Lógica proposicional

Matemática estructural y lógica

ISIS-1104

Lógica como sistema formal

Lógica como lenguaje formal

Lógica como lenguaje formal

- El alfabeto:

$$A = \{\neg, \wedge, \vee, (,), \mathit{True}, \mathit{False}, p, q, r, s, ...\}$$

Lógica como lenguaje formal

- El alfabeto:

$$A = \{\neg, \wedge, \vee, (,), \mathit{True}, \mathit{False}, p, q, r, s, \ldots\}$$

- La sintaxis:

$$\begin{aligned}
\texttt{sentence} &\rightarrow \texttt{atomic_sentence} \mid \texttt{complex_sentence} \\
\texttt{atomic_sentence} &\rightarrow \boldsymbol{\mathit{True}} \mid \boldsymbol{\mathit{False}} \mid \boldsymbol{p} \mid \boldsymbol{q} \mid \boldsymbol{r} \mid \boldsymbol{s} \mid \ldots \\
\texttt{complex_sentence} &\rightarrow \texttt{unary_op sentence} \\
\texttt{complex_sentence} &\rightarrow (\texttt{sentence binary_op sentence}) \\
\texttt{unary_op} &\rightarrow \neg \\
\texttt{binary_op} &\rightarrow \wedge \mid \vee
\end{aligned}$$

Lógica como lenguaje formal

- Fórmulas bien formadas:

Lógica como lenguaje formal

- Fórmulas bien formadas:
    - $\neg p$

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$
  - $(p \wedge \neg r)$

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$
  - $(p \wedge \neg r)$
  - $(p \wedge (q \vee r))$

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$
  - $(p \wedge \neg r)$
  - $(p \wedge (q \vee r))$
  - $\neg(p \vee q)$

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$
  - $(p \wedge \neg r)$
  - $(p \wedge (q \vee r))$
  - $\neg(p \vee q)$
- ¿Qué semántica podemos asignarle a este lenguaje?

Lógica como lenguaje formal

- Fórmulas bien formadas:
  - $\neg p$
  - $(\neg s \vee q)$
  - $(p \wedge \neg r)$
  - $(p \wedge (q \vee r))$
  - $\neg(p \vee q)$
- ¿Qué semántica podemos asignarle a este lenguaje?
- ¿Qué aparato deductivo podemos asignarle a este lenguaje?

Semántica

Semántica

- Todas las **fórmulas** representan **proposiciones** o **afirmaciones**.

Semántica

- Todas las **fórmulas** representan **proposiciones** o **afirmaciones**.
- *True*/*False* representa la proposición que siempre es verdadera/falsa.

Semántica

- Todas las **fórmulas** representan **proposiciones** o **afirmaciones**.
- *True*/*False* representa la proposición que siempre es verdadera/falsa.
- p, q, r, $\ldots$ son **variables proposicionales** y pueden usarse para representar cualquier proposición.

$$p \equiv \text{hoy está lloviendo}$$

Semántica

- Todas las **fórmulas** representan **proposiciones** o **afirmaciones**.
- *True*/*False* representa la proposición que siempre es verdadera/falsa.
- p, q, r, ... son **variables proposicionales** y pueden usarse para representar cualquier proposición.

$$p \equiv \text{hoy está lloviendo}$$

- $\neg$, $\wedge$ y $\vee$ son **operaciones lógicas**.

Semántica

- Todas las **fórmulas** representan **proposiciones** o **afirmaciones**.
- *True*/*False* representa la proposición que siempre es verdadera/falsa.
- p, q, r, ... son **variables proposicionales** y pueden usarse para representar cualquier proposición.

$$p \equiv \text{hoy está lloviendo}$$

- $\neg$, $\wedge$ y $\vee$ son **operaciones lógicas**.
- para entender el significado de las operaciones lógicas hacemos uso de **tablas de verdad**.

El operador $\neg$

El operador $\neg$

- Si p es una proposición cualquiera, $\neg p$ representa la **negación de** p o simplemente **no** p.

El operador $\neg$

- Si p es una proposición cualquiera, $\neg p$ representa la **negación de** p o simplemente **no** p.
- Su tabla de verdad es:

p	$\neg p$
True	*False*
False	*True*

El operador $\neg$

- Si p es una proposición cualquiera, $\neg p$ representa la **negación de** p o simplemente **no** p.
- Su tabla de verdad es:

p	$\neg p$
True	*False*
False	*True*

- Ejemplo:

$$p \equiv \text{ahora está lloviendo}$$

$$\neg p \equiv \text{ahora no está lloviendo}$$

El operador $\wedge$

El operador $\wedge$

- Si p y q son dos proposiciones, $(p \wedge q)$ representa la **conjunción entre** p **y** q o simplemente p **y** q.

El operador $\wedge$

- Si p y q son dos proposiciones, $(p \wedge q)$ representa la **conjunción entre** p **y** q o simplemente p **y** q.
- Su tabla de verdad es:

p	q	$(p \wedge q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

El operador $\wedge$

- Si p y q son dos proposiciones, $(p \wedge q)$ representa la **conjunción entre** p **y** q o simplemente p **y** q.
- Su tabla de verdad es:

p	q	$(p \wedge q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

- Ejemplo:

$$p \equiv \text{ahora está lloviendo}$$

$$q \equiv \text{tengo mi sombrilla}$$

$$p \wedge q \equiv \text{ahora está lloviendo y tengo mi sombrilla}$$

El operador $\vee$

El operador $\vee$

- Si p y q son dos proposiciones, $(p \vee q)$ representa la **disyunción entre** p **y** q o simplemente p **o** q.

El operador $\vee$

- Si p y q son dos proposiciones, $(p \vee q)$ representa la **disyunción entre** p **y** q o simplemente p **o** q.
- Su tabla de verdad es:

p	q	$(p \wedge q)$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

El operador $\vee$

- Si p y q son dos proposiciones, $(p \vee q)$ representa la **disyunción entre** p **y** q o simplemente p **o** q.
- Su tabla de verdad es:

p	q	$(p \wedge q)$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

- Ejemplo:

$$p \equiv \text{ahora está lloviendo}$$

$$q \equiv \text{tengo mi sombrilla}$$

$$p \vee q \equiv \text{ahora está lloviendo o tengo mi sombrilla}$$

Combinando operadores

Combinando operadores

- Ahora podemos decidir el **valor de verdad** de cualquier **fórmula** de nuestro lenguaje.

Combinando operadores

- Ahora podemos decidir el **valor de verdad** de cualquier **fórmula** de nuestro lenguaje.
- Por ejemplo, la tabla de verdad de $(\neg p \wedge q)$:

p	q	$\neg p$	$(\neg p \wedge q)$
True	*True*	*False*	*False*
True	*False*	*False*	*False*
False	*True*	*True*	*True*
False	*False*	*True*	*False*

Combinando operadores

- Ahora ustedes calculen la tabla de verdad de $(\neg p \vee q)$

Extendiendo nuestro lenguaje

Extendiendo nuestro lenguaje

- Si p y q son dos proposiciones, $(p \Rightarrow q)$ representa el **condicional material entre** p **y** q o simplemente **si** p **entonces** q.

Extendiendo nuestro lenguaje

- Si p y q son dos proposiciones, $(p \Rightarrow q)$ representa el **condicional material entre p y q** o simplemente **si p entonces q**.
- Su tabla de verdad es:

p	q	$(p \Rightarrow q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

Extendiendo nuestro lenguaje

- Si p y q son dos proposiciones, $(p \Rightarrow q)$ representa el **condicional material entre** p **y** q o simplemente **si** p **entonces** q.
- Su tabla de verdad es:

p	q	$(p \Rightarrow q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

- Hay que tener cuidado con la interpretación:

Extendiendo nuestro lenguaje

- Si p y q son dos proposiciones, $(p \Rightarrow q)$ representa el **condicional material entre** p **y** q o simplemente **si** p **entonces** q.
- Su tabla de verdad es:

p	q	$(p \Rightarrow q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

- Hay que tener cuidado con la interpretación:
  - Si Colombia gana el próximo mundial, entonces Bogotá queda en Sudamérica.

Extendiendo nuestro lenguaje

- Si p y q son dos proposiciones, $(p \Rightarrow q)$ representa el **condicional material entre** p **y** q o simplemente **si** p **entonces** q.
- Su tabla de verdad es:

p	q	$(p \Rightarrow q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

- Hay que tener cuidado con la interpretación:
  - Si Colombia gana el próximo mundial, entonces Bogotá queda en Sudamérica.
  - Si ahora está lloviendo, entonces yo tengo mi sombrilla.

Otra extensión: Equivalencias

Otra extensión: Equivalencias

- Como $(\neg p \vee q)$ y $(p \Rightarrow q)$ siempre tienen el **mismo valor de verdad**, decimos que son **equivalentes**.

$$((\neg p \vee q) \equiv (p \Rightarrow q))$$

Otra extensión: Equivalencias

- Como $(\neg p \vee q)$ y $(p \Rightarrow q)$ siempre tienen el **mismo valor de verdad**, decimos que son **equivalentes**.

$$((\neg p \vee q) \equiv (p \Rightarrow q))$$

- Sin embargo son **fórmulas** distintas porque estan construidas con distintos **símbolos**.

Otra extensión: Equivalencias

- Como $(\neg p \vee q)$ y $(p \Rightarrow q)$ siempre tienen el **mismo valor de verdad**, decimos que son **equivalentes**.

$$((\neg p \vee q) \equiv (p \Rightarrow q))$$

- Sin embargo son **fórmulas** distintas porque estan construidas con distintos **símbolos**.
- La tabla de verdad de $(p \equiv q)$ es:

p	q	$(p \equiv q)$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*True*

El resultado final

El resultado final

- El alfabeto:

$$A = \{\neg, \wedge, \vee, \Rightarrow, \equiv, (,), True, False, p, q, r, s, ...\}$$

El resultado final

- El alfabeto:

$$A = \{\neg, \wedge, \vee, \Rightarrow, \equiv, (,), \mathit{True}, \mathit{False}, p, q, r, s, ...\}$$

- La sintaxis:

$$\begin{aligned}
\texttt{sentence} &\rightarrow \texttt{atomic_sentence} \mid \texttt{complex_sentence} \\
\texttt{atomic_sentence} &\rightarrow \boldsymbol{\mathit{True}} \mid \boldsymbol{\mathit{False}} \mid \boldsymbol{p} \mid \boldsymbol{q} \mid \boldsymbol{r} \mid \boldsymbol{s} \mid ... \\
\texttt{complex_sentence} &\rightarrow \texttt{unary_op sentence} \\
\texttt{complex_sentence} &\rightarrow (\texttt{sentence binary_op sentence}) \\
\texttt{unary_op} &\rightarrow \neg \\
\texttt{binary_op} &\rightarrow \wedge \mid \vee
\end{aligned}$$

De la lógica proposicional, los paréntesis y otros demonios

El problema

El problema

- Tengamos en cuenta la siguiente fórmula bien formada

$$((((\neg p \wedge q) \wedge r) \Rightarrow ((r \vee t) \vee u)) \wedge v)$$

El problema

- Tengamos en cuenta la siguiente fórmula bien formada

$$((((\neg p \wedge q) \wedge r) \Rightarrow ((r \vee t) \vee u)) \wedge v)$$

- El uso de paréntesis hace difícil su lectura, entonces

$$\neg p \wedge q \wedge r \Rightarrow r \vee t \vee u \wedge v$$

El problema

- Tengamos en cuenta la siguiente fórmula bien formada

$$((((\neg p \wedge q) \wedge r) \Rightarrow ((r \vee t) \vee u)) \wedge v)$$

- El uso de paréntesis hace difícil su lectura, entonces

$$\neg p \wedge q \wedge r \Rightarrow r \vee t \vee u \wedge v$$

- Necesitamos una convención para evitar ambigüedad.

Precedencia de operadores

Precedencia de operadores

Tomaremos una convención prestada de la aritmética

Precedencia de operadores

Tomaremos una convención prestada de la aritmética

- Las operaciones $*$ y $/$ tienen precedencia sobre $+$ y $-$:

$$2 * 3 + 5 = (2 * 3) + 5$$

$$6 - 8/2 = 6 - (8/2)$$

Precedencia de operadores

Tomaremos una convención prestada de la aritmética

- Las operaciones $*$ y $/$ tienen precedencia sobre $+$ y $-$:

$$2 * 3 + 5 = (2 * 3) + 5$$

$$6 - 8/2 = 6 - (8/2)$$

- Las operaciones con la misma precedencia son asociativas a izquierda:

$$2 - 3 + 5 = (2 - 3) + 5$$

$$10/5 * 2 = (10/5) * 2$$

Precedencia de operadores

Precedencia de operadores

- La precedencia de operadores lógicos que tomaremos es

$$\neg \text{ precede a } {}^{\wedge}\!/\!_{\vee} \text{ precede a } \Rightarrow \text{ precede a } \equiv$$

Precedencia de operadores

- La precedencia de operadores lógicos que tomaremos es

$$\neg \text{ precede a } {}^{\wedge}\!/\!_{\vee} \text{ precede a } \Rightarrow \text{ precede a } \equiv$$

- Las operaciones son asociativas a izquierda.

Precedencia de operadores

- La precedencia de operadores lógicos que tomaremos es

$$\neg \text{ precede a } \wedge/\vee \text{ precede a } \Rightarrow \text{ precede a } \equiv$$

- Las operaciones son asociativas a izquierda.
- Ejemplo

$$p \Rightarrow \neg q \wedge r \vee s \text{ es lo mismo que } (p \Rightarrow ((\neg q \wedge r) \vee s))$$

Precedencia de operadores

- La precedencia de operadores lógicos que tomaremos es

$$\neg \text{ precede a } \wedge/\vee \text{ precede a } \Rightarrow \text{ precede a } \equiv$$

- Las operaciones son asociativas a izquierda.
- Ejemplo

$$p \Rightarrow \neg q \wedge r \vee s \text{ es lo mismo que } (p \Rightarrow ((\neg q \wedge r) \vee s))$$

- Dado que esto es una convención, no lo incluiremos dentro de la sintaxis de nuestro lenguaje.