GRAMÁTICAS FORMALES Y JERARQUÍA DE CHOMSKY

Un LENGUAJE FORMAL, sea finito o infinito, es un conjunto de palabras con connotación sintáctica.
Existen ciertas estructuras que tienen la habilidad de generar las palabras que forman un LF. Estas estructuras se denominan GRAMÁTICAS FORMALES (GFs).

2.1 GRAMÁTICA FORMAL (GF)

Una GF es, básicamente, un conjunto de PRODUCCIONES, es decir: reglas de re-escritura que se aplican para obtener cada una de las palabras del LF que la GF en cuestión genera.

Ejemplo 1
Sea el lenguaje L = {**a**}, formado por una sola palabra. Este lenguaje es generado por una gramática con una única producción: S −> a (se lee "S produce a").

Ejemplo 2
Sea el lenguaje L = {**a**, **b**}, formado por dos palabras. Este lenguaje es generado por la gramática con las siguientes producciones: S −> a y S −> b (cada producción genera una palabra del lenguaje).

Si bien en estos dos primeros ejemplos se observa que cada *producción* genera una palabra del lenguaje, esto no siempre es así.

Ejemplo 3
El lenguaje L = {**aa**, **ab**} puede ser generado por la GF con producciones S −> aa y S −> ab. Pero también puede ser generado por la GF con producciones S −> aT, T −> a y T −> b.
En el segundo caso, comenzamos aplicando la producción S −> aT para obtener el carácter a con el que comienzan ambas palabras de este LF; luego, si aplicamos la producción T −> a obtenemos la palabra **aa** y si aplicamos la producción T −> b obtenemos la palabra **ab**.

➔ Toda producción está formada por tres partes: el lado izquierdo, el lado derecho, y la *flecha,* que indica que el lado izquierdo de la producción "produce" (o es reemplazado por o equivale a) el lado derecho.

Ejemplo 4
En la producción S −> aT, el *lado izquierdo* está constituido por el símbolo S, mientras que su *lado derecho* está formado por la concatenación del carácter a con el símbolo T. La flecha indica que el símbolo S es reemplazado por la secuencia aT.

2.1.1 DEFINICIÓN FORMAL DE UNA GF

Toda GF es una 4-upla (V_N, V_T, P, S), donde:
- V_N es el vocabulario de no-terminales o *variables* (un conjunto finito),
- V_T es el vocabulario de terminales o caracteres del alfabeto sobre el cual se construye el LF que es generado por la gramática descripta (otro conjunto finito),
- P es el conjunto finito de producciones, y
- $S \in V_N$ es un no-terminal especial, llamado *símbolo inicial* o axioma, desde el cual siempre debe comenzar a aplicarse las producciones que generan todas las palabras de un determinado LF.

Ejemplo 5
Retomamos el Ejemplo 3, que describe las producciones de una gramática que genera el lenguaje {**aa**, **ab**}. La descripción formal de esta gramática es la 4-upla: G = ({S, T}, {a, b}, {S −> aT, T −> a, T −> b}, S}.
Dado que la variable S es el axioma, la generación de cualquier palabra comienza con una producción que tenga a S en su lado izquierdo; en este caso, hay una única producción con esta característica: S −> aT.
Esta única producción indica que el símbolo inicial S es reemplazado, obligatoriamente, por la secuencia aT, por lo que toda palabra generada por esta gramática debe comenzar con el carácter a.
Todavía no se ha obtenido una palabra del lenguaje, porque T es un no-terminal y sabemos que una palabra debe estar formada solo por caracteres o debe ser la palabra vacía. En consecuencia, debe haber una o más producciones que tengan a la variable T en su lado izquierdo, y se debe reemplazar a T por su lado derecho.
En este caso, la variable T tiene dos producciones, las que representan dos opciones: la producción T −> a significa que la variable T es reemplazada por el carácter a, mientras que la producción T −> b representa que el no-terminal T debe ser reemplazado por el carácter b.
Estas dos producciones para la variable T generan dos procesos diferentes: (1) si S −> aT y T −> a, se obtiene la palabra **aa**; (2) si S −> aT y T −> b, se obtiene la palabra **ab**.

Nota 1: Al describir GFs para aplicaciones teóricas, es muy común utilizar las siguientes convenciones:
- denominamos S al axioma,
- todo otro no-terminal es representado mediante una letra mayúscula,
- el vocabulario de terminales está formado por letras minúsculas o dígitos.

2.2 LA JERARQUÍA DE CHOMSKY

En 1956 y 1959, el lingüista norteamericano Noam Chomsky publicó dos trabajos sobre los Lenguajes Naturales que, aplicados al área de los Lenguajes Formales, produjeron lo que se conoce como Jerarquía de Chomsky.

Esta Jerarquía de Chomsky establece una clasificación de cuatro tipos de GFs que, a su vez, generan cuatro tipos diferentes de LFs.

Las GFs se clasifican según las restricciones que se imponen a sus producciones, y la Jerarquía de Chomsky establece estos cuatro niveles:
- Gramáticas Regulares o Gramáticas Tipo 3
- Gramáticas Independientes del Contexto o Gramáticas Tipo 2
- Gramáticas Sensibles al Contexto o Gramáticas Tipo 1
- Gramáticas Irrestrictas o Gramáticas Tipo 0

2.2.1 GRAMÁTICA REGULAR (GR)

Sus producciones tienen las siguientes restricciones:
- el lado izquierdo debe tener un solo no-terminal,
- el lado derecho debe estar formado por un solo terminal o un terminal seguido por un no-terminal.

Algunos autores (no todos) incluyen la posibilidad de que una GR pueda contener "producciones-épsilon", es decir: producciones cuyo lado derecho es ε.

Ejemplo 6

Sea la gramática G = ({S, X}, {a, b}, {S –> aX, X –> b}, S}.
Las producciones de esta gramática cumplen con las restricciones mencionadas arriba. Por lo tanto, esta GF es una Gramática Regular.

También es válida una GR en la que se invierte el orden en el lado derecho de aquellas producciones que tienen dos símbolos. Por lo tanto, una segunda definición para las GRs sería:

Una GF es Regular si sus producciones tienen las siguientes restricciones:
- el lado izquierdo debe tener un solo no-terminal,
- el lado derecho debe estar formado por un solo terminal, o un no-terminal seguido de un terminal.

Ejemplo 7
La GF ({S, X}, {a, b}, {S –> Xa, X –> b}, S} es Regular porque cumple con las restricciones de esta segunda definición.

En general: sean v y v' no-terminales y sea t un terminal. Entonces las producciones de una GR pueden tener estos formatos:

v –> t, v –> tv' y v –> ε o **v –> t, v –> v't y v –> ε**

Sin embargo, debemos tener cuidado con "la mezcla" de ambas definiciones porque la GF resultante no será una Gramática Regular.

Ejemplo 8
Sea G = ({S, X}, {a, b}, {S –> Xa, S –> bX, X –> b}, S}.
Esta GF no es Regular porque el no-terminal S produce "no-terminal seguido de terminal" y también produce "terminal seguido de no-terminal".

2.2.2 GRAMÁTICA INDEPENDIENTE DEL CONTEXTO (GIC)

A diferencia de las GRs, estas gramáticas no tienen restricciones con respecto a la forma del lado derecho de sus producciones, aunque sí se requiere que el lado izquierdo de cada producción siga siendo un único no-terminal.

Ejemplo 9
Supongamos que a la GR del Ejemplo 7 le agregamos la producción S –> ba.
Esta nueva producción posee dos terminales en su lado derecho y, como ya hemos visto, este no es un formato válido para una GR. La nueva GF es: G = ({S, X}, {a, b}, {S –> Xa, S –> ba, S –> bX, X –> b}, S}.
Esta gramática es una GIC.

En general: sean v y v' no-terminales y sea t un terminal. Entonces las producciones de una GIC corresponden a este formato general:

v –> (v' + t)*

donde la expresión (v' + t)* representa ε y cualquier secuencia de variables y/o terminales.

Ejemplo 10
Teniendo en cuenta la convención mencionada antes (letra mayúscula para no-terminales, y letras minúsculas o dígitos para terminales), he aquí una serie de producciones correctas para las GICs:
A –> ε

A −> a
B −> aa
B −> ZZ
B −> aZbbZa

Por las restricciones establecidas sobre ambos tipos de GFs, es fácil notar que toda GR también es una GIC; la inversa no es correcta.

La frase "independiente del contexto" refleja que, como el lado izquierdo de cada producción únicamente puede contener un solo no-terminal, la producción puede aplicarse sin importar el contexto donde se encuentre dicho no-terminal.

2.2.3 GRAMÁTICA IRRESTRICTA (GI)

Son las GFs más amplias. Sus producciones tienen la forma general:

$$\alpha \rightarrow \beta,$$

donde α y β pueden ser secuencias de no-terminales y/o terminales, con $\alpha \neq \varepsilon$.

Ejemplo 11 [de Hopcroft & Ullman, 1979]
La que sigue es una GI que genera el lenguaje $\{a^i / i \text{ es una potencia positiva de } 2\}$:
G = ({a}, {S, A, B, C, D, E}, {S −> ACaB, Ca −> aaC, CB −> DB, CB −> E, aD −> Da, AD −> AC,
aE −> Ea, AE −> ε}, S)

2.2.4 GRAMÁTICA SENSIBLE AL CONTEXTO (GSC)

Es una GI con la siguiente restricción en las longitudes: $|\beta| \geq |\alpha|$.

Ejemplo 12
La GF del Ejemplo anterior no es una GSC porque tiene dos producciones que violan la restricción impuesta sobre las longitudes: CB −> E y AE −> ε.

***

Cada tipo de GF genera un tipo de LF cuyo nombre deriva del nombre general de la correspondiente gramática.
Así, las GRs generan LENGUAJES REGULARES, las GICs generan LENGUAJES INDEPENDIENTES DEL CONTEXTO, las GSCs generan Lenguajes Sensibles al Contexto y las GIs generan Lenguajes Irrestrictos.

2.3 GRAMÁTICAS QUE GENERAN LENGUAJES FORMALES INFINITOS

El diseño de una GF que genere un LF infinito requiere que, al menos una de sus producciones sea recursiva, esto es: la variable que aparece en su lado izquierdo también debe encontrarse en su lado derecho.

Ejemplo 13
La GI del Ejemplo 11 genera un lenguaje infinito ya que una de sus producciones es recursiva: Ca −> aaC.

Ejemplo 14
La GR con producciones S −> aS y S −> a genera el LR infinito $\{a^n / n \geq 1\}$.

2.4 LA DERIVACIÓN

La derivación es el proceso que permite obtener cada una de las palabras de un LF a partir del axioma de una GF que lo genera, aplicando sucesivamente producciones convenientes de esa GF. En cuanto a la simbología utilizada: así como usamos la **->** en las producciones, utilizaremos **=>** en cada paso de una derivación.

Ejemplo 15
Sea la GIC con producciones S −> aSb y S −> ab. Esta GIC genera el lenguaje $\{a^n b^n / n \geq 1\}$. Una de las palabras de este lenguaje es **aaabbb**. Verifiquemos por derivación:

```
S => aSb => aaSbb => aaabbb.
```

En cambio, la cadena aaabb no es una palabra de este lenguaje y, por lo tanto, no la podremos derivar. Comprobemos esta situación:

```
S => aSb => aaSbb => ??
```

Nos falta obtener una sola a; sin embargo, no podemos agregarla a la cadena sin añadir también una b. Por lo tanto, no podemos derivar aaabb y, en consecuencia, ésta no es una palabra del lenguaje generado por la GIC dada.

2.5 RESUMEN

GRAMÁTICA FORMAL (GF)
DEFINICIÓN FORMAL DE UNA GF
NO-TERMINALES
TERMINALES
AXIOMA
PRODUCCIONES
JERARQUÍA DE CHOMSKY
GRAMÁTICA REGULAR
GRAMÁTICA INDEPENDIENTE DEL CONTEXTO
PRODUCCIONES RECURSIVAS
DERIVACIÓN

2.6 EJERCICIOS

(1) Sea el LF L = {**aa**, **ab**, **aba**}. Describa la Definición Formal de dos GFs que generen este LF: una debe ser una

GR y la otra una GIC (no GR).

(2) Sea el LF infinito $L = \{a^n bc^n / n \geq 1\}$. Describa la Definición Formal de una GIC que genere este LF.

(3) Dada la GIC construida en el punto anterior, utilice DERIVACIÓN para determinar si las siguierntes cadenas

son o no palabras del LF generado:

a) aaabccc
b) aabbcc
c) aaabcc
d) aabccc
e) aaaccc

Ejercicios de C

Dsearrolle la función de prototipo void token(char *s) que reciba un literal cadena de al menos 40 caracteres que contenga palabras, separadas por un espacio en blanco y muestre por el flujo estándar de salida cada una de las palabras que componen la cadena, una por linea
Ejemplo Si el literal dato es “sintaxis y semántica de los lenguajes” muestre:

sintaxis
y
semántica
de
los
lenguajes