AUTÓMATAS Y LENGUAJES FORMALES
PRELIMINARES MATEMÁTICOS

Francisco Hernández Quiroz

Departamento de Matemáticas
Facultad de Ciencias, UNAM
E-mail: fhq@ciencias.unam.mx
Página Web: www.matematicas.unam.mx/fhq

Posgrado en Ciencia e Ingeniería de la Computación

Inducción en definiciones y demostraciones

En matemáticas es común que un conjunto se defina *inductivamente*, es decir, a partir de un conjunto inicial y de reglas para crear nuevos elementos. Por ejemplo, las listas de naturales:

- [ ] es la lista vacía
- si $[a_1, \ldots, a_n]$ es una lista de $\mathbb{N}$ y $m \in \mathbb{N}$ entonces $[m, a_1, \ldots, a_n]$ es una lista de naturales.

Las definiciones inductivas vienen acompañadas de *demostraciones inductivas*, es decir, de técnicas de demostración que se basan en la estructura inductiva de un conjunto.

Números naturales

Los números naturales se pueden definir inductivamente:

- 0 es un número natural
- si n es un número natural, entonces el sucesor de n también es un número natural

El sucesor de un número k suele denotarse como $s(k)$, $suc(k)$ o simplemente como $k + 1$.

Inducción matemática

Si se quiere demostrar una propiedad P de los números naturales entonces se puede hacer utilizando su estructura inductiva:

1. Se demuestra que P vale para 0.
2. Se formula una hipótesis inductiva: "P vale para cierto número $m \in \mathbb{N}$".
3. Se demuestra que P vale para $m + 1$.

Ejemplo de inducción matemática

Queremos demostrar que $\sum_{i=0}^{n} i = \frac{n^2+n}{2}$. Entonces

1. Demostramos que $\sum_{i=0}^{0} i = \frac{0^2+0}{2}$ (obvio).
2. Formulamos la hipótesis $\sum_{i=0}^{m} i = \frac{m^2+m}{2}$.
3. Demostramos que $\sum_{i=0}^{m+1} i = \frac{(m+1)^2+m+1}{2}$.

Conjuntos bien ordenados

- Los conjuntos definidos inductivamente se pueden ordenar de acuerdo con el orden estructural

  $a <_S b$ sii a se construye a partir de b.

  En el caso de $\mathbb{N}$, $m <_S n$ sii $n = m + 1$.
- En muchos casos, $<_S$ ordena un conjunto de tal forma que
  - no existen cadenas infinitas descendentes $\cdots <_S a_{n+1} <_S a_n <_S \cdots <_S a_1$
  - para todo elemento, existe un número finito de predecesores en $<_S$.

Inducción bien fundada

Cuando hay un conjunto inductivo C bien ordenado por $<_S$, es posible demostrar una propiedad P de elementos de C inductivamente:

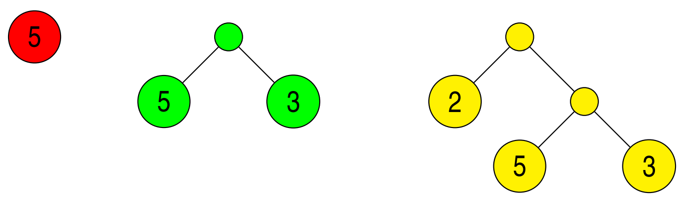
1. Se demuestra que P vale de todos los elementos minimales de C.
2. Se formula una hipótesis inductiva: "Sea $a \in C$. P vale para todos los $b \in C$ tales que $b <_S a$".
3. Se demuestra que P también vale para a.

Ejemplo de inducción bien fundada I

Definimos los árboles binarios de números naturales:

- Todos los números naturales son árboles binarios (los elementos básicos del conjunto)
- Si a y b son árboles binarios, entonces $(a \bullet b)$ es un árbol binario (caso inductivo)

Los árboles binarios también se pueden representar gráficamente:



Si $a = (b \bullet c)$, entonces $b <_S a$ y $c <_S a$.

Ejemplo de inducción bien fundada II

Considérense ahora las siguientes funciones:

$$\begin{aligned}
\text{Max}(n) &= n & \text{si n} \in \mathbb{N}\\
\text{Max}(a \bullet b) &= \text{máximo}\{\text{Max}(a), \text{Max}(b)\} &\\
\text{Card}(n) &= 1 & \text{si n} \in \mathbb{N}\\
\text{Card}(a \bullet b) &= \text{Card}(a) + \text{Card}(b) &\\
\text{Suma}(n) &= n & \text{si n} \in \mathbb{N}\\
\text{Suma}(a \bullet b) &= \text{Suma}(a) + \text{Suma}(b) &
\end{aligned}$$

Demostraremos que para cualquier árbol binario de naturales a

$$\text{Suma}(a) \leq \text{Max}(a) \times \text{Card}(a).$$

Ejemplo de inducción bien fundada III

1. Caso básico. Los minimales en este caso son los números naturales. Sea $n \in \mathbb{N}$. Entonces

$$\text{Suma}(n) = n = \text{Max}(n) = \text{Max}(n) \times 1 = \text{Max}(n) \times \text{Card}(n)$$

2. Hipótesis inductiva. Sea $a = (b \bullet c)$. Entonces

$$\text{Suma}(b) \leq \text{Max}(b) \times \text{Card}(b).$$

$$\text{Suma}(c) \leq \text{Max}(c) \times \text{Card}(c).$$

pues $b <_S a$ y $c <_S a$.

3. Debemos demostrar que

$$\text{Suma}(a) \leq \text{Max}(a) \times \text{Card}(a),$$

lo que se sigue de la hipótesis inductiva y las definiciones de Max, Card y Suma.

Relaciones y cerradura transitiva y reflexiva

Sea A un conjunto. Una relación R en A es un subconjunto de $A \times A$. La cerradura transitiva y reflexiva de R se define inductivamente:

$$\begin{aligned}
R^0 &= \{(a, a) \mid a \in A\}\\
R^1 &= R\\
R^{n+1} &= R^n \cup \{(a, b) \mid \exists c \text{ tal que } (a, c) \text{ y } (c, b) \in R^n\}\\
R^* &= \bigcup_{n \in \mathbb{N}} R^n\\
R^+ &= \bigcup_{n \in \mathbb{N}} R^{n+1}
\end{aligned}$$

R^+ es la *cerradura transitiva* y R^* es la *cerradura transitiva y reflexiva* de R.

Alfabeto

- Un alfabeto Σ es un conjunto finito de símbolos. Ejemplo: {a, b, . . . z}.
- Una cadena (finita) es una sucesión de símbolos de Σ. Ejemplo *acczi*.
- La “cadena” vacía no contiene símbolos. Se suele denotar con ϵ o λ.
- El conjunto de cadenas del alfabeto Σ se denota con Σ^*.
- Si se excluye a ϵ, entonces se llama Σ^+.

Concatenación

- Las cadenas se pueden concatenar. Por ejemplo, si $\alpha = aabc$ y $\beta = ccaz$, entonces
$$\alpha \cdot \beta = aabcccaz.$$
- Es muy común omitir el signo de concatenación $\cdot$
- ϵ es el elemento neutro de la operación de concatenación, pues
$$\alpha\epsilon = \epsilon\alpha = \alpha.$$

Lenguajes

Si Σ es un alfabeto, entonces un lenguaje L es un subconjunto de Σ^*.

Nota

- Dado que Σ es finito, Σ^* es un conjunto infinito del mismo tamaño que $\mathbb{N}$.
- El conjunto de subconjuntos de Σ^* se denota como $\mathcal{P}(\Sigma^*)$.
- Por la definición anterior, el conjunto de lenguajes en Σ^* es $\mathcal{P}(\Sigma^*)$.
- $\mathcal{P}(\Sigma^*)$ es infinito, de un tamaño mayor que el de $\mathbb{N}$.

Operaciones en lenguajes

Sean A y B dos lenguajes en Σ. Tenemos las siguientes operaciones:

- Unión $A \cup B$.
- Intersección $A \cap B$.
- Diferencia $A - B = \{\alpha \mid \alpha \in A \text{ y } \alpha \notin B\}$.
- Complemento $\sim A = \Sigma^* - A$.
- Concatenación $AB = \{\alpha\beta \mid \alpha \in A \text{ y } \beta \in B\}$.
- Estrella de Kleene $A^* = \bigcup_{n\in\mathbb{N}} A^n$ donde
$$\begin{aligned} A^0 &= \{\epsilon\} \\ A^{n+1} &= A^n A \end{aligned}$$

Gramáticas

Una gramática G es una cuarteta $\langle \Sigma, \Gamma, S, \rightarrow_G \rangle$:

- Σ es un alfabeto de símbolos terminales
- Γ es un alfabeto de símbolos no terminales
- $S \in \Gamma$ es el símbolo inicial
- $\rightarrow_G \subseteq (\Sigma \cup \Gamma)^* \Gamma^+ (\Sigma \cup \Gamma)^* \times (\Sigma \cup \Gamma)^*$ son las reglas de producción

A partir de G definimos la relación $\Rightarrow_G \subseteq (\Sigma \cup \Gamma)^*\Gamma^+(\Sigma \cup \Gamma)^* \times (\Sigma \cup \Gamma)^*$.

$\alpha \Rightarrow_G \beta$ sii existen $\gamma, \kappa, \theta, \lambda$ tales que

- $\alpha = \gamma\kappa\theta$
- $\beta = \gamma\lambda\theta$
- $\kappa \rightarrow_G \lambda$

Lenguaje generado por una gramática

Sea G una gramática. El lenguaje generado por G es

$$L_G = \{\alpha \in \Sigma^* \mid S \Rightarrow^*_G \alpha\}$$

Ejemplo. Sea G la siguiente gramática

- $\Sigma = \{a, b\}$
- $\Gamma = \{S\}$
- $\{S \rightarrow_G aSb, S \rightarrow_G \epsilon\}$

El lenguaje generado por la gramática anterior son todas las cadenas formadas por cierto número de a seguidas del mismo número de b

Jerarquía de Chomsky

Se pueden clasificar las gramáticas y los lenguajes que generan de acuerdo con ciertas restricciones:

- *Tipo 3*: Si todas las reglas de producción tienen la forma $A \rightarrow \alpha B$ o $A \rightarrow \alpha$, donde $A, B \in \Gamma$ y $\alpha \in \Sigma^*$
- *Tipo 2*: Si todas las reglas tienen la forma $A \rightarrow \alpha$, donde $A \in \Gamma$ y $\alpha \in (\Sigma \cup \Gamma)^*$
- *Tipo 1*: Si no hay una regla de producción con la forma $\alpha \rightarrow \epsilon$
- *Tipo 0*: Sin restricciones.

Esta clasificación se conoce como la *jerarquía de Chomsky*

Gramáticas y el curso

Estos tipos también se conocen con otros nombres:

- Tipo 3: lenguajes regulares
- Tipo 2: lenguajes independientes del contexto (o libres del contexto)
- Tipo 1: lenguajes dependientes del contexto (o sensibles del contexto)
- Tipo 0: lenguajes recursivamente enumerables

Estudiaremos los tipos 3, 2 y 0 en las primeras tres partes del curso.