Apuntes IIC1253 - Matemáticas Discretas

Examen de Título 2014

Gabriel Diéguez Franzani

Selección de diapositivas de
Gonzalo Díaz, Nicolás Rivera y Marcelo Arenas

23 de diciembre de 2013

Contenidos I

Contenidos II

1. Inducción

Inducción

- Consideraremos que los números naturales ($\mathbb{N}$) empiezan en el 0.
- Los tres principios de inducción presentados a continuación son equivalentes, y no necesitan demostración (de ahí el nombre "principio"), pues son inherentes a la definición de los números naturales.

Inducción Simple

Principio Simple de Inducción

Sea $P(n)$ una propiedad (predicado) definido para los números naturales. Si se cumple que:

i) $P(0)$,

ii) $\forall n \in \mathbb{N}(P(n) \ \rightarrow \ P(n+1))$,

entonces se cumple:

$$\forall n \in \mathbb{N}(P(n)).$$

Nota: La condición (i) se llama **base** de la inducción, (ii) es el **paso inductivo**, dentro del cual la expresión $P(n)$ es la **hipótesis inductiva** (HI) y la expresión $P(n+1)$ es la **tesis inductiva** (TI).

Ejercicio

Demuestre que para todo $n \in \mathbb{N}$ se cumple que:

$$1 + \cdots + n = \frac{n(n+1)}{2}.$$

Inducción Simple

B.I. Tomando $n = 0$, tenemos que $0 = \frac{0(0+1)}{2}$.

H.I. Suponemos que para n se cumple que $1 + \ldots + n = \frac{n(n+1)}{2}$.

T.I. Debemos demostrar que $1 + \ldots + n + (n+1) = \frac{(n+1)((n+1)+1)}{2}$.

Por H.I., sabemos que $1 + \ldots + n = \frac{n(n+1)}{2}$, y luego

$$1 + \ldots + n + (n+1) = \frac{n(n+1)}{2} + (n+1)$$

$$1 + \ldots + n + (n+1) = \frac{n(n+1) + 2(n+1)}{2}$$

$$1 + \ldots + n + (n+1) = \frac{(n+2)(n+1)}{2}$$

$$1 + \ldots + n + (n+1) = \frac{(n+1)(n+2)}{2}$$

$$1 + \ldots + n + (n+1) = \frac{(n+1)((n+1)+1)}{2} \square$$

Inducción fuerte

Principio Fuerte de Inducción

Sea $P(n)$ una propiedad (predicado) definida para los números naturales. Si $\forall n \in \mathbb{N}$ se cumple que:

i) $\forall k < n\,(P(k)) \;\rightarrow\; P(n)$,

entonces se cumple:

$$\forall n \in \mathbb{N}(P(n)).$$

Principio del Buen Orden

Principio del Buen Orden

Dado $S \subseteq \mathbb{N}$ tal que $S \neq \emptyset$, se cumple que S tiene un menor elemento, es decir:

$$\exists m \in S, \forall x \in S (m \leq x).$$

- Los principios anteriores se aplican todos a los números naturales.
- Esto se debe a que $\mathbb{N}$ es un conjunto que se puede construir a partir de un elemento base y un operador.
  - En este caso, el elemento base es el 0 y el operador el "sucesor".
- Esta construcción a partir de elementos base y operadores es lo que se conoce como una **definición inductiva**.
- Intuitivamente, en el caso de $\mathbb{N}$ podemos obtener todo natural a partir de sumarle 1 a otro natural (excepto el 0).

Inducción Estructural

Formalmente, tenemos la siguiente definición inductiva de $\mathbb{N}$:

Definición

$\mathbb{N}$ es el menor conjunto que cumple las siguientes reglas:

1. $0 \in \mathbb{N}$
2. Si $n \in \mathbb{N}$, entonces $n + 1 \in \mathbb{N}$

- Es importante la afirmación de "menor conjunto", dado que existen otros que cumplen las reglas.
- Notamos que esta definición está estrechamente relacionada con los principios de inducción: la propiedad debe demostrarse para el 0 (elemento base y primera regla), y luego usando el operador (segunda regla).

Inducción Estructural

- Esta noción de definición inductiva se puede usar para definir otros conjuntos.
- Podremos usar inducción para demostrar propiedades sobre tales conjuntos.
- Podremos definir nuevos objetos (funciones, operaciones, etc.) usando la definición inductiva del conjunto.

Definición Inductiva

Para definir inductivamente un conjunto necesitamos:

1. Establecer que el conjunto es el menor que cumple las reglas.
2. Un conjunto (no necesariamente finito) de elementos base, que se supondrá que inicialmente pertenecen al conjunto que se quiere definir.
3. Un conjunto finito de reglas de construcción de nuevos elementos del conjunto a partir de elementos que ya están en él.

Inducción Estructural: un ejemplo

Ejemplo

Queremos definir el conjunto $\mathcal{E}_{\mathbb{N}}$ de todas las expresiones aritméticas sobre los números naturales que se pueden construir usando los símbolos $+, *, (,)$. Por ejemplo,

$$
\begin{gathered}
(5 + 3 * 4) * 10 \\
7 \\
1 + 2 + 3 + 4
\end{gathered}
$$

son expresiones en $\mathcal{E}_{\mathbb{N}}$.

Inducción Estructural: un ejemplo

Definición de $\mathcal{E}_{\mathbb{N}}$

$\mathcal{E}_{\mathbb{N}}$ es el menor conjunto tal que:

1. Si $k \in \mathbb{N}$, entonces $k \in \mathcal{E}_{\mathbb{N}}$.
2. Si $E_1, E_2 \in \mathcal{E}_{\mathbb{N}}$, entonces $E_1 + E_2 \in \mathcal{E}_{\mathbb{N}}$.
3. Si $E_1, E_2 \in \mathcal{E}_{\mathbb{N}}$, entonces $E_1 * E_2 \in \mathcal{E}_{\mathbb{N}}$.
4. Si $E \in \mathcal{E}_{\mathbb{N}}$, entonces $(E) \in \mathcal{E}_{\mathbb{N}}$.

Inducción Estructural: un ejemplo

Podemos definir operadores sobre $\mathcal{E}_{\mathbb{N}}$ valiéndonos de su definición inductiva:
Dada una expresión, el operador $\#_L : \mathcal{E}_{\mathbb{N}} \to \mathbb{N}$ entrega la cantidad de paréntesis izquierdos de ella.

Definición de $\#_L$

1. $\#_L(k) = 0$ para todo $k \in \mathbb{N}$.
2. $\#_L(E_1 + E_2) = \#_L(E_1) + \#_L(E_2)$ para todas $E_1, E_2 \in \mathcal{E}_{\mathbb{N}}$.
3. $\#_L(E_1 * E_2) = \#_L(E_1) + \#_L(E_2)$ para todas $E_1, E_2 \in \mathcal{E}_{\mathbb{N}}$.
4. $\#_L((E)) = 1 + \#_L(E)$ para toda $E \in \mathcal{E}_{\mathbb{N}}$.

Inducción Estructural: demostraciones

La inducción estructural nos permite además demostrar propiedades sobre los conjuntos y operadores definidos inductivamente, usando inducción tal como en los números naturales.

Ejercicio

Defina el operador $\#_R$ que entrega la cantidad de paréntesis derechos de una expresión en $\mathcal{E}_{\mathbb{N}}$, y demuestre que para toda $E \in \mathcal{E}_{\mathbb{N}}$, se cumple que $\#_L(E) = \#_R(E)$.

Desarrollo en Ejercicios Resueltos.

2. Lógica proposicional

¿Por qué necesitamos la Lógica?

Necesitamos un lenguaje con una sintaxis precisa y una semántica bien definida.

Queremos usar este lenguaje en matemáticas.

- Definición de objetos matemáticos: conjunto, números naturales, números reales.
- Definición de teorías matemáticas: teoría de conjuntos, teoría de los número naturales.
- Definición del concepto de demostración.

También queremos usar este lenguaje en computación. ¿Por qué?

¿Por qué necesitamos la Lógica en computación?

Algunas aplicaciones:

- Bases de datos: Lenguajes de consulta, lenguajes para restricciones de integridad.
- Inteligencia artificial: Representación de conocimiento, razonamiento con sentido común.
- Ingeniería de software: Especificación de sistemas (lenguaje Z), verificación de propiedades.
- Teoría de la computación: complejidad descriptiva, algoritmos de aproximación.
- Criptografía: verificación de protocolos criptográficos.
- Procesamiento de lenguaje natural.
- ...

Lógica Proposicional: Sintaxis

Tenemos los siguientes elementos:

- Variables proposicionales (P): $p, q, r, \ldots$
- Conectivos lógicos: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$
- Símbolos de puntuación: (,)

Cada variable proposicional representa una proposición **completa** e **indivisible**, que puede ser **verdadera** o **falsa**.

Ejemplo

$$P = \{socrates\_es\_hombre,\ socrates\_es\_mortal\}.$$

Lógica Proposicional: Sintaxis

Conectivos lógicos son usados para construir expresiones que también pueden ser verdaderas o falsas.

Ejemplo

$$socrates\_es\_hombre \rightarrow socrates\_es\_mortal$$

$$socrates\_es\_hombre \rightarrow (\neg\, socrates\_es\_mortal)$$

Símbolos de puntuación son usados para evitar ambigüedades.

Sintaxis de la Lógica Proposicional: Definición

Dado: Conjunto P de variables proposicionales.

Definición

$L(P)$ es el menor conjunto que satisface las siguientes reglas:

1. $P \subseteq L(P)$.
2. *Si* $\varphi \in L(P)$, *entonces* $(\neg\varphi) \in L(P)$.
3. *Si* $\varphi, \psi \in L(P)$, *entonces* $(\varphi \vee \psi) \in L(P)$, $(\varphi \wedge \psi) \in L(P)$, $(\varphi \rightarrow \psi) \in L(P)$ *y* $(\varphi \leftrightarrow \psi) \in L(P)$.

Ejercicio

Verifique que $((\neg p) \rightarrow (q \vee r))$ es una fórmula.

Sintaxis de la Lógica Proposicional: Definición

La naturaleza de la definición es inductiva.

- Permite construir programas recursivos para chequear si una fórmula está bien construida.
- Permite definir inductivamente conceptos asociados a las fórmulas.
- Permite demostrar inductivamente propiedades de las fórmulas.

Inducción en la lógica proposicional: Ejercicios

1. Defina $v(\varphi)$ como el número de ocurrencias de variables proposicionales en φ.

2. Demuestre que para cada fórmula proposicional φ que no contiene el símbolo $\neg$ se tiene que $la(\varphi) \leq 4 \cdot v(\varphi)^2$.

   ¿Qué sucede si φ contiene el símbolo $\neg$?

   ¿Qué sucede si las fórmulas de la forma $(\neg(\neg\varphi))$ no son permitidas?

3. Demuestre que un prefijo propio de una fórmula no es una fórmula.

Semántica de la lógica proposicional

¿Cómo podemos determinar si una fórmula es verdadera o falsa?

Este valor de verdad depende de los valores de verdad asignados a las variables proposicionales y de los conectivos utilizados.

Valuación (asignación): $\sigma : P \to \{0, 1\}$.

Ejemplo

$\sigma(socrates\_es\_hombre) = 1$ y $\sigma(socrates\_es\_mortal) = 0$.

Semántica: Definición

Dado $\sigma : P \to \{0,1\}$, queremos extender σ:

$$\hat{\sigma} : L(P) \to \{0,1\}.$$

Definición

Dado $\varphi \in L(P)$,

- *Si $\varphi = p$, entonces $\hat{\sigma}(\varphi) := \sigma(p)$.*
- *Si $\varphi = (\neg\alpha)$, entonces*

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \textit{si } \hat{\sigma}(\alpha) = 0 \\ 0 & \textit{si } \hat{\sigma}(\alpha) = 1 \end{cases}$$

- *Si $\varphi = (\alpha \vee \beta)$, entonces*

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \textit{si } \hat{\sigma}(\alpha) = 1 \textit{ o } \hat{\sigma}(\beta) = 1 \\ 0 & \textit{si } \hat{\sigma}(\alpha) = 0 \textit{ y } \hat{\sigma}(\beta) = 0 \end{cases}$$

Semántica: Definición (continuación)

- Si $\varphi = (\alpha \wedge \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 1 \text{ y } \hat{\sigma}(\beta) = 1 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 0 \text{ o } \hat{\sigma}(\beta) = 0 \end{cases}$$

- Si $\varphi = (\alpha \rightarrow \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = 0 \text{ o } \hat{\sigma}(\beta) = 1 \\ 0 & \text{si } \hat{\sigma}(\alpha) = 1 \text{ y } \hat{\sigma}(\beta) = 0 \end{cases}$$

- Si $\varphi = (\alpha \leftrightarrow \beta)$, entonces

$$\hat{\sigma}(\varphi) = \begin{cases} 1 & \text{si } \hat{\sigma}(\alpha) = \hat{\sigma}(\beta) \\ 0 & \text{si } \hat{\sigma}(\alpha) \neq \hat{\sigma}(\beta) \end{cases}$$

Por simplicidad vamos a usar σ en lugar de $\hat{\sigma}$.

Semántica: Ejemplos

Supongamos que $\sigma(\mathit{socrates\_es\_hombre}) = 1$ y $\sigma(\mathit{socrates\_es\_mortal}) = 0$.

Entonces:

$$\sigma((\mathit{socrates\_es\_hombre} \rightarrow \mathit{socrates\_es\_mortal})) = 0$$

$$\sigma((((\mathit{socrates\_es\_hombre} \rightarrow \mathit{socrates\_es\_mortal}) \wedge \mathit{socrates\_es\_hombre}) \rightarrow \mathit{socrates\_es\_mortal})) = 1$$

Tablas de verdad

Cada fórmula se puede representar y analizar en una tabla de verdad.

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Ejercicio

Suponga que P contiene n variables. ¿Cuántas tablas de verdad distintas existen para $L(P)$?

Conectivos ternarios

Queremos definir el conectivo lógico: si p entonces q si no r.

p	q	r	si p entonces q si no r
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

¿Cómo se puede representar este conectivo usando $\neg$, $\wedge$ y $\rightarrow$?

Conectivos ternarios (continuación)

Solución: $(p \to q) \wedge ((\neg p) \to r)$.

p	q	r	si p entonces q si no r	$(p \to q) \wedge ((\neg p) \to r)$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

¿Por qué el conectivo es equivalente a la fórmula?

- Porque tienen la misma tabla de verdad

Satisfacción de una fórmula

Definición

Una fórmula φ es satisfacible si existe una valuación σ tal que $\sigma(\varphi) = 1$.

Ejemplo

Las siguientes fórmulas son satisfacibles:

$$(p \vee q) \rightarrow r$$

$$p \rightarrow \neg p$$

Las siguientes fórmulas no son satisfacibles:

$$p \wedge \neg p$$

$$(p \vee q) \leftrightarrow \neg(p \vee q)$$

Tautologías y contradicciones

Si una fórmula no es satisfacible, entonces decimos que un contradicción.

- $p \wedge \neg p$ es una contradicción

Definición

Una fórmula φ es una tautología si para toda valuación σ se tiene que $\sigma(\varphi) = 1$.

Ejemplo

Las siguientes fórmulas son tautologías:

$$p \vee \neg p$$

$$p \leftrightarrow p$$

Equivalencia de fórmulas

Definición

Dos fórmulas φ, ψ son equivalentes, denotado como $\varphi \equiv \psi$, si para toda valuación σ se tiene que $\sigma(\varphi) = \sigma(\psi)$.

Una definición alternativa de la noción de equivalencia:

φ, ψ son equivalentes si $\varphi \leftrightarrow \psi$ es una tautología.

Ejercicio

1. Demuestre que las definiciones anteriores coinciden.
2. Defina la noción de equivalencia usando tablas de verdad.
    - ¿Qué pasa si las fórmulas no usan las mismas variables? ¿Puede ocurrir esto?

Algunas equivalencias útiles

Ley de la doble negación:

$$\neg(\neg\varphi) \equiv \varphi$$

Leyes de De Morgan:

$$\begin{aligned} \neg(\varphi \wedge \psi) &\equiv (\neg\varphi) \vee (\neg\psi) \\ \neg(\varphi \vee \psi) &\equiv (\neg\varphi) \wedge (\neg\psi) \end{aligned}$$

Leyes de conmutatividad:

$$\begin{aligned} \varphi \wedge \psi &\equiv \psi \wedge \varphi \\ \varphi \vee \psi &\equiv \psi \vee \varphi \end{aligned}$$

Algunas equivalencias útiles

Leyes de asociatividad:

$$\begin{aligned}(\varphi \wedge \psi) \wedge \theta &\equiv \varphi \wedge (\psi \wedge \theta)\\ (\varphi \vee \psi) \vee \theta &\equiv \varphi \vee (\psi \vee \theta)\end{aligned}$$

Leyes de distributividad:

$$\begin{aligned}\varphi \wedge (\psi \vee \theta) &\equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)\\ \varphi \vee (\psi \wedge \theta) &\equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)\end{aligned}$$

Ley de implicancia:

$$\varphi \rightarrow \psi \equiv (\neg\varphi) \vee \psi$$

Ley de doble implicancia:

$$\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$$

Formas normales: DNF

Decimos que una fórmula φ está en forma normal disyuntiva (DNF) si φ es de la forma:

$$\bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} l_{i,j} \right),$$

donde cada $l_{i,j}$ es un literal, es decir, una variable proposicional o la negación de una variable proposicional.

Ejemplo

$(p \wedge q) \vee (\neg p \wedge r \wedge s)$

Formas normales: DNF

Teorema

Toda fórmula es equivalente a una fórmula en DNF.

Ya demostramos este teorema, ¿cierto?

Formas normales: CNF

Decimos que una fórmula φ está en forma normal conjuntiva (CNF) si φ es de la forma:

$$\bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{n_i} l_{i,j} \right),$$

donde cada $l_{i,j}$ es un literal.

Ejemplo

$(p \vee \neg q) \wedge (\neg p \vee \neg r \vee s) \wedge (\neg r \vee s)$

Formas normales: CNF

Teorema

Toda fórmula es equivalente a una fórmula en CNF.

Ejercicio

Haga dos demostraciones del teorema.

- En la primera sólo utilice las leyes de equivalencia. ¿Qué leyes necesita utilizar?
- En la segunda utilice el resultado de que toda fórmula es equivalente a una fórmula en DNF. ¿Qué leyes de equivalencia necesita utilizar en este caso?

Conectivos funcionalmente completos

Definición

Un conjunto de conectivos es funcionalmente completo si es posible definir cada fórmula usando sólo estos conectivos.

Ya demostramos que $\{\neg, \vee, \wedge\}$ es funcionalmente completo. ¿Son $\{\neg, \vee\}$ y $\{\neg, \wedge\}$ funcionalmente completos?

Ejercicio

1. Demuestre que $\{\neg, \rightarrow\}$ es funcionalmente completo.
2. Demuestre que $\{\neg\}$ no es funcionalmente completo.
3. ¿Es $\{\wedge, \vee, \rightarrow, \leftrightarrow\}$ funcionalmente completo?

La noción de consecuencia lógica

Una valuación σ satisface un conjunto de fórmulas Σ si para cada $\varphi \in \Sigma$, se tiene que $\sigma(\varphi) = 1$.

Notación: $\sigma(\Sigma) = 1$.

¿Cuándo decimos que una fórmula ψ se deduce desde Σ?

Definición

ψ *es consecuencia lógica de* Σ *si para cada valuación* σ *tal que* $\sigma(\Sigma) = 1$, *se tiene que* $\sigma(\psi) = 1$.

Notación: $\Sigma \models \psi$

La noción de consecuencia lógica: Ejemplos

Modus ponens:

$$\{p,\ p \to q\} \models q$$

Demostración por partes:

$$\{p \vee q \vee r,\ p \to s,\ q \to s,\ r \to s\} \models s$$

Ejercicio

1. Demuestre que si $\Sigma \models \alpha \wedge \beta$, entonces $\Sigma \models \alpha$ y $\Sigma \models \beta$.
2. ¿Es cierto que si $\Sigma \models \alpha \vee \beta$, entonces $\Sigma \models \alpha$ o $\Sigma \models \beta$?

3. Lógica de Primer Orden

Lógica de primer orden

Dos de los objetivos de la lógica proposicional:

- Poder modelar el proceso de razonamiento.
- Poder formalizar la noción de demostración.

¿Podemos expresar el siguiente argumento en lógica proposicional?

$$\begin{array}{l} \text{Todos los hombres son mortales.} \\ \text{Sócrates es hombre.} \\ \hline \text{Por lo tanto, Sócrates es mortal.} \end{array}$$

¿Podemos demostrar que para el conjunto de los números naturales es cierto que todo número es par o impar?

Lógica de primer orden

El poder expresivo de la lógica proposicional es limitado.

- ¿Por qué usamos esta lógica?

Vamos a introducir una lógica más expresiva.

- Tiene algunas de las buenas propiedades de la lógica proposicional, pero no todas.

Para expresar el argumento mostrado al principio necesitamos cuantificadores: para todo y existe.

Lógica de primer orden: Vocabulario

Una fórmula en lógica de primer orden está definida sobre algunas constantes, funciones y predicados.

Notación

Un vocabulario $\mathcal{L}$ es la unión de tres conjuntos:

$$\begin{array}{lcl} \textit{constantes} & : & \{c_1, \ldots, c_\ell, \ldots\}, \\ \textit{funciones} & : & \{f_1, \ldots, f_m, \ldots\}, \\ \textit{relaciones} & : & \{R_1, \ldots, R_n, \ldots\}. \end{array}$$

Notación

La aridad de una función f (relación R) es el número de argumentos de f (de R).

- *Cada función tiene una aridad mayor a 0.*
- *Cada relación tiene una aridad mayor o igual a 0.*

Lógica de primer orden: Vocabulario

Ejemplo

Para los números naturales $\mathcal{L}$ es la unión de

$$\begin{array}{lcl} \text{constantes} & : & \{0, 1\}, \\ \text{funciones} & : & \{s, +, \cdot\}, \\ \text{relaciones} & : & \{<\}. \end{array}$$

s es una función unaria, $+$ y $\cdot$ son funciones binarias y $<$ es una relación binaria.

Lógica de primer orden: Sintaxis

Las fórmulas de la lógica de primer orden se construyen usando:

- Conectivos lógicos: $\neg$, $\vee$, $\wedge$, $\rightarrow$ y $\leftrightarrow$.
- Paréntesis: (y).
- Relación binaria $=$.
- Variables.
- Cuantificadores: $\forall$ y $\exists$.

Veamos algunos ejemplos, antes de introducir formalmente la sintaxis de la lógica de primer orden.

Sintaxis de la lógica de primer orden: Ejemplos

Ejemplo

Sea $\mathcal{L} = \{0,\, 1,\, s,\, +,\, \cdot,\, <\}$.

- $1 = s(0)$.
  Para la igualdad usamos notación infija: No escribimos $= (1, s(0))$.
- $\forall x\ x < s(x)$.
  Usamos notación infija para funciones y relaciones comunes.
- $\forall x \exists y\ x = y + y$.
- $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$.

Sintaxis de la lógica de primer orden: Términos

Desde ahora en adelante: Suponemos dada una lista infinita de variables.

Definición

El conjunto de $\mathcal{L}$-términos es el menor conjunto que satisface las siguientes condiciones:

- *Cada constante c en $\mathcal{L}$ es un $\mathcal{L}$-término.*
- *Cada variable x es un $\mathcal{L}$-término.*
- *Si $t_1, \ldots, t_n$ son $\mathcal{L}$-términos y f es una función n-aria en $\mathcal{L}$, entonces $f(t_1, \ldots, t_n)$ es un $\mathcal{L}$-término.*

Ejemplos

0, $s(s(s(1)))$ y $s(0) \cdot s(x)$

Sintaxis de la lógica de primer orden: Fórmulas

Definición

El conjunto de $\mathcal{L}$-fórmulas es el menor conjunto que satisface las siguientes condiciones:

- *Si t_1 y t_2 son $\mathcal{L}$-términos, entonces $t_1 = t_2$ es una $\mathcal{L}$-fórmula.*
- *Si $t_1, \ldots, t_n$ son $\mathcal{L}$-términos y R es una relación n-aria en $\mathcal{L}$, entonces $R(t_1, \ldots, t_n)$ es una $\mathcal{L}$-fórmula.*
- *Si φ y ψ son $\mathcal{L}$-fórmulas, entonces $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$ y $(\varphi \leftrightarrow \psi)$ son $\mathcal{L}$-fórmulas.*
- *Si φ es una $\mathcal{L}$-fórmula y x es una variable, entonces $(\exists x\ \varphi)$ y $(\forall x\ \varphi)$ son $\mathcal{L}$-fórmulas.*

Notación

$t_1 = t_2$ *y* $R(t_1, \ldots, t_n)$ *son llamadas fórmulas atómicas.*

Inducción en la lógica de primer orden

Principio de inducción: Para cada subconjunto A del conjunto de $\mathcal{L}$-fórmulas tal que

Caso base	:	para cada par t_1, t_2 de $\mathcal{L}$-términos, $t_1 = t_2 \in A$
	:	para cada predicado n-ario R y cada secuencia de $\mathcal{L}$-términos $t_1, \ldots, t_n$, $R(t_1, \ldots, t_n) \in A$
Caso inductivo	:	si $\varphi, \psi \in A$, entonces $(\neg\varphi) \in A$ y $(\varphi \star \psi) \in A$, donde $\star \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$
	:	si $\varphi \in A$ y x es una variable, entonces $(\exists x\, \varphi) \in A$ y $(\forall x\, \varphi) \in A$

se tiene que A es igual al conjunto de $\mathcal{L}$-fórmulas

Lógica de primer orden: Semántica

Notación

Omitimos paréntesis si no se produce una ambigüedad.

¿Es $\forall x \exists y \; x = y + y$ cierta en $\mathcal{L} = \{0, 1, s, +, \cdot, <\}$?

- Si pensamos en los números naturales es falsa.
- Pero $\mathcal{L}$ también puede usarse como vocabulario para los números reales, y en este conjunto la fórmula es cierta.

El valor de verdad de una fórmula depende de la **interpretación que se da a las constantes, funciones y relaciones.**

- Tenemos que introducir la noción de estructura.

Semántica de la lógica de primer orden: Estructuras

Una $\mathcal{L}$-estructura interpreta todos los componentes de $\mathcal{L}$ en un dominio.

Definición

Una $\mathcal{L}$-estructura $\mathfrak{A}$ contiene:

- *Un dominio A no vacío.*
- *Para cada constante $c \in \mathcal{L}$, una interpretación $c^{\mathfrak{A}} \in A$ de c.*
- *Para cada función m-aria $f \in \mathcal{L}$, una interpretación $f^{\mathfrak{A}} : A^m \to A$ de f.*
- *Para cada relación n-aria $R \in \mathcal{L}$, una interpretación $R^{\mathfrak{A}} \subseteq A^n$ de R.*

Notación

$\mathfrak{A} = \langle A, c^{\mathfrak{A}}, \ldots, f^{\mathfrak{A}}, \ldots, R^{\mathfrak{A}}, \ldots \rangle$

Algunos ejemplos de estructuras

Ejemplo

Los números naturales son representados por la estructura:

$$\mathfrak{N} \;=\; \langle \mathbb{N}, 0^{\mathfrak{N}}, 1^{\mathfrak{N}}, s^{\mathfrak{N}}, +^{\mathfrak{N}}, \cdot^{\mathfrak{N}}, <^{\mathfrak{N}} \rangle.$$

Los números reales son representados por la estructura:

$$\mathfrak{R} \;=\; \langle \mathbb{R}, 0^{\mathfrak{R}}, 1^{\mathfrak{R}}, s^{\mathfrak{R}}, +^{\mathfrak{R}}, \cdot^{\mathfrak{R}}, <^{\mathfrak{R}} \rangle.$$

Ahora podemos decir que $\mathfrak{N}$ no satisface $\forall x \exists y \; x = y + y$ y que $\mathfrak{R}$ si satisface esta fórmula.

Semántica de la lógica de primer orden: Variables libres

Necesitamos introducir la noción de variable libre.

El conjunto de variables de un $\mathcal{L}$-término t se define como:

- Si $t = c$ es una constante, entonces $V(t) = \emptyset$.
- Si $t = x$ es una variable, entonces $V(t) = \{x\}$.
- Si $t = f(t_1, \ldots, t_n)$, entonces $V(t) = V(t_1) \cup \cdots \cup V(t_n)$.

Ejemplo

$$\begin{aligned} V(f(g(x,y), s(0))) &= V(g(x,y)) \cup V(s(0)) \\ &= V(x) \cup V(y) \cup V(0) \\ &= \{x\} \cup \{y\} \cup \emptyset \\ &= \{x, y\} \end{aligned}$$

Semántica de la lógica de primer orden: Variables libres

El conjunto de variables de una $\mathcal{L}$-fórmula φ se define como:

- Si $\varphi \; = \; t_1 = t_2$, entonces $V(\varphi) = V(t_1) \cup V(t_2)$.
- Si $\varphi = R(t_1, \ldots, t_n)$, entonces $V(\varphi) = V(t_1) \cup \cdots \cup V(t_n)$.
- Si $\varphi = (\neg\psi)$, entonces $V(\varphi) = V(\psi)$.
- Si $\varphi = (\psi \star \theta)$ ($\star \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$), entonces $V(\varphi) = V(\psi) \cup V(\theta)$.
- Si $\varphi = (\exists x \; \psi)$ o $\varphi = (\forall x \; \psi)$, entonces $V(\varphi) = \{x\} \cup V(\psi)$.

Ejemplo

$$\begin{aligned}
V((\exists x \; P(x)) \vee (\forall y \; Q(s(y)))) &= V(\exists x \; P(x)) \cup V(\forall y \; Q(s(y))) \\
&= \{x\} \cup V(P(x)) \cup \{y\} \cup V(Q(s(y))) \\
&= \{x\} \cup V(x) \cup \{y\} \cup V(s(y)) \\
&= \{x\} \cup \{x\} \cup \{y\} \cup V(y) \\
&= \{x, y\}
\end{aligned}$$

Semántica de la lógica de primer orden: Variables libres

Definición

El conjunto de variables libres de una $\mathcal{L}$-fórmula φ se define como:

- *Si φ es una fórmula atómica, entonces $VL(\varphi) = V(\varphi)$.*
- *Si $\varphi = (\neg\psi)$, entonces $VL(\varphi) = VL(\psi)$.*
- *Si $\varphi = (\psi \star \theta)$ ($\star \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$), entonces $VL(\varphi) = VL(\psi) \cup VL(\theta)$.*
- *Si $\varphi = (\exists x\ \psi)$ o $\varphi = (\forall x\ \psi)$, entonces $VL(\varphi) = VL(\psi) \setminus \{x\}$.*

Variable libre: No aparece cuantificada.

Semántica de la lógica de primer orden: Variables libres

Ejemplo

$$\begin{aligned} VL(P(x) \wedge \exists y\ Q(x, y)) &= \{x\} \\ VL(P(z) \wedge \exists z\ R(z)) &= \{z\} \end{aligned}$$

Notación

- *Si* φ *es una fórmula, entonces usamos* $\varphi(x_1, \ldots, x_k)$ *para indicar que* $VL(\varphi) = \{x_1, \ldots, x_k\}$.
- *Decimos que* φ *es una oración si* $VL(\varphi) = \emptyset$.

Semántica de la lógica de primer orden: Definición

Si una fórmula contiene variables libres, entonces no podemos decir directamente que es verdadera o falsa en una estructura.

- ¿Es $x < s(0)$ cierta en $\mathfrak{N}$?

El valor de verdad de una fórmula con variables libres depende de los valores dados a estas variables.

- Si x es 0, entonces $x < s(0)$ es cierta en $\mathfrak{N}$. Pero si x es 1, entonces es falsa.

Semántica de la lógica de primer orden: Definición

Dada una estructura $\mathfrak{A}$ con dominio A, una asignación σ es una función que asigna a cada variable un valor en A.

Extendemos σ para dar valores a los términos:

- Si $t = c$ es una constante, entonces $\hat{\sigma}(t) = c^{\mathfrak{A}}$.
- Si $t = x$ es una variable, entonces $\hat{\sigma}(t) = \sigma(x)$.
- Si $t = f(t_1, \ldots, t_n)$, entonces $\hat{\sigma}(t) = f^{\mathfrak{A}}(\hat{\sigma}(t_1), \ldots, \hat{\sigma}(t_n))$.

Semántica de la lógica de primer orden: Definición

Ejemplo

Si $\sigma(x) = 7$ es una asignación para $\mathfrak{N}$, entonces

$$\begin{aligned}
\hat{\sigma}(s(1) \cdot s(x)) &= \hat{\sigma}(s(1)) \cdot^{\mathfrak{N}} \hat{\sigma}(s(x)) \\
&= s^{\mathfrak{N}}(\hat{\sigma}(1)) \cdot^{\mathfrak{N}} s^{\mathfrak{N}}(\hat{\sigma}(x)) \\
&= s^{\mathfrak{N}}(1^{\mathfrak{N}}) \cdot^{\mathfrak{N}} s^{\mathfrak{N}}(\sigma(x)) \\
&= 2 \cdot^{\mathfrak{N}} s^{\mathfrak{N}}(7) \\
&= 2 \cdot^{\mathfrak{N}} 8 \\
&= 16
\end{aligned}$$

Por simplicidad, usamos σ en lugar de $\hat{\sigma}$.

Semántica de la lógica de primer orden: Definición

Vamos a definir la semántica de la lógica de primer orden.

Dado: Un vocabulario $\mathcal{L}$, una $\mathcal{L}$-estructura $\mathfrak{A}$ con dominio A y una asignación σ para $\mathfrak{A}$.

Definición

Decimos que $(\mathfrak{A}, \sigma)$ *satisface una* $\mathcal{L}$*-fórmula* φ*, denotado como* $(\mathfrak{A}, \sigma) \models \varphi$*, si y sólo si:*

- $\varphi \;=\; t_1 = t_2$ *y* $\sigma(t_1) = \sigma(t_2)$
- $\varphi \;=\; R(t_1, \ldots, t_n)$ *y* $(\sigma(t_1), \ldots, \sigma(t_n)) \in R^{\mathfrak{A}}$
- $\varphi \;=\; (\neg\psi)$ *y no es cierto que* $(\mathfrak{A}, \sigma) \models \psi$
- $\varphi \;=\; (\psi \vee \theta)$*, y* $(\mathfrak{A}, \sigma) \models \psi$ *o* $(\mathfrak{A}, \sigma) \models \theta$

Semántica de la lógica de primer orden: Definición

- $\varphi \ = \ (\psi \wedge \theta)$, $(\mathfrak{A}, \sigma) \models \psi$ y $(\mathfrak{A}, \sigma) \models \theta$
- $\varphi \ = \ (\psi \rightarrow \theta)$, y $(\mathfrak{A}, \sigma) \not\models \psi$ o $(\mathfrak{A}, \sigma) \models \theta$
- $\varphi \ = \ (\psi \leftrightarrow \theta)$, y ambos $(\mathfrak{A}, \sigma) \models \psi$, $(\mathfrak{A}, \sigma) \models \theta$, o ambos $(\mathfrak{A}, \sigma) \not\models \psi$, $(\mathfrak{A}, \sigma) \not\models \theta$
- $\varphi \ = \ (\exists x \ \psi)$ y existe $a \in A$ tal que $(\mathfrak{A}, \sigma[x/a]) \models \psi$, donde

$$\sigma[x/a](y) = \begin{cases} a & y = x \\ \sigma(y) & y \neq x \end{cases}$$

- $\varphi \ = \ (\forall x \ \psi)$ y para todo $a \in A$ se tiene que $(\mathfrak{A}, \sigma[x/a]) \models \psi$

Nota: Si φ es una oración, podemos decir que $\mathfrak{A} \models \varphi$.

Fórmulas satisfacibles

Decimos que una $\mathcal{L}$-fórmula φ es satisfacible si existe una $\mathcal{L}$-estructura $\mathfrak{A}$ y una asignación σ para $\mathfrak{A}$ tal que $(\mathfrak{A}, \sigma) \models \varphi$.

- Si φ es oración, entonces φ es satisfacible si existe $\mathfrak{A}$ tal que $\mathfrak{A} \models \varphi$.

Si una fórmula no es satisfacible, entonces decimos que es una contradicción.

Ejercicio

Construya fórmulas satisfacibles y otras contradictorias.

Fórmulas válidas

Decimos que una $\mathcal{L}$-fórmula φ es válida si para toda $\mathcal{L}$-estructura $\mathfrak{A}$ y toda asignación σ para $\mathfrak{A}$ se tiene que $(\mathfrak{A}, \sigma) \models \varphi$.

- Si φ es oración, entonces φ es válida si para todo $\mathfrak{A}$ se tiene que $\mathfrak{A} \models \varphi$.

Ejercicio

Construya fórmulas válidas.

Equivalencia de fórmulas

Definición

Dos $\mathcal{L}$-fórmulas φ, ψ son equivalentes, denotado como $\varphi \equiv \psi$, si para toda $\mathcal{L}$-estructura $\mathfrak{A}$ y todo asignación σ para $\mathfrak{A}$, se tiene que:

$$(\mathfrak{A}, \sigma) \models \varphi \quad \textit{si y sólo si} \quad (\mathfrak{A}, \sigma) \models \psi$$

Si en la definición anterior φ, ψ son oraciones, entonces son equivalentes si para toda $\mathcal{L}$-estructura $\mathfrak{A}$:

$$\mathfrak{A} \models \varphi \quad \text{si y sólo si} \quad \mathfrak{A} \models \psi$$

Algunas equivalencias útiles

Todas las equivalencias para la lógica proposicional siguen siendo válidas en este contexto.

- doble negación, leyes de De Morgan, conmutatividad de $\wedge$ y $\vee$, asociatividad de $\wedge$ y $\vee$, distributividad de $\wedge$ sobre $\vee$ y de $\vee$ sobre $\wedge$, implicancia, doble implicancia, ...

Tenemos nuevas equivalencias útiles:

$$
\begin{aligned}
\forall x\, \varphi &\equiv \neg(\exists x\, \neg\varphi) \\
\exists x\, \varphi &\equiv \neg(\forall x\, \neg\varphi) \\
\forall x\, (\varphi \wedge \psi) &\equiv (\forall x\, \varphi) \wedge (\forall x\, \psi) \\
\exists x\, (\varphi \vee \psi) &\equiv (\exists x\, \varphi) \vee (\exists x\, \psi)
\end{aligned}
$$

La noción de consecuencia lógica

Sea Σ un conjunto de $\mathcal{L}$-oraciones. Una estructura $\mathfrak{A}$ satisface Σ si para cada $\varphi \in \Sigma$ se tiene que $\mathfrak{A} \models \varphi$.

Notación: $\mathfrak{A} \models \Sigma$

Definición

Una $\mathcal{L}$-oración φ es consecuencia lógica de un conjunto Σ de $\mathcal{L}$-oraciones si para cada $\mathcal{L}$-estructura $\mathfrak{A}$:

$$\text{si } \mathfrak{A} \models \Sigma, \text{ entonces } \mathfrak{A} \models \varphi$$

Notación: $\Sigma \models \varphi$

4. Teoría de Conjuntos

- Un *conjunto* es una colección de *elementos*. Un elemento puede *pertenecer* a un conjunto o no.
- Un conjunto A puede definirse por *extensión*, es decir, enumerando sus elementos explícitamente.

Ejemplo

Definimos el siguiente conjunto por extensión:

$$A = \{1, 2, 3\}.$$

- Un conjunto se puede definir por *comprensión*.

Notación: Definimos un conjunto A por comprensión de la siguiente forma:

$$A = \{x \in U \mid P(x)\}.$$

Léase: A es el conjunto formado por los elementos x en U, tales que $P(x)$. Aqui, U representa un conjunto universo.

Entonces, dada la definición anterior, la siguiente proposición es verdadera:

$$\forall x(x \in A \leftrightarrow P(x)).$$

Ejemplo

$$A = \{x \in \mathbb{N} \mid \exists m \in \mathbb{N}(2m = x)\}$$

Definición

Dados dos conjuntos A y B, A es **subconjunto** de B ($A \subseteq B$) si y sólo si:

$$\forall x(x \in A \to x \in B).$$

Ejercicio

Demuestre que $A \subseteq B$, dado lo siguiente:

$$A = \{m \in \mathbb{Z} \mid \exists r \in \mathbb{Z}(m = 6r + 12)\},$$
$$B = \{n \in \mathbb{Z} \mid \exists s \in \mathbb{Z}(n = 3s)\},$$

Definición

Dos conjuntos A y B son **iguales** si y sólo si $A \subseteq B$ y $B \subseteq A$. Se escribe $A = B$.

Pregunta: ¿Qué propiedades tiene la siguiente proposición?

$$A = B \leftrightarrow (A \subseteq B) \wedge (A \subseteq B).$$

Ejercicio

Demuestre que los siguientes conjuntos son iguales:

$$A = \{m \in \mathbb{Z} \mid \exists a \in \mathbb{Z}(m = 2a)\},$$
$$B = \{n \in \mathbb{Z} \mid \exists b \in \mathbb{Z}(n = 2b - 2)\},$$

Definición

Definimos el **conjunto vacío** como el conjunto que no tiene elementos. Por extensión:

$$\emptyset = \{\}.$$

Ejercicio

- Dado un conjunto A, determine si $\emptyset \subseteq A$.
- Dado un conjunto A, determine si $A \subseteq A$.

Operaciones sobre conjuntos

Definición

Dados dos conjuntos A y B, definimos los siguientes conjuntos:

- El conjunto unión:

$$A \cup B = \{x \mid x \in A \vee x \in B\},$$

- El conjunto intersección:

$$A \cap B = \{x \mid x \in A \wedge x \in B\},$$

- El conjunto diferencia:

$$A - B = A \backslash B = \{x \mid x \in A \wedge x \notin B\}.$$

Operaciones sobre conjuntos

Definición

Dados dos conjuntos A y B, definimos el siguiente conjunto:

- El conjunto potencia:

$$\mathcal{P}(A) = \{x \mid x \subseteq A\}.$$

Ejercicio

Dado $A = \{1, \{2, 3\}, 4\}$, escriba $P(A)$.

Definición

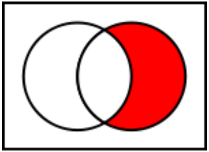
Un **universo** U es un conjunto dentro del cual enmarcamos una discusión. En este caso, se asume que todos los elementos pertenecen a U.

Definición

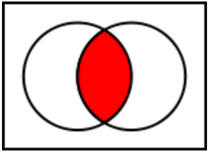
Dado un universo U y un conjunto A (entonces $A \subseteq U$), definimos el **conjunto complemento** de A:

$$A^c = \{x \in U \mid x \notin A\},$$

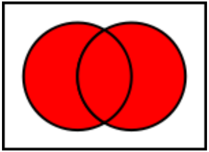
Diagramas de Venn



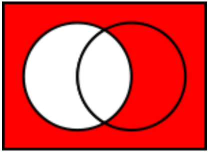
$B - A$



$A \cap B$



$A \cup B$



A^c

A continuacion vemos una serie de propiedades de las operaciones entre conjuntos:

Teorema

Dado un universo U, tenemos la ley del doble complemento:

$$(A^c)^c = A.$$

Teorema

Las leyes de Morgan:

$$(A \cup B)^c = A^c \cap B^c,$$
$$(A \cap B)^c = A^c \cup B^c.$$

Teorema

Las leyes de conmutatividad:

$$A \cup B = B \cup A,$$
$$A \cap B = B \cap A.$$

Teorema

Las leyes de asociatividad:

$$A \cup (B \cup C) = (A \cup B) \cup C,$$
$$A \cap (B \cap C) = (A \cap B) \cap C.$$

Teorema

Las leyes de distributividad:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C),$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$$

Teorema

Las leyes de idempotencia:

$$A \cup A = A,$$
$$A \cap A = A.$$

Teorema

Las leyes de elemento neutro:

$$A \cup \emptyset = A,$$
$$A \cap U = A.$$

Teorema

Las leyes de elemento inverso:

$$A \cup A^c = U,$$
$$A \cap A^c = \emptyset.$$

Teorema

Las leyes de dominación:

$$A \cup U = U,$$
$$A \cap \emptyset = \emptyset.$$

Teorema

Las leyes de absorción:

$$A \cup (A \cap B) = A,$$
$$A \cap (A \cup B) = A.$$

Definición

Dos conjuntos A y B se dicen **disjuntos** ssi $A \cap B = \emptyset$.

Definición

Los conjuntos $A_1, \ldots, A_n$ se dicen **mutuamente disjuntos** ssi:

$$A_i \cap A_j = \emptyset \quad \forall i \neq j.$$

Definición

Sea $P = \{A_1, \ldots, A_n\}$ un conjunto de conjuntos no-vacíos y sea A un conjunto cualquiera. P es una **partición** A ssi:

- $A_1, \ldots, A_n$ son mutuamente disjuntos,
- $A = \bigcup_{i=1}^{n} A_i$.

5. Relaciones

Par ordenado

Queremos definir un *par ordenado* de la forma (a, b). Dos pares ordenados (a, b) y (c, d) deberan ser iguales ssi $(a = c) \wedge (b = d)$.

Definición

Un **par ordenado** (a, b) es un conjunto:

$$(a, b) = \{\{a\}, \{a, b\}\}.$$

Teorema

Dos pares ordenados $p = (a, b)$ y $q = (c, d)$ son iguales ssi $(a = c) \wedge (b = d)$

Ejercicio

Si definimos un par ordenado de la siguiente forma:
$(a, b) = \{a, \{b\}\}$, muestre que no se cumple el teorema anterior.

Definición

Una **n-tupla** se define de la siguiente forma:

$$(a_1, a_2, \ldots, a_n) = ((a_1, a_2), a_3), \ldots), a_n).$$

Definición

Dados los conjuntos A y B, se define el **producto cartesiano** como el siguiente conjunto:

$$A \times B = \{(a, b) \mid (a \in A) \wedge (b \in B)\}.$$

Definición

Dados los conjuntos $A_1, \ldots, A_n$, se define el **producto cartesiano n-dimensional**:

$$A_1 \times \ldots \times A_n = \{(a_1, \ldots, a_n) \mid (a_1 \in A_1) \wedge \ldots \wedge (a_n \in A_n)\}.$$

Definición

Una **relación binaria** R sobre los conjuntos A y B es un subconjunto de $A \times B$.

Ejemplo

Considerando N y E del ejemplo anterior, si para cada alumno de este curso definimos el par $(nombre, edad)$, el conjunto de estos pares ordenados es una relación sobre N y E. Llamemos **ALUMNOS** a esta relación.

Definición

Dados los conjuntos $A_1, \ldots, A_n$, una **relación n-aria** R sobre $A_1, \ldots, A_n$ es un subconjunto de $A_1 \times \ldots \times A_n$.

Ejemplo

Podemos redefinir la operación $+$ como una relación 3-aria llamada R_+ sobre $\mathbb{N}, \mathbb{N}, \mathbb{N}$, de la siguiente forma. Sean $a, b, c \in \mathbb{N}$,

$$(a, b, c) \in R_+ \;\Leftrightarrow\; a + b = c \;\Leftrightarrow\; +(a, b) = c.$$

Asi, la relación R_+ contiene 3-tuplas como:

$$R_+ = \{(1, 0, 1), (3, 15, 18), (0, 109, 109), \ldots\},$$

y no contiene tuplas como:

$$(1, 0, 16), (5, 15, 18) \notin R_+.$$

Relaciones binarias

Dado: conjunto A

R es una relación binaria sobre A si $R \subseteq A \times A$.

- Para indicar que $a, b \in A$ están relacionados a través de R usamos las notaciones: $R(a, b)$ y aRb

Ejemplo

Si $A = \mathbb{N}$, las siguientes son relaciones binarias sobre A:

$$\begin{aligned} R_1 &= \{(i,j) \in \mathbb{N} \times \mathbb{N} \mid i = j\} \\ R_2 &= \{(i,j) \in \mathbb{N} \times \mathbb{N} \mid i < j\} \end{aligned}$$

En este capítulo sólo vamos a considerar relaciones binarias, usamos el termino relación para referirnos a ellas.

Propiedades de las relaciones

Definición

Una relación R sobre A es:

- *Refleja: Para cada $a \in A$, se tiene $R(a, a)$*
- *Irrefleja: Para cada $a \in A$, no se tiene $R(a, a)$*

Ejercicio

De ejemplos de relaciones reflejas e irreflejas sobre $\mathbb{N}$.

Propiedades de las relaciones

Definición

Una relación R sobre A es:

- *Simétrica:* *Para cada* $a, b \in A$, *si* $R(a, b)$ *entonces* $R(b, a)$
- *Asimétrica:* *Para cada* $a, b \in A$, *si* $R(a, b)$ *entonces no es cierto* $R(b, a)$
- *Antisimétrica:* *Para cada* $a, b \in A$, *si* $R(a, b)$ *y* $R(b, a)$, *entonces* $a = b$

Ejercicio

De ejemplos de relaciones simétricas, asimétricas y antisimétricas sobre $\mathbb{N}$.

Propiedades de las relaciones

Definición

Una relación R sobre A es:

- *Transitiva: Para cada $a, b, c \in A$, si $R(a, b)$ y $R(b, c)$, entonces $R(a, c)$*
- *Conexa: Para cada $a, b \in A$, se tiene $R(a, b)$ o $R(b, a)$*

Ejercicio

De ejemplos de relaciones transitivas y conexas sobre $\mathbb{N}$.

Relaciones de equivalencia

Definición

Una relación R sobre A es una relación de equivalencia si R es refleja, simétrica y transitiva.

Ejemplo

Sea $A = \mathbb{N} \times \mathbb{N}$ y $\sim$ una relación definida de la siguiente forma:

$$(a, b) \sim (c, d) \quad \Leftrightarrow \quad a + d = c + b$$

Demuestre que $\sim$ es una relación de equivalencia.

Clases de equivalencia

Definición

Dada una relación de equivalencia R sobre A y un elemento $b \in A$, la clase de equivalencia de b bajo R se define como:

$$[b]_R \quad = \quad \{c \in A \mid R(b, c)\}$$

Ejercicio

Suponga que $\sim$ es definida como en la transparencia anterior. Para cada $(a, b) \in A$, ¿que representa $[(a, b)]_\sim$?

Ordenes parciales y totales

Dado: Relación R sobre un conjunto A

Definición

R es un orden parcial sobre A si R es refleja, antisimétrica y transitiva. Si R es además conexa, entonces en un orden total sobre A.

Ejercicio

De ejemplos de ordenes parciales y totales.

6. Funciones y Cardinalidad

Recordando

Definición

Una relación $f \subseteq A \times B$ es llamada una función de A en B (denotada $f : A \to B$) si, dado $x \in A$ existe un único $y \in B$ tal que $(x, y) \in f$. (Denotamos por $f(x)$ al único $y \in B$ tal que $(x, y) \in f$).

Al conjunto A solemos llamarle dominio de f, al conjunto $\{f(x) : x \in A\}$ solemos llamarle el recorrido de f. Además, note que $f = g$ ssi $\forall x \in A : f(x) = g(x)$

Más recuerdos

Una función $f : A \to B$ es

- Inyectiva (1-1) si, dados $x, y \in A$,

$$f(x) = f(y) \to x = y.$$

- Epiyectiva (sobre) si dado $y \in B$ existe $x \in A$ tal que $f(x) = y$.
- Biyectiva si es inyectiva y epiyectiva.

$f : \mathbb{R} \to \mathbb{R}$ definida por $f(x) = e^x$ es inyectiva. $g : \mathbb{R} \to \mathbb{R}$ definida por $g(x) = x^3 + x^2$ es epiyectiva. $h : \mathbb{R} \to \mathbb{R}$ definida poe $h(x) = x^3$ es biyectiva.

Cardinalidad

Definición

Dos conjuntos A y B tienen igual **cardinalidad** (o bien son **equinumerosos**) si y sólo si existe una biyección entre ellos, i.e. existe una función $f : A \to B$ que es uno-a-uno y sobre.

Se define la relacion de **equinumerosidad** $\sim$ entre conjuntos de la siguiente forma: $A \sim B$ si y sólo si A y B tienen la misma cardinalidad.

Teorema

La relación $\sim$ es una relación de equivalencia.

Definición

Un conjunto A se dice **finito** si y sólo si existe un natural n tal que $A \sim n$.

Alternativamente, A se dice **finito** si y sólo si existe un $n \in \mathbb{N}$ tal que el conjunto $\{1, 2, \ldots, n\}$ es equinumeroso con A.

Definición

Un conjunto A que no es finito se dice **infinito**.

Definición

Un conjunto equinumeroso con $\mathbb{N}$ se dice **numerable**.

Ejemplo

El conjunto P de numeros naturales pares es numerable. Considere la función $f : \mathbb{N} \to P$:

$$f(n) = 2n$$

Se debe demostrar que f es una biyección.

Un teorema más general

Definición

Un conjunto A es menos numeroso que un conjunto B si:

- *existe una función inyectiva $f : A \to B$; y*
- *no existe una biyección $g : A \to B$*

Ejemplo

En las transparencias anteriores demostramos que $\mathbb{N}$ es menos numeroso que $\mathbb{R}$.

Teorema de Cantor

Sea A un conjunto cualquiera. No existe una biyección entre A y $\mathcal{P}(A)$.

Ejercicio

Demuestre el Teorema de Cantor.

7. Teoría de números y Criptografía

Divisibilidad

Definición

Dados dos números naturales, $d, n \in \mathbb{N}$, $d \mid n$ (d **divide** a n) si y sólo si:

$$\exists k \in \mathbb{N}(n = dk).$$

Teorema

Dado cualquier entero n y entero positivo d, existen enteros únicos q y r tales que

$$n = dq + r \ \wedge \ 0 \leq r < d.$$

A q se le dice el cuociente y a r se le dice el resto.

Definición

Sea $n \in \mathbb{Z}$ y $d \in \mathbb{N} - \{0\}$. Se define $n \,\%\, d$ o bien $n \ \mathbf{mod} \ d$ como el resto de la division entre n y d.

Ejemplo

$5 \,\%\, 3 = 2$ ya que $5 = 3k + 2$ (con $k = 1$).

Congruencia Módulo n

Definición

Sean $a, b, n \in \mathbb{Z}$ con $n > 1$. $a \equiv_{\textbf{mod } n} b$ (a y b **soncongruentes módulo** n) si y sólo si:

$$a \textbf{ mod } n = b \textbf{ mod } n.$$

Definición

En la definición anterior se habla de $\equiv_{\textbf{mod } n}$ como un operador. Ahora definimos un predicado asociado:

$$\equiv_{\textbf{mod } n} (a, b) \quad \Leftrightarrow \quad a \equiv_{\textbf{mod } n} b,$$

y una relación binaria, $\equiv_{\textbf{mod } n} \subseteq \mathbb{Z} \times \mathbb{Z}$:

$$(a, b) \in \equiv_{\textbf{mod } n} \quad \Leftrightarrow \quad a \equiv_{\textbf{mod } n} b,$$

Teorema

Sean $a, b, n \in \mathbb{Z}$ con $n > 1$. Las siguientes afirmaciones son equivalentes:

- $a \equiv_{\mathbf{mod}\ n} b$,
- $n \mid (a - b)$,
- $\exists k \in \mathbb{Z}(a = b + kn)$,
- a y b tienen el mismo resto r cuando son divididos por n.
- $a \ \mathbf{mod}\ n = b \ \mathbf{mod}\ n$.

Ejemplo

Sea $n = 7$. Se cumple, entonces:

$$\begin{aligned} 5 \equiv_{\mathbf{mod}\ n} 12 \quad &\Leftrightarrow \quad 7 \mid (5 - 12) \quad \Leftrightarrow \quad \exists k(5 = 12 + 7k) \\ &\Leftrightarrow \quad \text{5 y 12 tienen mismo resto al dividirse por 7} \\ &\Leftrightarrow \quad 5 \ \mathbf{mod}\ 7 = 12 \ \mathbf{mod}\ 7. \end{aligned}$$

Teorema

La congruencia módulo n, $\equiv_{\textbf{mod } n}$, es una relación de equivalencia.

Notación: Sea $n \in \mathbb{Z}$ con $n > 1$. Dado un $a \in \mathbb{Z}$, la clase de equivalencia de a se escribe $[a]_n$:

$$[a]_n = [a]_{\equiv_{\textbf{mod } n}},$$

es decir,

$$[a]_n = \{m \in \mathbb{Z} \mid m \equiv_{\textbf{mod } n} a\}.$$

Definición

Un entero d es una **combinación lineal** de los enteros a y b si y sólo si:

$$\exists s, t \in \mathbb{Z}\big(d = as + bt\big).$$

Definición

Un entero d es el **máximo comun divisor** de a y b ($d = \mathbf{mcd}(a, b)$) ssi:

- d es divisor de a y de b $\big(a \textbf{ mod } d = b \textbf{ mod } d = 0\big)$.
- $\forall c \in \mathbb{Z}$, si c es divisor de a y de b, entonces $c \leq d$.

Teorema

$$\forall a, b \in \mathbb{Z} - \{0\} \left(d = \mathbf{mcd}(a, b) \ \rightarrow \ \exists s, t \in \mathbb{Z}(d = as + bt) \right).$$

Con palabras: $\forall a, b$ enteros, distintos de cero, si $d = \mathbf{mcd}(a, b)$ entonces existen enteros s y t tales que $d = as + bt$.

Teorema

$$\forall a, b \in \mathbb{Z} - \{0\} \left(d = \mathbf{mcd}(a, b) \ \rightarrow \ \exists s, t \in \mathbb{Z}(d = as + bt) \right).$$

Finalmente, $c = d = \mathbf{mcd}(a, b) = as + bt$.

Lema

Dados $a, b \in \mathbb{Z}$ distintos de cero, y si q y r son enteros que cumplen $a = bq + r$, entonces:

$$\mathbf{mcd}(a, b) = \mathbf{mcd}(b, r).$$

Ejercicio

Estudiar la desmostración (Lema 4.8.2, Epp).

Cálculo de máximo común divisor

De lo anterior, concluimos la siguiente identidad:

$$\text{MCD}(a, b) = \begin{cases} a & b = 0 \\ \text{MCD}(b, a \bmod b) & b > 0 \end{cases}$$

Podemos usar esta identidad para generar un algoritmos recursivo para calcular el máximo común divisor.

- Este algoritmo puede ser extendido para calcular s y t tales que $\text{MCD}(a, b) = s \cdot a + t \cdot b$
- Vamos a usar este algoritmo para calcular los coeficiente d y e usados en RSA.

Inverso modular

Definición

b es inverso de a en módulo n si $a \cdot b \equiv 1 \mod n$

Para el caso de RSA: d es inverso de e en módulo $\phi(N)$

- En el ejemplo: 37 es inverso de 13 en módulo 60

¿Todo número tiene inverso modular?

- No: 2 no tiene inverso en módulo 4

¿Bajo qué condiciones a tiene inverso en módulo n?

- ¿Cómo podemos calcular este inverso?

Inverso modular: Existencia

Teorema

a tiene inverso en módulo n si y sólo si $MCD(a, n) = 1$

Criptografía

Es el estudio de metodos para enviar y recibir mensajes en privado.

Algoritmo

Cifrado Cesar (Caesar cipher): un ejemplo de algoritmo criptográfico.

Input: Letra $M \in \{0, \ldots, 26\}$
$C \leftarrow (M + 3)$ **mod** 26

Ejemplo

Dado el texto "`holamundo`", interpretado como 9 mensajes consecutivos, tendremos:

$$M[0..9] = \texttt{krodpxqgr}.$$

Notación: Aunque un mensaje consista en muchos mensajes consecutivos, se denotará el conjunto como un solo mensaje: M.

En criptografía se habla de un agente A que desea comuncarse con un agente B. Usualmente a A se le conoce tambien como *Alice* y a B como *Bob*.

Ejemplo

Alice desea enviar el mensaje $M = \texttt{holamundo}$ a Bob. Para eso, Alice selecciona el numero $d = 3$, tal que:

$$C = (M + d) \textbf{ mod } 26.$$

Con eso, Alice obtiene el texto $C = \texttt{krodpxqgr}$.

Nota: Aquí se abusa de la notación, usando M y C tanto para cada letra individual como para el texto completo.

Definiciones informales

Considerando el ejemplo anterior, definimos lo siguiente:

- Al texto M se le conoce como *texto plano* (*plaintext*).
- Al texto C se le conoce como *texto cifrado* (*ciphertext*).
- A la variable d se le conoce como *llave* (*key*).

Ejemplo (continuación)

Bob recibe el texto cifrado $C = \texttt{krodpxqgr}$.

$$C = (M + d) \ \mathbf{mod}\ 26.$$

Con eso, Alice obtiene el texto $C = \texttt{krodpxqgr}$

- Bob recibe el texto cifrado $C = \texttt{krodpxqgr}$. Que necesita para recuperar el texto plano?

Ejemplo (continuación)

Bob recibe el texto cifrado $C = \texttt{krodpxqgr}$. Usa la llave d para obtener el mensaje original:

$$M = (C - d) \textbf{ mod } 26.$$

Con eso, Alice obtiene el texto $C = \texttt{krodpxqgr}$

Ejercicio

Demuestre que para cualquier mensaje cifrado $C = (M + d) \textbf{ mod } 26$, el mesaje recuperado $M' = (C - d) \textbf{ mod } 26$ coincide con el mensaje original: $M' = M$.

- Como puede Alice hacerle llegar a Bob la llave?

Definición informal

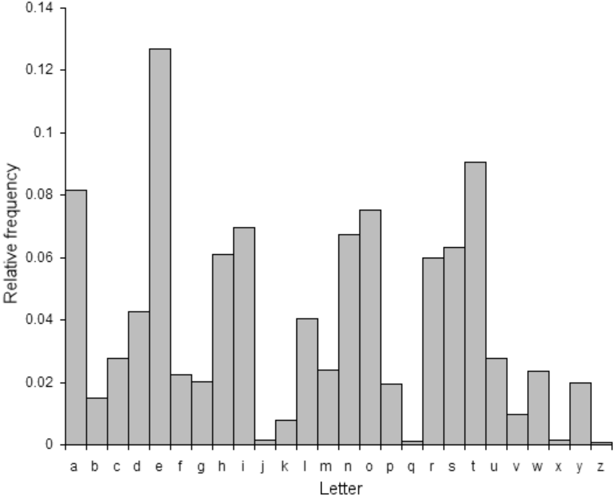
Una **llave pública** es una llave que debe ser compartida para ser usada.

Ademas de Alice y Bob, introducimos un tercer agente al problema: *Eve* (del inglés *eavesdropper*, que es una persona que escucha una conversación ajena).

En principio, debemos suponer que Eve tiene acceso a toda la información que es compartida pubilcamente. En el caso del cifrado de Cesar, esto incluye el texto cifrado C y la llave pública d.

- Suponiendo que Eve no tiene la llave d. Puede recuperar el texto plano M?

Si Eve tiene un texto cifrado suficientemente largo, puede analizar la *frecuencia* de las letras (imagen tomada de `http://en.wikipedia.org/wiki/Caesar_cipher`):



Motivación: Criptografía de clave pública

Se tiene dos funciones E y D:

- E sirve para encriptar y D para desencriptar: $D(E(M)) = M$
- E no puede usarse para desencriptar: $E(E(M)) \neq M$
- E y D están relacionadas, pero es difícil descubrir D a partir de E

Vamos a ver un ejemplo de este tipo de sistemas: RSA

El sistema criptográfico RSA

Algoritmo:

1. Adivine dos números primos distintos P y Q
2. Sean $N = P \cdot Q$ y $\phi(N) = (P-1) \cdot (Q-1)$
3. Sean e y d dos números tales que $(e \cdot d) \bmod \phi(N) = 1$
4. Entonces:

$$\begin{aligned} E(M) &= M^e \bmod N \\ D(M) &= M^d \bmod N \end{aligned}$$

Decimos que (e, N) es la clave pública y (d, N) es la clave privada

El sistema criptográfico RSA

Ejemplo

Sean $P = 7$ y $Q = 11$

- Se tiene que $N = 77$ y $\phi(N) = 60$

Sean $e = 13$ y $d = 37$

- Se tiene que $(13 \cdot 37) \bmod 60 = 1$

Entonces: $E(M) = M^{13} \bmod 77$ y $D(M) = M^{37} \bmod 77$

Para $M = 5$:

$$
\begin{array}{lcccc}
E(5) & = & 5^{13} \bmod 77 & = & 26 \\
D(E(5)) & = & 26^{37} \bmod 77 & = & 5
\end{array}
$$

¿Por qué funciona RSA?

¿Qué propiedades debemos demostrar que son ciertas?

- $D(E(M)) = M$ para todo $M \in \{0, \ldots, N-1\}$

¿Qué problemas debemos demostrar que pueden ser resueltos de manera eficiente?

- Generar primos P y Q: Verificar si un número es primo
- Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$
- Calcular funciones E y D

¿Qué problemas no pueden ser resueltos de manera eficiente?

- Dado (e, N) calcular d: Encontrar los divisores de N

Vamos a estudiar algunos conceptos de Teoría de Números necesarios para mostrar que RSA funciona y puede ser implementado.

RSA funciona correctamente

Sean E y D construidas como fue mencionado en las transparencias anteriores.

- $N = P \cdot Q$, $E(M) = M^e \bmod N$ y $D(M) = M^d \bmod N$

Teorema (Rivest-Shamir-Adleman)

Para cada $M \in \{0, \ldots, N-1\}$, se tiene que $D(E(M)) = M$.

Demostración: Sabemos que

$$\begin{aligned} D(E(M)) &= (M^e \bmod N)^d \bmod N \\ &= (M^e)^d \bmod N \\ &= M^{e \cdot d} \bmod N \end{aligned}$$

Por lo tanto, tenemos que demostrar que $M^{e \cdot d} \equiv M \bmod N$

RSA funciona correctamente: Demostración

Sabemos que $e \cdot d \equiv 1 \bmod \phi(N)$

- Por lo tanto: $e \cdot d = k \cdot \phi(N) + 1$

Tenemos que demostrar que $M^{k \cdot \phi(N)+1} \equiv M \bmod N$

- El siguiente lema es fundamental para la demostración

Lema

$M^{k \cdot \phi(N)+1} \equiv M \, mod \, P$ y $M^{k \cdot \phi(N)+1} \equiv M \, mod \, Q$

Demostración: Primero suponemos que $P|M$.

RSA funciona correctamente: Demostración

Entonces:
$$\begin{aligned} M^{k\cdot\phi(N)+1} \bmod P &= (M \bmod P)^{k\cdot\phi(N)+1} \bmod P \\ &= 0^{k\cdot\phi(N)+1} \bmod P \\ &= 0 \end{aligned}$$

Por lo tanto: $M^{k\cdot\phi(N)+1} \equiv M \bmod P$

En segundo lugar, suponemos que $P \nmid M$.

- Sea $R = M \bmod P$

Dado que $R \in \{1, \ldots, P-1\}$, por teorema de Fermat:
$$R^{P-1} \equiv 1 \bmod P$$

Por lo tanto, dado que $R \equiv M \bmod P$:
$$M^{P-1} \equiv 1 \bmod P$$

RSA funciona correctamente: Demostración

De esto concluimos que:

$$
\begin{aligned}
M^{k\cdot\phi(N)+1} \bmod N &= ((M^{P-1})^{k\cdot(Q-1)} \cdot M) \bmod P \\
&= ((M^{P-1} \bmod P)^{k\cdot(Q-1)} \cdot M) \bmod P \\
&= (1^{k\cdot(Q-1)} \cdot M) \bmod P \\
&= M \bmod P
\end{aligned}
$$

Concluimos que $M^{k\cdot\phi(N)+1} \equiv M \bmod P$

- De la misma forma se demuestra que $M^{k\cdot\phi(N)+1} \equiv M \bmod Q$ $\square$

RSA funciona correctamente: Demostración

Del lema concluimos que:

$$\begin{aligned} M^{k\cdot\phi(N)+1} - M &= \alpha \cdot P \\ M^{k\cdot\phi(N)+1} - M &= \beta \cdot Q \end{aligned}$$

Por lo tanto: $\alpha \cdot P = \beta \cdot Q$

Entonces, dado que P y Q son primos distintos tenemos que $P|\beta$

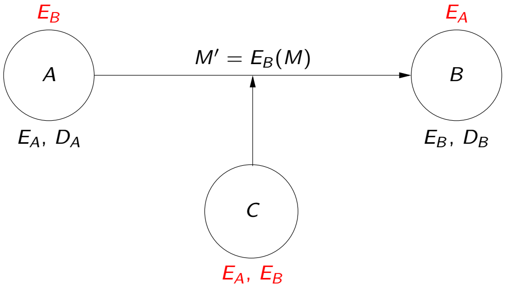
- $\beta = \gamma \cdot P$

Concluimos que $M^{k\cdot\phi(N)+1} - M = \gamma \cdot P \cdot Q$

- Vale decir: $M^{k\cdot\phi(N)+1} \equiv M \bmod N$ $\square$

RSA: Autentificación y firma digitales

Escenario usual para RSA:



Dos preguntas a responder:

- Autentificación: ¿Cómo puede saber A si E_B es efectivamente la clave pública de B?
- Firma digital: ¿Cómo puede saber B si el mensaje M' fue efectivamente enviado por A?

RSA: Autentificación y firma digitales

Una propiedad fundamental de RSA: $E(D(M)) = M$

- Usamos esta propiedad para resolver los problemas de autentificación y firma digital

Autentificación: Suponemos que existe una organización certificadora I

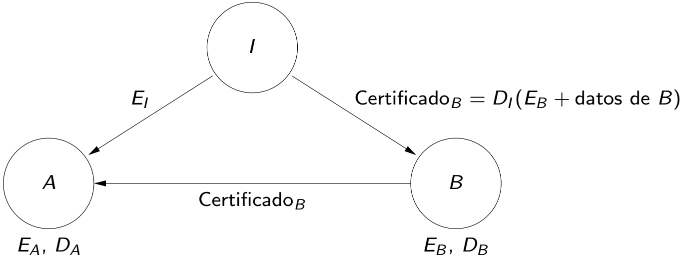
- Todos tienen acceso seguro a E_I

¿Cómo puede ser implementada la organización certificadora?

- ¿Por qué podemos suponer que tenemos acceso seguro a E_I?

RSA: Autentificación

Organización certificadora funciona de la siguiente forma:



Tenemos entonces el siguiente protocolo:

- I entrega a B un certificado Certificado_B
- Para saber la clave pública de B (y los datos relevantes de B), A utiliza $E_I(\text{Certificado}_B)$

RSA: Firma digital

Envío de mensajes firmados: A envía tanto el mensaje M' como una firma F:

$$\underset{E_A,\ D_A}{A} \xrightarrow[M' = E_B(M),\ F = D_A(M')]{} \underset{E_B,\ D_B}{B}$$

B recibe M' y F:

- Verificación: ¿Es cierto que $M' = E_A(M')$?
- Mensaje a leer: $D_B(M')$

Cálculo de exponentes e y d en RSA

Recuerde que en RSA: $N = P \cdot Q$ y $\phi(N) = (P-1) \cdot (Q-1)$

- Tenemos que generar e y d tales que $e \cdot d \equiv 1 \bmod \phi(N)$

Tenemos los ingredientes necesarios para generar e y d:

> genere al azar un número e
> **while** $\text{MCD}(e, \phi(N)) > 1$ **do**
>   genere al azar un número e
> calcule s y t tales que $1 = s \cdot \phi(N) + t \cdot e$
> sea $d \in \{0, \ldots, \phi(N) - 1\}$ tal que $d \equiv t \bmod \phi(N)$
> **return** (e, d)

RSA: Implementación (continuación)

Como mencionamos anteriormente, para poder implementar RSA necesitamos algoritmos eficientes para los siguientes problemas:

(1) Generar primos P y Q

(2) Generar números e y d tales que $(e \cdot d) \bmod \phi(N) = 1$

(3) Calcular funciones E y D

Ya resolvimos (2) y (3), nos falta resolver (1).

- Lamentablemente, esto está fuera del alcance de este curso ...

8. Grafos

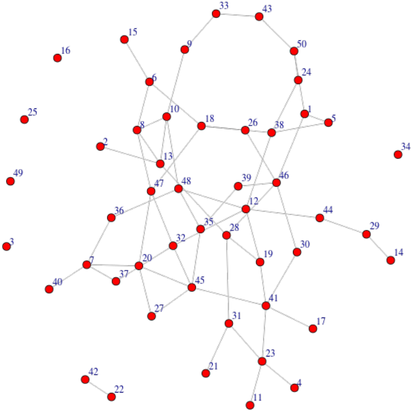
Definiciones

- Un grafo no dirigido (o simplemente un grafo) G es un par (V, E) donde V es un conjunto (cuyos elementos son llamados vértices) y E es otro conjunto (cuyos elementos son llamados aristas) y los elementos de E son de la forma $e = \{x, y\}$ donde $x, y \in V$.
- Una arista de la forma $e = \{x, x\}$ es llamado un loop. Un grafo sin loop es llamado un grafo simple.

En este curso asumiremos que todos los grafos son simples, a menos que se indique lo contrario (por lo tanto grafo significa grafo no dirigido y simple). También asumiremos que todo grafo es finito, es decir la cardinalidad de V y de E es finita. Además asumiremos que $V \neq \emptyset$.

Dibujito

Un dibujito ayuda a representar y entender bien las cosas. $V = \{1, 2, ..., 50\}$



Definiciones

- Dos vértices x e y son adyacentes, denotado $x \sim y$, si $e = \{x, y\} \in E$, además diremos que x e y son los extremos de la arista e.
- Definimos la vecindad de $x \in V$, denotada $N(x)$, como

$$N(x) = \{y \in V : x \sim y\},$$

además a $|N(x)|$ le llamamos el grado de x, denotado por $d(x)$.

Lema: Handshaking Lemma

Dado un grafo $G = (V, E)$ se tiene que

$$\sum_{x \in V} d(x) = 2|E|$$

Demostración: Pizarra

El resultado anterior, a pesar de lo simple, tiene algunos resultados importantes.

Consecuencias del Lema

1. Todo grafo de n vértices puede tener a lo más $\frac{n(n-1)}{2}$ aristas.
2. Todo grafo tiene una cantidad par de vértices de grado impar.
3. Un grafo G de n vértices se dice k-regular si todos los vértices tienen grado k. Un grafo k-regular tiene $\frac{nk}{2}$ aristas.
4. Si G es un grafo k-regular de n vértices entonces nk tiene que ser par.

Definiciones

- Una caminata es una lista $x_1, x_2, ..., x_k$ donde $\{x_i, x_{i+1}\} \in E$ para todo $i \in \{1, .., k-1\}$. Decimos que la caminata empieza en x_1 y termina en x_k.
- Decimos que x e y están conectados si existe una caminata que empieza en x y termina en y. Además decimos que cada vértice está conectado con él mismo.
- Un grafo es conexo si para todo par $x, y \in V$ están conectados. En otro caso se dice disconexo.
- Un subgrafo es un subconjunto de un grafo $G = (V, E)$ es un grafo $G' = (V', E')$ donde $V' \subseteq V$ y $E' \subseteq E$ para $e \in E'$ se tiene que $e \subseteq V$. Denotamos $H \subseteq G$
- Una componente conexa H es un subgrafo de G tal que H es conexo y es maximal, es decir para todo $H \subseteq H' \subseteq G$ H' es disconexo.

Definicion

Dado un grafo no dirigido $G = (V, E)$, un conjunto de vértices $C \subseteq V$ es un **clique** de G ssi $\forall u, v \in C((u, v) \in E)$.

Definicion

Dado un grafo $G = (V, E)$, un conjunto de vertices $C \subseteq V$ es un **conjunto independiente** de G ssi $\forall u, v \in C((u, v) \notin E)$.

Definicion

Dado un grafo $G = (V, E)$, un **camino de largo** n en G es una n-tupla $c = (u_1, \ldots, u_n) \in V^n$ donde para todo $1 \leq i \leq n-1$ se cumple que $(u_i, u_{i+1}) \in E$.

Definicion

Dado un grafo $G = (V, E)$, un **camino simple** es un camino $c = (u_1, \ldots, u_n)$ para el cual se cumple que:

$$\forall i, j \in \{1, \ldots, n\}\big(i \neq j \ \rightarrow \ u_i \neq u_j\big).$$

Definicion

Dado un grafo $G = (V, E)$, un camino $c = (u_1, \ldots, u_n)$ es un **ciclo** si y sólo si $u_1 = u_n$.

Definicion

Dado un grafo $G = (V, E)$ y dos nodos $u, v \in V$, v es **alcanzable** desde u si y sólo si existe un camino en G que comienza en u y termina en v.

Definicion

Dado un grafo $G = (V, E)$, una **n-coloracion** c de G es una funcion $c : V \to \{1, \ldots, n\}$ que cumple la siguiente condicion:

$$\forall u, v \in V\big((u, v) \in E \to c(u) \neq c(v)\big).$$

Definicion

un grafo $G = (V, E)$ se dice **n-colorable** si existe una n-coloracion de G.

Más definiciones

- Grafo completo: un grafo en que todos los nodos están conectados con todos los demás; equivalentemente, cada par de nodos está conectado por una arista; o también, cada nodo tiene grado $n - 1$.
- Grafo bipartito: un grafo $G = (V, E)$ cuyos nodos pueden ser divididos en dos conjuntos disjuntos U y V tales que toda arista en E conecta a un nodo en U con uno en V; i.e., U y V son conjuntos independientes. Un grafo bipartito no tiene ciclos de largo impar, y es 2-coloreable (más aún, un grafo que no es bipartito no es 2-coloreable).

Muchas veces, los elementos de V no importan (en realidad en todo lo que hemos hecho no importa que es V (si es un conjunto de número o vacas, da lo mismo). Para formalizar esta idea definimoslo que es ser isomorfo:

Definición

Dos grafos $G = (V, E)$ y $G' = (V', E')$ son isomorfos si existe una biyección $f : V \to V'$ tal que $\{x, y\} \in E$ ssi $\{f(x), f(y)\} \in E'$.

Demostrar que dos grafos son isomorfos es difícil y en general hay que dar la biyección. Pero lo importante es el concepto de que el nombre de los vértices no importa para estudiar ciertas propiedades.

Ejercicio: Muestre que ser isomorfos es relación de equivalencia.

Nombres

Algunas clases de isomorfismo importantes.

1. P_n (camino de largo n) es un grafo conexo de n vértices en que todos los vértices tienen grado 2 excepto dos vértices que tienen grado 1, llamados extremos.
2. C_n (ciclo de largo n) es un grado de conexo de n vértices en que todos los vértices tienen grado 2.
3. K_n (Grafo completo de tamaño n) es un grafo de n vértices en que todos los vértices tienen grado $n-1$.
4. $K_{n,m}$ (Grafo bipartito completo) Sean V_1 y V_2 dos conjuntos disjuntos cardinalidad n y m respectivamente. $K_{n,m}$ es el grafo que se forma considerando $V = V_1 \cup V_2$ y $E = \{\{x, y\} : x \in V_1, y \in V_2\}$.

Árboles

Los árboles son importantes en la vida y en las matemáticas discretas.

Definiciones

- Un grafo sin ciclos se llama acíclico
- Sea G un grafo acíclico. Si G tiene más de una componente se llama un bosque, si G tiene exactamente una componente se llama un árbol.

Definición

Sea $T = (V, E)$ un árbol. Un vértice $v \in V$ se dice una hoja si $d(v) = 1$.

lema

Todo árbol T con más de un arista tiene al menos dos hojas.

Los árboles pueden ser descritos de muchas formas. Ejercicio: Suponga que T es un grafo con n vértices. Muestre que las siguientes proposiciones son todas equivalentes:

1. T es un árbol.
2. T es acíclico y tiene exactamente $n - 1$ aristas.
3. T es conexo y tiene $n - 1$ aristas.
4. T es conexo y eliminar cualquier arista desconecta T.
5. Dados dos vértices de T existe exactamente un camino que los conecta

Árboles

Definición: Árbol con raíz

Un **árbol con raíz** es un árbol $T = (V, E)$ en que uno de sus vértices $r \in V$ se ha distinguido de los demás, y se le llama **raíz** del árbol.

Definiciones

- El largo del único camino entre r y un vértice x se llama **profundidad** de x.
- El máximo de las profundidades de los vértices de T es la **altura** del árbol.
- El conjunto de vértices que aparecen en el único camino de r a x se llaman **ancestros** de x.
- El **padre** de x es su ancestro de mayor profundidad. Análogamente, x es **hijo** de su padre.

Árboles

Definición: Árbol binario

Un árbol con raíz $T = (V, E)$ es un **árbol binario** si todo vértice tiene grado a lo más 3, o equivalentemente, si todo vértice tiene a los más 2 hijos. Si todo vértice que no es hoja tiene exactamente 2 hijos, es un árbol binario completo.

Teorema

- Un árbol binario de altura h tiene a lo más 2^h hojas.
- Un árbol binario completo de altura h tiene exactamente 2^h hojas.

9. Análisis de algoritmos

Qué es el Análisis de Algoritmos?
Es una disciplina de la Ciencia de la Computación que tiene básicamene dos objetivos:

- Entender porque los algoritmos terminan y al final de su ejecución obtenemos el resultado que decian hacer (correción)
- Estimar la cantidad de recursos que el algoritmo necesita para su ejecución. (complejidad)

Por qué es importante el Análisis de Algoritmos?

- porque nos ayuda a entender bien los algoritmos para poder reutilizarlos parcial o totalmente.
- porque nos sirve para saber qué mejorar de un algoritmo o como implementarlo de forma más eficiente.

Un *programa* o *algoritmo* tiene:

- **Precondiciones:** Representan el input del programa.
- **Postcondiciones:** Representan el output del programa.

Ejemplo

Para un programa que multiplica dos numeros, podriamos tener:

- **Pre:** $m, n \in \mathbb{N}$.
- **Post:** $p = mn$.

Correccion de algoritmos

Para demostrar que un algoritmo es correcto, se deben demostrar dos cosas:

- Que el algoritmo se detiene.
- Que la ejecucion del algoritmo causa que las postcondiciones sean verdaderas.

Correccion de un loop

while(G **)**
cuerpo del loop
end while

Definicion

Para un loop, se define la *invariante del loop* $I(n)$ como un predicado que es verdadero en cada paso de la iteracion.

Definicion

Para un loop, se define la *condicion del loop* G como un predicado que debe ser verdadero para ejecutar la siguiente iteracion.

Correccion de un loop

La correccion de un **while** se hace por induccion:

- **Propiedad de base:** La precondicion del loop debe implicar que $I(0)$ es verdadero.
- **Propiedad inductiva:** Para todo natural $k > 0$, si G y $I(k)$ son verdaderos antes de la iteracion, entonces $I(k+1)$ es verdadero despues de la iteracion.
- **Termino finito:** Existe un k para el cual G es falso.
- **Correccion:** Inmediatamente despues de terminado el loop (G es falso), para $k = N$ si $I(N)$ es verdadero, entonces las postcondiciones son verdaderas.

Ejercicio

Sin usar el operador producto, escriba un algoritmo que multiplica dos numeros naturales:

- **Pre:** $n, m \in \mathbb{N}$.
- **Post:** $p = nm$.

Luego, demuestre que el algoritmo se detiene y que es correcto.

Respuesta: Consideramos el siguiente algoritmo:

$i \leftarrow 0$, $p \leftarrow 0$
while $(i \neq m)$
  $p \leftarrow p + n$
  $i \leftarrow i + 1$
end while

$i \leftarrow 0$, $p \leftarrow 0$
while $(i \neq m)$
  $p \leftarrow p + n$
  $i \leftarrow i + 1$
end while

Ahora, definimos la invariante del loop $I(k) : (i = k) \wedge (p = kn)$.
Propiedad base: $I(0) : (i = 0) \wedge (p = 0)$ es verdadero.
Propiedad inductiva: Debemos demostrar que si $G \wedge I(k)$ es verdadero al iniciar una iteracion, entonces $(k + 1)$ es verdadero despues de la iteracion.
Sabemos que $i \neq m$, $p = kn$ y $i = k$.
Hacemos:

$$\begin{aligned} p &\leftarrow p + n = kn + n = (k+1)n, \\ i &\leftarrow i + 1 = k + 1. \end{aligned}$$

Complejidad de un algoritmo

Definicion

Dado un algoritmo A se define la *complejidad* de A como una funcion $T_A : \mathbb{N} \to \mathbb{N}$. La funcion $T_A(n)$ recibe el tamaño n del input y retorna la cantidad de pasos que debe realizar A para terminar, en el peor caso.

Nota: Usualmente no es importante la forma especifica de la funcion T_A, sino que el conjunto asintotico en el que esta.

Notación asintótica

En muchos casos, nos interesa conocer el *orden* de un algoritmo en lugar de su complejidad exacta.

- Queremos decir que un algoritmo es lineal o cuadrático, en lugar de decir que su complejidad es $3n^2 + 17n + 22$

Vamos a desarrollar notación para hablar del orden de un algoritmo.

Vamos a considerar funciones de la forma $f : \mathbb{N} \to \mathbb{R}_0^+$, donde $\mathbb{R}^+ = \{r \in \mathbb{R} \mid r > 0\}$ y $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$

- Incluyen a las funciones definidas en las transparencias anteriores, y también sirven para modelar el tiempo de ejecución de un algoritmo

La notación $O(f)$

Sea $f : \mathbb{N} \to \mathbb{R}_0^+$

Definición

$$O(f) \;=\; \{g : \mathbb{N} \to \mathbb{R}_0^+ \mid (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N}) \\ (\forall n \geq n_0)\,(g(n) \leq c \cdot f(n))\}$$

Ejercicio

Demuestre que $3n^2 + 17n + 22 \in O(n^2)$

Las notaciones $\Omega(f)$ y $\Theta(f)$

Definición

$$\begin{aligned}\Omega(f) = \ & \{g : \mathbb{N} \to \mathbb{R}_0^+ \mid (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N}) \\ & \qquad (\forall n \geq n_0)\,(c \cdot f(n) \leq g(n))\} \\ \Theta(f) = \ & O(f) \cap \Omega(f)\end{aligned}$$

Ejercicios

1. Demuestre que $3n^2 + 17n + 22 \in \Theta(n^2)$
2. Demuestre que $g \in \Theta(f)$ si y sólo si existen $c, d \in \mathbb{R}^+$ y $n_0 \in \mathbb{N}$ tal que para todo $n \geq n_0$: $c \cdot f(n) \leq g(n) \leq d \cdot f(n)$

Ejemplo

Un algoritmo de busqueda lineal puede correr en tiempo $T_A(n) = 2n + 2$ (en peor caso!). Entonces, se dice que su complejidad es $T_A(n) = 2n + 2$. Sin embargo, usualmente basta con notar que $T_A(n) \in O(n)$.

Se dice, entonces, que la complejidad del algoritmo de busqueda lineal esta en $O(n)$. A veces, se dice que la complejidad de *A es* $O(n)$ (menos preciso). Por ultimo, a veces se dice que el algoritmo es *lineal*.

Ecuaciones de recurrencia

Suponga que tiene una lista ordenada (de menor a mayor) L de números naturales

- L tiene n elementos, para referirnos al elemento i-ésimo $(1 \leq i \leq n)$ usamos la notación $L[i]$

¿Cómo podemos verificar si un número a está en L?

Ecuaciones de recurrencia: Búsqueda binaria

Para verificar si un número a está en L usamos el siguiente algoritmo:

encontrar(a, L, i, j)
    if $i > j$ **then return** no
    else if $i = j$ **then**
        if $L[i] = a$ **then return** i
        else return no
    else
        $p = \lfloor \frac{i+j}{2} \rfloor$
        if $L[p] < a$ **then return encontrar**$(a, L, p+1, j)$
        else if $L[p] > a$ **then return encontrar**$(a, L, i, p-1)$
        else return p

Llamada inicial al algoritmo: **encontrar**$(a, L, 1, n)$

Ecuaciones de recurrencia: Búsqueda binaria

¿Cuál es la complejidad del algoritmo?

- ¿Qué operaciones vamos a considerar?
- ¿Cuál es el peor caso?

Si contamos sólo las comparaciones, entonces la siguiente expresión define la complejidad del algoritmo:

$$T(n) = \begin{cases} 1 & n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + 1 & n > 1 \end{cases}$$

Esta es una ecuación de recurrencia.

Ecuaciones de recurrencia: Búsqueda binaria

¿Cómo podemos solucionar una ecuación de recurrencia?

- Técnica básica: sustitución de variables

Para la ecuación anterior usamos la sustitución $n = 2^k$.

- Vamos a resolver la ecuación suponiendo que n es una potencia de 2
- Vamos a estudiar condiciones bajo las cuales el resultado para una potencia de 2 puede ser extendido a todo n
    - Estas condiciones van a servir para cualquier potencia

Ecuaciones de recurrencia: sustitución de variables

Si realizamos la sustitución $n = 2^k$ en la ecuación:

$$T(n) = \begin{cases} 1 & n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + 1 & n > 1 \end{cases}$$

obtenemos:

$$T(2^k) = \begin{cases} 1 & k = 0 \\ T(2^{k-1}) + 1 & k > 0 \end{cases}$$

Ecuaciones de recurrencia: sustitución de variables

Extendiendo la expresión anterior obtenemos:

$$\begin{aligned} T(2^k) &= T(2^{k-1}) + 1 \\ &= (T(2^{k-2}) + 1) + 1 \\ &= T(2^{k-2}) + 2 \\ &= (T(2^{k-3}) + 1) + 2 \\ &= T(2^{k-3}) + 3 \\ &= \cdots \end{aligned}$$

Deducimos la expresión general para $k - i \geq 0$:

$$T(2^k) = T(2^{k-i}) + i$$

Ecuaciones de recurrencia: sustitución de variables

Considerando $i = k$ obtenemos:

$$\begin{aligned} T(2^k) &= T(1) + k \\ &= 1 + k \end{aligned}$$

Dado que $k = \log_2 n$, obtenemos que $T(n) = \log_2 n + 1$ para n potencia de 2.

- Vamos a definir notación para decir esto de manera formal

Notaciones asintóticas condicionales

Sea P un predicado sobre los números naturales.

Definición

$$\begin{aligned} O(f \mid P) \ = \ \{g : \mathbb{N} \to \mathbb{R}_0^+ \mid & (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N}) \\ & (\forall n \geq n_0)\,(n \in P \to g(n) \leq c \cdot f(n))\} \end{aligned}$$

Las notaciones $\Omega(f \mid P)$ y $\Theta(f \mid P)$ son definidas de manera análoga.

Búsqueda en una lista ordenada: complejidad del algoritmo

Sea $\text{POTENCIA}_2 = \{2^i \mid i \in \mathbb{N}\}$

Para la función T que define la complejidad del procedimiento **encontrar**, tenemos que:

$$T \quad \in \quad \Theta(\log_2 n \mid \text{POTENCIA}_2)$$

¿Podemos concluir que $T \in \Theta(\log_2 n)$?

- Vamos a estudiar este problema, pero antes vamos a ver un segundo ejemplo.

Ordenamiento de una lista

Ahora queremos ordenar una lista L de números naturales

- Utilizamos el algoritmo **mergesort**

Si contamos sólo las comparaciones, entonces la siguiente expresión define la complejidad de **mergesort**:

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + (n-1) & n > 1 \end{cases}$$

Orden de **mergesort**

Nuevamente utilizamos la sustitución $n = 2^k$, obteniendo:

$$T(2^k) \quad = \quad \begin{cases} 0 & k = 0 \\ 2 \cdot T(2^{k-1}) + (2^k - 1) & k > 0 \end{cases}$$

Desarrollando esta expresión obtenemos:

$$\begin{aligned}
T(2^k) &= 2 \cdot T(2^{k-1}) + (2^k - 1) \\
&= 2 \cdot (2 \cdot T(2^{k-2}) + (2^{k-1} - 1)) + (2^k - 1) \\
&= 2^2 \cdot T(2^{k-2}) + 2^k - 2 + 2^k - 1 \\
&= 2^2 \cdot T(2^{k-2}) + 2 \cdot 2^k - (1 + 2) \\
&= 2^2 \cdot (2 \cdot T(2^{k-3}) + (2^{k-2} - 1)) + 2 \cdot 2^k - (1 + 2) \\
&= 2^3 \cdot T(2^{k-3}) + 2^k - 2^2 + 2 \cdot 2^k - (1 + 2) \\
&= 2^3 \cdot T(2^{k-3}) + 3 \cdot 2^k - (1 + 2 + 2^2) \\
&= \cdots
\end{aligned}$$

Orden de **mergesort**

Deducimos la expresión general para $k - i \geq 0$:

$$\begin{aligned} T(2^k) &= 2^i \cdot T(2^{k-i}) + i \cdot 2^k - \sum_{j=0}^{i-1} 2^j \\ &= 2^i \cdot T(2^{k-i}) + i \cdot 2^k - 2^i + 1 \end{aligned}$$

Considerando $i = k$ obtenemos:

$$\begin{aligned} T(2^k) &= 2^k \cdot T(1) + k \cdot 2^k - 2^k + 1 \\ &= k \cdot 2^k - 2^k + 1 \end{aligned}$$

Dado que $k = \log_2 n$, concluimos que:

$$T \in \Theta(n \cdot \log_2 n \mid \text{POTENCIA}_2)$$

Teorema Maestro

El teorema maestro es una de las principales herramientas para encontrar solución a las recurrencias que aparecen en algoritmos del tipo Dividir para Conquistar.

Teorema

Sean a_1, a_2, b, c, d constantes reales positivas y supongamos que $T(n)$ satisface la ecuación de recurrencia:

$$T(n) = \begin{cases} c_0, & \text{si } 0 \leq n < \frac{b}{b-1} \\ a_1 T(\lceil \frac{n}{b} \rceil) + a_2 T(\lfloor \frac{n}{b} \rfloor) + cn^d, & \text{si } n \geq \frac{b}{b-1} \end{cases}$$

entonces, llamando $a = a_1 + a_2$ se tiene.

$$T(n) \in \begin{cases} \Theta(n^d), & \text{si } a < b^d, \\ \Theta(n^d \log(n)), & \text{si } a = b^d, \\ \Theta(n^{\log_b(a)}), & \text{si } a > b^d \end{cases}$$

Apuntes IIC1253 - Matemáticas Discretas

Examen de Título 2014

Gabriel Diéguez Franzani

Selección de diapositivas de
Gonzalo Díaz, Nicolás Rivera y Marcelo Arenas

23 de diciembre de 2013