Expresiones Regulares y Derivadas Formales

Las Derivadas Sucesivas.

Universidad de Cantabria

Esquema

Derivadas Sucesivas

Recordemos que los lenguajes de los prefijos dan información sobre los lenguajes.

Derivadas Sucesivas

La idea es “derivar las derivadas” y utilizar el siguiente resultado:

Lemma

Sea $L \subseteq \Sigma^*$ un lenguaje sobre el alfabeto Σ generado por una gramática regular $G := (V, \Sigma, q_0, P)$. Sea $a \in \Sigma$ un símbolo del alfabeto. Entonces, la siguiente gramática $G_a = (V_a, \Sigma, q_a, P_a)$ genera el lenguaje $a \cdot L$ donde:

- *q_a es una nueva variable (no presente en V) y $V_a := V \cup \{q_a\}$.*
- *$P_a := P \cup \{q_a \mapsto aq_0\}$.*

Derivadas Sucesivas

Definición (Derivadas Sucesivas de una Expresión Regular)

Sea $\Sigma = \{a_1, \ldots, a_n\}$ *un alfabeto finito,* $\omega \in \Sigma^*$ *una palabra sobre el alfabeto y* α *una expresión regular. Definiremos la derivada* $D_\omega(\alpha)$ *mediante el proceso siguiente:*

- *Si* $\omega = \lambda$ *es la palabra vacía,* $D_\lambda(\alpha) = \alpha$.
- *Si* $|\omega| = 1$ *(es una palabra de longitud* 1*) y, por tanto,* $\omega = a_i \in \Sigma$*, definimos* $D_\omega(\alpha) = D_{a_i}(\alpha)$*, conforme a la definición de derivada anterior.*
- *Si* $\omega = a\omega_1$ *con* $a \in \Sigma$ *y* $\omega_1 \in \Sigma^*$*, definimos*

$$D_\omega(\alpha) = D_{a_i}(D_{\omega_1}(\alpha)).$$

Derivadas Sucesivas

Al conjunto de derivadas sucesivas la denotaremos:

$$Der(\alpha) := \{D_\omega(\alpha) \ : \ \omega \in \Sigma^*\}.$$

El conjunto de derivadas sucesivas es un conjunto finito, por lo tanto es posible calcularlo.

Algoritmo

La entrada de nuestro algoritmo es una expresión regular α sobre un alfabeto finito Σ.

Algoritmo

Hallar todos los elementos del conjunto
$Der(\alpha) := \{D_\omega(\alpha) \ : \ \omega \in \Sigma^*\}$.

Algoritmo

Definir un conjunto finito V de variables, biyectable al conjunto $Der(\alpha)$. Llamaremos a la variable inicial q_0 y se relacionará con la expresión α.

Algoritmo

Definir $P_1 := 1$ y

$$P_2 := \begin{cases} \{q \mapsto \lambda\}, & \text{si } \lambda \in L(\alpha) \\ \emptyset, & \text{en caso contrario} \end{cases}$$

Algoritmo

- **Mientras** $P_2 \neq P_1$ **hacer**
  - $P_1 := P_2$
  - Para cada $\beta \in Der(\alpha)$ **hacer**
    - Para cada $a \in \Sigma$ **hacer**
    - Hallar $\gamma := D_a(\beta)$, $p := E(\gamma)$ y $q := E(\beta)$ en V.
    - Si $\lambda \in L(\gamma)$, hacer $P_2 := P_2 \cup \{q \mapsto a\}$.
    - Si $\gamma \neq \emptyset, \lambda$, hacer $P_2 := P_2 \cup \{q \mapsto ap\}$.
    - **tomar siguiente** α
  - **tomar siguiente** β
- **finaliza mientras**

La gramática $G = (V, \Sigma, q_0, P_2)$.

Ejemplo

Tomemos la expresión regular $a(a+b)^*b$, empecemos calculando el conjunto de las derivadas:

$$Der(\alpha) = \{\emptyset, a(a+b)^*b, (a+b)^*b, (a+b)^*b+\lambda\}$$

Ejemplo

$$D_a(a(a+b)^*b) = (a+b)^*b, \quad D_b(a(a+b)^*b) = \emptyset.$$

$$D_a((a+b)^*b) = (a+b)^*b, \quad D_b((a+b)^*b) = (a+b)^*b + \lambda.$$

$$D_a((a+b)^*b+\lambda) = (a+b)^*b, \quad D_b((a+b)^*b+\lambda) = (a+b)^*b+\lambda.$$

Ejemplo

Relacionemos todas las variables con las expresiones regulares:

$$q_0 = a(a+b)^*b,\ q_1 = (a+b)^*b,\ q_2 = (a+b)^*b + \lambda.$$

Estas son las producciones

$$\begin{aligned} q_0 &\mapsto aq_1, \\ q_1 &\mapsto aq_1 | bq_2 | b, \\ q_2 &\mapsto aq_1 | bq_2. \end{aligned}$$

Ejemplo

Relacionemos todas las variables con las expresiones regulares:

$$q_0 = a(a+b)^*b,\ q_1 = (a+b)^*b,\ q_2 = (a+b)^*b + \lambda.$$

Estas son las producciones

$$\begin{aligned} q_0 &\mapsto aq_1, \\ q_1 &\mapsto aq_1 | bq_2 | b, \\ q_2 &\mapsto aq_1 | bq_2. \end{aligned}$$

Resultados Adicionales

Usando las propiedades adicionales de las expresiones regulares podemos suponer $\alpha = \alpha_1 + \ldots + \alpha_n$ (Forma Disyuntiva Normal).

Resultados Adicionales

Lemma

Sea L_1 y L_2 dos lenguajes (regulares) sobre el alfabeto Σ generados respectivamente por gramáticas $G_1 = (V_1, \Sigma, q_1, P_1)$ y $G_2 = (V_2, \Sigma, q_2, P_2)$, entonces $L_1 \cup L_2$ es también un lenguaje (regular) generado por una gramática. La gramática que genera la unión es una nueva gramática $G = (V, \Sigma, q_0, P)$ dada por las reglas siguientes:

1. $q_0 \notin V_1 \cup V_2$, *es decir es una nueva variable.*
2. $V := V_1 \cup V_2 \cup \{q_0\}$.
3. $P := P_1 \cup P_2 \cup \{q_0 \mapsto q_1 \mid q_2\}$.

Resultados Adicionales

En el caso de unión finita $L = L_1 \cup \cdots \cup L_m$, el Lema anterior se puede extender de la forma obvia. Por tanto, la unión finita de lenguajes generados por gramáticas (resp. regulares) es un lenguaje generado por una gramática (resp. regulares).

Ejemplo

El lenguaje $a^* + b^*$ tiene una gramática que se puede hallar utilizando ese lema:

$$\begin{aligned} q_0 &\mapsto q_1 | q_2, \\ q_1 &\mapsto a q_1 | \lambda, \\ q_2 &\mapsto b q_2 | \lambda. \end{aligned}$$

Ejemplo

La gramática que se obtiene aplicando el algoritmo es:

$$
\begin{aligned}
q_0 &\mapsto aq_1|bq_2|\lambda, \\
q_1 &\mapsto aq_1|a, \\
q_2 &\mapsto bq_2|b.
\end{aligned}
$$

Posibles Transformaciones

La primera gramática puede ser transformada, simplemente sustituyendo las producciones de q_1 en q_0, lo que quiere decir que el número de gramáticas que generan el mismo lenguaje es infinito.

Conclusiones

Las gramáticas que estamos buscando no son únicas. Para utilizar el lema que se ha presentado es necesario tener la expresión regular en Forma Disyuntiva Normal.