hecho sucede que $(M^+)^k = M^+$ para cualquier número natural $k \geq 1$. Al final sucede que $(M^+)^+ = M^+$ como era de esperarse. Este ejemplo simple muestra nuevamente el efecto que puede generar la inclusión de la palabra vacía en un lenguaje.

4.3.1. Especificación de lenguajes

Ya se había mencionado la estrategia de especificación de lenguajes que parte de algunos lenguajes iniciales bien especificados y que se combinan mediante las operaciones de lenguajes para obtener nuevos lenguajes. Se señaló que era razonable comenzar con algunos de los lenguajes finitos pequeños sobre algún alfabeto V dado, debido a que todos los elementos de tales conjuntos pueden ser listados explícitamente. Posteriormente se pueden formar combinaciones finitas con estos lenguajes iniciales usando las operaciones para lenguajes. A manera de ejemplo de esta estrategia considérese el caso siguiente. Sea el alfabeto $V = \{a, b, c\}$. Considérense ahora los lenguajes finitos iniciales siguientes:

$$\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}.$$

Con estos lenguajes ahora se pueden formar los lenguajes siguientes:

$$L_1 = \{a\}\{a, c\} = \{aa, ac\}.$$

$L_2 = \{a, b, c\}L_1 = \{a, b, c\}\{aa, ac\} = \{aaa, aac, baa, bac, caa, cac\}.$

$L_3 = L_1\{a, b, c\} = \{aa, ac\}\{a, b, c\} = \{aaa, aab, aac, aca, acb, acc\}.$

$$\begin{aligned} L_4 &= L_2 \cup L_3 \\ &= \{aaa, aac, baa, bac, caa, cac\} \cup \{aaa, aab, aac, aca, acb, acc\} \\ &= \{aaa, aac, baa, bac, caa, cac, aab, aca, acb, acc\}. \end{aligned}$$

$$\begin{aligned} L_5 &= L_2 \cap L_3 \\ &= \{aaa, aac, baa, bac, caa, cac\} \cap \{aaa, aab, aac, aca, acb, acc\} \\ &= \{aaa, aac\}. \end{aligned}$$

$$\begin{aligned} L_6 &= L_2 - L_3 \\ &= \{aaa, aac, baa, bac, caa, cac\} - \{aaa, aab, aac, aca, acb, acc\} \\ &= \{baa, bac, caa, cac\}. \end{aligned}$$

$$\begin{aligned} L_7 &= L_3 - L_2 \\ &= \{aaa, aab, aac, aca, acb, acc\} - \{aaa, aac, baa, bac, caa, cac\} \\ &= \{aab, aca, acb, acc\}. \end{aligned}$$

Nótese que todos estos lenguajes están completamente especificados, sus elementos están listados explícitamente. Ciertamente, los lenguajes anteriores son finitos y muy pequeños, pero la estrategia ilustrada, en principio, permite construir lenguajes mucho más grandes y, de hecho, permite construir lenguajes infinitos. Para esto, las operaciones de cerradura son de suma importancia, como se ilustra a continuación.

Considérese nuevamente el alfabeto $V = \{a, b, c\}$ y sobre él considérense ahora los lenguajes siguientes:

$M_1 = \emptyset^* = \{\epsilon\}.$

$M_2 = M_1 \cup \{a, b\} = \{\varepsilon, a, b\}.$

$M_3 = M_2\{c, b\} = \{\varepsilon, a, b\}\{c, b\} = \{c, b, ac, ab, bc, bb\}.$

$M_4 = \{c, b\}M_2 = \{c, b\}\{\varepsilon, a, b\} = \{c, ca, cb, b, ba, bb\}.$

$M_5 = M_3 \cap M_4 = \{c, b, bb\}.$

$M_6 = M_5{}^* = \{\omega \in V^* \mid |\omega|_a = 0\}.$

Claramente, el último lenguaje construido en el ejemplo anterior es un lenguaje infinito; y también debe ser claro que este lenguaje está completamente especificado.

Otros ejemplos de lenguajes son los siguientes:

$N_1 = \{a, b\}^*\{c\} \cup \{b, c\}.$

$N_2 = \{a, b, c\}^*\{a\}\{a\}.$

Obsérvese que N_1 es el conjunto de todas las palabras formadas con los símbolos a, b que terminan con una sola c, junto con las palabras b y c. Por su parte, N_2 es el conjunto de todas las palabras formadas con los símbolos a, b, c que terminan con la secuencia $aa = a^2$. La operación de complemento de lenguajes permite construir conjuntos como el siguiente:

$N_3 = \overline{(\{a,b,c\}^*\{a\}\{a\})}$.

Éste es el conjunto de todas las palabras formadas con los símbolos a, b, c que no terminan con la secuencia $aa = a^2$.

Se debe mencionar que muchos de los lenguajes construidos en los ejemplos anteriores se pueden obtener realizando otras combinaciones de lenguajes iniciales y operaciones. El objetivo principal del enfoque no es mostrar que un cierto lenguaje se puede construir unívocamente a partir de ciertos lenguajes iniciales, sino mostrar que es factible resolver el problema de especificación de lenguajes empleando operaciones para lenguajes. En otro documento (Los Lenguajes Regulares y sus Propiedades) se va emplear una variante de este método para construir una familia muy importante de lenguajes formales.

4.4. Otras operaciones

4.4.1. Cocientes de palabras

En muchas situaciones teóricas y prácticas es importante conocer los prefijos y los sufijos de las palabras de un lenguaje respecto a una palabra fija dada. Antes de establecer las operaciones sobre lenguajes que dependen de los prefijos y sufijos es conveniente repasar estos conceptos. Así, para cualesquiera dos palabras u, v sobre un cierto alfabeto V se pueden establecer las relaciones siguientes:

u es un *prefijo* de v si existe una palabra z sobre V tal que $v = uz$.

u es un *sufijo* de v si existe una palabra z sobre V tal que $v = zu$.

u es un *factor* (o *subpalabra*, aunque este término está en desuso) de v si existen palabras z, z' sobre V tales que $v = zuz'$.

u es una *subpalabra dispersa* de v si v como una secuencia de letras contiene a u como una subsecuencia, esto es, existen palabras $z_1, \ldots, z_t$ y $y_0, \ldots, y_t$ sobre V tales que $u = z_1 \ldots z_t$ y $v = y_0 z_1 y_1 \ldots z_t y_t$.

Las relaciones anteriores se mantienen incluso cuando $u = \varepsilon$ o $u = v$; cuando estos casos triviales son excluidos, las relaciones se denominan *propias*. Sólo la primera relación va a ser de interés en lo que sigue.

Si sucede que $v = uz$, entonces se puede escribir $u = vz^{-1}$ y $z = u^{-1}v$, y se dice que u es el *cociente por la derecha de* v *por* z, y que z es el *cociente por la izquierda de* v *por* u. Coloquialmente, el cociente por la derecha de v por z es la subsecuencia que queda si a v se le quita su parte final z. Similarmente, el cociente por la izquierda de v por u es la subsecuencia que queda si a v se le quita su parte inicial u. A continuación se extienden estas nociones a las palabras de un cierto lenguaje.

Cociente por la derecha

Dados la palabra $u \in V^*$ y el lenguaje $L \subseteq V^*$, se define el *cociente por la derecha de* L *por* u, el cual se denota $u^{-1}L$, o bien $u \setminus L$, al lenguaje formado por todos los finales de las palabras de L cuyo prefijo es u; tales factores se les suele llamar los buenos finales de u en L. Esto es, se define al lenguaje:

$$u^{-1}L = u \setminus L = \{v \in V^* \mid uv \in L\}.$$

Esta definición también se puede parafrasear de la manera siguiente. Si $w = uv$ es una palabra de L, entonces v es el cociente por la izquierda de w por u. De esta manera, $u^{-1}L$ es el conjunto de todos los cocientes por la izquierda de las palabras de L por u. Nótese lo paradójico, y enredado, de la situación. Para obtener el cociente por la derecha de un lenguaje por una palabra dada se deben realizar las operaciones de cociente por la izquierda de todas las palabras del lenguaje por la palabra dada.

Cociente por la izquierda

Análogamente a lo anterior, dados $u \in V^*$ y $L \subseteq V^*$ se define el *cociente por la izquierda de* L *por* u, el cual se denota Lu^{-1}, o bien $u \,/\, L$, como el lenguaje formado por todos los buenos comienzos de las palabras de L cuyo sufijo es u. De esta manera, se establece el lenguaje:

$$Lu^{-1} = u/L = \{v \in V^* \mid vu \in \Sigma^*\}.$$

Y, de manera similar al caso anterior, si $w = vu$ es una palabra de L, entonces v es el cociente por la derecha de w por u. De esta manera, Lu^{-1} es el conjunto de todos los cocientes por la derecha de las palabras de L por u. Así, para obtener el cociente por la

izquierda de un lenguaje por una palabra dada se deben realizar las operaciones de cociente por la derecha de todas las palabras del lenguaje por la palabra dada.

A continuación se presentan unas cuantas propiedades del cociente por la derecha de un lenguaje por una palabra dada; las propiedades del cociente por la izquierda son análogas y se omiten.

En lo que sigue se debe considerar que a es una palabra unitaria y u, v son palabras arbitrarias sobre el alfabeto V, mientras que L y L_i, para $i = 1, 2$, son lenguajes sobre el mismo alfabeto.

Si sucede que $L_1 \subseteq L_2$, entonces se tiene que $u^{-1}L_1 \subseteq u^{-1}L_2$.

Sea $v \in u^{-1}L_1$. Esto quiere decir que $uv \in L_1$. Y esto, a su vez, quiere decir que $uv \in L_2$, ya que $L_1 \subseteq L_2$. Así pues, se tiene que $v \in u^{-1}L_2$ como se quería probar.

Sucede que $u^{-1}(L_1 \cup L_2) = u^{-1}L_1 \cup u^{-1}L_2$.

Sea $v \in u^{-1}(L_1 \cup L_2)$. De esto se deduce que $uv \in (L_1 \cup L_2)$, y de esto se deduce, a su vez, que $uv \in L_1$ o que $uv \in L_2$. De la primera inclusión se deduce que $v \in u^{-1}L_1$, mientras que de la segunda se deduce que $v \in u^{-1}L_2$. De todo esto se deduce que $v \in u^{-1}L_1 \cup u^{-1}L_2$, que es lo que se quería probar.

Sucede que $u^{-1}(L_1 \cap L_2) = u^{-1}L_1 \cap u^{-1}L_2$.

La prueba es similar a la del resultado anterior.

Sucede que $a^{-1}(L_1L_2) = \begin{cases} (a^{-1}L_1)L_2 & \text{si} \quad \varepsilon \notin L_1; \\ (a^{-1}L_1)L_2 \cup a^{-1}L_2 & \text{si} \quad \varepsilon \in L_1. \end{cases}$

El primer resultado se prueba como sigue. Si $\varepsilon \notin L_1$, entonces toda palabra de L_1 es de longitud mayor o igual a uno; debido a esto, la operación planteada consiste en calcular los buenos finales de la palabra a en L_1 y concatenar lo obtenido con las palabras de L_2. Es decir, se obtiene $(a^{-1}L_1)L_2$.

El segundo resultado se prueba usando un razonamiento similar al anterior. Ya que $\varepsilon \in L_1$, considérese el lenguaje $L' = L_1 - \{\varepsilon\}$, el cual no contiene a la palabra vacía. Con esta definición, y teniendo en cuenta la propiedad distributiva de la concatenación con la unión y el resultado probado antes, se puede establecer lo siguiente:

$$\begin{aligned} a^{-1}(L_1L_2) &= a^{-1}\big((L' \cup \{\varepsilon\})L_2\big) \\ &= a^{-1}(L'L_2 \cup L_2) \\ &= a^{-1}(L'L_2) \cup a^{-1}L_2. \end{aligned}$$

Como L' no contiene a la palabra vacía, la última expresión coincide con $a^{-1}(L')L_2 \cup a^{-1}L_2$. Y como la única diferencia entre L' y L_1 es la palabra vacía y ésta no interviene en el cálculo de los buenos finales se establece el resultado buscado.

Sucede que $a^{-1}L^* = (a^{-1}L)L^*$.

Este resultado se prueba en dos partes. En la primera se asume que $\varepsilon \notin L$, con lo cual se puede establecer que $L^* = LL^* \cup \{\varepsilon\}$. Esto implica que $a^{-1}L^* = a^{-1}(LL^* \cup \{\varepsilon\}) = a^{-1}(LL^*) = (a^{-1}L)L^*$, ya que $a^{-1}\{\varepsilon\} = \emptyset$.

En la segunda parte se asume que $\varepsilon \in L$. Debido a esto, se considera entonces al lenguaje $L_1 = L - \{\varepsilon\}$; nótese que $L^* = {L_1}^*$ a pesar de que L_1 no contiene a la palabra vacía. De esta manera, se deduce que $a^{-1}L_1 = a^{-1}L$, ya que la única diferencia entre L y L_1 es la palabra vacía. Así pues, se tiene que $a^{-1}L^* = a^{-1}{L_1}^* = (a^{-1}L_1){L_1}^* = (a^{-1}L_1)L^* = (a^{-1}L)L^*$.

4.4.2. Cocientes de lenguajes

Las nociones de cocientes de palabras por una palabra presentadas inicialmente se han extendido a cocientes de las palabras de un lenguaje por una palabra. Ahora se extiende nuevamente la noción para considerar cocientes de las palabras de un lenguaje por las palabras de otro lenguaje. Estas nuevas nociones ya se presentan de manera muy breve.

Cociente por la derecha

Considérese que L_1 y L_2 son dos lenguajes sobre un mismo alfabeto V. Se define al *cociente por la derecha* de las palabras de L_2 por las palabras de L_1, que se denota ${L_1}^{-1}L_2$, como el conjunto siguiente:

$$ {L_1}^{-1}L_2 = \{v \in V^* \mid uv \in L_2, u \in L_1\}. $$

Es decir, se ha definido al conjunto de todos los buenos finales en L_2 para palabras de L_1.

Cociente por la izquierda

Nuevamente considérese que L_1 y L_2 son dos lenguajes sobre un mismo alfabeto V. Se define ahora al *cociente por la izquierda* de las palabras de L_2 por las palabras de L_1, que se denota $L_2{L_1}^{-1}$, como el conjunto siguiente:

$$ L_2{L_1}^{-1} = \{v \in \Sigma^* \mid vu \in L_2, u \in L_1\}. $$

Es decir, se ha definido al conjunto de todos los buenos comienzos en L_2 para palabras de L_1.

4.4.3. Sustitución

Dados dos alfabetos V_1 y V_2, se define una función s tal que a cada palabra unitaria sobre V_1(o símbolo si se procede con precaución) se le hace corresponder un lenguaje sobre V_2. Esto es, se tiene que:

$$s: {V_1}^1 \to \mathcal{P}({V_2}^*).$$

Para que esta función sea de utilidad en la teoría de los lenguajes formales es necesario hacer lo siguiente. Se considera una extensión natural de la aplicación anterior a palabras sobre V_1 estableciendo la función siguiente:

$$s': {V_1}^* \to \mathcal{P}({V_2}^*),$$

donde

$$s'(\varepsilon) = \{\varepsilon\},$$

$$s'(uv) = s'(u)s'(v).$$

Es a esta aplicación extendida a la que se le denomina *sustitución*. Nótese que la definición anterior implica que:

$$s'(a) = s(a).$$

Esta es la razón por la que la definición inicial de s se establece en términos de palabras unitarias: en una sustitución está involucrada una concatenación de palabras.

Dicho de manera muy simple, una sustitución es una operación para lenguajes en las que las palabras de un lenguaje sobre un alfabeto V_1 con sustituidas por lenguajes sobre otro alfabeto V_2. Con esto, la operación de concatenación de palabras sobre el alfabeto V_1 se transforma en la concatenación de lenguajes sobre el alfabeto V_2.

A manera de ejemplo sencillo de lo anterior considérense los alfabetos $V_1 = \{a, b, c\}$ y $V_2 = \{0, 1\}$, y la sustitución s dada por:

$$s(a) = \{01, 001\}, s(b) = \left\{1^i \mid i > 0\right\}, s(c) = \{\varepsilon\}.$$

Con esto, la palabra $baca$ sobre V_1 se transforma en:

$$
\begin{aligned}
s'(baca) &= s'(bac)s(a) \\
&= s'(bac)\cdot\{01,001\} \\
&= s'(ba)s(c)\cdot\{01,001\} \\
&= s'(ba)\cdot\{\varepsilon\}\cdot\{01,001\} \\
&= s'(b)s(a)\cdot\{\varepsilon\}\cdot\{01,001\} \\
&= s'(b)\cdot\{01,001\}\cdot\{\varepsilon\}\cdot\{01,001\} \\
&= s'(\varepsilon)s(b)\cdot\{01,001\}\cdot\{\varepsilon\}\cdot\{01,001\} \\
&= s'(\varepsilon)\cdot\{1^i \mid i>0\}\cdot\{01,001\}\cdot\{\varepsilon\}\cdot\{01,001\} \\
&= \{\varepsilon\}\cdot\{1^i \mid i>0\}\cdot\{01,001\}\cdot\{\varepsilon\}\cdot\{01,001\}.
\end{aligned}
$$

Así, al final se encuentra que $s'(baca) = \{1^i0101, 1^i01001, 1^i00101, 1^i001001 \mid i>0\}$. En efecto, una palabra se sustituye por todo un lenguaje.

Se puede considerar una extensión adicional de la aplicación anterior al considerar ahora a todas las palabras de un lenguaje L; en este caso se dice que se realiza una sustitución en el lenguaje. Tal sustitución en el lenguaje L queda definida como sigue:

$S(L) = \bigcup_{\omega\in L} s'(\omega)$.

Sin considerar los detalles, a continuación se presenta una muy breve lista con algunas propiedades importantes que satisfacen las sustituciones.

Propiedades

$S(L_1 \cup L_2) = S(L_1) \cup S(L_2)$.

$S(L_1L_2) = S(L_1)S(L_2)$.

$S(L^*) = \big(S(L)\big)^*$.

4.4.4. Homomorfismo

Como se acaba de señalar, en una sustitución se hace corresponder toda palabra sobre un alfabeto con un lenguaje sobre otro alfabeto. En un homomorfismo, en cambio, a cada palabra unitaria (o símbolo) sobre un alfabeto se le hace corresponder una palabra sobre otro alfabeto; en su versión extendida, un homomorfismo hace corresponder una palabra sobre un alfabeto con otra palabra sobre otro alfabeto.

La definición de un *homomorfismo* se establece como sigue. Dados dos alfabetos V_1 y V_2, un homomorfismo es una función (una aplicación) $h: {V_1}^1 \to {V_2}^*$. Una extensión de esta aplicación a palabras corresponde a la función $h': {V_1}^* \to {V_2}^*$ definida como sigue:

$h'(\varepsilon) = \varepsilon$;

$h'(uv) = h'(u)h'(v)$.

Un homomorfismo se denomina *alfabético* cuando para toda palabra unitaria (o símbolo) a de V_1 es asociada con una palabra de longitud no mayor a uno sobre V_2. Esto es, un homomorfismo alfabético es una aplicación h' tal que $h'(a) \in {V_2}^{\leq 1}$. Si sucede que la palabra asociada sobre V_2 también es unitaria, esto es $h'(a) \in {V_2}^1$, entonces se dice que el homomorfismo es *estrictamente alfabético*.

De la misma manera que se define la sustitución para lenguajes, también se puede definir un homomorfismo para lenguajes procediendo como sigue:

$h(L) = \{h(\omega) \mid \omega \in L\}$.

En este caso, L es un lenguaje sobre el alfabeto V_1, mientras que $h(L)$ es un lenguaje sobre el alfabeto V_2.

Los homomorfismos son usados en la demostración de muchos teoremas importantes de la teoría de los lenguajes formales y de la teoría de los autómatas.

Homomorfismo inverso

Dado un homomorfismo $h: {V_1}^* \to {V_2}^*$, se define su *homomorfismo inverso* de la manera siguiente:

$h^{-1}(v) = \{u \in {V_1}^* \mid h(u) = v\}$.

Nótese que v es una palabra sobre V_2, mientras que $h^{-1}(v)$ es un lenguaje sobre V_1. Esto es, el homomorfismo inverso es una aplicación $h^{-1}: {V_2}^* \to \mathcal{P}({V_1}^*)$.

Una definición homóloga para lenguajes es la siguiente:

$h^{-1}(L) = \{u \in {V_1}^* \mid h(u) \in L\}$.

Nótese que L es un lenguaje sobre V_2. La extensión establece una relación entre lenguajes sobre cada alfabeto.

Los homomorfismos inversos también son una herramienta útil en la demostración de muchos teoremas de la teoría de los lenguajes formales y de la teoría de los autómatas.

5. SEMIANILLOS

Muchos de los resultados (teoremas) más importantes que se presentan en las teorías de los lenguajes formales y autómatas se basan en la construcción de un cierto objeto matemático con propiedades específicas. Muchas veces, sin embargo, se encuentra

que la construcción es muy satisfactoria desde un punto de vista intuitivo (y sobre todo, desde un punto de vista ingenieril), pero la demostración de la validez y eficacia de la construcción puede no ser satisfactoria desde un punto de vista matemático. El enfoque algebraico a la teoría de los lenguajes formales y de los autómatas trata de corregir tal situación. En especial, los métodos algebraicos permiten separar claramente los aspectos intuitivos de los aspectos rigurosos de una demostración. Además, proporcionan demostraciones que no sólo son más compactas, sino que destacan más claramente los aspectos computacionales que están involucrados en el teorema que se está demostrando. Se podría pensar de primera instancia que tal enfoque riguroso es poco valioso para los ingenieros en computación, pero esto es una falsa percepción de la profesión. Entre las tareas más importantes que realiza un ingeniero en computación está analizar la eficacia de la operación de sistemas procesadores de datos, y los métodos de análisis más importantes con los que cuenta se basan en los resultados de la teoría de los lenguajes formales y autómatas. Así pues, métodos rigurosos de pruebas matemáticas, proporcionan métodos rigurosos y confiables de análisis en la práctica.

El enfoque algebraico se basa fundamentalmente en las estructuras algebraicas denominadas semigrupos (y monoides) y semianillos; adicionalmente se hace uso de otras estructuras como los grupos, los anillos, las series de potencias formales y las matrices. Estos temas, en realidad, forman parte de un curso intermedio o avanzado de lenguajes formales y autómatas por la sencilla razón de que se requiere del conocimiento previo de la teoría para poder establecer todo el potencial de los nuevos métodos. Sin embargo, no está por demás que en un curso introductorio se mencionen las bases de tales métodos. Esto es lo que se hace a continuación.

Un *monoide* es un sistema algebraico que consiste de un conjunto M, una operación binaria asociativa $\circ$ sobre M y un elemento neutro (la identidad) 1 tal que $1 \circ a = a \circ 1 = a$ para todo $a \in M$. Un monoide es un semigrupo con un elemento neutro. Un monoide se denomina *conmutativo* si y sólo $a \circ b = b \circ a$ para todo $a, b \in M$. La operación binaria usualmente se denota por yuxtaposición, simplemente ab, y suele denominarse *producto*.

Si en un contexto dado, la operación binaria y el elemento neutro de un monoide M se sobreentienden, entonces simplemente se le refiere por el nombre M; de no ser el caso, se usa la notación completa $(M,\circ,1)$.

El tipo más importante de monoide para nuestras consideraciones es el *monoide libre* V^* generado por un conjunto no vacío V. Este monoide tiene como elementos a todas las *cadenas finitas*, también referidas como *palabras*, $x_1 \ldots x_n$, con $x_i \in V$, y el producto $w_1 \cdot w_2$ se forma al escribir la cadena w_2 inmediatamente después de la cadena w_1. El elemento neutro de V^*, también denominado la *palabra vacía*, se denota ε. Dicho sea de paso, el semigrupo libre sobre V es el subsemigrupo de V^* que contiene todos los elementos a excepción de la palabra vacía; usualmente se le denota V^+.

Los miembros de V son denominados *símbolos* o *letras*; el conjunto V mismo se denomina un *alfabeto*. En el caso de un conjunto V finito, los subconjuntos de V^* se denominan *lenguajes* (*formales*) sobre V.

Un *semianillo* es un conjunto A junto con dos operaciones binarias $+$ y $\cdot$ y dos elementos constantes 0 y 1 tales que:

1) $(A,+,0)$ es un monoide conmutativo;
2) $(A,\cdot,1)$ es un monoide;
3) se satisfaces las leyes distributivas $a \cdot (b+c) = a \cdot b + a \cdot c$ y $(a+b) \cdot c = a \cdot c + b \cdot c$ para todos los elementos a, b, c; y
4) $a \cdot 0 = 0 \cdot a = 0$ para todo elemento a.

La operación $+$ suele denominarse suma, mientras que $\cdot$ se denomina producto; estos nombres no necesariamente se refieren a las conocidas operaciones aritméticas para números. Usualmente el símbolo $\cdot$ se omite, escribiendo ab en lugar de $a \cdot b$. También suele emplearse un orden de operaciones, en el que $\cdot$ se aplica antes que $+$; así, $a + bc$ se debe interpretar como $a + (bc)$.

Si las dos operaciones binarias y los dos elementos constantes de un semianillo A se sobreentienden, entonces simplemente se le refiere con el nombre A. De no ser el caso, entonces es necesario emplear la notación completa $(A,+,\cdot,0,1)$.

Comparado con un anillo, en un semianillo se omite el requerimiento de inversos para la suma; dicho de otra manera, en un semianillo sólo se requiere de un monoide conmutativo y no de un grupo conmutativo como en el caso del anillo. Dicho de paso, el

requerimiento de inversos aditivos en un anillo implica la existencia de un cero multiplicativo, el cual tiene que ser especificado explícitamente. Todo esto quiere decir que un semianillo, como su nombre lo implica, es un sistema algebraico más simple que un anillo; más simple sí, pero no menos interesante o trivial. Y sólo para completar, si la operación de multiplicación del semianillo también es conmutativa, entonces todo el semianillo se denomina *conmutativo*.

Mucha de la teoría de anillos continúa teniendo sentido cuando se aplica a semianillos generales. Un ejemplo típico de esto lo constituye el semianillo de los números naturales (o enteros positivos) $\mathbb{N}$. Algo similar ocurre con los enteros, racionales, reales y complejos, todos ellos con sus operaciones aritméticas convencionales. Un semianillo que es interesante en las ciencias de la computación es el semianillo Booleano $\mathbb{B} = \{0, 1\}$ en el que se cumple $1 + 1 = 1 \cdot 1 = 1$. Pero sin duda el semianillo que más nos interesa es el siguiente. Si V es un alfabeto y si para todo $L_1, L_2 \in V^*$ se define el producto:

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ y } w_2 \in L_2\},$$

entonces $(\mathcal{P}(V^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ es un semianillo sobre $\mathcal{P}(V^*)$, el conjunto potencia de V^*. Si V es un alfabeto finito, entonces $\mathcal{P}(V^*)$ se denomina el *semianillo de los lenguajes formales sobre* V.

En la teoría de los semianillos se establecen las coincidencias y diferencias con la teoría de los anillos, y se estudian aquellos semianillos particulares que presentan propiedades distinguibles. Tal es el caso del semianillo de los lenguajes formales tal como se acaba de definir, o considerando otras operaciones de conjuntos además de la unión o variantes del producto de lenguajes. Quedan pues estos temas para un curso intermedio o avanzado de lenguajes formales.

Referencias Bibliográficas

Daniel I. A. Cohen. *Introduction to Computer Theory.* John Wiley & Sons, Inc. New York 1986.

Peter A. Fejer and Dan A. Simovici. *Mathematical Foundations of Computer Science.* Volume 1: Sets, Relations, and Induction. Texts and Monographs in Computer Science, Volume X. Springer-Verlag, New York 1990.

John E. Hopcroft, Rajeev Motwani y Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*, tercera edición. Pearson Addison-Wesley, Boston 2008.

Dean Kelley. *Teoría de Autómatas y Lenguajes Formales.* Prentice Hall, Madrid 1995.

Werner Kuich. *Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata.* Chapter 9 in Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Grzegorz Rozenberg and Arto Salomaa, eds. Springer-Verlag 1997.

Mark V. Lawson. *Finite Automata.* Chapman & Hall/CRC Boca Raton Florida 2003.

Alexandru Mateescu and Arto Salomaa. *Formal Languages: an Introduction and a Synopsis.* Chapter 1 in Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Grzegorz Rozenberg and Arto Salomaa, eds. Springer-Verlag 1997.

Sheng Yu. *Regular Languages.* Chapter 2 in Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Grzegorz Rozenberg and Arto Salomaa, eds. Springer-Verlag 1997.