Consecuencia lógica y satisfacibilidad

Clase 03

IIC 1253

Prof. Cristian Riveros

Outline

Modelación en lógica proposicional

Si Pedro estudia para la I1, entonces obtendrá un buena nota.

Pedro y Sofía estudiaron para la I1.

Por lo tanto, Pedro obtendrá una buena nota.

¿cómo formalizamos esta **deducción** el lógica proposicional?

¿Cuáles son nuestras proposiciones **básicas**?

$$\begin{aligned} PE &:= \text{Pedro estudia para la I1} \\ SE &:= \text{Sofía estudia para la I1} \\ BN &:= \text{Pedro obtiene una buena nota.} \end{aligned}$$

¿Cuáles son nuestras proposiciones **compuestas**?

$$\begin{aligned} PE \rightarrow BN &:= \text{Si Pedro estudia para la I1, entonces obtendrá un buena nota.} \\ PE \wedge SE &:= \text{Pedro y Sofía estudiaron para la I1} \end{aligned}$$

Modelación en lógica proposicional

$$\begin{array}{l} PE \rightarrow BN \\ PE \wedge SE \\ \hline BN \end{array}$$

$$\begin{array}{lcl} PE & := & \text{Pedro estudia para la I1} \\ SE & := & \text{Sofía estudia para la I1} \\ BN & := & \text{Pedro obtiene una buena nota.} \end{array}$$

¿por qué podemos hacer esta **deducción**?

PE	SE	BN	$PE \rightarrow BN$	$PE \wedge SE$	BN
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	1	1	1

Modelación en lógica proposicional (otro ejemplo)

$$\begin{array}{l} PE \;\vee\; SE \\ \neg PE \;\vee\; SE \\ \hline SE \end{array}$$

PE := Pedro estudia para la I1
SE := Sofía estudia para la I1
BN := Pedro obtiene una buena nota.

¿por qué podemos hacer esta **deducción**?

PE	SE	$PE \vee SE$	$\neg PE \vee SE$	SE
0	0	0	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	1	1

Modelación en lógica proposicional (anti-ejemplo)

$$\begin{array}{l} PE \rightarrow BN \\ PE \vee SE \\ \hline BN \quad \times \end{array}$$

PE := Pedro estudia para la I1
SE := Sofía estudia para la I1
BN := Pedro obtiene una buena nota.

¿por qué esta **deducción** es **errónea**?

PE	SE	BN	$PE \rightarrow BN$	$PE \vee SE$	BN	
0	0	0	1	0	0	
0	0	1	1	0	1	
0	1	0	1	1	0	×
0	1	1	1	1	1	
1	0	0	0	1	0	
1	0	1	1	1	1	
1	1	0	0	1	0	
1	1	1	1	1	1	

Consecuencia lógica

Sea $\Sigma = \{\varphi_1, \ldots, \varphi_m\}$ un conjunto de formulas con variables $p_1, \ldots, p_n$.

Definición

- Diremos que φ es **consecuencia lógica** de Σ si, y solo si, para toda valuación $v_1, \ldots, v_n$ se tiene que:

  si para todo $i \leq m$ $\varphi_i(v_1, \ldots, v_n) = 1$, entonces $\varphi(v_1, \ldots, v_n) = 1$.

- Si φ es consecuencia lógica de Σ, entonces escribiremos $\Sigma \vDash \varphi$.

- Diremos que $\varphi_1, \ldots, \varphi_m$ son **premisas** y φ la **conclusión**.

Ejemplo

- $\{ q \to p, \ q \wedge s \} \vDash p$
- $\{ p \vee q, \ \neg p \} \vDash q$

Algunas consecuencias lógicas clásicas

Consecuencias lógicas

1. **Modus ponens**: $\{p, p \to q\} \models q$

p	q	p	$p \to q$	q
0	0	0	1	0
0	1	0	1	1
1	0	1	0	0
1	1	1	1	1

2. **Modus tollens**: $\{\neg q, p \to q\} \models \neg p$

p	q	$\neg q$	$p \to q$	$\neg p$
0	0	1	1	1
0	1	0	1	1
1	0	1	0	0
1	1	0	1	0

Algunas consecuencias lógicas clásicas

Consecuencias lógicas

3. **Resolución**: $\{\ p \vee q,\ \neg q \vee r\ \} \models\ p \vee r$

p	q	r	$p \vee q$	$\neg q \vee r$	$p \vee r$
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Sobre consecuencia lógica

¿cuales de las siguientes afirmaciones son verdaderas?

- $\{1\} \models \varphi$ entonces φ es una **tautología**. ✓
- Si α es una **contradicción**, entonces $\{\alpha\} \models \varphi$ para toda formula φ. ✓
- Si $\{\varphi_1, \ldots, \varphi_m\} \models \varphi$,
  entonces $\{\varphi_1, \ldots, \varphi_m, \alpha\} \models \varphi$ para toda formula α. ✓

Demuestre estas afirmaciones.

Algunas reglas de consecuencia lógica

1. **Modus ponens**: $\{\ p,\ p \to q\ \} \vDash q$
2. **Modus tollens**: $\{\ \neg q,\ p \to q\ \} \vDash \neg p$
3. **Silogismo**: $\{\ p \to q,\ q \to r\ \} \vDash p \to r$
4. **Silogismo disyuntivo**: $\{\ p \vee q,\ \neg p\ \} \vDash q$
5. **Conjunción**: $\{\ p,\ q\ \} \vDash p \wedge q$
6. **Simplificación conjuntiva**: $\{\ p \wedge q\ \} \vDash p$
7. **Aplificación disyuntiva**: $\{\ p\ \} \vDash p \vee q$
8. **Demostración condicional**: $\{\ p \wedge q,\ p \to (q \to r)\ \} \vDash r$
9. **Demostración por casos**: $\{\ p \to r,\ q \to r\ \} \vDash (p \vee q) \to r$

Demuestre cada una de las consecuencias lógicas

Composición y consecuencia lógica

Sean $\Sigma = \{\varphi_1(p_1, \ldots, p_n), \ldots, \varphi_m(p_1, \ldots, p_n)\}$ y $\beta_1, \ldots \beta_n$ formulas proposicionales.

Definición

La **composición** $\Sigma(\beta_1, \ldots, \beta_n)$ es el conjunto resultante de componer cada formula en Σ con $\beta_1, \ldots, \beta_n$, esto es:

$$\Sigma(\beta_1, \ldots, \beta_n) = \{\varphi_1(\beta_1, \ldots, \beta_n), \ldots, \varphi_m(\beta_1, \ldots, \beta_n)\}$$

Teorema

Sean Σ un conjunto de formulas y $\varphi(p_1, \ldots, p_n)$, $\beta_1, \ldots, \beta_n$ formulas.

Si $\Sigma \vDash \varphi$, entonces $\Sigma(\beta_1, \ldots, \beta_n) \vDash \varphi(\beta_1, \ldots, \beta_n)$.

Demuestre este teorema (muy similar al caso de equivalencia lógica)

¿cómo demostramos que $\Sigma \models \varphi$?

1. Verificando todas las valuaciones (tabla de verdad).

2. **Deducimos** φ desde Σ usando alguna de las reglas anteriores.

¿cómo demostramos que $\Sigma \models \varphi$?

Ejemplo

¿ es verdad que $\{\ p,\ p \to q,\ s \lor r,\ \neg s \land \neg t\ \} \models q \land r$?

1.	p	(Premisa)
2.	$p \to q$	(Premisa)
3.	q	(Modus Ponens 1 y 2)
4.	$s \lor r$	(Premisa)
5.	$\neg s \to r$	(equivalencia con 4.)
6.	$\neg s \land \neg t$	(Premisa)
7.	$\neg s$	(Simplificación conjuntiva 6)
8.	r	(Modus Ponens 5 y 7)
9.	$q \land r$	(Conjunción 3 y 8)

¿alguna estrategia mejor?

Outline

Satisfacción de un conjunto de formulas

Definiciones

- $\varphi(p_1, \ldots, p_n)$ se dice **satisfacible** si existe una valuación $v_1, \ldots, v_n$:

$$\varphi(v_1, \ldots, v_n) = 1$$

- Σ se dice **satisfacible** si existe una valuación $v_1, \ldots, v_n$:

$$\Sigma(v_1, \ldots, v_n) = 1$$

- Σ es **inconsistente** si NO es satisfacible.

¿qué propiedad cumple la tabla de verdad de una formula satisfacible?
¿y la del conjunto?

Satisfacción de un conjunto de formulas

Definiciones

- $\varphi(p_1, \ldots, p_n)$ se dice **satisfacible** si existe una valuación $v_1, \ldots, v_n$:

$$\varphi(v_1, \ldots, v_n) \;=\; 1$$

- Σ se dice **satisfacible** si existe una valuación $v_1, \ldots, v_n$:

$$\Sigma(v_1, \ldots, v_n) \;=\; 1$$

- Σ es **inconsistente** si NO es satisfacible.

¿cuál de las siguientes formulas/conjuntos son satisfacibles?

- $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$
- $\{\ \neg q \vee p,\ \ q \vee \neg r,\ \ \neg p \vee r\ \}$
- $\{\ \neg q \vee p,\ \ \neg p \vee r,\ \ \neg r \vee q,\ \ p \vee q \vee r,\ \ \neg p \vee \neg q \vee \neg r\ \}$

Consecuencia lógica vs satisfacibilidad

Teorema

$\{\varphi_1, \ldots, \varphi_m\} \vDash \varphi$ si, y solo si, $\{\varphi_1, \ldots, \varphi_m, \neg\varphi\}$ es **inconsistente**.

Demostración ($\Rightarrow$)

Suponga que $\{\varphi_1, \ldots, \varphi_m\} \vDash \varphi$.

PD: para toda $v_1, \ldots, v_n$ se cumple que $\{\varphi_1, \ldots, \varphi_m, \neg\varphi\}(v_1, \ldots, v_n) = 0$.

Tomamos una valuación cualquiera $v_1, \ldots, v_n$ y:

1. Si $\{\varphi_1, \ldots, \varphi_m\}(v_1, \ldots, v_n) = 0$,

   $\therefore \ \{\varphi_1, \ldots, \varphi_m, \neg\varphi\}(v_1, \ldots, v_n) = 0$ ✓

2. Si $\{\varphi_1, \ldots, \varphi_m\}(v_1, \ldots, v_n) = 1$, entonces:
   - $\varphi(v_1, \ldots, v_n) = 1$ (¿por qué?)
   - $\neg\varphi(v_1, \ldots, v_n) = 0$

   $\therefore \ \{\varphi_1, \ldots, \varphi_m, \neg\varphi\}(v_1, \ldots, v_n) = 0$ ✓

Consecuencia lógica vs satisfacibilidad

Teorema

$\{\varphi_1, \ldots, \varphi_m\} \vDash \varphi$ si, y solo si, $\{\varphi_1, \ldots, \varphi_m, \neg\varphi\}$ es **inconsistente**.

Demostración ($\Leftarrow$)

Suponga que $\{\varphi_1, \ldots, \varphi_m, \neg\varphi\}$ es **inconsistente**.

PD1: para toda $v_1, \ldots, v_n$,
si $\{\varphi_1, \ldots, \varphi_m\}(v_1, \ldots, v_n) = 1$, entonces $\varphi(v_1, \ldots, v_n) = 1$.

Sea $v_1, \ldots, v_n$ una valuación cualquiera tal que $\{\varphi_1, \ldots, \varphi_m\}(v_1, \ldots, v_n) = 1$.

PD2: $\varphi(v_1, \ldots, v_n) = 1$.

$$
\begin{array}{lll}
\{\varphi_1, \ldots, \varphi_m\}(v_1, \ldots, v_n) = 1 & \text{entonces} & \neg\varphi(v_1, \ldots, v_n) = 0 \quad \text{(¿por qué?)} \\
 & \text{entonces} & \varphi(v_1, \ldots, v_n) = 1
\end{array}
$$

$\square$

Satisfacibilidad y representación de problemas

Problema

Dada una fórmula φ, queremos verificar si φ es **satisfacible**.

¿cómo podemos hacer esto **eficientemente**?

- El problema de satisfacción es un problema fundamental tanto en ciencia de la computación como en ingeniería.
- Muchos otros problemas pueden ser resueltos usando este problema.

Sudoku

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

(Libro Rosen)

¿tiene solución este sudoku?

Sudoku

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Reglas del Sudoku

1. Cada fila debe tener los dígitos $\{1, 2, \ldots, 9\}$.
2. Cada columna debe tener los dígitos $\{1, 2, \ldots, 9\}$.
3. Cada cuadrado 3x3 debe tener los dígitos $\{1, 2, \ldots, 9\}$.
4. Cada celda puede contener un solo dígito.

¿cómo usamos satisfacibilidad para resolver sudoku?

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Para un sudoku S cualquiera, vamos a construir un conj, de formulas Σ_S:

S tiene solución si, y solo si, Σ_S es **satisfacible**.

¿cuáles son nuestras variables proposicionales?

¿cuáles son nuestras variables proposicionales?

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Para $1 \leq i \leq 9$, $1 \leq j \leq 9$ y $1 \leq n \leq 9$, definimos la variable $p_{i,j,n}$ tal que:

$p_{i,j,n}$ será **verdadero** ssi colocamos el número n en el casillero (i,j)

(¿cuántas variables tenemos?)

¿cómo nos aseguramos que una valuación
para las variables $p_{i,j,n}$ codifique una solución para S?

¿cómo codificamos los valores iniciales del tablero?

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Para cada casillero (i, j) no-vacío en S con valor n, tendremos una formula:

$$p_{i,j,n}$$

Ejemplo: para nuestro tablero sudoku

$$p_{1,2,2} \ , \ p_{1,3,9} \ , \ p_{1,7,4} \ , \ p_{2,4,5} \ , \ p_{2,7,1} \ , \ \ldots , \ p_{9,8,6}$$

¿cómo codificamos las otras restricciones?

1. ¿cómo verificamos que cada fila contiene todos los números?

$$\bigwedge_{i=1}^{9} \left(\bigwedge_{n=1}^{9} \left(\bigvee_{j=1}^{9} p_{i,j,n} \right) \right)$$

2. ¿cómo verificamos que cada columna contiene todos los números?

$$\bigwedge_{j=1}^{9} \left(\bigwedge_{n=1}^{9} \left(\bigvee_{i=1}^{9} p_{i,j,n} \right) \right)$$

¿cómo codificamos las otras restricciones?

3. ¿cómo verificamos que cada cuadrado de 3x3 tiene todos los números?

$$\bigwedge_{i=0}^{2} \bigwedge_{j=0}^{2} \bigwedge_{n=1}^{9} \left(\bigvee_{k=1}^{3} \bigvee_{l=1}^{3} p_{3i+k,3j+l,n} \right)$$

4. ¿cómo verificamos que cada celda tenga un solo valor?

$$\bigwedge_{i=1}^{9} \bigwedge_{j=1}^{9} \left(\bigwedge_{n=1}^{9} \left(p_{i,j,n} \rightarrow \bigwedge_{\substack{1 \leq n' \leq 9, \\ n \neq n'}} \neg p_{i,j,n'} \right) \right)$$

¿cómo usamos satisfacibilidad para resolver sudoku?

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Para un tablero S, considere el conjunto Σ_S de todas las formulas definidas anteriormente. Entonces tendremos que:

$$S \text{ tiene solución} \quad \text{si, y solo si,} \quad \Sigma_S \text{ es satisfacible.}$$

Demuestre esta última afirmación.