Lenguajes Formales y Monoides

Universidad de Cantabria

Esquema

1. Lenguajes Formales y Monoides
2. Operaciones con Palabras
3. Sistemas de Transición

La operación esencial sobre Σ^* es la concatenación o adjunción de palabras:

$$\cdot : \Sigma^* \times \Sigma^* \longrightarrow \Sigma^*$$

$$(x, y) \longmapsto x \cdot y$$

es decir, si $x = x_1 \cdots x_n$ e $y = y_1 \cdots y_m$, entonces

$$x \cdot y = x_1 \cdots x_n y_1 \cdots y_m.$$

Denotemos por $\lambda \in \Sigma^*$ la palabra vacía (para distinguirla del lenguaje vacío $\emptyset$).

Teorema

$(\Sigma^*, \cdot)$ es un monoide (semigrupo unitario), donde λ es el elemento neutro, y la longitud define un morfismo de monoides entre Σ^* y el conjunto de los números naturales. El monoide Σ^* es abeliano si y solamente si el cardinal de Σ es uno.

Teorema

$(\Sigma^*, \cdot)$ es un monoide (semigrupo unitario), donde λ es el elemento neutro, y la longitud define un morfismo de monoides entre Σ^* y el conjunto de los números naturales. El monoide Σ^* es abeliano si y solamente si el cardinal de Σ es uno.

Discusión en Clase

Si Σ es un alfabeto finito, el conjunto Σ^* es numerable, esto es, es biyectable con el conjunto $\mathbb{N}$ de los números naturales.

Ejemplo: Si el conjunto $\Sigma = \{0, 1\}$, entonces la biyección viene dada por

$$\begin{aligned} B : \Sigma^* &\mapsto \mathbb{N} \\ \omega &\mapsto 2^{l(\omega)} + b(\omega) \end{aligned}$$

donde $l(\omega)$ es la longitud de la palabra y $b(\omega)$ es el número natural que tiene como representación en binario la palabra ω.

Discusión en Clase

Si Σ es un alfabeto finito, el conjunto Σ^* es numerable, esto es, es biyectable con el conjunto $\mathbb{N}$ de los números naturales.
Ejemplo: Si el conjunto $\Sigma = \{0, 1\}$, entonces la biyección viene dada por

$$\begin{array}{rcl} B : \Sigma^* & \mapsto & \mathbb{N} \\ \omega & \mapsto & 2^{l(\omega)} + b(\omega) \end{array}$$

donde $l(\omega)$ es la longitud de la palabra y $b(\omega)$ es el número natural que tiene como representación en binario la palabra ω.

Discusión en Clase

Si Σ es un alfabeto finito, el conjunto Σ^* es numerable, esto es, es biyectable con el conjunto $\mathbb{N}$ de los números naturales.
Ejemplo: Si el conjunto $\Sigma = \{0, 1\}$, entonces la biyección viene dada por

$$\begin{array}{rccl} B : \Sigma^* & \mapsto & \mathbb{N} \\ \omega & \mapsto & 2^{l(\omega)} + b(\omega) \end{array}$$

donde $l(\omega)$ es la longitud de la palabra y $b(\omega)$ es el número natural que tiene como representación en binario la palabra ω.

Discusión en Clase

Esto es muy importante, ya que a partir de ahora se podrá ordenar las palabras de menor a mayor. También queda definido que es la palabra más pequeña que no esté en un lenguaje.

Ejemplo

Sea $\Sigma = \{0, 1\}$. *Se pide*

- *la palabra más corta que tenga un número par de ceros.*
- *la palabra más corta que empiece por 1, acabe por 0 y tenga longitud impar.*
- *el número de palabras de longitud 17 que cumplen cada una de estas condiciones.*

Operaciones Básicas con Palabras

Además de la concatenación de palabras, existen las siguientes operaciones:

- *Reverso de una Palabra:* Se trata de una biyección

$$^R : \Sigma^* \longrightarrow \Sigma^*,$$

dada mediante:

  - Si $\omega = \lambda$, $\lambda^R = \lambda$,
  - Si $\omega = x_1 \cdots x_n \in \Sigma^*$, con $x_i \in \Sigma$, se define

$$\omega^R := x_n x_{n-1} \cdots x_1 \in \Sigma^*.$$

- *Potencia de Palabras.* Se define recursivamente a partir de la concatenación. Dada una palabra $\omega \in \Sigma^*$ y un número natural $n \in \mathbb{N}$, se define la potencia ω^n, mediante:
  - Si $n = 0$, entonces $\omega^0 = \lambda$,
  - Para $n \geq 1$, se define

$$\omega^n := \omega \cdot \omega^{n-1}.$$

Operaciones Básicas con Palabras

Teorema

Dado un alfabeto finito Σ y dos palabras $\omega_1, \omega_2 \in \Sigma^*$ entonces tenemos que

- $(\omega_1\omega_2)^R = \omega_2^R\omega_1^R$.
- $(\omega_1^n)^R = (\omega_1^R)^n$

Operaciones Básicas con Palabras

Además de la concatenación de palabras, existen las siguientes operaciones:

- *Reverso de una Palabra:* Se trata de una biyección

$$^R : \Sigma^* \longrightarrow \Sigma^*,$$

dada mediante:

  - Si $\omega = \lambda$, $\lambda^R = \lambda$,
  - Si $\omega = x_1 \cdots x_n \in \Sigma^*$, con $x_i \in \Sigma$, se define

$$\omega^R := x_n x_{n-1} \cdots x_1 \in \Sigma^*.$$

- *Potencia de Palabras.* Se define recursivamente a partir de la concatenación. Dada una palabra $\omega \in \Sigma^*$ y un número natural $n \in \mathbb{N}$, se define la potencia ω^n, mediante:
  - Si $n = 0$, entonces $\omega^0 = \lambda$,
  - Para $n \geq 1$, se define

$$\omega^n := \omega \cdot \omega^{n-1}.$$

Operaciones Básicas con Palabras

Teorema

Dado un alfabeto finito Σ y dos palabras $\omega_1, \omega_2 \in \Sigma^*$ entonces tenemos que

- $(\omega_1\omega_2)^R = \omega_2^R\omega_1^R$.
- $(\omega_1^n)^R = (\omega_1^R)^n$

Un lenguaje que tendrá cierta relevancia en futuras discusiones posteriores es el *Palíndromo* $\mathcal{P} \subseteq \{0,1\}^*$ y que viene dado por la siguiente igualdad:

$$\mathcal{P} := \{\omega \in \{0,1\}^* \ : \ \omega^R = \omega\}.$$

Un lenguaje que tendrá cierta relevancia en futuras discusiones posteriores es el *Palíndromo* $\mathcal{P} \subseteq \{0,1\}^*$ y que viene dado por la siguiente igualdad:

$$\mathcal{P} := \{\omega \in \{0,1\}^* \ : \ \omega^R = \omega\}.$$

Ejemplos: Ana, arenera, arepera, anilina, Oruro, oso, radar, reconocer, rotor, salas, seres, somos, sometemos

Operaciones Elementales con Lenguajes

Operaciones básicas con lenguajes formales, fijado alfabeto Σ

- Unión de Lenguajes Dados $L_1, L_2 \subseteq \Sigma^*$:

$$L_1 \cup L_2 := \{\omega \in \Sigma^* \ : \ [\omega \in L_1] \vee [\omega \in L_2]\}.$$

- Concatenación de Lenguajes: Dados $L_1, L_2 \subseteq \Sigma^*$:

$$L_1 \cdot L_2 := \{\omega_1 \cdot \omega_2 \in \Sigma^* \ : \ \omega_1 \in L_1, \ \omega_2 \in L_2\}.$$

- Potencia de Lenguajes:
  - Si $n = 0$, $L^0 = \{\lambda\}$.
  - Si $n \geq 1$, $L^n := L \cdot (L^{n-1})$.

Algunos Comentarios

- $L_1 \cdot L_2$ no es, en general, igual a $L_2 \cdot L_1$.
- No es cierto que $L_1 \cdot L_2 = L_2 \cdot L_1$ implica $L_1 = L_2$.
  El ejemplo más sencillo de esto es
  $\Sigma = \{a\},\ L_1 = \{a\},\ L_2 = \{aa\}$.
- Se cumple la propiedad distributiva. Para lenguajes L_1, L_2 y L_3 contenidos en Σ^*:

$$L_1 \cdot (L_2 \cup L_3) = L_1 \cdot L_2 \cup L_1 \cdot L_3.$$

$$(L_1 \cup L_2) \cdot L_3 = L_1 \cdot L_3 \cup L_2 \cdot L_3.$$

Vamos a probar esta última propiedad.

Otras Operaciones entre Lenguajes

Ejemplo

Si $A = \{0, 1\}$ *y* $B = \{1, 2\}$*, entonces* $AB = \{01, 02, 11, 12\}$

- *Responded si es verdad que si el número de elementos de A es m y el número de elementos de B es n entonces el número de elementos de AB es nm.*
- *Definid el conjunto* $A = \{\omega \in \Sigma^* \mid 100\omega = \omega 100\}$.

Otras Operaciones entre Lenguajes

- *Clausura transitiva o monoide generado por un lenguaje:* Dado un lenguaje $L \subseteq \Sigma^*$ se define el monoide L^* que genera mediante la igualdad siguiente:

$$L^* := \bigcup_{n \in \mathbb{N}} L^n.$$

- *Clausura positiva de un lenguaje:* Dado un lenguaje $L \subseteq \Sigma^*$ se define la clausura positiva L^+ que genera mediante la igualdad siguiente:

$$L^+ := \bigcup_{n \geq 1} L^n.$$

Sistemas de Transición

Un proceso algorítmico se basa en “una serie de pasos hasta obtener un resultado".
Surge así la noción de *Sistema de Transición (Deductivo/ de Producciones/ semi–Thue...)*
Necesitamos caracterizar estos “pasos” de una manera formal.

Definición

Llamaremos sistema de transición a todo par $(S, \rightarrow)$*, donde* S *es un conjunto (que se denomina espacio de configuraciones) y* $\rightarrow \subseteq S \times S$ *es una relación.*

Una *sucesión de computación* en el sistema de transición $(S, \rightarrow)$ es una sucesión finita de elementos de S:

$$s_1, \ldots, s_n$$

donde $(s_i, s_{i+1}) \in \rightarrow$.
Escribir $s_i \rightarrow s_{i+1}$ es equivalente a $(s_i, s_{i+1}) \in \rightarrow$. Con ello, una computación en el sistema de transición $(S, \rightarrow)$ es simplemente:

$$s_1 \rightarrow \cdots \rightarrow s_n$$

Se dice que el sistema de transición es determinista si cada $s \in S$ tiene un sólo sucesor a lo sumo y es indeterminista en caso contrario.

Definición

Dada una configuración $s \in S$, diremos que una configuración $s' \in S$ es deducible de s y lo denotaremos por $s \vdash s'$, si existe una sucesión de computación

$$s = s_1 \to \cdots \to s_n = s'$$

La relación que debe existir entre los datos de un problema y su resolución es de ser deducible para algún sistema de transición. En cada caso clarificaremos los sistemas de transición esenciales del modelo de cálculo introducido (es decir, la acción dinámica del modelo definido).

Conclusiones

Los sistemas de transición deterministas son una abstracción de como representar un modelo de computación, aunque no de todos.
Existe similitud entre sistemas de transición y grafos (potencialmente con un número infinito de nodos). Un grafo orientado es simplemente un sistema de transición con un conjunto de configuraciones finito.

Sistemas de rescritura

Definición

Un sistema de reescritura, tambien conocido como un sistema semi-Thue utiliza la estructura de monomio de las palabra para definir una relación de rescritura que se aplica a todos los substrings. Esto está dado por un conjunto de reglas R que se definen como reglas de reescritura.

Sistemas de rescritura

Ejemplo

Sea $\Sigma = \{0, 1\}$, *y sean las reglas* $R = \{01 \mapsto \lambda, 10 \mapsto \lambda\}$

$0100010 \mapsto 00010 \mapsto 00$

Sistemas de rescritura

Notad que ahora podemos relacionar un sistema de transición con un sistema de reescritura. Las configuraciones son palabras y una computación está dada por aplicar una relación de reescritura.