Tema 3: Semántica proposicional

José A. Alonso Jiménez
Joaquín Borrego Díaz
Antonia Chávez González

Dpto. de Ciencias de la Computación e Inteligencia Artificial

Universidad de Sevilla

Sintaxis de la lógica proposicional

- **Alfabeto proposicional:**
  - símbolos proposicionales.
  - conectivas lógicas:
    $\neg$ (negación),
    $\wedge$ (conjunción),
    $\vee$ (disyunción),
    $\rightarrow$ (condicional),
    $\leftrightarrow$ (equivalencia).
  - símbolos auxiliares: "(" y ")".
- **Fórmulas proposicionales:**
  - símbolos proposicionales
  - $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, $(F \leftrightarrow G)$
- **Eliminación de paréntesis:**
  - Eliminación de paréntesis externos.
  - Precedencia: $\neg, \wedge, \vee \rightarrow, \leftrightarrow$
  - Asociatividad: $\wedge$ y $\vee$ asocian por la derecha
- **Sintaxis en OTTER**

Usual	$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
OTTER	`-`	`&`	`|`	`->`	`<->`

Semántica: Verdad e interpretación

- **Valores de verdad:**
  - 1: verdadero (en MACE, `T`)
  - 0: falso (en MACE, `F`)
- **Interpretaciones:**
  - Ejemplo 1:
    I_1(rojo_sobre_amarillo) = 1
    I_1(amarillo_sobre_morado) = 1
    I_1(amarillo_sobre_rojo) = 0
  - Ejemplo 1:
    I_2(rojo_sobre_amarillo) = 1
    I_2(amarillo_sobre_morado) = 0
    I_2(amarillo_sobre_rojo) = 1
- **Funciones de verdad:**

i	$\neg i$
1	0
0	1

i	j	$i \wedge j$	$i \vee j$	$i \longrightarrow j$	$i \leftrightarrow j$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Modelo de una fórmula

- **Significado de una fórmula en una interpretación**
  - **Fórmula:** $F = (p \vee q) \wedge (\neg q \vee r)$
  - **Significado de F en la interpretación I:**
    $I(p) = I(r) = 1, I(q) = 0$

$$\begin{array}{ccccccc}
(p & \vee & q) & \wedge & (\neg q & \vee & r) \\
(1 & \vee & 0) & \wedge & (\neg 0 & \vee & 1) \\
 & 1 & & \wedge & (1 & \vee & 1) \\
 & 1 & & \wedge & & 1 & \\
 & & & 1 & & &
\end{array}$$

    Luego, F es un válida en I

  - **Significado de F en la interpretación J:**
    $J(r) = 1, J(p) = J(q) = 0$

$$\begin{array}{ccccccc}
(p & \vee & q) & \wedge & (\neg q & \vee & r) \\
(0 & \vee & 0) & \wedge & (\neg 0 & \vee & 1) \\
 & 0 & & \wedge & (1 & \vee & 1) \\
 & 0 & & \wedge & & 1 & \\
 & & & 0 & & &
\end{array}$$

    Luego, F no es válida en J

- **Modelo de una fórmula**
  - $I \models F$ **syss significado**$(F, I) = 1$
  - I **es un modelo de** F $(I \models F)$
  - J **no es un modelo de** F $(J \not\models F)$

Comprobación de modelo con MACE

- **Comprobación de modelos: Ejemplo 1**
  - **Problema: Determinar si la la interpretación I definida por**
    $$I(p) = I(r) = 1, I(q) = 0$$
    es un modelo de la fórmula
    $$(p \vee q) \wedge (\neg q \vee r)$$
  - **Entrada** (`comprobacion_modelo_fla_1.in`)

```
formula_list(usable).
 (p | q) & (-q | r).
end_of_list.

list(mace_constraints).
 assign(p,T).
 assign(q,F).
 assign(r,T).
end_of_list.
```

  - **Cálculo con MACE:**

```
mace -p <comprobacion_modelo_fla_1.in
        >comprobacion_modelo_fla_1.out
```

  - **Salida** (`comprobacion_modelo_fla_1.out`)

```
======================= Model #1 at 0.00 seconds:
 p: T
 q: F
 r: T
end_of_model
```

  - **Conclusión: $I \models F$**

Comprobación de modelo con MACE

• Comprobación de modelos: Ejemplo 2

- **Problema: Determinar si la la interpretación J definida por**
  $$J(r) = 1, J(p) = J(q) = 0$$
  es un modelo de la fórmula
  $$(p \vee q) \wedge (\neg q \vee r)$$
- **Entrada (`comprobacion_modelo_fla_2.in`):**

```
formula_list(usable).
 (p | q) & (-q | r).
end_of_list.

list(mace_constraints).
 assign(p,F).
 assign(q,F).
 assign(r,T).
end_of_list.
```

- **Cálculo con MACE:**

```
mace -p <comprobacion_modelo_fla_2.in
        >comprobacion_modelo_fla_2.out
```

- **Salida (`comprobacion_modelo_fla_2.out`)**

```
The search is complete.  No models were found.
```

- **Conclusión: $J \not\models F$**

Fórmulas satisfacibles e insatisfacibles

- **Fórmulas satisfacibles e insatisfacibles:**
  - **Def.:** F es satisfacible syss F tiene modelo
  - **Def.:** F es insatisfacible syss F no tiene modelo
  - **Ejemplo:**
    $(p \rightarrow q) \land (q \rightarrow r)$ es satisfacible
    $I(p) = I(q) = I(r) = 0$
    $p \land \neg p$ es insatisfacible

p	$\neg p$	$p \land \neg p$
1	0	0
0	1	0

Satisfacibilidad con MACE

- **Satisfacibilidad con MACE: Ejemplo 1**
  - **Problema: Determinar si la fórmula**
    $$(p \rightarrow q) \land (q \rightarrow r)$$
    es satisfacible.
  - **Entrada:** `satisfacibilidad_1.in`

```
formula_list(sos).
 (p -> q) & (q -> r).
end_of_list.
```

  - **Ejecución:**

```
mace -p <satisfacibilidad_1.in >satisfacibilidad_1.out
```

  - **Salida:** `satisfacibilidad_1.out`

```
=== Model #1 at 0.00 seconds:
 p: F
 q: F
 r: F
end_of_model
```

  - **Conclusión: la fórmula $(p \rightarrow q) \land (q \rightarrow r)$ es satisfacible (y un modelo es I con $I(p) = I(q) = I(r) = F$)**

Satisfacibilidad con MACE

- **Satisfacibilidad con MACE: Ejemplo 2**
  - **Problema: Determinar si la fórmula**

    $p \wedge \neg p$

    es satisfacible.
  - **Entrada:** `satisfacibilidad_2.in`

```
formula_list(sos).
 p & -p.
end_of_list.
```

  - **Ejecución:**

```
mace -p <satisfacibilidad_2.in >satisfacibilidad_2.out
```

  - **Salida:** `satisfacibilidad_2.out`

```
The search is complete.  No models were found.
```

  - **Conclusión: fórmula $p \wedge \neg p$ no es satisfacible.**

Fórmulas válidas

- **Fórmula válida (tautología):**
  - Def.: F es válida syss
    toda interpretación de F es modelo de F
  - Representación: $\models F$
  - Ejemplo: $\models (p \rightarrow q) \vee (q \rightarrow p)$

p	q	r	$(p \rightarrow q)$	$(q \rightarrow r)$	$(p \rightarrow q) \vee (q \rightarrow r)$
1	1	1	1	1	1
1	1	0	1	0	1
1	0	1	0	1	1
1	0	0	0	1	1
0	1	1	1	1	1
0	1	0	1	0	1
0	0	1	1	1	1
0	0	0	1	1	1

  - Ejemplo: $\not\models (p \rightarrow q)$

p	q	$p \rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

Satisfacibilidad y validez

- Los problemas de satisfacibilidad y validez:
  - Problema de la satisfacibilidad:
    Dada F determinar si es satisfacible.
  - Problema de la validez:
    Dada F determinar si es válida
- Relaciones entre validez y satisfacibilidad:
  - F es válida $\iff$ $\neg F$ es insatisfacible
  - F es válida $\implies$ F es satisfacible
  - F es satisfacible $\not\Longrightarrow$ $\neg F$ es insatisfacible
    $p \rightarrow q$ es satisfacible
    $I(p) = I(q) = 1$
    $\neg(p \rightarrow q)$ es satisfacible
    $I(p) = 1, I(q) = 0$
- El problema de la satisfacibilidad es NP–completo

Validez con MACE

- **Validez con MACE: Ejemplo 1**
  - **Problema: Determinar si la fórmula**
    $$(p \rightarrow q) \lor (q \rightarrow p)$$
    es válida.
  - **Entrada (`validez_1.in`):**

```
formula_list(usable).
 -((p -> q) | (q -> p)).
end_of_list.
```

  - **Ejecución:**

```
mace -p <validez_1.in >validez_1.out
```

  - **Salida (`validez_1.out`):**

```
The search is complete.  No models were found.
```

  - **Conclusión: la fórmula $(p \rightarrow q) \lor (q \rightarrow p)$ es válida.**

Validez con MACE

- **Validez con MACE: Ejemplo 2**
  - **Problema: Determinar si la fórmula**

    $$p \rightarrow q$$

    es válida.
  - **Entrada** (`validez_2.in`):

```
formula_list(usable).
 -(p -> q).
end_of_list.
```

  - **Ejecución:**

```
mace -p <validez_2.in >validez_2.out
```

  - **Salida** (`validez_2.out`):

```
=== Model #1 at 0.00 seconds:
 p: T
 q: F
end_of_model
```

  - **Conclusión: la fórmula $p \rightarrow q$ no es válida, un contraejemplo es la interpretación I con $I(p) = T, I(q) = F$.**

Modelos de un conjunto de fórmulas

- **Modelo de un conjunto de fórmulas:**
  - **Def.:** I es modelo de S syss para toda $F \in S, I \models F$
  - **Representación:** $I \models S$
  - **Ejemplo:**
    $S = \{(p \vee q) \wedge (\neg q \vee r), q \rightarrow r\}$
    $I_1(p) = 1, I_1(q) = 0, I_1(r) = 1 \Longrightarrow I_1 \models S$
    $I_2(p) = 0, I_2(q) = 1, I_2(r) = 0 \Longrightarrow I_2 \not\models S$

Comprobación de modelos con MACE

- **Comprobación de modelos: Ejemplo 1**
  - **Problema: Determinar si la interpretación I definida por**
    $$I(p) = 1, I_1(q) = 0, I_1(r) = 1$$
    es un modelo del conjunto de fórmulas
    $$S = \{(p \vee q) \wedge (\neg q \vee r), q \rightarrow r\}$$
  - **Entrada** (`comprobacion_modelo_conjunto_1.in`)**:**

```
formula_list(usable).
 (p | q) & (-q | r).
 q -> r.
end_of_list.

list(mace_constraints).
 assign(p,T).
 assign(q,F).
 assign(r,T).
end_of_list.
```

  - **Ejecución:**

```
mace -p <comprobacion_modelo_conjunto_1.in
        >comprobacion_modelo_conjunto_1.out
```

  - **Salida** (`comprobacion_modelo_conjunto_1.out`)**:**

```
==== Model #1 at 0.01 seconds:
 p: T
 q: F
 r: T
end_of_model
The set is satisfiable (1 model(s) found).
```

  - **Conclusión: $I \models S$.**

Comprobación de modelos con MACE

• Comprobación de modelos: Ejemplo 2

- **Problema: Determinar si la interpretación I definida por**

  $I(p) = 0, I_1(q) = 1, I_1(r) = 0$

  es un modelo del conjunto de fórmulas

  $S = \{(p \lor q) \land (\neg q \lor r), q \to r\}$

- **Entrada** (`comprobacion_modelo_conjunto_2.in`):

```
formula_list(usable).
 (p | q) & (-q | r).
 q -> r.
end_of_list.

list(mace_constraints).
 assign(p,F).
 assign(q,T).
 assign(r,F).
end_of_list.
```

- **Ejecución:**

```
mace -p <comprobacion_modelo_conjunto_2.in
        >comprobacion_modelo_conjunto_2.out
```

- **Salida** (`comprobacion_modelo_conjunto_2.out`):

```
The search is complete. No models were found.
```

- **Conclusión:** $I \not\models S$.

Conjuntos consistentes

- **Conjunto consistente de fórmulas:**
  - **Def.: S es consistente syss S tiene modelo**
  - **Def.: S es inconsistente syss S no tiene modelo**
  - $\{(p \lor q) \land (-q \lor r), p \to r\}$ **es consistente**
    $\{I_4, I_6, I_8\}$
  - $\{(p \lor q) \land (-q \lor r), p \to r, \neg r\}$ **es inconsistente**

	p	q	r	$(p \lor q)$	$(\neg q \lor r)$	$(p \lor q) \land (\neg q \lor r)$	$p \to r$	$\neg r$
I_1	0	0	0	0	1	0	1	1
I_2	0	0	1	0	1	0	1	0
I_3	0	1	0	1	0	0	1	1
I_4	0	1	1	1	1	1	1	0
I_5	1	0	0	1	1	1	0	1
I_6	1	0	1	1	1	1	1	0
I_7	1	1	0	1	0	0	0	1
I_8	1	1	1	1	1	1	1	0

Consistencia con MACE

- $\{(p \vee q) \wedge (-q \vee r), p \rightarrow r\}$ **es consistente**
  - **Entrada:** `ej-consistencia.in`

```
formula_list(sos).
 (p | q) & (-q | r).
 p -> r.
end_of_list.
```

  - **Ejecución:** `mace -p <ej-consistencia.in`
  - **Salida:**

```
=== Model #1 at 0.00 seconds:
 p: T
 q: F
 r: T
end_of_model
```

- $\{(p \vee q) \wedge (-q \vee r), p \rightarrow r, \neg r\}$ **inconsistente**
  - **Entrada:** `ej-inconsistencia.in`

```
formula_list(sos).
 (p | q) & (-q | r).
 p -> r.
 -r.
end_of_list.
```

  - **Ejecución:** `mace -p <ej-inconsistencia.in`
  - **Salida:**

```
The search is complete.  No models were found.
```

Consecuencia lógica

- **Consecuencia lógica:**
  - **Def.: F es consecuencia de S syss todos los modelos de S son modelos de F**
  - **Representación: $S \models F$**
  - **Ejemplo: $\{p \to q, q \to r\} \models p \to r$**

	p	q	r	$p \to q$	$q \to r$	$p \to r$
I_1	0	0	0	1	1	1
I_2	0	0	1	1	1	1
I_3	0	1	0	1	0	1
I_4	0	1	1	1	1	1
I_5	1	0	0	0	1	0
I_6	1	0	1	0	1	1
I_7	1	1	0	1	0	0
I_8	1	1	1	1	1	1

  - **Ejemplo: $\{p\} \not\models p \land q$**

p	q	$p \land q$
1	1	1
1	0	0
0	1	0
0	0	0

Consecuencia, validez y consistencia

- **Relación entre consecuencia lógica, validez, satisfacibilidad y consistencia:**
  Las siguientes condiciones son equivalentes:
  - $\{F_1, \ldots, F_n\} \models G$
  - $\models F_1 \land \ldots \land F_n \to G$
  - $\neg(F_1 \land \ldots \land F_n \to G)$ es insatisfacible
  - $\{F_1, \ldots, F_n, \neg G\}$ es inconsistente

Consecuencia lógica con MACE

- **Ejemplo 1: $\{p \leftrightarrow q, r \leftrightarrow (p \wedge q)\} \not\models p \wedge r$**
  - **Entrada:**

```
formula_list(sos).
 p <->q.
 r <-> (p & q).
 -(p & r).
end_of_list.
```

  - **Salida:**

```
==== Model #1 at 0.00 seconds:
 p: F
 q: F
 r: F
end_of_model
```

- **Ejemplo 2: $\{p \leftrightarrow q, r \leftrightarrow (p \wedge q)\} \models p \leftrightarrow r$**
  - **Entrada:**

```
formula_list(sos).
 p <->q.
 r <-> (p & q).
 -(p <-> r).
 end_of_list.
```

  - **Salida:**

```
The search is complete.  No models were found.
```

Problema de los animales con MACE

- **Base de conocimiento**
  - Base de reglas:
    * R1: Si el animal tiene pelos es mamífero.
    * R2: Si el animal da leche es mamífero.
    * R3: Si el animal es un mamífero y tiene pezuñas es ungulado.
    * R4: Si el animal es un mamífero y rumia es ungulado.
    * R5: Si el animal es un ungulado y tiene cuello largo es una jirafa.
    * R6: Si el animal es un ungulado y tiene rayas negras es una cebra.
  - Base de hechos:
    * H1: El animal tiene pelos.
    * H2: El animal tiene pezuñas.
    * H3: El animal tiene rayas negras.
  - Consecuencia
    * El animal es una cebra.

Problema de los animales con MACE

- **Solución con MACE**
  - **Entrada** (`ej-animales.in`)

```
formula_list(sos).
tiene_pelos | da_leche -> es_mamifero.
es_mamifero & (tiene_pezuñas | rumia) -> es_ungulado.
es_ungulado & tiene_cuello_largo -> es_jirafa.
es_ungulado & tiene_rayas_negras -> es_cebra.

tiene_pelos & tiene_pezuñas & tiene_rayas_negras.

-es_cebra.
end_of_list.
```

  - **Salida:**

```
> mace -p <ej-animales.in
The search is complete.  No models were found.
```

Problema de los animales con MACE

- **Modelo del problema de los animales**
  - **Entrada** (ej-animales-2.in)

```
formula_list(sos).
tiene_pelos | da_leche -> es_mamifero.
es_mamifero & (tiene_pezuñas | rumia) -> es_ungulado.
es_ungulado & tiene_cuello_largo -> es_jirafa.
es_ungulado & tiene_rayas_negras -> es_cebra.

tiene_pelos & tiene_pezuñas & tiene_rayas_negras.

% -es_cebra.
end_of_list.
```

  - **Salida:**

```
> mace -p <ej-animales-2.in
==== Model #1 at 0.01 seconds:
 tiene_pelos: T
 es_mamifero: T
 da_leche: F
 tiene_pezugnas: T
 es_ungulado: T
 rumia: F
 tiene_cuello_largo: F
 es_jirafa: F
 tiene_rayas_negras: T
 es_cebra: T
end_of_model
```

Problemas del coloreado de pentágonos

- **Problemas del coloreado de pentágono con dos colores**
  - **Enunciado: Demostrar que es imposible colorear los vértices de un pentágono de rojo o azul de forma que los vértices adyacentes tengan colores distintos.**
  - **Representación: Se numeran los vértices consecutivos del pentágono con los números 1, 2, 3, 4 y 5. Se usan los símbolos r_i ($1 \leq i \leq 5$) para representar que el vértice i es rojo y los símbolos a_j ($1 \leq j \leq 5$) para representar que el vértice j es azul.**
  - **Entrada:**

```
formula_list(usable).
% El vértice i (1 <= i <= 5) es azul o rojo:
a1 | r1.                  a2 | r2.
a3 | r3.                  a4 | r4.
a5 | r5.

% Dos vértices adyacentes no pueden ser azules:
-(a1 & a2).               -(a2 & a3).
-(a3 & a4).               -(a4 & a5).
-(a5 & a1).

% Dos vértices adyacentes no pueden ser rojos:
-(r1 & r2).               -(r2 & r3).
-(r3 & r4).               -(r4 & r5).
-(r5 & r1).
end_of_list.
```

  - **Salida:**

```
The search is complete.  No models were found.
```

Problemas del coloreado de pentágonos

- **Problemas del coloreado de pentágono con dos colores**
  - **Enunciado: Demostrar que es posible colorear los vértices de un pentágono de rojo, azul o negro de forma que los vértices adyacentes tengan colores distintos.**
  - **Representación: Se numeran los vértices consecutivos del pentágono con los números 1, 2, 3, 4 y 5. Se usan los símbolos r_i ($1 \leq i \leq 5$) para representar que el vértice i es rojo, los símbolos a_j ($1 \leq j \leq 5$) para representar que el vértice j es azul y los símbolos n_j ($1 \leq j \leq 5$) para representar que el vértice j es negro.**
  - **Entrada**

```
formula_list(usable).
% El vértice i (1 <= i <= 5) es azul, rojo o negro:
a1 | r1 | n1.          a2 | r2 | n2.
a3 | r3 | n3.          a4 | r4 | n4.
a5 | r5 | n5.

% Dos vértices adyacentes no pueden ser azules:
-(a1 & a2).            -(a2 & a3).
-(a3 & a4).            -(a4 & a5).
-(a5 & a1).

% Dos vértices adyacentes no pueden ser rojos:
-(r1 & r2).            -(r2 & r3).
-(r3 & r4).            -(r4 & r5).
-(r5 & r1).
```

Problemas del coloreado de pentágonos

```
% Dos vértices adyacentes no pueden ser negros:
-(n1 & n2).           -(n2 & n3).
-(n3 & n4).           -(n4 & n5).
-(n5 & n1).
end_of_list.
```

- **Salida:**

```
======================= Model #1 at 0.00 seconds:
 a1: T
 r1: F
 n1: F
 a2: F
 r2: T
 n2: F
 a3: T
 r3: F
 n3: F
 a4: F
 r4: T
 n4: F
 a5: F
 r5: F
 n5: T
end_of_model
```

- **Solución**

```
           a1
          /  \
        n5    r2
        |      |
        r4 -- a3
```

Bibliografía

- Bundy, A. *The Computer Modelling of Mathematical Reasoning* (Academic Press, 1983)
  - Cap. 2 "Arguments about propositions"
  - Cap. 3: "The internal structure of propositions"
- Chang, C.L. y Lee, R.C.T *Symbolic Logic and Mechanical Theorem Proving* (Academic Press, 1973)
  - Cap. 2 "The propositional logic"
- Genesereth, M.R. *Computational Logic* (27 March 2000)
  - Cap. 2 "Propositional logic"
  - Cap. 3 "Truth table method"
- Nilsson, N.J. *Inteligencia artificial (Una nueva síntesis)* (McGraw–Hill, 2000)
  - Cap. 13 "El cálculo proposicional"
- Russell, S. y Norvig, P. *Inteligencia artificial (un enfoque moderno)* (Prentice Hall Hispanoamericana, 1996)
  - Cap. 6.4 "Lógica propositiva: un tipo de lógica muy sencillo"