Gramáticas

Horst H. von Brand
vonbrand@inf.utfsm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Contenido

Otra manera de representar lenguajes

Gramáticas y sus lenguajes

La jerarquía de Chomsky

Gramáticas regulares

Gramáticas de contexto libre

Resumen

Motivación

Con todo lo flexible y útiles que son los lenguajes regulares, quedan cortos para tareas de interés. Al ser no-regular el simple lenguaje $\{a^n b^n : n \geq 0\}$, representar por ejemplo expresiones aritméticas (mi profesor de matemáticas de media siempre nos advertía «todo lo que se abre se cierra», cada abre paréntesis debe ir con su respectivo cierre paréntesis, y claramente no hay límite al número de paréntesis) o las estructuras arbitrariamente anidadas de un lenguaje de programación como C o Python. Requerimos un formalismo más poderoso.

Gramáticas

La idea de *gramáticas* proviene de las gramáticas usadas para describir lenguajes naturales, como el castellano. Se tomaron ideas de allí para describir lenguajes artificiales, como Algol 60, lenguaje de programación que influyó decisivamente en los lenguajes en uso actual.

Gramáticas

Definición

Una *gramática* $G = (N, \Sigma, P, S)$ consta de:

- N: El alfabeto de *no-terminales*.
- Σ: El alfabeto de *terminales*. Exigimos que $N \cap \Sigma = \varnothing$.
- P: Conjunto de *producciones*. Una producción tiene la forma $\alpha \to \beta$, donde $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$ y $\beta \in (N \cup \Sigma)^*$.
- S: El *símbolo de partida*, $S \in N$.

Comúnmente se denota $V = N \cup \Sigma$, el *vocabulario* de la gramática. A palabras en V^* se les llama *formas sentenciales* de G.

Gramáticas

En castellano: el lado izquierdo de la producción son terminales y no-terminales, debe contener al menos un no-terminal.
El lado derecho son terminales y no-terminales, sin restricciones.

Relación de derivación de G

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática, $V = N \cup \Sigma$. La *relación de derivación de* G entre formas sentenciales de G se define mediante:

$$u \Rightarrow_G v \iff u = x\alpha y \wedge v = x\beta y \wedge (\alpha \to \beta) \in P$$

En castellano: reemplaza el lado izquierdo de una producción por el lado derecho respectivo. Pueden haber varias producciones aplicables.
Normalmente omitiremos el subíndice G, la gramática se subentiende.

El lenguaje generado por G

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática. El *lenguaje generado por* G se define como:

$$\mathscr{L}(G) = \{\alpha \in \Sigma^* : S \Rightarrow_G^* \alpha\}$$

En castellano: son las palabras formadas solo por terminales que se derivan en cero o más pasos del símbolo de partida S.

¿Vale la pena este lío?

Demostraremos que la gramática $G = (\{S\}, \{a, b\}, P, S)$ con producciones:

$$S \to aSb$$
$$S \to \varepsilon$$

genera nuestro lenguaje no-regular símbolo $\{a^n b^n : n \geq 0\}$.

¿Vale la pena este lío?

Primeramente, $\varepsilon \in \mathscr{L}(G)$, ya que $S \to \varepsilon$ es una producción.

Enseguida, demostramos por inducción que $S \Rightarrow^* a^n S b^n$ para todo $n \geq 0$:

Base: Para $n = 0$, $S \Rightarrow^* S$ es evidente.

Inducción: Suponiendo que vale para $n = k$, usando solo la primera producción tenemos:

$$S \Rightarrow^* a^k S b^k \Rightarrow a^{k+1} S b^{k+1}$$

Finalmente, usando la segunda producción:

$$S \Rightarrow^* a^n S b^n \Rightarrow a^n b^n$$

Es claro que no hay otras formas sentenciales de G que se derivan de S, y el lenguaje generado es el que se indica.

Algunas convenciones

Para abreviar descripción de gramáticas, adoptamos:

- Usamos letras mayúsculas $(A, B, C, \ldots)$ como no-terminales.
- Letras minúsculas $(a, b, c, \ldots)$ y otros símbolos $(+, -, *, /, (,), \ldots)$ son terminales. Excluimos $\rightarrow$ y $|$, que tienen otros usos.

Algunas convenciones

- Abreviamos producciones con el mismo lado izquierdo:

$$\alpha \to \beta_1$$
$$\alpha \to \beta_2$$
$$\vdots$$
$$\alpha \to \beta_n$$

como:

$$\alpha \to \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$$

Algunas convenciones

Con las anteriores, no hace falta indicar explícitamente los conjuntos de terminales y no-terminales. (Sí, tener terminales o no-terminales que no aparecen en ninguna producción es perfectamente legal. Pero tiene poco sentido práctico.)

Adoptamos la convención que el no-terminal al lado izquierdo de la primera producción es el símbolo de partida.

Gramática regalona

Adoptaremos como mascota la gramática siguiente:

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to a \mid (E)$$

Representa expresiones aritméticas con átomos a, operaciones + y $*$, y paréntesis. Cumple lo que exigía el profesor: cada '(' se balancea con un ')'. El lenguaje que genera no es regular.

Otro ejemplo

Otro ejemplo de gramática es:

$$S \to (S)S \mid \varepsilon$$

Esta gramática genera las palabras formadas por paréntesis debidamente anidados.
La demostración es similar a la usada para derivar la función generatriz de números de Catalan.

Otro ejemplo

Primeramente, ε es una secuencia de paréntesis balanceados, y $\varepsilon \in \mathscr{L}(G)$ por la producción $S \to \varepsilon$.
Enseguida, por inducción si S deriva una secuencia de paréntesis balanceados, la producción $S \to (S)S$ dará una secuencia de paréntesis balanceados.
Finalmente, una secuencia de paréntesis balanceados comienza con '(', ese debe estar balanceado con un ')', después de una secuencia balanceada (resultado de un S); luego del ')' que cierra el primer '(' viene nuevamente una secuencia balanceada (otro S).

Las consideraciones anteriores, bastante informales, se pueden formalizar en una demostración por inducción. Gramáticas, inducción y recursividad van de la mano.

La jerarquía de Chomsky

La siguiente clasificación de gramáticas parece arbitraria, más adelante veremos justificaciones para algunas de sus distinciones.
Note que cada nivel de la jerarquía introduce restricciones adicionales, las gramáticas forman una cadena de inclusiones propias. Lo mismo ocurre con los lenguajes generados. Algunas de las diferencias se demostrarán luego.
Los nombres de los tipos de gramáticas los justificaremos luego.

La jerarquía de Chomsky

Tipo 0: *Irrestrictas*. Como las definimos.

Tipo 1: *Sensibles al contexto*. Todas las producciones cumplen:

$$\alpha \to \beta \qquad |\alpha| \leq |\beta|$$

Tipo 2: *De contexto libre*. Todas las producciones cumplen:

$$A \to \alpha \qquad A \in N, \alpha \in (N \cup \Sigma)^+$$

Tipo 3: *Regulares*. Todas las producciones son de una de las formas:

$$A \to \alpha B \qquad A, B \in N, \alpha \in \Sigma^*$$

$$A \to \beta \qquad A \in N, \beta \in \Sigma^+$$

Justificación de los nombres

Las gramáticas de tipo 3 (regulares) generan (casi) los lenguajes regulares. De allí el nombre.
Las gramáticas de tipo 2 (de contexto libre) se llaman así porque sus producciones reemplazan el lado izquierdo de $A \to \beta$ por el derecho, sin importar dónde aparece.
Las gramáticas de tipo 1 (sensibles al contexto) pueden expresarse mediante producciones de la forma:

$$\alpha A \beta \to \alpha \gamma \beta \qquad A \in N, \alpha, \beta \in V^*, \gamma \in V^+$$

Es decir, reemplaza A por γ en el contexto dado por α, β.

Las gramáticas regulares generan lenguajes regulares

Teorema

Las gramáticas regulares generan los lenguajes regulares que no contienen ε.

Las gramáticas regulares generan lenguajes regulares

Demostración

Es claro que ninguna gramática regular puede generar ε. Demostramos implicancia en ambas direcciones.

Si L es regular y $\varepsilon \notin L$, es $L = \mathscr{L}(M)$ para un DFA $M = (Q, \Sigma, \delta, q_0, F)$, donde q_0 no es final. Sin pérdida de generalidad, podemos suponer $Q \cap \Sigma = \varnothing$. La gramática $G = (Q, \Sigma, P, q_0)$ con producciones:

$$q \to ap \qquad \delta(q,a) = p$$
$$q \to a \qquad \delta(q,a) \in F$$

genera $\mathscr{L}(M)$.

Las gramáticas regulares generan lenguajes regulares

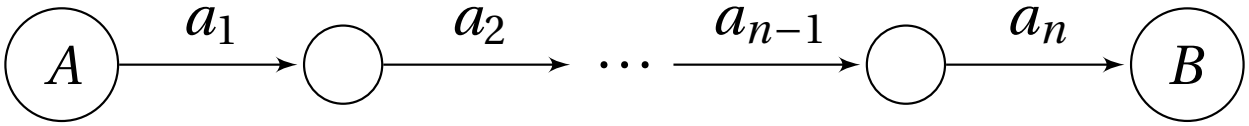
Demostración

Supongamos ahora que $L = \mathscr{L}(G)$ para una gramática regular G. Construiremos un NFA $M = (Q, \Sigma, \delta, S, \{F\})$, donde parte de los estados son símbolos no-terminales, tal que $L = \mathscr{L}(M)$. Para poder usar el nombre F para el estado final exigiremos que $F \notin N$.

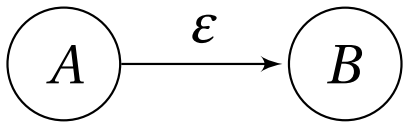
Las gramáticas regulares generan lenguajes regulares

Demostración

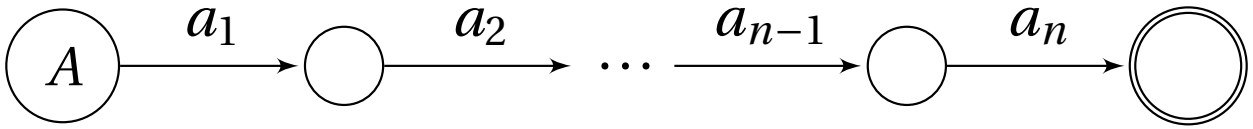
Para la producción $A \to a_1 a_2 \ldots a_n B$ construimos el retazo:



Para la producción $A \to B$ basta con:



Para $A \to a_1 a_2 \ldots a_n$ construimos:



Las gramáticas regulares generan lenguajes regulares

Demostración

Finalmente, identificamos los estados rotulados con no-terminales, designando como inicial el estado que corresponde al símbolo de partida S.
Vemos que el autómata traza posibles derivaciones en G conforme lee símbolos, es claro que acepta exactamente el lenguaje que la gramática genera. □

Las gramáticas regulares generan lenguajes regulares

La irritante restricción de no generar ε podemos evitarla permitiendo producciones $A \to \varepsilon$, con el ajuste obvio a la construcción del teorema precedente. Informalmente, suelen llamarse *gramáticas regulares* si las producciones son de las formas:

$$A \to \alpha B \qquad A, B \in N, \alpha \in \Sigma^*$$
$$A \to \beta \qquad A \in N, \beta \in \Sigma^*$$

La restricción, que no hace gran diferencia práctica, es para mantener la jerarquía.

Gramática de contexto libre que genera no regular

La gramática:

$$S \to aSb \mid ab$$

es de contexto libre. Como antes, demostramos que genera $L = \{a^n b^n : n \geq 1\}$. Pero $L \cup \{\varepsilon\}$ es nuestro niño símbolo no-regular, con lo que L no es regular.

Con esto demostramos que el primer peldaño de la jerarquía de Chomsky no es arbitrario.

Resumen

- Definimos gramáticas y los lenguajes que generan.
- Introdujimos la *relación de derivación* de la gramática.
- Describimos la jerarquía de Chomsky, definiendo las gramáticas de cada nivel por las formas de sus producciones. Note que Tipo $i \subsetneq$ Tipo j si $i > j$.
- Demostramos que las gramáticas regulares (relajando la forma de las producciones) generan exactamente los lenguajes regulares.
- Demostramos que hay gramáticas de contexto libre que generan lenguajes no regulares.