

MODELOS DE COMPUTACIÓN
GRADO EN INGENIERÍA INFORMÁTICA

Memoria de prácticas

Autor

Vladislav Nikolov Vasilev

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

CURSO 2018-2019

Contents

1	Prácticas	2
1.1	Práctica 1	2
1.1.1	Ejercicio 1	2
1.1.2	Ejercicio 2	3
1.1.3	Ejercicio 3	3
1.1.4	Ejercicio 4	5
1.2	Práctica 2	9
1.3	Práctica 3	10
1.4	Práctica 4	11
2	Ejercicios voluntarios	12

1 Prácticas

1.1 Práctica 1

1.1.1 Ejercicio 1

Enunciado. Calcula una gramática libre de contexto que genere el lenguaje $L = \{a^n b^m c^m d^{2n} \mid n, m \geq 0\}$.

Solución

Se define la gramática como una cuádrupla con la forma $G = (V, T, P, S)$, siendo V el conjunto de variables, T el conjunto de elementos terminales, P las reglas de producción y S el símbolo inicial. Se puede definir cada uno de los conjuntos de la siguiente forma:

$$V = \{S, X, Y\}$$

$$T = \{a, b, c, d\}$$

$$P = \{S \rightarrow aXdd \mid bYc \mid \varepsilon, X \rightarrow aXdd \mid bYc \mid add \mid \varepsilon, Y \rightarrow bYc \mid bc \mid \varepsilon\}$$

$$S = \{S\}$$

Ésta es una gramática de **tipo 2**, ya que a la izquierda aparece una variable sola, sin símbolos terminales, y a la derecha aparece la variable con símbolos terminales tanto por la derecha como por la izquierda, impidiendo por tanto que sea regular por la izquierda o por la derecha.

Con ésta gramática, primero se escoge si se van a empezar a generar una a con la secuencia dd al final, o si directamente se comenzará a generar la secuencia b seguida de c . Si se escoge la primera opción, se ponen tantas a al principio y dd al final como sea necesario, y después se puede escoger si se sigue con las b y c , o si se termina sin poner ninguno de los símbolos anteriores. Si se decide comenzar a poner b y c desde el principio o después de poner todas las a y dd que se quieran, se ponen todas las b y c que se quieran, hasta que se decida terminar la secuencia.

Gracias a las reglas de producción se pueden satisfacer todas las restricciones del lenguaje, ya que por cada a se genera dd , y por cada b se genera c . Además, se puede aceptar la cadena vacía o que alguna de las partes no esté.

1.1.2 Ejercicio 2

Enunciado. Describir una gramática que genere los números decimales escritos con el formato [signo][cifra][punto][cifra]. Por ejemplo, +3.45433, -453.23344, ...

Solución

La solución más sencilla que se puede ofrecer a este problema consiste en utilizar una gramática libre de contexto, como se mostrará a continuación. No obstante, el problema también es resoluble mediante una gramática regular, aumentando sin embargo el número de producciones y de variables necesarias.

Definimos la gramática como una cuádrupla con la forma $G = (V, T, P, S)$, siendo V el conjunto de variables, T el conjunto de elementos terminales, P las reglas de producción y S el símbolo inicial. Se puede definir cada uno de los conjuntos de la siguiente forma:

$$\begin{aligned} V &= \{S, X\} \\ T &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., +, -\} \\ P &= \left\{ \begin{array}{c} S \rightarrow +X.X \mid -X.X \\ X \rightarrow 0X \mid 1X \mid 2X \mid 3X \mid 4X \mid 5X \mid 6X \mid 7X \mid 8X \mid 9X \mid \\ \quad 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{array} \right\} \\ S &= \{S\} \end{aligned}$$

Primero se genera el signo y el punto, pudiendo escoger si el número es positivo o negativo. Después, en la parte entera y decimal se van generando números en el rango $[0, 9]$, pudiendo escoger cuál es el siguiente número o cuando terminar de insertar números.

1.1.3 Ejercicio 3

Enunciado. Calcular una gramática libre de contexto que genere el lenguaje $L = \{0^i 1^j 2^k \mid \text{tal que } i \neq j \text{ o } j \neq k\}$.

Solución

Definimos la gramática como una cuádrupla con la forma $G = (V, T, P, S)$, siendo V el conjunto de variables, T el conjunto de elementos terminales, P las reglas de producción y S el símbolo inicial. Se puede definir cada uno de los conjuntos de la siguiente forma:

$$\begin{aligned}
V &= \{S, X, Y, Z, P, R, A, B, M, K, U, D, N, L, C, Q\} \\
T &= \{0, 1, 2\} \\
P &= \left\{ \begin{array}{l} S \rightarrow 0X1 \mid 0Y2 \mid 1Z2 \mid 0A1P2 \mid 0R1B2, \ X \rightarrow 0X \mid X1 \mid \varepsilon \\ Y \rightarrow 0Y \mid Y2 \mid \varepsilon, \ Z \rightarrow 1Z \mid Z2 \mid \varepsilon, \ P \rightarrow P2 \mid \varepsilon, \ R \rightarrow 0R \mid \varepsilon, \\ A \rightarrow 0M \mid N1, \ M \rightarrow OMU \mid \varepsilon, \ U \rightarrow 1 \mid \varepsilon, \ N \rightarrow CN1 \mid \varepsilon, \ C \rightarrow 0 \mid \varepsilon, \\ B \rightarrow 1K \mid L2, \ K \rightarrow 1KD \mid \varepsilon, \ D \rightarrow 2 \mid \varepsilon, \ L \rightarrow QL2 \mid \varepsilon, \ Q \rightarrow 1 \mid \varepsilon \end{array} \right\} \\
S &= \{S\}
\end{aligned}$$

Ya que hay muchas reglas y puede no llegar a quedar claro para qué es cada una, vamos a ir comentándolas para que no queden dudas sobre el por qué de cada una de ellas.

La primera de ellas, $S \rightarrow 0X1$, indica que solo se van a producir los símbolos 0 y 1, dándose por tanto la condición $j \neq k$, ya que no hay ningún símbolo 2. X puede ser sustituido por tantos 0 o 1 como se desee, lo cuál corresponde a la producción $X \rightarrow 0X \mid X1 \mid \varepsilon$.

Después tenemos la producción $S \rightarrow 0Y2$, la cuál es parecida a la anterior, solo que esta vez se producen solo los símbolos 0 y 2, satisfaciendo por tanto las condiciones $i \neq j$ y $j \neq k$ simultáneamente. La variable Y puede ser sustituida por tantos 0 o 2 como se desee, lo cuál se corresponde con la producción $Y \rightarrow 0Y \mid Y2 \mid \varepsilon$.

La regla $S \rightarrow 1Z2$ permite producir los símbolos 1 y 2. En este caso, esta regla permite satisfacer la restricción $i \neq j$, ya que no se produce ningún símbolo 0. La variable Z puede ser sustituida por tantos 1 y 2 como se desee. Ésto se corresponde con la producción $Z \rightarrow 1Z \mid Z2 \mid \varepsilon$.

La regla $S \rightarrow 0A1P2$ permite producir los símbolos 0, 1 y 2, cumpliendo sin embargo la restricción $i \neq j$, introduciendo la desigualdad por tanto en la parte de los 0 y los 1, es decir, obligando que el número de 0 y de 1 sea diferente y permitiendo producir tantos 2 como se desee. La variable A se puede sustituir con la regla $A \rightarrow 0M \mid N1$, escogiendo si se quieren más 0 que 1 (se escogería OM) o más 1 que 0 (se escogería en este caso $N1$). Al haber escogido estas reglas, se asegura que como mínimo hay un símbolo más de ese tipo. La variable M puede ser sustituida por $M \rightarrow OMU \mid \varepsilon$, permitiendo poner tantos 0 como se deseen y poniendo por cada uno una variable U , la cuál puede ser sustituida luego por $U \rightarrow 1 \mid \varepsilon$, poniendo o no tantos 1 como U haya. Hay que tener en cuenta que el número de 1 será siempre menor que el número de 0, ya que al principio, con $A \rightarrow 0M$ se puso un 0 extra, y como la regla $M \rightarrow OMU$ produce una variable que pueda ser sustituida por 1 por cada 0 nuevo que se coloca, se asegura que se cumplirá la desigualdad $i \neq j$ como se mencionó anteriormente, siendo en este caso

$i > j$ ya que $num(1) \leq num(0) - 1$. Algo similar ocurre con la regla $A \rightarrow N1$, ya que permite producir más 1 que 0 de la misma forma que antes. Primero se introduce un 1 extra y después se sustituye la variable N por $N \rightarrow CN1 \mid \varepsilon$, permitiendo poner tantos 1 como se deseen y permitiendo poner un 0 por cada nuevo 1 que se añade (lo cuál se corresponde a la regla $C \rightarrow 0 \mid \varepsilon$). Aquí ocurre lo mismo que en el caso anterior, ya que cumplirá la restricción $i \neq j$ verificando que $i < j$, debido a que $num(0) \leq num(1) - 1$.

Finalmente tenemos la regla $S \rightarrow 0R1B2$, la cuál es similar a la anterior mencionada debido a que permite producir los símbolos 0, 1 y 2, satisfaciendo sin embargo la desigualdad $j \neq k$, lo cuál significa que se producen tantos 0 como se deseen, pero el número de 1 y de 2 tiene que ser distinto. La variable R puede ser sustituida por $R \rightarrow 0R \mid \varepsilon$, es decir, por tantos 0 como se desee. La variable B puede ser sustituida por $B \rightarrow 1K \mid L2$, permitiendo en el primer caso que haya más 1 que 2, y en el segundo caso que haya más 2 que 1. La variable K puede ser sustituida por $K \rightarrow 1KD \mid \varepsilon$, poniendo un símbolo 1, después otra variable K y finalmente una variable D , la cuál puede ser sustituida mediante la regla $D \rightarrow 2 \mid \varepsilon$, poniendo como mucho tantos 2 como variables D haya. Con esto se cumple la desigualdad $j \neq k$ ya que se ha puesto un 1 extra al principio, de forma que $j > k$ y $num(2) \leq num(1) - 1$. En el caso de querer más 2 que 1, se escogería $B \rightarrow L2$, sustituyendo luego la variable L por $L \rightarrow QL2 \mid \varepsilon$, poniendo una variable Q , una variable L y un 2. La variable Q sería sustituida luego mediante la regla $Q \rightarrow 1 \mid \varepsilon$, permitiendo poner un o ningún 1 por cada variable Q . En este caso se cumpliría que $j \neq k$ ya que $j < k$ porque se cumple que $num(1) \leq num(2) - 1$.

1.1.4 Ejercicio 4

Enunciado. Una empresa de videojuegos “*The fantastic platform*” están planteando diseñar una gramática capaz de generar niveles de un juego de plataformas, cada uno de los niveles siguiendo las siguientes restricciones:

- Hay 2 grupos de enemigos: grupos grandes (g) y grupos pequeños (p).
- Hay 2 tipos de monstruos: fuertes (f) y débiles (d).
- Los grupos grandes de enemigos tienen, al menos, 1 monstruo fuerte y 1 débil. Y los 2 primeros monstruos pueden ir en cualquier orden. A partir del tercer monstruo, irán primero los débiles y después los fuertes.
- Los grupos pequeños tienen como mucho 1 monstruo fuerte.
- Al final de cada nivel habrá una sala de recompensas (x).

Por ejemplo, la cadena terminal “*gfdddddfffpdddfx*” representa que el nivel tiene (*gfdddddfff*) un grupo grande con un monstruo fuerte, 4 débiles y otros 3 fuertes; después tiene (*pddddf*) un grupo pequeño con 3 débiles y uno fuerte.

Elaborar una gramática que genere estos niveles con sus restricciones. Cada palabra del lenguaje es un solo nivel. ¿A qué tipo de gramática dentro de la jerarquía de Chomsky pertenece la gramática diseñada?

¿Sería posible diseñar una gramática de tipo 3 para dicho problema?

Solución

Definimos la gramática como una cuádrupla con la forma $G = (V, T, P, S)$, siendo V el conjunto de variables, T el conjunto de elementos terminales, P las reglas de producción y S el símbolo inicial. Se puede definir cada uno de los conjuntos de la siguiente forma:

$$\begin{aligned}
 V &= \{S, G, X, Y, Z, V, R, P, B\} \\
 T &= \{g, p, f, d, x\} \\
 P &= \left\{ \begin{array}{l} S \rightarrow gGx \mid pPx, G \rightarrow ddX \mid dfY \mid fdY \mid ffZ, X \rightarrow dX \mid fR, \\ Y \rightarrow X \mid V \mid R \mid gG \mid pP \mid \varepsilon, Z \rightarrow dX \mid dV, V \rightarrow dV \mid gG \mid pP \mid \varepsilon, \\ R \rightarrow fR \mid gG \mid pP \mid \varepsilon, P \rightarrow dP \mid dB \mid fB, B \rightarrow dB \mid gG \mid pP \mid \varepsilon \end{array} \right\} \\
 S &= \{S\}
 \end{aligned}$$

La gramática es de **tipo 2**, ya que a la izquierda aparecen solo variables y a la derecha aparecen variables con símbolos terminales tanto por la derecha como por la izquierda o sin símbolos terminales, impidiendo por tanto que sea regular, tanto por la derecha o regular por la izquierda.

Una vez dicho esto, se va a proceder a explicar cómo funcionan las reglas de producción. Con $S \rightarrow gGx \mid pPx$ se escoge con qué grupo empezar primero: si uno grande (gGx) o uno pequeño (pPx).

Si se escoge empezar con un grupo grande, como da igual en qué orden están los dos primeros enemigos, se puede sustituir la variable G mediante la regla $G \rightarrow ddX \mid dfY \mid fdY \mid ffZ$. Si con la regla anterior se han producido dos enemigos débiles, se sustituye la variable X mediante la regla $X \rightarrow dX \mid fR$, que permite poner primero todos los enemigos débiles que se quieran y poner uno fuerte al final. Después de poner el fuerte, la variable R se puede sustituir mediante la regla $R \rightarrow fR \mid gG \mid pP \mid \varepsilon$, que permite poner tantos enemigos fuertes como se quieran, y después un grupo grande, uno pequeño o terminar de poner grupos de enemigos. Si por el contrario, al principio del grupo grande se escogen poner dos enemigos fuertes, entonces se tiene que poner como mínimo un enemigo débil. Esto se hace a través de la variable Z , que se sustituye mediante la regla $Z \rightarrow dX \mid dV$, que pone un enemigo débil y permite escoger si seguir con la variable X para poner enemigos débiles o fuertes, o seguir con la variable V , que se sustituye mediante

la regla $V \rightarrow dV \mid gG \mid pP \mid \varepsilon$ y permite poner tantos enemigos débiles como se quieran o poner un nuevo grupo grande, pequeño o terminar de poner grupos. Si por el contrario se escoge poner un enemigo fuerte y uno débil, como ya se cumple la restricción del grupo grande, se puede sustituir la variable Y mediante la regla $Y \rightarrow X \mid V \mid R \mid gG \mid pP \mid \varepsilon$, que permite poner enemigos fuertes y débiles, solo fuertes, solo débiles, poner un grupo grande, uno pequeño o terminar de insertar grupos.

Si en cambio se escoge empezar con un grupo pequeño, la variable P puede ser sustituida mediante la regla $P \rightarrow dP \mid dB \mid fB$, que permite poner tantos enemigos débiles como se quieran y se puede escoger si se quiere un enemigo fuerte, o si por el contrario solo se van a producir débiles. En todo caso, si se produce un enemigo fuerte o si no se decide producir se escoge el camino de la variable B , la cuál puede ser sustituida con la regla $B \rightarrow dB \mid gG \mid pP \mid \varepsilon$, permitiendo de nuevo poner cuantos enemigos débiles se desee, y después poner un grupo grande, uno pequeño o terminar de insertar grupos de enemigos.

Como se puede comprobar, estas reglas permiten crear niveles de forma flexible, ya que se pueden combinar los grupos grandes y los pequeños en el orden que se quiera. Además, también permite generar los enemigos de una forma versátil, permitiendo muchas combinaciones posibles.

Respecto a la segunda pregunta, es posible diseñar una gramática de tipo 3 para este problema. Esto se debe al hecho de que, aunque la gramática obtenida inicialmente sea de tipo de 2, no se garantiza que el lenguaje sea de tipo 2, si no que también puede ser de tipo 3. Para ello, definimos la gramática como una cuádrupla $G = (V, T, P, S)$, donde V son las variables, T los símbolos terminales, P las reglas de producción y S el símbolo inicial. Cada uno de los conjuntos tendría la siguiente forma:

$$\begin{aligned}
 V &= \{S, G, X, Y, Z, V, R, P, B\} \\
 T &= \{g, p, f, d, x\} \\
 P &= \left\{ \begin{array}{l} S \rightarrow gG \mid pP, G \rightarrow ddX \mid dfY \mid fdY \mid ffZ, X \rightarrow dX \mid fR, \\ Y \rightarrow dX \mid dV \mid fR \mid gG \mid pP \mid x, Z \rightarrow dX \mid dV, V \rightarrow dV \mid gG \mid pP \mid x, \\ R \rightarrow fR \mid gG \mid pP \mid x, P \rightarrow dP \mid dB \mid fB, B \rightarrow dB \mid gG \mid pP \mid x \end{array} \right\} \\
 S &= \{S\}
 \end{aligned}$$

Como se puede comprobar fácilmente, esta gramática es de **tipo 3**, ya que a la izquierda aparece la variable sola y a la derecha aparece, o bien un símbolo terminal, o bien uno o más símbolos terminales acompañados de una variable a la derecha. Por tanto, se trata de una gramática regular por la derecha.

Se puede comprobar fácilmente que esta gramática produce las mismas palabras que la anterior. Las diferencias son que el símbolo de recompensa de nivel x se genera cuando no se quieren generar más grupos de enemigos en vez de al principio como se hacía antes. Esto también implica que todos los ε se han sustituido por el símbolo x . Adicionalmente, para que el lenguaje fuese regular por la derecha, a las producciones de Y que anteriormente solo implicaban un cambio de variable se les ha añadido un símbolo terminal que además cumple las restricciones impuestas por el problema (una d para las variables X y V y una f para la variable R).

1.2 Práctica 2

1.3 Práctica 3

1.4 Práctica 4

2 Ejercicios voluntarios