

Lógica proposicional (o booleana)

Stalin Muñoz y David A. Rosenblueth

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)
Universidad Nacional Autónoma de México (UNAM)

contenido

contenido

- 1 conceptos preliminares
- 2 sintaxis
- 3 semántica
- 4 inferencia

contenido

contenido

- 1 conceptos preliminares
- 2 sintaxis (lenguaje)
- 3 semántica (significado de lenguaje)
- 4 inferencia (cómo razonar)

2. preliminares (proposiciones)

proposición:

una afirmación que puede ser cierta o falsa.

2. preliminares (proposiciones)

proposición:

una afirmación que puede ser cierta o falsa.

ejemplos de proposiciones:

- El Sr. A iba en el tren.
- El Sr. A tiene un arma.
- El Sr. A cometió el homicidio.

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos*

“**y**”, “**o**”, “**si... entonces...**” ...

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos* “**y**”, “**o**”, “**si... entonces...**” ...

ejemplo 1:

- **Si** *el Sr. A iba en el tren* **y** *el Sr. A tiene un arma*,
entonces *el Sr. A cometió el homicidio*.

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos* “**y**”, “**o**”, “**si... entonces...**” ...

ejemplo 1:

- **Si** *el Sr. A iba en el tren* **y** *el Sr. A tiene un arma*,
entonces *el Sr. A cometió el homicidio*.
- *El Sr. A iba en el tren.*

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos* “y”, “o”, “**si... entonces...**” ...

ejemplo 1:

- **Si** *el Sr. A iba en el tren* **y** *el Sr. A tiene un arma*, **entonces** *el Sr. A cometió el homicidio*.
- *El Sr. A iba en el tren.*
- Por lo tanto:
Si *el Sr. A tiene un arma*, **entonces** *el Sr. A cometió el homicidio*.

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos*
“**y**”, “**o**”, “**si... entonces...**” ...

ejemplo 2:

- **Si** llovió **y** el Sr. K está mojado,
entonces el Sr. K salió de su casa.

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos* “y”, “o”, “**si... entonces...**” ...

ejemplo 2:

- **Si** llovió **y** el Sr. K está mojado, **entonces** el Sr. K salió de su casa.
- Llovió.

2. preliminares (oraciones)

oración:

combinamos proposiciones en *oraciones* con *conectivos lógicos* “y”, “o”, “**si... entonces...**” ...

ejemplo 2:

- **Si** *llovió y el Sr. K está mojado*, **entonces** *el Sr. K salió de su casa*.
- *Llovió*.
- Por lo tanto:
Si *el Sr. K está mojado*, **entonces** *el Sr. K salió de su casa*.

2. preliminares (oraciones)

oración:

combinamos proposiciones en oraciones con *conectivos lógicos* “y”, “o”, “si... entonces...” ...

estructura lógica

- ambos razonamientos tienen la misma estructura lógica
- a la lógica no le interesa tanto el significado de cada proposición, sino más bien la estructura lógica del razonamiento

3. sintaxis (fórmulas)

sintaxis de fórmulas

conjunto \mathcal{P} de variables proposicionales

fórmula es \perp

o \top

o p

o \neg *fórmula*

o *fórmula* \wedge *fórmula*

o *fórmula* \vee *fórmula*

o *fórmula* \rightarrow *fórmula*

o (*fórmula*)

donde $p \in \mathcal{P}$.

3. sintaxis (fórmulas)

sintaxis de fórmulas

conjunto \mathcal{P} de variables proposicionales

<i>fórmula</i> es \perp	“fondo”
o \top	“cima”
o p	variable proposicional
o \neg <i>fórmula</i>	negación (“no”)
o <i>fórmula</i> \wedge <i>fórmula</i>	conjunción (“y”)
o <i>fórmula</i> \vee <i>fórmula</i>	disyunción (“o”)
o <i>fórmula</i> \rightarrow <i>fórmula</i>	implicación (“si...entonces...”)
o (<i>fórmula</i>)	paréntesis

donde $p \in \mathcal{P}$.

4. semántica (tablas de verdad)

$$\frac{\perp}{\text{falso}}$$

$$\frac{\top}{\text{cierto}}$$

α	$\neg\alpha$
<i>falso</i>	<i>cierto</i>
<i>cierto</i>	<i>falso</i>

α	β	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	\dots
<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>cierto</i>	
<i>falso</i>	<i>cierto</i>	<i>falso</i>	<i>cierto</i>	<i>cierto</i>	
<i>cierto</i>	<i>falso</i>	<i>falso</i>	<i>cierto</i>	<i>falso</i>	
<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	

4. semántica (tablas de verdad)

tabla de verdad de fórmula arbitraria

a	b	c	$((a \wedge b) \rightarrow c) \wedge a$
<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>
<i>falso</i>	<i>falso</i>	<i>cierto</i>	<i>falso</i>
<i>falso</i>	<i>cierto</i>	<i>falso</i>	<i>falso</i>
<i>falso</i>	<i>cierto</i>	<i>cierto</i>	<i>falso</i>
<i>cierto</i>	<i>falso</i>	<i>falso</i>	<i>cierto</i>
<i>cierto</i>	<i>falso</i>	<i>cierto</i>	<i>cierto</i>
<i>cierto</i>	<i>cierto</i>	<i>falso</i>	<i>falso</i>
<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>

4. semántica (equivalencia lógica)

equivalencia lógica

$\alpha \equiv \beta$ cuando α y β tienen la misma tabla de verdad.

$$\underbrace{((a \wedge b) \rightarrow c) \wedge a}_{\alpha} \equiv \underbrace{(a \wedge (\neg b)) \vee (a \wedge b \wedge c)}_{\beta}$$

propiedades

$$\alpha \vee \beta \equiv \neg(\neg\alpha \wedge \neg\beta)$$

$$\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$$

4. semántica (formas normales)

literal:

variable proposicional o negación de variable proposicional

ℓ es p o $\neg p$ donde $p \in \mathcal{P}$.

forma normal disyuntiva = FND

$$\underbrace{(\ell_{1,1} \wedge \dots \wedge \ell_{1,n_1})}_{\text{término}} \vee (\ell_{2,1} \wedge \dots \wedge \ell_{2,n_2}) \vee \dots \vee (\ell_{m,1} \wedge \dots \wedge \ell_{m,n_m})$$

forma normal conjuntiva = FNC

$$\underbrace{(\ell_{1,1} \vee \dots \vee \ell_{1,n_1})}_{\text{cláusula}} \wedge (\ell_{2,1} \vee \dots \vee \ell_{2,n_2}) \wedge \dots \wedge (\ell_{m,1} \vee \dots \vee \ell_{m,n_m})$$

4. semántica (validez y satisfactibilidad)

fórmula satisfactible

tabla de verdad con *cierto* (al menos) un renglón del lado derecho: *fórmula satisfactible*.

tautología (o validez)

tabla de verdad con *cierto* en todos los renglones del lado derecho: *tautología* o *fórmula válida*.

satisfactibilidad y validez son *duales*

α es satisfactible si y solo si $\neg\alpha$ no es tautología.

4. semántica (modelo y consecuencia lógica)

modelo

modelo \mathcal{M} para una fórmula = renglón de la tabla de verdad

conjunto de modelos que hacen cierta una fórmula

$\mathbb{M}_{cierto}(\alpha)$ = conjunto de modelos que hacen cierta una fórmula

consecuencia lógica

“lógicamente implica”

$\alpha_1, \alpha_2, \dots, \alpha_n \models \beta$ si y solo si

$$\mathbb{M}_{cierto}(\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n) \subseteq \mathbb{M}_{cierto}(\beta)$$

4. semántica (ejemplo de consecuencia lógica)

a	b	c	$((a \wedge b) \rightarrow c) \wedge a$	$b \rightarrow c$
<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>	<i>cierto</i>
<i>falso</i>	<i>falso</i>	<i>cierto</i>	<i>falso</i>	<i>cierto</i>
<i>falso</i>	<i>cierto</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>
<i>falso</i>	<i>cierto</i>	<i>cierto</i>	<i>falso</i>	<i>cierto</i>
<i>cierto</i>	<i>falso</i>	<i>falso</i>	<i>cierto</i>	<i>cierto</i>
<i>cierto</i>	<i>falso</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>
<i>cierto</i>	<i>cierto</i>	<i>falso</i>	<i>falso</i>	<i>falso</i>
<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>	<i>cierto</i>

$$(a \wedge b) \rightarrow c, a \models b \rightarrow c$$

$\mathbb{M}_{cierto}(((a \wedge b) \rightarrow c) \wedge a) \subseteq \mathbb{M}_{cierto}(b \rightarrow c)$ i.e., preserva la verdad

4. semántica (oración)

ejemplo 1:

- **Si** *el Sr. A iba en el tren* **y** *el Sr. A tiene un arma*,
entonces *el Sr. A cometió el homicidio*.

$$(a \wedge b) \rightarrow c$$

- *El Sr. A iba en el tren.*

$$a$$

- Por lo tanto:

Si *el Sr. A tiene un arma*, **entonces** *el Sr. A cometió el homicidio*.

$$b \rightarrow c$$

5. inferencia (reglas de inferencia)

deducción natural

$$\frac{(a \wedge b) \rightarrow c \quad a}{b \rightarrow c}$$

$$\frac{\alpha \wedge \beta}{\alpha} \text{ eliminación de } \wedge$$

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta} \text{ modus ponens}$$

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta} \text{ introducción de } \wedge$$

y otras más

“produce”

$\alpha_1, \alpha_2, \dots, \alpha_n \vdash \beta$ si existen aplicaciones de reglas de inferencia que permiten convertir $\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n$ en β

5. inferencia (sistema de inferencia)

sistema de inferencia:

conjunto de reglas de inferencia

sistema de inferencia correcto:

si $\alpha_1, \alpha_2, \dots, \alpha_n \vdash \beta$, entonces $\alpha_1, \alpha_2, \dots, \alpha_n \models \beta$

sistema de inferencia completo:

si $\alpha_1, \alpha_2, \dots, \alpha_n \models \beta$, entonces $\alpha_1, \alpha_2, \dots, \alpha_n \vdash \beta$

5. inferencia (SAT)

dada α , ¿es α satisfactible?

α en forma normal disyuntiva = FND

$$\underbrace{(\ell_{1,1} \wedge \dots \wedge \ell_{1,n_1}) \vee (\ell_{2,1} \wedge \dots \wedge \ell_{2,n_2}) \vee \dots \vee (\ell_{m,1} \wedge \dots \wedge \ell_{m,n_m})}_{\text{término}}$$

fácil

α en forma normal conjuntiva = FNC

$$\underbrace{(\ell_{1,1} \vee \dots \vee \ell_{1,n_1}) \wedge (\ell_{2,1} \vee \dots \vee \ell_{2,n_2}) \wedge \dots \wedge (\ell_{m,1} \vee \dots \vee \ell_{m,n_m})}_{\text{cláusula}}$$

difícil

5. inferencia (SAT)

problema 1: SAT

dada α en FNC, ¿es α satisfactible?

problema 2: verificación de modelos

dados α y un modelo \mathcal{M} (renglón de la tabla de verdad de α),
¿ α vale *cierto* en \mathcal{M} ?

NP-completez

¿es más fácil verificación de modelos que SAT?

¡es sorprendente que no sabemos!

SAT es NP-completo (al igual que muchos otros problemas)
(todos los algoritmos conocidos toman un tiempo exponencial)

5. inferencia (complejidad de satisfactibilidad)

satisfactibilidad para lógica proposicional

satisfactibilidad para lógica proposicional es NP-completa

(todos los algoritmos conocidos toman un tiempo exponencial y nadie ha demostrado que existan algoritmos más eficientes)

5. Algoritmo DPLL

Algoritmo SAT para formulas en Forma Normal Conjuntiva

DPLL (Davis-Putnam-Logemann-Loveland):

un algoritmo de *búsqueda con retroceso* para resolver el problema SAT.

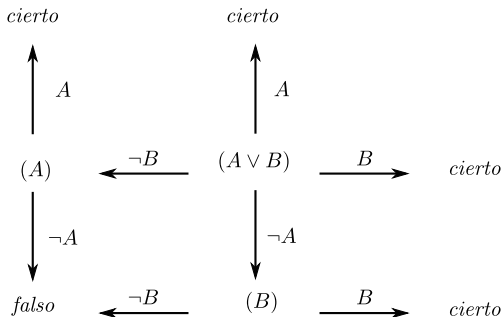
Búsqueda con retroceso

- 1 asigna de manera parcial una solución candidata,
- 2 refina la solución parcial mientras sea prometedora,
- 3 retrocede al punto de última elección al determinar que la solución parcial no conducirá a una solución completa, y
- 4 elige otras soluciones parciales mientras la solución completa no se encuentre.

5. Algoritmo DPLL

Objetivo

Simplificar la fórmula hasta llegar al valor de verdad *cierto* .



5. Algoritmo DPLL

Simplificar fórmulas con una literal

Si asignamos a la variable v el valor de verdad *cierto*, entonces formamos una literal $\ell = v$. Si por el contrario le asignamos el valor de *falso*, se forma la literal complementaria $\ell = \neg v$

- 1 Toda cláusula que contenga la literal ℓ se elimina de la formula simplificada porque es *cierta*.
- 2 Si una cláusula contiene la literal complementaria ℓ' borramos ℓ' de la cláusula porque es *falsa*.

Por ejemplo si $\phi = (A \vee B \vee \neg C) \wedge (\neg A \vee \neg B)$, entonces $\phi(B) = (\neg A)$.

5. Algoritmo DPLL

Al simplificar la fórmula:

- una fórmula vacía es satisfactible ya que todas las cláusulas de la formula original se borraron al ser ciertas,
- una cláusula vacía representa una cláusula falsa, ya que todas las literales se borraron, y
- la cláusula vacía implica que la formula es falsa, dado que está en conjunción con las otras cláusulas de la fórmula.

5. Algoritmo DPLL

- *cláusula unitaria* = cláusula que tiene una sola literal.

Se simplifica la fórmula asumiendo que la literal de una cláusula unitaria es *cierta*.

- *literal pura* = la literal complementaria no aparece en la fórmula.

Por ejemplo, en la fórmula siguiente A es pura, pero tanto B como C no lo son: $(A \vee B \vee C) \wedge (A \vee \neg B \wedge \neg C) \wedge (\neg C)$.

Si la formula original es satisfactible, al simplificarla con la literal pura, la formula resultante también es satisfactible.

5. Algoritmo DPLL

DPLL(ϕ)

Entrada: ϕ : Cadena con la fórmula en FNC.

Salida: *cierto* si la fórmula es satisfactible, falso de otra forma.

- 1 Si ϕ es vacía regresa *cierto*.
- 2 Si existe una cláusula vacía en ϕ regresa *falso*.
- 3 Si (ℓ) es una cláusula unitaria en ϕ regresa $\phi(\ell)$
- 4 Si existe una literal pura ℓ en ϕ regresa $\phi(\ell)$
- 5 Selecciona una variable v que aparece en ϕ
- 6 Si DPLL($\phi(v)$) regresa *cierto*
- 7 En caso contrario regresa DPLL($\phi(\neg v)$)

5. Algoritmo DPLL

Ejemplo

- DA = Detective en vagón A
- $S1A$ = Sospechoso 1 en vagón A
- $CS1$ = Cuchillo Sospechoso 1
- $SS1$ = Sangre en pañuelo de Sospechoso 1
- $S2A$ = Sospechoso 2 en vagón A
- $CS2$ = Cuchillo Sospechoso 2
- $SS2$ = Sangre en pañuelo de Sospechoso 2
- VA = Víctima en vagón A
- $S1MV$ = Sospechoso 1 mata a la víctima
- $S2MV$ = Sospechoso 2 mata a la víctima

5. Algoritmo DPLL

Ejemplo continuación

- El sospechoso 1 mató a la víctima o el sospechoso 2 mató a la víctima: $(S1MV \vee S2MV)$
- Si el Sospechoso 1 mató a la victima entonces tiene un cuchillo y sangre en su pañuelo además el sospechoso y la víctima están en el mismo vagón pero en distinto vagón al detective:

$$S1MV \rightarrow (CS1 \wedge SS1) \wedge ((S1A \wedge VA \wedge \neg DA) | (\neg S1A \wedge \neg VA \wedge DA))$$
- Similarmente para el sospechoso 2:

$$S2MV \rightarrow (CS2 \wedge SS2) \wedge ((S2A \wedge VA \wedge \neg DA) | (\neg S2A \wedge \neg VA \wedge DA))$$
- Alguien no estaba en el vagón A:

$$(\neg S1A \vee \neg S2A \vee \neg VA \vee \neg DA)$$

5. Algoritmo DPLL

Ejemplo continuación FNC

- $S1MV \rightarrow (CS1 \wedge SS1) \wedge ((S1A \wedge VA \wedge \neg DA) \vee (\neg S1A \wedge \neg VA \wedge DA))$
- $\neg S1MV \vee ((CS1 \wedge SS1) \wedge ((S1A \wedge VA \wedge \neg DA) \vee (\neg S1A \wedge \neg VA \wedge DA)))$
- $\neg S1MV \vee (CS1 \wedge SS1 \wedge (S1A \vee \neg S1A) \wedge (S1A \vee \neg VA) \wedge (S1A \vee DA) \wedge (VA \vee \neg S1A) \wedge (VA \vee \neg VA) \wedge (VA \vee DA) \wedge (\neg DA \vee \neg S1A) \wedge (\neg DA \vee \neg VA) \wedge (\neg DA \vee DA))$
- $(CS1 \vee \neg S1MV) \wedge (SS1 \vee \neg S1MV) \wedge (S1A \vee \neg VA \vee \neg S1MV) \wedge (S1A \vee DA \vee \neg S1MV) \wedge (VA \vee \neg S1A \vee \neg S1MV) \wedge (VA \vee DA \vee \neg S1MV) \wedge (\neg DA \vee \neg S1A \vee \neg S1MV) \wedge (\neg DA \vee \neg VA \vee \neg S1MV)$

5. Algoritmo DPLL

Formula completa en FNC

$$\begin{aligned}
 & (S1MV \vee S2MV) \wedge (CS1 \vee \neg S1MV) \wedge (SS1 \vee \neg S1MV) \wedge \\
 & (S1A \vee \neg VA \vee \neg S1MV) \wedge (S1A \vee DA \vee \neg S1MV) \wedge (VA \vee \neg S1A \vee \\
 & \neg S1MV) \wedge (VA \vee DA \vee \neg S1MV) \wedge (\neg DA \vee \neg S1A \vee \neg S1MV) \wedge \\
 & (\neg DA \vee \neg VA \vee \neg S1MV) \wedge (CS2 \vee \neg S2MV) \wedge (SS2 \vee \neg S2MV) \wedge \\
 & (S2A \vee \neg VA \vee \neg S2MV) \wedge (S2A \vee DA \vee \neg S2MV) \wedge (VA \vee \neg S2A \vee \\
 & \neg S2MV) \wedge (VA \vee DA \vee \neg S2MV) \wedge (\neg DA \vee \neg S1A \vee \neg S2MV) \wedge \\
 & (\neg DA \vee \neg VA \vee \neg S2MV) \wedge (\neg S1A \vee \neg S2A \vee \neg VA \vee \neg DA)
 \end{aligned}$$