Matemáticas Discretas

Lógica proposicional

Nicolás Alvarado
nfalvarado@mat.uc.cl

Bernardo Barías
bjbarias@uc.cl

Sebastián Bugedo
bugedo@uc.cl

Gabriel Diéguez
gsdieguez@ing.puc.cl

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

23 de agosto de 2023

Objetivos

1. Formular enunciados formales en notación matemática usando lógica, conjuntos, relaciones, funciones, cardinalidad, y otras herramientas, desarrollando definiciones y teoremas al respecto, así como demostrar o refutar estos enunciados, usando variadas técnicas.
2. Aplicar inducción como técnica para demostración de propiedades en conjuntos discretos y como técnica de definición formal de objetos discretos.
3. Modelar formalmente un problema usando lógica, conjuntos, relaciones, y las propiedades necesarias, y demostrar propiedades al respecto de su modelo.

Contenidos

¿Qué es lógica?

Definición

Lógica es el uso y estudio del **razonamiento válido**.

Wikipedia

Introducción

- La lógica (en su sentido más amplio) estudia la **inferencia**.
  - Cómo obtener conclusiones a partir de premisas.
- Originalmente, hacía esto en el lenguaje natural.

Ejemplo

¿Es el siguiente razonamiento válido?

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

La lógica debería poder usarse para demostrar que sí.

Introducción

¿Qué pasa si usamos el lenguaje natural para definir cosas? Por ejemplo:

- Podemos representar los números naturales usando oraciones del lenguaje natural: "dos mil dieciocho", "el segundo número"...
- El diccionario de la RAE tiene una cantidad finita de palabras.
- El número de oraciones con a lo más 50 palabras también es finito.
- Por lo tanto, deben existir números naturales que no pueden ser representados con oraciones de a lo más 50 palabras.
- Por PBO, debe existir un menor número que cumpla lo anterior.

Introducción

Sea n el siguiente número natural:

El primer número natural que no puede ser definido por una oración con a lo más cincuenta palabras del diccionario de la Real Academia Española.

La oración anterior define a n con 25 palabras. **¡Contradicción!**

¿Qué pasó?

¿Por qué necesitamos la lógica?

- Necesitamos un lenguaje con una **sintaxis** precisa y una **semántica** bien definida.
  - **Qué** objetos pertenecen al lenguaje y qué **significan**.
- Queremos usar este lenguaje en matemáticas para hacer cosas que hasta ahora hemos hecho de manera intuitiva.
  - Definición de objetos matemáticos (conjuntos, números naturales, números reales. . .).
  - Definición de teorías matemáticas (teoría de conjuntos, teoría de los naturales. . .).
  - Definición del concepto de demostración.
- Pero también queremos usar este lenguaje en computación.

¿Por qué necesitamos la lógica en computación?

- Bases de datos: lenguajes de consulta, lenguajes para restricciones.
- Inteligencia artificial: representación de conocimiento, razonamiento.
- Ingeniería de software: especificación de sistemas, verificación de propiedades.
- Teoría de la computación: complejidad descriptiva, algoritmos de aproximación.
- Criptografía: verificación de protocolos.
- Procesamiento de lenguaje natural.
- ...

¿Cuántas lógicas existen?

- Lógica proposicional
- Lógica de predicados
- Lógica de primer orden
- Lógica de segundo orden
- Lógicas descriptivas
- Lógicas temporales
- Lógica intuicionista
- Lógica trivalente
- etc ...

¿Qué lógicas veremos en el curso?

- **Lógica proposicional**
- **Lógica de predicados**
- Lógica de primer orden
- Lógica de segundo orden
- Lógica de segundo orden monádica
- Lógica temporal lineal
- Lógica intuicionista
- Lógica trivalente
- etc ...

Más en..

- IIC2213 - Lógica para ciencias de la computación
- IIC3263 - Teoría de modelos finitos

Sintaxis de la Lógica proposicional

Usaremos los siguientes elementos:

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

Las variables proposicionales representan proposiciones **completas** e **indivisibles**, que pueden ser **verdaderas** o **falsas**. En general llamaremos P al conjunto de ellas.

Ejemplo

$P = \{\text{Juan\_cursa\_IIC1253}, \text{Juan\_aprobo\_IIC1103}\}$

Sintaxis de la Lógica proposicional

Los conectivos lógicos se usan para construir expresiones más complejas que también pueden ser verdaderas o falsas.

Ejemplo

$$\text{Juan\_cursa\_IIC1253} \rightarrow \text{Juan\_aprobo\_IIC1103}$$
$$(\neg \text{Juan\_cursa\_IIC1253}) \rightarrow \text{Juan\_aprobo\_IIC1103}$$

Los símbolos de puntuación se usan para evitar ambigüedades.

Sintaxis de la Lógica proposicional

Definición

Sea P un conjunto de variables proposicionales. El conjunto de todas las **fórmulas** de lógica proposicional sobre P, denotado por $L(P)$, es el menor conjunto que cumple las siguientes reglas:

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

¡Podemos usar Inducción estructural!

Ejercicios

1. Defina la función $largo(\varphi)$ como el largo de una fórmula en lógica proposicional (variables, conectivos y paréntesis).
2. Defina la función $var(\varphi)$ como el número de ocurrencias de variables proposicionales en φ.
3. Demuestre que si φ no contiene el símbolo $\neg$, se cumple que $largo(\varphi) \leq 4 \cdot var(\varphi)^2$.
   - ¿Qué pasa si contiene el símbolo $\neg$?

Ejercicio

Defina la función $largo(\varphi)$ como el largo de una fórmula en lógica proposicional (variables, conectivos y paréntesis).

1. $largo(p) = 1$, con $p \in P$.
2. $largo((\neg\varphi)) = 3 + largo(\varphi)$, con $\varphi \in L(P)$.
3. $largo((\varphi \star \psi)) = 3 + largo(\varphi) + largo(\psi)$, con $\varphi, \psi \in L(P)$ y $\star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Ejercicio

Defina la función $var(\varphi)$ como el número de ocurrencias de variables proposicionales en φ.

1. $var(p) = 1$, con $p \in P$.
2. $var((\neg\varphi)) = var(\varphi)$, con $\varphi \in L(P)$.
3. $var((\varphi \star \psi)) = var(\varphi) + var(\psi)$, con $\varphi, \psi \in L(P)$ y $\star \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

Sintaxis de la Lógica proposicional

Ejercicio

Demuestre que si φ no contiene el símbolo $\neg$, se cumple que $largo(\varphi) \leq 4 \cdot var(\varphi)^2$.

Demostración: usaremos inducción estructural.

BI: $largo(p) = 1 \leq 4 = 4 \cdot var(p)^2$.

HI: Supongamos que la propiedad se cumple para $\varphi, \psi \in L(P)$ tales que no tienen el símbolo $\neg$; es decir, $largo(\varphi) \leq 4 \cdot var(\varphi)^2$ y $largo(\psi) \leq 4 \cdot var(\psi)^2$.

TI: Necesitamos demostrar que $largo((\varphi \star \psi)) \leq 4 \cdot var((\varphi \star \psi))^2$.

Desarrollando: $largo((\varphi \star \psi)) = 3 + largo(\varphi) + largo(\psi)$

$$\overset{HI}{\leq} 3 + 4 \cdot var(\varphi)^2 + 4 \cdot var(\psi)^2 = 3 + 4(var(\varphi)^2 + var(\psi)^2)$$

$$= 3 + 4((var(\varphi) + var(\psi))^2 - 2 \cdot var(\varphi) \cdot var(\psi))$$

$$= 3 - 8 \cdot var(\varphi) \cdot var(\psi) + 4 \cdot (var(\varphi) + var(\psi))^2$$

Aplicando la definición de var obtenemos que $largo((\varphi \star \psi)) \leq 3 - 8 \cdot var(\varphi) \cdot var(\psi) + 4 \cdot var((\varphi \star \psi))^2$. Ahora, como $var(\theta) \geq 1$ para cualquier $\theta \in L(P)$, tenemos que $8 \cdot var(\varphi) \cdot var(\psi) \geq 8$ y por lo tanto $3 - 8 \cdot var(\varphi) \cdot var(\psi) \leq 0$. Finalmente, aplicando esta última desigualdad obtenemos que $largo((\varphi \star \psi)) \leq 4 \cdot var((\varphi \star \psi))^2$ como queríamos demostrar.

Por inducción estructural, se sigue que la propiedad es cierta para toda fórmula que no contiene $\neg$. $\square$

Ejercicio

¿Qué pasa si contiene el símbolo $\neg$?

Como la regla que permite ocupar el conectivo $\neg$ no agrega variables pero sí aumenta el largo, podríamos ocuparla y aumentar arbitrariamente el largo de una fórmula sin aumentar el número de variables, con lo que la propiedad no se cumpliría luego de algún número de $\neg$ agregadas. Una idea para limitar esto es no permitir el uso de más de una negación; i.e., no permitir fórmulas del tipo $(\neg(\neg\varphi))$ y sucesivas. Esto se fundamenta más adelante.

¿Cuándo una fórmula es verdadera?

- Depende del "mundo" en que la estemos interpretando.
- Un "mundo" particular le dará una interpretación a cada variable proposicional.
    - Un valor verdadero o falso.
- Con estos valores y los conectivos utilizados, podremos determinar el valor de verdad de la fórmula.

Definición

Una **valuación** o **asignación de verdad** para las variables proposicionales en un conjunto P es una función $\sigma : P \to \{0, 1\}$.

Convención: $0 =$ falso, $1 =$ verdadero.

Semántica de la Lógica proposicional

Dados un conjunto P y una asignación de verdad σ para P, necesitamos extender la noción de valuación a las fórmulas en $L(P)$. ¿Cómo lo hacemos?

Definición

La función $\hat{\sigma} : L(P) \to \{0,1\}$ se define como:

1. Si $p \in P$, entonces $\hat{\sigma}(p) = \sigma(p)$.
2. $\hat{\sigma}((\neg\varphi)) = \begin{cases} 1 & \text{si } \hat{\sigma}(\varphi) = 0 \\ 0 & \text{si } \hat{\sigma}(\varphi) = 1 \end{cases}$
3. $\hat{\sigma}((\varphi \wedge \psi)) = \begin{cases} 1 & \text{si } \hat{\sigma}(\varphi) = 1 \text{ y } \hat{\sigma}(\psi) = 1 \\ 0 & \text{si } \hat{\sigma}(\varphi) = 0 \text{ o } \hat{\sigma}(\psi) = 0 \end{cases}$

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

Semántica de la Lógica proposicional

Definición

La función $\hat{\sigma} : L(P) \to \{0,1\}$ se define como:

1. Si $p \in P$, entonces $\hat{\sigma}(p) = \sigma(p)$.
2. $\hat{\sigma}((\neg\varphi)) = \begin{cases} 1 & \text{si } \hat{\sigma}(\varphi) = 0 \\ 0 & \text{si } \hat{\sigma}(\varphi) = 1 \end{cases}$
3. $\hat{\sigma}((\varphi \wedge \psi)) = \begin{cases} 1 & \text{si } \hat{\sigma}(\varphi) = 1 \text{ y } \hat{\sigma}(\psi) = 1 \\ 0 & \text{si } \hat{\sigma}(\varphi) = 0 \text{ o } \hat{\sigma}(\psi) = 0 \end{cases}$

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

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

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

Semántica de la Lógica proposicional

- $\hat{\sigma}(\varphi)$ es la **evaluación** de φ dada la asignación σ y representa su valor de verdad.
- Por simplicidad, usaremos σ en lugar de $\hat{\sigma}$.

Ejemplo

Si tenemos una valuación σ tal que

$$\sigma(\text{Juan\_cursa\_IIC1253}) = 1$$
$$\sigma(\text{Juan\_aprobo\_IIC1103}) = 0$$

entonces

$$\sigma((\text{Juan\_cursa\_IIC1253} \rightarrow \text{Juan\_aprobo\_IIC1103})) = 0$$

Definición

Dos fórmulas $\varphi, \psi \in L(P)$ son **lógicamente equivalentes** (denotado como $\varphi \equiv \psi$) si para toda valuación σ se tiene que $\sigma(\varphi) = \sigma(\psi)$.

Ejercicio

Demuestre que $(p \to q) \equiv ((\neg p) \vee q)$.

Ejercicio

Demuestre que $(p \to q) \equiv ((\neg p) \vee q)$.

<u>Demostración:</u> Como tenemos 2 variables proposicionales, hay 4 valuaciones posibles. Podemos probarlas todas y verificar que siempre son iguales para ambas fórmulas. Una manera de hacer esto es ponerlas en una tabla como la siguiente:

	p	q	$(p \to q)$	$((\neg p) \vee q)$
σ_1 :	0	0	1	1
σ_2 :	0	1	1	1
σ_3 :	1	0	0	0
σ_4 :	1	1	1	1

Verificamos que para cada valuación posible el valor de verdad de las fórmulas es el mismo, con lo que demostramos que son equivalentes (por definición). $\square$

Tablas de verdad

Las fórmulas se pueden representar y analizar en una tabla de verdad.

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

Ejercicio

Si P tiene n variables proposicionales, ¿cuántas tablas de verdad distintas hay para $L(P)$?

Respuesta: Hay 2^n valuaciones posibles para n variables, por lo que cada tabla tiene 2^n filas. Cada una de estas filas puede tomar valor 0 o 1, por lo que podemos tener 2^{2^n} tablas distintas.

Tablas de verdad

Note que $\varphi \equiv \psi$ si y sólo si sus respectivas columnas en una tabla de verdad son iguales.

- Podemos usar tablas de verdad para comparar fórmulas, como en el ejercicio que ya hicimos. Otro ejemplo:

p	q	r	$(p \wedge q) \wedge r$	$p \wedge (q \wedge r)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Tablas de verdad

p	q	r	$(p \wedge q) \wedge r$	$p \wedge (q \wedge r)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Concluimos que $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$.

- Es decir, el conectivo $\wedge$ es asociativo.
- Podemos omitir paréntesis cuando tengamos una cadena de ellos.
  - Escribimos simplemente $p \wedge q \wedge r$.

Leyes de equivalencia

Doble negación

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

Leyes de De Morgan

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

Conmutatividad

$\varphi \wedge \psi \equiv \psi \wedge \varphi$
$\varphi \vee \psi \equiv \psi \vee \varphi$

Leyes de equivalencia

Asociatividad

$\varphi \wedge (\psi \wedge \theta) \equiv (\varphi \wedge \psi) \wedge \theta$
$\varphi \vee (\psi \vee \theta) \equiv (\varphi \vee \psi) \vee \theta$

Distributividad

$\varphi \wedge (\psi \vee \theta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$
$\varphi \vee (\psi \wedge \theta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)$

Idempotencia

$\varphi \wedge \varphi \equiv \varphi$
$\varphi \vee \varphi \equiv \varphi$

Leyes de equivalencia

Absorción

$\varphi \wedge (\varphi \vee \psi) \equiv \varphi$
$\varphi \vee (\varphi \wedge \psi) \equiv \varphi$

Implicancia

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

Doble implicancia

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

Ejercicios

1. Demuestre las leyes enunciadas.
2. ¿Es $\rightarrow$ asociativo?

Leyes de equivalencia

Ejercicio

1. Demuestre las leyes enunciadas.

Ya demostramos la ley asociativa para $\wedge$ y la ley de implicancia. El resto se dejan como ejercicios.

Ejercicios

2. ¿Es $\rightarrow$ asociativo?

Esto es lo mismo que preguntarnos si $\varphi \rightarrow (\psi \rightarrow \theta) \equiv (\varphi \rightarrow \psi) \rightarrow \theta$, lo cual no es cierto. Por ejemplo, si tomamos una valuación σ tal que $\sigma(\varphi) = 0$, en la primera fórmula el valor de verdad de θ depende del valor de verdad de ψ, mientras que en la segunda fórmula forzamos que sea 1, al hacer la primera parte de la implicación cierta. Otra forma de comprobarlo es haciendo la tabla de verdad para ambas fórmulas.

Notación

- Omitiremos paréntesis externos y otros que no produzcan ambigüedad.
  - Ejemplo: escribimos $p \wedge q \wedge r$.
- Usaremos los siguientes operadores:

$$\bigvee_{i=1}^{n} \varphi_i = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n$$

$$\bigwedge_{i=1}^{n} \varphi_i = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$$

Conectivos y fórmulas

Las tablas de verdad también nos sirven para definir conectivos.

Ejemplo

El conectivo XOR se define con la siguiente tabla de verdad:

p	q	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Ejercicio

- ¿Cuántos conectivos binarios posibles tenemos?

Conectivos y fórmulas

Tenemos una fórmula $\varphi \in L(P)$, con $P = \{p, q, r\}$, y lo único que conocemos de ella es que cumple la siguiente tabla de verdad:

p	q	r	φ
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

- ¿Podemos construir una fórmula que sea lógicamente equivalente a φ?
- Dada una tabla con n variables proposicionales, ¿podemos generalizar el argumento anterior?

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

Podemos generalizar el argumento anterior. Sea φ una fórmula con n variables proposicionales $p_1, \ldots, p_n$. Tenemos 2^n posibles valuaciones $\sigma_1, \ldots, \sigma_{2^n}$. La tabla de verdad de φ se vería como:

$$
\begin{array}{ccccccc|c}
 & p_1 & p_2 & \cdots & p_{n-1} & p_n & & \varphi \\
\hline
\sigma_1: & 0 & 0 & \ldots & 0 & 0 & & \sigma_1(\varphi) \\
\sigma_2: & 0 & 0 & \ldots & 0 & 1 & & \sigma_2(\varphi) \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots & & \vdots \\
\sigma_{2^n}: & 1 & 1 & \ldots & 1 & 1 & & \sigma_{2^n}(\varphi)
\end{array}
$$

Para cada σ_j con $j \in \{1, \ldots, 2^n\}$ consideremos la siguiente fórmula:

$$
\varphi_j = \left(\bigwedge_{\substack{i=1\ldots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\ldots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right)
$$

Conectivos y fórmulas

$$\varphi_j = \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right)$$

Notemos que φ_j representa a la fila j de la tabla de verdad de φ. Ahora hacemos la disyunción entre las valuaciones que hacen verdad a φ:

$$\bigvee_{\substack{j=1\dots 2^n \\ \sigma_j(\varphi)=1}} \varphi_j = \bigvee_{\substack{j=1\dots 2^n \\ \sigma_j(\varphi)=1}} \left(\left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=1}} p_i \right) \wedge \left(\bigwedge_{\substack{i=1\dots n \\ \sigma_j(p_i)=0}} (\neg p_i) \right) \right)$$

Conectivos y fórmulas

Teorema

Podemos representar cualquier tabla de verdad con una fórmula.

Corolario

Podemos representar cualquier tabla de verdad con una fórmula que sólo usa $\neg, \wedge$ y $\vee$.

Conectivos funcionalmente completos

Definición

Un conjunto de conectivos lógicos se dice **funcionalmente completo** si toda fórmula en $L(P)$ es lógicamente equivalente a una fórmula que sólo usa esos conectivos.

- Ya demostramos que $\{\neg, \wedge, \vee\}$ es funcionalmente completo.

Ejercicios

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

Conectivos funcionalmente completos

Ejercicio

Demuestre que $\{\neg, \rightarrow\}$ es funcionalmente completo.

<u>Demostración:</u> Como sabemos que $C = \{\neg, \wedge, \vee\}$ es funcionalmente completo, demostraremos por inducción que toda fórmula construida usando sólo los conectivos anteriores es lógicamente equivalente a otra fórmula que solo usa $\neg$ y $\rightarrow$. Con esto, queda demostrado que $C' = \{\neg, \rightarrow\}$ es funcionalmente completo.

BI: Si $\varphi = p$, con $p \in P$, la propiedad se cumple trivialmente.

HI: Supongamos que $\varphi, \psi \in L(P)$, que sólo usan conectivos en C, son tales que $\varphi \equiv \varphi'$ y $\psi \equiv \psi'$, donde φ', ψ' sólo usan conectivos en C'.

TI: Consideraremos una fórmula θ construida con los pasos inductivos para los operadores en C:

$i)$ $\theta = (\neg\varphi) \overset{HI}{\equiv} (\neg\varphi')$, y como φ' sólo usa conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'.

Ejercicio

Demuestre que $\{\neg, \rightarrow\}$ es funcionalmente completo.

TI:

$ii)$ $\theta = \varphi \wedge \psi \overset{HI}{\equiv} \varphi' \wedge \psi'$

Usando las leyes de doble negación, De Morgan y de implicancia:

$$\theta \equiv \neg(\neg(\varphi' \wedge \psi')) \equiv \neg((\neg\varphi') \vee (\neg\psi')) \equiv \neg(\varphi' \rightarrow (\neg\psi'))$$

Y como φ', ψ' sólo usan conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'.

Conectivos funcionalmente completos

Ejercicio

Demuestre que $\{\neg, \rightarrow\}$ es funcionalmente completo.

TI:
$iii)$ $\theta = \varphi \vee \psi \overset{HI}{\equiv} \varphi' \vee \psi'$

Usando la ley de implicancia:

$\theta \equiv (\neg\varphi') \rightarrow \psi'$

Y como φ', ψ' sólo usan conectivos en C', θ es equivalente a una fórmula que sólo usa conectivos en C'. $\square$

Ejercicio

Demuestre que $\{\neg\}$ no es funcionalmente completo.

Demostración: Dado $P = \{p\}$, demostraremos por inducción que toda fórmula en $L(P)$ construida usando sólo p y $\neg$ es lógicamente equivalente a p o a $\neg p$. Como ninguna de estas fórmulas es equivalente a $p \wedge \neg p$, se concluye que $\{\neg\}$ no puede ser funcionalmente completo.

BI: Si $\varphi = p$, con $p \in P$, la propiedad se cumple trivialmente.

HI: Supongamos que $\varphi \in L(P)$ construida usando sólo p y $\neg$ es equivalente a p o a $\neg p$.

TI: El único caso inductivo que tenemos que demostrar es $\psi = \neg\varphi$, pues sólo podemos usar el conectivo $\neg$. Por HI, sabemos que para toda valuación σ se cumple que $\sigma(\varphi) = \sigma(p)$ o $\sigma(\varphi) = \sigma(\neg p)$. Podemos hacer una tabla de verdad:

φ	$\psi = \neg\varphi$
p	$\neg p$
$\neg p$	p

Concluimos que ψ es equivalente a p o a $\neg p$. Por lo tanto, por el argumento dado al principio, tenemos que $\{\neg\}$ no es funcionalmente completo. $\square$

Matemáticas Discretas

Lógica proposicional

Nicolás Alvarado
nfalvarado@mat.uc.cl

Bernardo Barías
bjbarias@uc.cl

Sebastián Bugedo
bugedo@uc.cl

Gabriel Diéguez
gsdieguez@ing.puc.cl

Departamento de Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

23 de agosto de 2023