Teoría de la Computación

Clase 10: Gramáticas independientes del contexto

Mauro Artigiani

27 agosto 2021

Universidad del Rosario, Bogotá

Introducción

Repaso

En el curso hemos hablado de una clase de lenguajes: los *lenguajes regulares*. Hemos visto que se pueden describir como los lenguajes reconocidos por autómatas de estados finitos o como expresiones regulares.

Repaso

En el curso hemos hablado de una clase de lenguajes: los *lenguajes regulares*. Hemos visto que se pueden describir como los lenguajes reconocidos por autómatas de estados finitos o como expresiones regulares.

También hemos visto que no todo lenguaje es regular.

Gramáticas independientes del contexto

En las próximas sesiones hablaremos de una clase más amplia de lenguajes, llamados lenguajes independientes del contexto. Los lenguajes independientes del contexto han sido definidos inicialmente en lingüística para analizar los lenguajes humanos.

Gramáticas independientes del contexto

En las próximas sesiones hablaremos de una clase más amplia de lenguajes, llamados lenguajes independientes del contexto. Los lenguajes independientes del contexto han sido definidos inicialmente en lingüística para analizar los lenguajes humanos.

La idea es de descomponer un lenguaje en componentes pequeñas, como *verbos*, *nombres*, *etc. . .* y de describir como cada elemento se conecta a los demás.

Gramáticas independientes del contexto

En las próximas sesiones hablaremos de una clase más amplia de lenguajes, llamados lenguajes independientes del contexto. Los lenguajes independientes del contexto han sido definidos inicialmente en lingüística para analizar los lenguajes humanos.

La idea es de descomponer un lenguaje en componentes pequeñas, como *verbos*, *nombres*, *etc...* y de describir como cada elemento se conecta a los demás.

Esto nos permite generar frases en el lenguaje en manera recursiva. Las reglas que nos permiten esta deducción se llaman una gramática independientes del contexto.

Parser

Sin embargo, una de las aplicaciones más importantes de los lenguajes independientes del contexto es en la construcción de *parser* de un lenguaje de programación.

Parser

Sin embargo, una de las aplicaciones más importantes de los lenguajes independientes del contexto es en la construcción de *parser* de un lenguaje de programación. Un parser extrae de un código su significado, usualmente se hace esto antes de traducir el código en código máquina.

Parser

Sin embargo, una de las aplicaciones más importantes de los lenguajes independientes del contexto es en la construcción de *parser* de un lenguaje de programación. Un parser extrae de un código su significado, usualmente se hace esto antes de traducir el código en código máquina.

Más adelante veremos un poco como, a partir de una gramática independientes del contexto se pueda construir un parser en manera (relativamente) sencilla.

Gramáticas independientes del contexto

Ejemplo: gramática

La siguiente es una gramática independiente del contexto, G_1:

$$A \to 0A1$$
$$A \to B$$
$$B \to \#$$

Ejemplo: gramática

La siguiente es una gramática independiente del contexto, G_1:

$$A \to 0A1$$
$$A \to B$$
$$B \to \#$$

Una gramática es una colección de reglas de substitución, o producciones de la forma

$$\texttt{variable} \to \texttt{cadena de variables y terminales},$$

por ejemplo, en G_1 las variables son A y B; los terminales son 0, 1, y $\#$.

Ejemplo: gramática

La siguiente es una gramática independiente del contexto, G_1:

$$A \rightarrow 0A1$$
$$A \rightarrow B$$
$$B \rightarrow \#$$

Una gramática es una colección de reglas de substitución, o producciones de la forma

$$\texttt{variable} \rightarrow \texttt{cadena de variables y terminales},$$

por ejemplo, en G_1 las variables son A y B; los terminales son 0, 1, y $\#$. La variable en la primera regla de substitución es llamada variable inicial.

Ejemplo: derivación

Dada una gramática, se pueden producir frases de la siguiente manera:

1. Escribimos la variable inicial;

Ejemplo: derivación

Dada una gramática, se pueden producir frases de la siguiente manera:

1. Escribimos la variable inicial;
2. Elegimos una regla de substitución para remplazar la variable inicial con una cadena de variables y terminales;

Ejemplo: derivación

Dada una gramática, se pueden producir frases de la siguiente manera:

1. Escribimos la variable inicial;
2. Elegimos una regla de substitución para remplazar la variable inicial con una cadena de variables y terminales;
3. Seguimos con el paso 2 hasta que no queden variables.

Ejemplo: derivación

Dada una gramática, se pueden producir frases de la siguiente manera:

1. Escribimos la variable inicial;
2. Elegimos una regla de substitución para remplazar la variable inicial con una cadena de variables y terminales;
3. Seguimos con el paso 2 hasta que no queden variables.

La secuencia de substituciones se llama una derivación. Por ejemplo, si $G_1 = \{A \to 0A1, A \to B, B \to \#\}$, una derivación de $000\#111 \in G_1$ es:

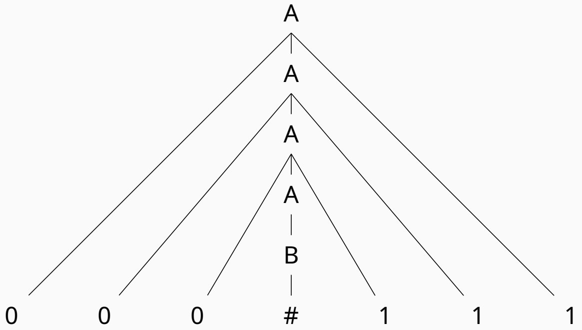
$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111.$$

Ejemplo: árbol de parsing

Una manera muy útil de representar la derivación

$$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111.$$

es utilizar un árbol de parsing:



Lenguajes independientes del contexto

Todas las frases que se pueden generar a partir de una gramática independiente del contexto dada forman su lenguaje.

Lenguajes independientes del contexto

Todas las frases que se pueden generar a partir de una gramática independiente del contexto dada forman su lenguaje. Por ejemplo $L(G_1) = \{0^n\#1^n, n \geq 0\}$.

Lenguajes independientes del contexto

Todas las frases que se pueden generar a partir de una gramática independiente del contexto dada forman su lenguaje. Por ejemplo $L(G_1) = \{0^n\#1^n, n \geq 0\}$.

Todos los lenguajes que pueden ser descritos por una gramática independiente del contexto se llaman lenguajes independientes del contexto `CFL`.

Definición formal de gramática independiente del contexto

Definición

Somos listos para definir formalmente una gramática independiente del contexto (o `CFG`).

Definición

Una gramática independiente del contexto es una 4-tupla (V, Σ, R, S), donde:

- V es un conjunto finito de variables;
- Σ es un conjunto finito, disyunto de V, de terminales;
- R es el conjunto finito de reglas, cada regla asocia una variable a una cadena de variables y terminales;
- $S \in V$ es la variable inicial.

Definición

Si u y v son dos cadenas de terminales y A es una variable, con una regla $A \to w$, decimos que uwv se deriva directamente de uAv, escrito $uAv \Rightarrow uwv$.

Definición

Si u y v son dos cadenas de terminales y A es una variable, con una regla $A \to w$, decimos que uwv se deriva directamente de uAv, escrito $uAv \Rightarrow uwv$. Más en general, decimos que v deriva u si $u = v$ o existe una secuencia $u_1, u_2, \ldots, u_k$, con $k \geq 0$ tal que

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_k \Rightarrow v,$$

lo que se denota $u \stackrel{*}{\Rightarrow} v$.

Definición

Si u y v son dos cadenas de terminales y A es una variable, con una regla $A \to w$, decimos que uwv se deriva directamente de uAv, escrito $uAv \Rightarrow uwv$. Más en general, decimos que v deriva u si $u = v$ o existe una secuencia $u_1, u_2, \ldots, u_k$, con $k \geq 0$ tal que

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_k \Rightarrow v,$$

lo que se denota $u \stackrel{*}{\Rightarrow} v$.

El lenguaje de una gramática es $\{w \in \Sigma^*, S \stackrel{*}{\Rightarrow} w\}$.

Un ejemplo

Sea $G = (V, \Sigma, R, \langle \texttt{expr} \rangle)$, donde $V = \{\langle \texttt{expr} \rangle, \langle \texttt{term} \rangle, \langle \texttt{factor} \rangle\}$ y $\Sigma = \{\mathsf{a}, +, \times, (,)\}$. Las reglas de G son

$$\begin{aligned}
\langle \texttt{expr} \rangle &\to \langle \texttt{expr} \rangle + \langle \texttt{term} \rangle | \langle \texttt{term} \rangle \\
\langle \texttt{term} \rangle &\to \langle \texttt{term} \rangle \times \langle \texttt{factor} \rangle | \langle \texttt{factor} \rangle \\
\langle \texttt{factor} \rangle &\to (\langle \texttt{expr} \rangle) | \mathsf{a}
\end{aligned}$$

Un ejemplo

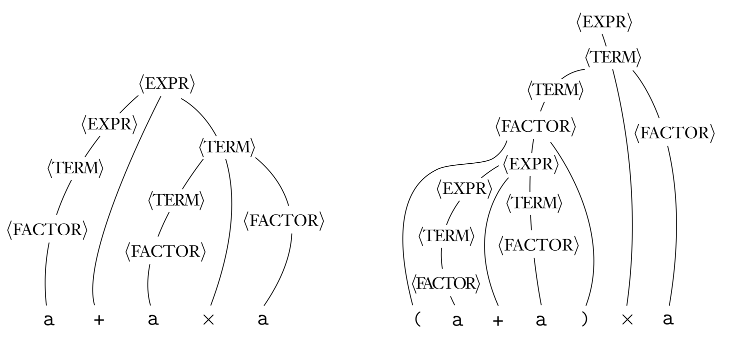
Sea $G = (V, \Sigma, R, \langle \texttt{expr} \rangle)$, donde $V = \{\langle \texttt{expr} \rangle, \langle \texttt{term} \rangle, \langle \texttt{factor} \rangle\}$ y $\Sigma = \{\mathtt{a}, +, \times, (,)\}$. Las reglas de G son

$$\begin{aligned}
\langle \texttt{expr} \rangle &\to \langle \texttt{expr} \rangle + \langle \texttt{term} \rangle | \langle \texttt{term} \rangle \\
\langle \texttt{term} \rangle &\to \langle \texttt{term} \rangle \times \langle \texttt{factor} \rangle | \langle \texttt{factor} \rangle \\
\langle \texttt{factor} \rangle &\to (\langle \texttt{expr} \rangle) | \mathtt{a}
\end{aligned}$$

Esta gramática describe un pedazo de la aritmética.

Un ejemplo

Nótese que G factoriza las expresiones:



Un ejemplo más complicado

Consideramos la gramática

$$
\begin{aligned}
\langle\texttt{frase}\rangle &\to \langle\texttt{frase-nombre}\rangle \mid \langle\texttt{frase-verbo}\rangle \\
\langle\texttt{frase-nombre}\rangle &\to \langle\texttt{nombre-c}\rangle \mid \langle\texttt{nombre-c}\rangle\langle\texttt{frase-prep}\rangle \\
\langle\texttt{frase-verbo}\rangle &\to \langle\texttt{verbo-c}\rangle \mid \langle\texttt{verbo-c}\rangle\langle\texttt{frase-prep}\rangle \\
\langle\texttt{frase-prep}\rangle &\to \langle\texttt{prep}\rangle\langle\texttt{nombre-c}\rangle \\
\langle\texttt{nombre-c}\rangle &\to \langle\texttt{articulo}\rangle\langle\texttt{nombre}\rangle \\
\langle\texttt{articulo}\rangle &\to \texttt{a}|\texttt{the} \\
\langle\texttt{nombre}\rangle &\to \texttt{boy}|\texttt{girl}|\texttt{flower} \\
\langle\texttt{verbo}\rangle &\to \texttt{touches}|\texttt{likes}|\texttt{sees} \\
\langle\texttt{prep}\rangle &\to \texttt{with}
\end{aligned}
$$

que genera un pedazo de la lengua inglés, como por ejemplo: `the boy sees a flower`.

Construir `CFG`

Como con los autómatas, diseñar una `CFG` es algo creativo. De todos modos, hay estrategias que se pueden utilizar.

Construir CFG

Como con los autómatas, diseñar una CFG es algo creativo. De todos modos, hay estrategias que se pueden utilizar.

Consideramos el lenguaje formado por $\{0^n1^n, n \geq 0\} \cup \{1^n0^n, n \geq 0\}$. Podemos construir una gramática por el primer pedazo definiendo

$$S_1 \to 0S_11|\varepsilon.$$

De manera parecida, una gramática por el segundo pedazo es

$$S_2 \to 1S_20|\varepsilon.$$

Finalmente una gramática para todo el lenguaje es

$$S \to S_1|S_2$$
$$S_1 \to 0S_11|\varepsilon$$
$$S_2 \to 1S_20|\varepsilon$$

Construir CFG

Si hay una cadena del tipo 0^n1^n, donde se necesitaría tener en memoria una cantidad infinita de elementos de una cadena para compararlos con los de la cadena después, pueden añadir la regla $R \to 0R1$.

Construir CFG

Si hay una cadena del tipo $0^n 1^n$, donde se necesitaría tener en memoria una cantidad infinita de elementos de una cadena para compararlos con los de la cadena después, pueden añadir la regla $R \to 0R1$.

Para terminar, a veces uno puede utilizar la recursividad para ayudarse.

Gramática para un autómata

Construir CFG

Si un lenguaje independiente del contexto es también un lenguaje regular, podemos construir un DFA que lo reconoce y traducir el autómata en una CFG en la siguiente manera:

1. Para cualquier estado q_i del autómata definimos una variable R_i.

Construir CFG

Si un lenguaje independiente del contexto es también un lenguaje regular, podemos construir un DFA que lo reconoce y traducir el autómata en una CFG en la siguiente manera:

1. Para cualquier estado q_i del autómata definimos una variable R_i.
2. Si hay la transición $\delta(q_i, a) = q_j$, añadimos la regla $R_i \to aR_j$.

Construir `CFG`

Si un lenguaje independiente del contexto es también un lenguaje regular, podemos construir un `DFA` que lo reconoce y traducir el autómata en una `CFG` en la siguiente manera:

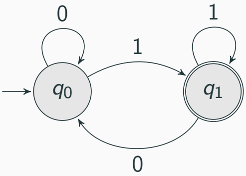
1. Para cualquier estado q_i del autómata definimos una variable R_i.
2. Si hay la transición $\delta(q_i, a) = q_j$, añadimos la regla $R_i \to aR_j$.
3. Si q_i es un estado final del autómata, añadimos la regla $R_i \to \varepsilon$.

Construir `CFG`

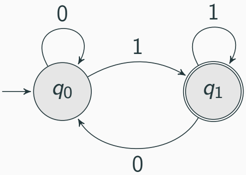
Si un lenguaje independiente del contexto es también un lenguaje regular, podemos construir un `DFA` que lo reconoce y traducir el autómata en una `CFG` en la siguiente manera:

1. Para cualquier estado q_i del autómata definimos una variable R_i.
2. Si hay la transición $\delta(q_i, a) = q_j$, añadimos la regla $R_i \to aR_j$.
3. Si q_i es un estado final del autómata, añadimos la regla $R_i \to \varepsilon$.
4. Finalmente, decimos que la variable inicial es la variable R_0 (dado que q_0 era el estado inicial del `DFA`)

Ejemplo

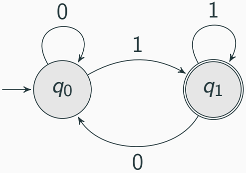


Ejemplo



- $R_0 \rightarrow 0R_0 \mid 1R_1$

Ejemplo



- $R_0 \rightarrow 0R_0 \mid 1R_1$
- $R_1 \rightarrow 1R_1 \mid 0R_0 \mid \varepsilon$

Resumen

Resumen

Hoy aprendimos:

- Qué es un lenguaje independiente del contexto y como se puede describir a través de una gramática independiente del contexto;
- Qué es un arból de parsing;
- Unos "trucos" para construir gramáticas independientes del contexto;
- Cómo encontrar la gramática para un `DFA`.