Notas de Clase para IL

2. Definición de la Lógica Proposicional
Ejercicios resueltos

Rafael Farré, Robert Nieuwenhuis,
Pilar Nivela, Albert Oliveras, Enric Rodríguez

3 de septiembre de 2009

1. Ejercicios

1. (dificultad 1) ¿Cuántas interpretaciones posibles hay en función de $|\mathcal{P}|$?
(nota: si S es un conjunto, $|S|$ denota su *cardinalidad*, es decir, el número de elementos de S).

Para definir una interpretación I para el conjunto de símbolos de predicado $\mathcal{P}$, podemos escoger para cada $p \in \mathcal{P}$ si $I(p) = 0$ o $I(p) = 1$. Hay en total por lo tanto $2^{|\mathcal{P}|}$ interpretaciones. ■

2. (dificultad 1) Demuestra que $p \wedge \neg p$ es insatisfactible.

Por definición de satisfactibilidad, la fórmula $p \wedge \neg p$ es insatisfactible si y sólo si no hay ninguna interpretación que la satisfaga. Tomemos una interpretación I cualquiera y veamos que $eval_I(p \wedge \neg p) = 0$:

$$\begin{array}{rcll} eval_I(p \wedge \neg p) & = & min(eval_I(p), eval_I(\neg p)) & [\text{ por definición de } eval_I \\ & & & \quad \text{para la conjunción } \wedge\ ] \\ & = & min(eval_I(p), 1 - eval_I(p)) & [\text{ por definición de } eval_I \\ & & & \quad \text{para la negación } \neg\ ] \end{array}$$

Como $eval_I(p) \in \{0, 1\}$ se da uno de los siguientes dos casos:

a) $eval_I(p) = 0$: entonces $min(eval_I(p), 1 - eval_I(p)) = min(0, 1) = 0$

b) $eval_I(p) = 1$: entonces $min(eval_I(p), 1 - eval_I(p)) = min(1, 0) = 0$.

Luego en ambos casos se cumple $eval_I(p \wedge \neg p) = 0$, para toda I. ■

3. (dificultad 1) Demuestra que $p \vee \neg p$ es una tautología.

Por definición de tautología, la fórmula $p \vee \neg p$ es una tautología si y sólo si cualquier interpretación la satisface. Tomemos una interpretación I cualquiera y veamos que $eval_I(p \vee \neg p) = 1$:

$$\begin{array}{rcll} eval_I(p \vee \neg p) & = & max(eval_I(p),\ eval_I(\neg p)) & [\text{ definición de } eval_I\ ] \\ & = & max(eval_I(p),\ 1 - eval_I(p)) & [\text{ definición de } eval_I\ ] \end{array}$$

Debido a que $eval_I(p) \in \{0, 1\}$, este máximo será siempre 1. ■

4. (dificultad 1) Escribe la tabla de verdad para las fórmulas $(p \wedge \neg(q \vee r))$ y $(\neg p \vee (q \vee r))$. Di, por cada una de las fórmulas si es tautología o insatisfactible y averigua las posibles relaciones de consecuencia o equivalencia lógica entre ellas.

Las tablas de verdad son:

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

- Tautología: ninguna de ellas es tautología ya que existe al menos una interpretación que no las satisface.
- Insatisfactibilidad: ninguna de ellas es insatisfactible ya que existe al menos una interpretación que las satisface.
- Consecuencia lógica: $p \wedge \neg(q \vee r)$ no es consecuencia lógica de $\neg p \vee (q \vee r)$, ya que, por ejemplo, la primera interpretación de la tabla es un modelo de $\neg p \vee (q \vee r)$ pero no de $p \wedge \neg(q \vee r)$. Similarmente, la quinta interpretación de la tabla demuestra que $\neg p \vee (q \vee r)$ no es consecuencia de $p \wedge \neg(q \vee r)$.
- Equivalencia lógica: las fórmulas no son equivalentes ya que ninguna es consecuencia lógica de la otra.

■

5. (dificultad 1) Sean F y G dos fórmulas. ¿Es cierto que $F \vee G$ es tautología si y sólo si alguna de las dos fórmulas F o G lo es? Demuéstralo.

$\Longrightarrow$) Esta implicación no es cierta: no es cierto que si $F \vee G$ es tautología entonces F es tautología o G es tautología, tal como lo demuestra el siguiente contraejemplo.

Contraejemplo: $p \vee \neg p$ es una tautología, pero ni p ni $\neg p$ son tautologías.

$\Longleftarrow$) Esta implicación sí que es cierta: si F es una tautología, entonces $F \vee G$ es una tautología.

En efecto, para toda interpretación I se cumple:

$$eval_I(F \vee G) = max(eval_I(F), eval_I(G)) = max(1, eval_I(G)) = 1.$$

De manera similar se ve que si G es una tautología, entonces $F \vee G$ es una tautología.

Puesto que la implicación $\Longrightarrow$) no es cierta, el *si y sólo si* del enunciado no es cierto. ■

6. (dificultad 2) Sea F una fórmula. Demuestra que F es tautología si y sólo si $\neg F$ es insatisfactible.

La siguiente cadena de equivalencias demuestra el enunciado:

F es tautología

ssi	para toda I, $I \models F$	[por def. de tautología]
ssi	para toda I, $eval_I(F) = 1$	[por def. de satisfacción $\models$]
ssi	para toda I, $1 - eval_I(F) = 0$	[por def. de $-$]
ssi	para toda I, $eval_I(\neg F) = 0$	[por def. de $eval_I$]
ssi	para toda I, $I \not\models \neg F$	[por def. de satisfacción $\models$]
ssi	$\neg F$ es insatisfactible	[por def. de insatisfactibilidad]

■

7. (dificultad 2) Sean F y G dos fórmulas. Demuestra que F es consecuencia lógica de G si y sólo si $G \wedge \neg F$ es insatisfactible.

F es consecuencia lógica de G

ssi	para toda I, $I \not\models G$ o $I \models F$	[def. de consec.]
ssi	para toda I, $eval_I(G) = 0$ o $eval_I(F) = 1$	[def. de $\models$]
ssi	para toda I, $eval_I(G) = 0$ o $1 - eval_I(F) = 0$	[def. de $-$]
ssi	para toda I, $eval_I(G) = 0$ o $eval_I(\neg F) = 0$	[def. de $eval$]
ssi	para toda I, $min(eval_I(G), eval_I(\neg F)) = 0$	[def. de min]
ssi	para toda I, $eval_I(G \wedge \neg F) = 0$	[def. de $eval$]
ssi	para toda I, $I \not\models G \wedge \neg F$	[def. de $\models$]
ssi	$G \wedge \neg F$ es insatisfactible	[def. de satisfactible]

■

8. (dificultad 2) Sean F y G dos fórmulas. Demuestra que F es lógicamente equivalente a G si y sólo si $G \leftrightarrow F$ es tautología si y sólo si $(G \wedge \neg F) \vee (F \wedge \neg G)$ es insatisfactible.

9. (dificultad 2) Da un ejemplo de tres fórmulas F_1, F_2, y F_3 tales que $F_1 \wedge F_2 \wedge F_3$ sea insatisfactible y donde cualquier conjunción de todas ellas menos una sea satisfactible. Generalízalo a n fórmulas.

Consideremos por ejemplo los símbolos de predicado $\mathcal{P} = \{p, q\}$, y las fórmulas:

- $F_1 = p$
- $F_2 = q$
- $F_3 = \neg p \vee \neg q$

La única interpretación I que satisface tanto F_1 como F_2 deberá tener $I(p) = I(q) = 1$, por lo que no satisface F_3, es decir, la fórmula $F_1 \wedge F_2 \wedge F_3$ es insatisfactible.

Sin embargo, vemos que cualquier conjunción de todas ellas menos una es satisfactible con las interpretaciones I_{12}, I_{13}, I_{23} definidas como:

- $I_{12}(p) = 1$, $I_{12}(q) = 1$;
- $I_{13}(p) = 1$, $I_{13}(q) = 0$;
- $I_{23}(p) = 0$, $I_{23}(q) = 1$.

Se puede ver fácilmente que $I_{12} \models F_1 \wedge F_2$, que $I_{13} \models F_1 \wedge F_3$ y que $I_{23} \models F_2 \wedge F_3$. Esto se generaliza a n fórmulas definidas sobre los símbolos de predicado p_1, ..., p_{n-1}: para $1 \leq i \leq n-1$ definimos $F_i = p_i$, y definimos $F_n = \neg p_1 \vee \ldots \vee \neg p_{n-1}$. ■

10. (dificultad 1) Demuestra que son tautologías:

    a) $(p \wedge (p \rightarrow q)) \rightarrow q$

    b) $((p \rightarrow q) \wedge \neg q) \rightarrow \neg q$

11. (dificultad 2) Dada una fórmula F, si $c(F)$ denota el número de sus conectivas binarias (contando repeticiones) y $p(F)$ el número de sus símbolos de predicado (contando repeticiones), demuestra que $p(F) = c(F) + 1$.

Sea F la fórmula del problema. Demostremos el enunciado por inducción sobre $|F|$, la longitud de la fórmula F como cadena de símbolos. Para $|F| = 1$, necesariamente F es un símbolo de predicado, y por lo tanto $p(F) = 1$, $c(F) = 0$; en consecuencia, $p(F) = c(F) + 1$.

Supongamos ahora que $|F| > 1$. Distinguimos dos casos según la estructura sintáctica de F. En primer lugar, si F es de la forma $\neg F_1$, entonces $c(F) = c(F_1)$ y $p(F) = p(F_1)$. Además, por hipótesis de inducción, $p(F_1) = c(F_1) + 1$. Por lo tanto, $p(F) = p(F_1) = c(F_1) + 1 = c(F) + 1$.

En segundo lugar, si F es de la forma $(F_1 \otimes F_2)$, donde $\otimes$ denota $\wedge$ o $\vee$, entonces $c(F) = c(F_1) + 1 + c(F_2)$, y $p(F) = p(F_1) + p(F_2)$. Por otro lado, por hipótesis de inducción, $p(F_1) = c(F_1) + 1$ y $p(F_2) = c(F_2) + 1$. Por lo tanto, $p(F) = p(F_1) + p(F_2) = c(F_1) + 1 + c(F_2) + 1 = c(F) + 1$. ■

12. (dificultad 1) Demuestra de la forma más sencilla que encuentres:
    $(\neg q \vee \neg r \vee p) \wedge (p \vee q) \wedge (r \vee p) \quad \not\equiv \quad p \vee q$

Sea I la interpretación definida como $I(p) = 0$, $I(q) = 1$, y $I(r) = 0$. Como $eval_I(r \vee p) = 0$, entonces $eval_I((\neg q \vee \neg r \vee p) \wedge (p \vee q) \wedge (r \vee p)) = 0$; por otro lado, $eval_I(p \vee q) = 1$. Como no tienen los mismos modelos, las dos fórmulas no pueden ser lógicamente equivalentes. ■

13. (dificultad 1) Demuestra de la forma más sencilla que encuentres:
    $(p \vee q) \wedge (r \vee p) \wedge (\neg q \vee \neg r \vee p) \quad \equiv \quad p$

Calculemos la tabla de verdad de las dos fórmulas y comprobemos que tienen el mismo conjunto de modelos:

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

■

14. (dificultad 3) Sea F una fórmula que no contiene el símbolo $\neg$. Demuestra que si el número de símbolos de predicado es $k + 1$ entonces la longitud de la fórmula es $4k + 1$.

15. (dificultad 2) Si $F \to G$ es tautología y F es tautología, entonces ¿G es tautología? Demuéstralo.

16. (dificultad 2) Si $F \to G$ es satisfactible y F es satisfactible, entonces ¿G es satisfactible? Demuéstralo.

17. (dificultad 2) Si $F \to G$ es tautología y F satisfactible, entonces ¿G es satisfactible? Demuéstralo.

18. (dificultad 2) Demuestra las siguientes equivalencias entre fórmulas:

$F \wedge F$	$\equiv F$	idempotencia de $\wedge$
$F \vee F$	$\equiv F$	idempotencia de $\vee$
$F \wedge G$	$\equiv G \wedge F$	conmutatividad de $\wedge$
$F \vee G$	$\equiv G \vee F$	conmutatividad de $\vee$
$F \wedge (F \vee G)$	$\equiv F$	absorción 1
$F \vee (F \wedge G)$	$\equiv F$	absorción 2
$(F \wedge G) \wedge H$	$\equiv F \wedge (G \wedge H)$	asociatividad de $\wedge$
$(F \vee G) \vee H$	$\equiv F \vee (G \vee H)$	asociatividad de $\vee$
$(F \wedge G) \vee H$	$\equiv (F \vee H) \wedge (G \vee H)$	distributividad 1
$(F \vee G) \wedge H$	$\equiv (F \wedge H) \vee (G \wedge H)$	distributividad 2
$\neg\neg F$	$\equiv F$	doble negación
$\neg(F \wedge G)$	$\equiv \neg F \vee \neg G$	ley de De Morgan 1
$\neg(F \vee G)$	$\equiv \neg F \wedge \neg G$	ley de De Morgan 2
Si F es tautología entonces:		
$F \wedge G$	$\equiv G$	ley de tautología 1
$F \vee G$	$\equiv F$	ley de tautología 2
Si F es insatisfactible entonces:		
$F \wedge G$	$\equiv F$	ley de insatisfactible 1
$F \vee G$	$\equiv G$	ley de insatisfactible 2

Para comprobar que dos fórmulas son lógicamente equivalentes basta ver que se evalúan de la misma manera en cualquier interpretación I. A continuación demostramos algunas de las propiedades utilizando este argumento:

- $F \vee G \equiv G \vee F$

$$\begin{aligned} eval_I(F \vee G) &= max(eval_I(F),\ eval_I(G)) && \text{[def. } eval_I \text{ de } \vee] \\ &= max(eval_I(G),\ eval_I(F)) && [max \text{ es conmutativo}] \\ &= eval_I(G \vee F) && \text{[def. } eval_I \text{ de } \vee] \end{aligned}$$

- $(F \vee G) \vee H \equiv F \vee (G \vee H)$

$$\begin{aligned} eval_I((F \vee G) \vee H) &= max(eval_I(F \vee G),\ eval_I(H)) \\ &= max(max(eval_I(F),\ eval_I(G)),\ eval_I(H)) \\ &= max(eval_I(F),\ max(eval_I(G),\ eval_I(H))) \\ &= max(eval_I(F),\ eval_I(G \vee H)) \\ &= eval_I(F \vee (G \vee H)) \end{aligned}$$

  En este caso, la tercera igualdad es debido a la asociatividad de max, mientras que todos los otros son aplicaciones de la definición de evaluación de una disyunción.

- $(F \vee G) \wedge H \equiv (F \wedge H) \vee (G \wedge H)$

  Por un lado

$$\begin{aligned} & eval_I((F \vee G) \wedge H) \\ &= min(eval_I(F \vee G),\ eval_I(H)) \\ &= min(max(eval_I(F),\ eval_I(G)),\ eval_I(H)) \end{aligned}$$

  Por otro:

$$\begin{aligned} & eval_I((F \wedge H) \vee (G \wedge H)) \\ &= max\{eval_I(F \wedge H), eval_I(G \wedge H)\} \\ &= max(min(eval_I(F),\ eval_I(H)),\ min(eval_I(G),\ eval_I(H))) \end{aligned}$$

  Distiguimos dos casos según el valor de $eval_I(H)$.
  • Si $eval_I(H) = 0$:

$$\begin{aligned} eval_I((F \vee G) \wedge H) &= min(max(eval_I(F),\ eval_I(G)),\ 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} eval_I((F \wedge H) \vee (G \wedge H)) &= max(min(eval_I(F),\ 0), \\ &\qquad min(eval_I(G),\ 0)) \\ &= max(0, 0) = 0 \end{aligned}$$

- Si $eval_I(H) = 1$:

$$\begin{aligned} eval_I((F \vee G) \wedge H) &= min(max(eval_I(F),\ eval_I(G)),\ 1) \\ &= max(eval_I(F),\ eval_I(G)) \end{aligned}$$

$$\begin{aligned} eval_I((F \wedge H) \vee (G \wedge H)) &= max(min(eval_I(F),\ 1), \\ &\qquad min(eval_I(G),\ 1)) \\ &= max(eval_I(F),\ eval_I(G)) \end{aligned}$$

- Si F es tautología entonces $F \wedge G \equiv G$.

$$\begin{aligned} eval_I(F \wedge G) &= min(eval_I(F),\ eval_I(G)) && \text{[def. } eval_I \text{ de } \wedge] \\ &= min(\ 1,\ eval_I(G)) && [F \text{ es tautología}] \\ &= eval_I(G) && [\text{propiedad de } min] \end{aligned}$$

- Si F es insatisfactible entonces $F \wedge G \equiv F$.

$$\begin{aligned} eval_I(F \wedge G) &= min(eval_I(F),\ eval_I(G)) && \text{[def. } eval_I] \\ &= 0 && [F \text{ insatisfactible}] \\ &= eval_I(F) && [F \text{ insatisfactible}] \end{aligned}$$

■

19. (dificultad 2) Dadas fórmulas F, A, B, ¿es cierto que $F \wedge (A \vee B)$ es satisfactible si y sólo si al menos una de las fórmulas $F \wedge A$ y $F \wedge B$ lo es? Demuéstralo.

20. (dificultad 2) Demuestra que si $A \wedge B \models D$ y $A \wedge C \models D$ entonces $A \wedge (B \vee C) \models D$. ¿Es cierto el recíproco?

21. (dificultad 3) Una *relación binaria* sobre un conjunto S es un subconjunto del producto cartesiano $S \times S$, es decir, un conjunto de pares de elementos de S. Sea $\approx$ una relación binaria sobre S que escribiremos con notación infija, es decir, si para dos elementos a y b de S tenemos $(a, b) \in \approx$, escribiremos $a \approx b$. Se dice que $\approx$ es una *relación de equivalencia* en S si se cumplen las siguientes condiciones:

    a) Para cada $x \in S$ tenemos $x \approx x$ (*reflexividad*).
    b) Para cada $x, y \in S$, si $x \approx y$ entonces $y \approx x$ (*simetría*).
    c) Para cada $x, y, z \in S$, si $x \approx y$ y $y \approx z$ entonces $x \approx z$ (*transitividad*).

    Demuestra que la equivalencia lógica $\equiv$ es una relación de equivalencia en el conjunto de las fórmulas proposicionales definidas sobre un conjunto $\mathcal{P}$ de símbolos. Demuestra también que es una relación compatible con las operaciones lógicas $\neg$, $\wedge$ y $\vee$; es decir:

    a) Si $F \equiv G$ entonces $\neg F \equiv \neg G$.
    b) Si $F_1 \equiv G_1$ y $F_2 \equiv G_2$ entonces $F_1 \wedge F_2 \equiv G_1 \wedge G_2$.

c) Si $F_1 \equiv G_1$ y $F_2 \equiv G_2$ entonces $F_1 \vee F_2 \equiv G_1 \vee G_2$.

Veamos que $\equiv$ es reflexiva: dada una fórmula F, se tiene que demostrar que $F \equiv F$, lo cual es obvio ya que para toda interpretación I se tiene $eval_I(F) = eval_I(F)$.

Veamos que $\equiv$ es simétrica: dadas fórmulas F y G, se tiene que demostrar que $F \equiv G$ implica $G \equiv F$. Claramente esto también es así: para toda interpretación I, si se cumple que $eval_I(F) = eval_I(G)$ también se cumple $eval_I(G) = eval_I(F)$.

Veamos que $\equiv$ es transitiva: dadas fórmulas F, G y H, se tiene que demostrar que $F \equiv G$ y $G \equiv H$ implican $F \equiv H$. Fácilmente tenemos que para toda interpretación I, si se cumple que $eval_I(F) = eval_I(G)$ y que $eval_I(G) = eval_I(H)$, entonces se cumple $eval_I(F) = eval_I(H)$.

Veamos ahora que $\equiv$ es compatible con $\neg$, $\wedge$ y $\vee$. Si tomamos F y G tales que $F \equiv G$ entonces para toda interpretación I se tiene $\neg F \equiv \neg G$ ya que:

$$\begin{aligned} eval_I(\neg F) &= 1 - eval_I(F) && [eval_I \text{ de } \neg] \\ &= 1 - eval_I(G) && [F \equiv G] \\ &= eval_I(\neg G) && [eval_I \text{ de } \neg] \end{aligned}$$

Para las conectivas binarias tomemos $F_1 \equiv G_1$ y $F_2 \equiv G_2$ y es también inmediato ver que

$$\begin{aligned} eval_I(F_1 \wedge F_2) &= min(eval_I(F_1),\ eval_I(F_2)) && [eval_I \text{ de } \wedge] \\ &= min(eval_I(G_1),\ eval_I(G_2)) && [F_1 \equiv F_2 \text{ y } G_1 \equiv G_2] \\ &= eval_I(G_1 \wedge G_2) && [eval_I \text{ de } \wedge] \end{aligned}$$

y que

$$\begin{aligned} eval_I(F_1 \vee F_2) &= max(eval_I(F_1),\ eval_I(F_2)) && [eval_I \text{ de } \vee] \\ &= max(eval_I(G_1), eval_I(G_2)) && [F_1 \equiv F_2 \text{ y } G_1 \equiv G_2] \\ &= eval_I(G_1 \vee G_2) && [eval_I \text{ de } \vee] \end{aligned}$$

■

22. (dificultad 2) Definimos recursivamente las subfórmulas de una fórmula F de la forma siguiente:

    - Si F es un símbolo de predicado p, entonces p es su única subfórmula.
    - Si F es $\neg G$, entonces las subfórmulas de F son F y las subfórmulas de G.
    - Si F es de la forma $(G \wedge H)$ o de la forma $(G \vee H)$, entonces las subfórmulas de F son F y las subfórmulas de G y las de H (contando con repetición, es decir, todas las ocurrencias).

Demuestra que para toda fórmula F, el número de subfórmulas de F, denotado $nsf(F)$, cumple que $nsf(F) \leq |F|$, donde $|F|$ es el número de símbolos (conectivas, paréntesis y símbolos de predicado) de F.

La demostración es por inducción sobre $|F|$.

Caso base: Suponemos que $|F| = 1$. Esto implica que F es p, con p un símbolo de predicado. Como $nsf(F) = 1$ y $|F| = 1$ se cumple $nsf(F) \leq |F|$.

Paso de inducción: Supongamos que $|F| > 1$ y que el resultado es cierto para todas las G con $|G| < |F|$. Hemos de demostrar que es cierto para F.

Si $|F| > 1$ entonces F debe ser de una de las siguientes formas:

- a) F es $\neg G$
- b) F es $(G \vee H)$
- c) F es $(G \wedge H)$

a) Si F es de la forma $\neg G$ entonces se cumple:

$$\begin{array}{rlll} nsf(F) = & 1 + nsf(G) & \text{[def. de } nsf \text{ cuando } F \text{ es } \neg G] \\ \leq & 1 + |G| & \text{[HI sobre } G \text{ pues } |G| < |F|] \\ = & |F| & \text{[def de } |F| \text{ cuando } F \text{ es } \neg G] \end{array}$$

b) Si F es de la forma $(G \vee H)$ entonces se cumple:

$$\begin{array}{rll} nsf(F) = & 1 + nsf(G) + nsf(H) & \text{[def. de } nsf \text{ cuando } F \text{ es } (G \vee H)] \\ \leq & 1 + |G| + |H| & \text{[HI sobre } G, H \\ & & \text{pues } |G| < |F|, |H| < |F|] \\ < & 3 + |G| + |H| & \text{[propiedades aritméticas]} \\ = & |F| & \text{[def de } |F| \text{ cuando } F \text{ es } (G \vee H)] \end{array}$$

c) Demostración como el caso b), pero reemplazando $\vee$ por $\wedge$.

Luego en todos los casos hemos demostrado que $nsf(F) \leq |F|$.

■

23. (dificultad 3) Demuestra que si en una fórmula F sustituimos una ocurrencia de una subfórmula G por otra G' lógicamente equivalente a G, obtenemos una nueva fórmula F' que es lógicamente equivalente a F. Este resultado tiene un interés obvio para poder manipular fórmulas, por lo que tiene nombre: es el *Lema de Sustitución*.

Sea $n(F)$ la suma del número de conectivas $\wedge$, $\vee$ y $\neg$ de F menos el número de conectivas $\wedge$, $\vee$ y $\neg$ de G. Como G es una subfórmula de F, entonces $n(F) \geq 0$. Demostremos el lema por inducción sobre $n(F)$.

- Caso base: si $n(F) = 0$, entonces necesariamente G es idénticamente F. Por lo tanto, si sustituimos G por G', entonces F' es idénticamente G'. Como $G \equiv G'$ por hipótesis, entonces $F \equiv F'$.
- Caso inductivo: supongamos que $n(F) > 0$, y que el lema es cierto para cualquier fórmula H tal que H contiene una ocurrencia de G como subfórmula y $n(H) < n(F)$. Como $n(F) > 0$, G es distinta de F. Distinguimos varios casos según la estructura de F:
  - Supongamos F de la forma $\neg F_1$ para una cierta fórmula F_1. Entonces necesariamente la ocurrencia de G que se sustituye ocurre también como subfórmula en F_1. Denotamos por F_1' la fórmula resultante de sustituir la ocurrencia de G por G' en F_1. Entonces F' es $\neg F_1'$. Además, como $n(F_1) < n(F)$, podemos aplicar la hipótesis de inducción y obtenemos que $F_1 \equiv F_1'$. Por el ejercicio anterior, $\neg F_1 \equiv \neg F_1'$, y por lo tanto $F \equiv F'$.
  - Supongamos F de la forma $F_1 \otimes F_2$ para ciertas fórmulas F_1, F_2, y $\otimes \in \{\wedge, \vee\}$. Sin pérdida de generalidad podemos suponer también que la ocurrencia de G que se sustituye por G' ocurre como subfórmula de F_1. Definimos F_1' como la fórmula resultante de sustituir G por G' en F_1. Como $n(F_1) < n(F)$, por hipótesis de inducción $F_1 \equiv F_1'$. Por el ejercicio anterior, $F_1 \otimes F_2 \equiv F_1' \otimes F_2$, y como F es $F_1 \otimes F_2$ y F' es $F_1' \otimes F_2$, se tiene $F \equiv F'$. El caso en que la G que se sustituye por G' ocurre como subfórmula en F_2 es análogo.

■

24. (dificultad 2) Demuestra utilizando los resultados de los dos ejercicios anteriores que $q \vee (p \wedge (\neg p \vee r))$ es lógicamente equivalente a $(q \vee p) \wedge ((q \vee r) \vee (p \wedge \neg p))$.

Utilizaremos el Lema de Sustitución para reemplazar subfórmulas por otras que, de acuerdo con un ejercicio anterior, son equivalentes. Una posible manera de demostrar la equivalencia entre ambas fórmulas es la siguiente:

$$
\begin{array}{lcll}
q \vee (p \wedge (\neg p \vee r)) & \equiv & q \vee ((\neg p \vee r) \wedge p) & [\text{ conmutatividad de } \wedge \text{ }] \\
 & \equiv & q \vee ((\neg p \wedge p) \vee (r \wedge p)) & [\text{ distributividad 2 }] \\
 & \equiv & q \vee (r \wedge p) & [\text{ ley insatisfactible 2 }] \\
 & \equiv & (r \wedge p) \vee q & [\text{ conmutatividad de } \vee \text{ }] \\
 & \equiv & (r \vee q) \wedge (p \vee q) & [\text{ distributividad 1 }]
\end{array}
$$

$$
\begin{array}{lcll}
(q \vee p) \wedge ((q \vee r) \vee (p \wedge \neg p)) & \equiv & (q \vee p) \wedge (q \vee r) & [\text{ ley insatisfactible 2 }] \\
 & \equiv & (q \vee r) \wedge (q \vee p) & [\text{ commutatividad } \wedge \text{ }] \\
 & \equiv & (r \vee q) \wedge (q \vee p) & [\text{ commutatividad } \vee \text{ }] \\
 & \equiv & (r \vee q) \wedge (p \vee q) & [\text{ commutatividad } \vee \text{ }]
\end{array}
$$

Finalmente, la equivalencia ha quedado demostrada ya que

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

y la relación de equivalencia lógica, como toda relación de equivalencia, es transitiva (ver ejercicio anterior). ■

25. (dificultad 2) Demuestra que para todas las fórmulas A, B, C se cumple que $A \wedge B \models C$ si y sólo si $A \models (B \to C)$.

$(A \wedge B) \models C$	ssi	$(A \wedge B) \wedge \neg C$ es insatisfactible	[ej. anterior]
	ssi	$A \wedge (B \wedge \neg C)$ es insatisfactible	[asociativ. $\wedge$]
	ssi	$A \wedge \neg\neg(B \wedge \neg C)$ es insatisfactible	[doble neg.]
	ssi	$A \wedge \neg(\neg B \vee C)$ es insatisfactible	[De Morgan]
	ssi	$A \models (\neg B \vee C)$	[ej. anterior]
	ssi	$A \models (B \to C)$	[def. $\to$]

■

26. (dificultad 2) Supongamos que $|\mathcal{P}| = 100$ y que nos interesa determinar si una fórmula F construida sobre $\mathcal{P}$ es satisfactible o no. Si el algoritmo está basado en un análisis de la tabla de verdad, y evaluar F en una interpretación I dada cuesta un microsegundo (10^{-6} segundos), ¿cuántos años tardará? Más adelante veremos técnicas que muchas veces funcionan mejor.

27. (dificultad 2) Una *función booleana* de n entradas es una función $f\colon \{0,1\}^n \to \{0,1\}$, es decir, una función que toma como entrada una cadena de n bits y devuelve un bit. ¿Cuántas funciones booleanas de n entradas hay?

Una función booleana queda determinada por el valor que toma en todas sus posibles entradas. Si la función toma como entrada cadenas de n bits, sabemos que existen 2^n posibles entradas (n bits i dos posibilidades para cada uno de ellos). Para determinar una función booleana tenemos que dar un valor (0 o 1) para cada una de estas entradas, lo que da lugar a 2^{2^n} funciones booleanas de n entradas. ■

28. (dificultad 2) Cada fórmula F representa una única función booleana: la que devuelve 1 exactamente para aquellas cadenas de bits I tales que $eval_I(F) = 1$. Por eso, dos fórmulas son lógicamente equivalentes si y sólo si representan la misma función booleana. ¿Cuántas funciones booleanas (o cuántas fórmulas lógicamente no-equivalentes) hay en función de $|\mathcal{P}|$?

Como una fórmula definida sobre $\mathcal{P}$ puede considerarse como una función booleana con $|\mathcal{P}|$ entradas, de acuerdo con el ejercicio anterior hay $2^{2^{|\mathcal{P}|}}$ funciones booleanas distintas. ■

29. (dificultad 2) Consideremos los siguientes símbolos de predicado:

p es *María abusa de la comida basura*

q es *Tomás abusa de la comida basura*

r es *María está enferma*

s es *Tomás está enfermo*

t es *María fuma*

u es *Tomás fuma*

a) Traduce las siguientes fórmulas a tu idioma:

1) $p \wedge \neg q$
2) $(\neg p \wedge \neg t) \rightarrow \neg r$
3) $(t \vee u) \rightarrow (t \wedge u)$
4) $s \leftrightarrow (q \wedge u)$

b) Traduce las siguientes frases a fórmulas de la lógica proposicional. Si en algún caso hay varias fórmulas posibles, indica si son lógicamente equivalentes o no. Nótese que, si no lo son, estamos ante un caso de ambigüedad del lenguaje natural.

1) Tomás no fuma salvo que María lo haga.
2) Ni Tomás ni María abusan de la comida basura.
3) María esta sana si y sólo si no fuma y no abusa de la comida basura
4) María no fuma si Tomás no lo hace
5) Si María no fuma entonces está sana si no abusa de la comida basura

---

a) Las traducciones son:

1) María abusa de la comida basura y Tomás no
2) Si María no abusa de la comida basura y no fuma entonces no está enferma
3) Si María o Tomás fuman entonces María y Tomás fuman
4) Tomás está enfermo si y sólo si abusa de la comida basura y fuma

b) Posibles traducciones a lógica proposicional son:

1) $u \rightarrow t$ o bien
$\neg t \rightarrow \neg u$

Ambas fórmulas son lógicamente equivalentes. Son abreviaciones de $\neg u \vee t$ y $\neg(\neg t) \vee \neg u$, respectivamente, y sus tablas de verdad demuestran la equivalencia:

t	u	$\neg t$	$\neg(\neg t)$	$\neg u$	$\neg u \vee t$	$\neg(\neg t) \vee \neg u$
0	0	1	0	1	1	1
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	1	0	1	0	1	1

2) $\neg q \wedge \neg p$
3) $\neg r \leftrightarrow (\neg t \wedge \neg p)$, o bien
   $(\neg r \leftrightarrow \neg t) \wedge \neg p$.
   En este caso, podemos ver que no son lógicamente equivalentes ya que si tomamos la interpretación I tal que $I(r) = 1$, $I(t) = 0$ y $I(p) = 1$, tenemos que $eval_I(\neg r \leftrightarrow (\neg t \wedge \neg p)) = 1$ mientras que $eval_I((\neg r \leftrightarrow \neg t) \wedge \neg p) = 0$.
4) $\neg u \rightarrow \neg t$
5) $\neg t \rightarrow (\neg p \rightarrow \neg r)$, o bien
   $\neg p \rightarrow (\neg t \rightarrow \neg r)$

   En este caso ambas fórmulas son lógicamente equivalentes. Son abreviaciones de $t \vee (p \vee \neg r)$ y $p \vee (t \vee \neg r)$ y sus tablas de verdad muestran su equivalencia:

p	r	t	$\neg r$	$p \vee \neg r$	$t \vee \neg r$	$t \vee (p \vee \neg r)$	$p \vee (t \vee \neg r)$
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	0	0	0	0	0
0	1	1	0	0	1	1	1
1	0	0	1	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	1	0	1	1
1	1	1	0	1	1	1	1

■

30. (dificultad 3) Demuestra para $n \geq 1$ las dos siguientes equivalencias lógicas:
    $(F_1 \wedge \ldots \wedge F_n) \vee G \equiv (F_1 \vee G) \wedge \ldots \wedge (F_n \vee G)$
    $(F_1 \vee \ldots \vee F_n) \wedge G \equiv (F_1 \wedge G) \vee \ldots \vee (F_n \wedge G)$

31. (dificultad 3) Escribe en una tabla de verdad las 16 funciones booleanas de 2 entradas. ¿Cuántas de ellas sólo dependen de una de las dos entradas? ¿Cuántas dependen de cero entradas? ¿Las otras, vistas como conectivas lógicas, tienen algún nombre? Ya sabemos que podemos expresar cualquier función booleana con el conjunto de tres conectivas $\{\wedge, \vee, \neg\}$, es decir, cualquier función booleana es equivalente a una fórmula construida sobre estas tres conectivas. ¿Es cierto esto también para algún conjunto de sólo dos de las 16 funciones? (Hay varias maneras, pero basta con dar una sola.)

32. (dificultad 3) Demuestra que cualquier función booleana de dos entradas se puede expresar con sólo *nor* o bien con sólo *nand*, donde $nor(F, G)$ es $\neg(F \vee G)$, y $nand(F, G)$ es $\neg(F \wedge G)$.

33. (dificultad 3) Tres estudiantes A, B y C son acusados de introducir un virus en la salas de ordenadores de la FIB. Durante el interrogatorio, las declaraciones son las siguientes:

- A dice: "B lo hizo y C es inocente"
- B dice: "Si A es culpable entonces C también lo es"
- C dice: "Yo no lo hice, lo hizo al menos uno de los otros"

a) ¿Son las tres declaraciones contradictorias?

b) Asumiendo que todos son inocentes, ¿quién o quiénes mintieron en la declaración?

c) Asumiendo que nadie mintió, ¿quién es inocente y quién es culpable?

Sea a la variable que significa que el estudiante A es culpable, b la variable que significa que el estudiante B es culpable, y c la variable que significa que el estudiante C es culpable. Similarmente, abusando la notación denotamos por A la declaración de A, por B la declaración de B, y por C la declaración de C. Entonces $A = b \wedge \neg c$, $B = \neg a \vee c$, $C = \neg c \wedge (a \vee b)$. Observamos que la interpretación I tal que $I(a) = 0$, $I(b) = 1$ y $I(c) = 0$ es tal que $I \models A \wedge B \wedge C$, de modo que las tres declaraciones no son contradictorias. Además, si todos fuesen inocentes, es decir tomando la interpretación I' tal que $I'(a) = I'(b) = I'(c) = 0$, tendríamos que $I' \not\models A$, $I' \models B$, $I' \not\models C$; de modo que tanto A como C habrían mentido. Finalmente, la interpretación I nos dice que un posible escenario es que B haya sido el único culpable. Nótese que $A \wedge B \wedge C \equiv b \wedge \neg c \wedge (\neg a \vee c) \wedge (a \vee b)$, y que $A \wedge B \wedge C \models \neg c \wedge (\neg a \vee c) \models \neg a$. Como también tenemos $A \wedge B \wedge C \models b$ y $A \wedge B \wedge C \models \neg c$, vemos que I es el único modelo de $A \wedge B \wedge C$. De modo que si todo el mundo dijo la verdad, el único culpable es B. ■

34. (dificultad 2) Inventa y define formalmente alguna otra lógica distinta a la lógica proposicional. Por ejemplo, si las interpretaciones son funciones $I : \mathcal{P} \to \{0, 1, \perp\}$ que también pueden dar "indefinido" $\perp$, se puede adaptar la noción de satisfacción de manera razonable, aunque la respuesta ya no será binaria: la "evaluación" de una fórmula F en una interpretación I puede dar 1 (I satisface F) o 0 (I no satisface F) o $\perp$ (indefinido).

La lógica que vamos a definir mantendrá la misma sintaxis que la lógica proposicional, pero con semántica distinta. Para este caso, una interpretación será una función $I : \mathcal{P} \to \{0, 1, \perp\}$. Entonces, definiremos la evaluación de una fórmula en una interpretación I como:

- Si la fórmula es un símbolo p de $\mathcal{P}$, entonces $eval_I(p) = I(p)$.
- En caso contrario, $eval_I(F \wedge G)$, $eval_I(F \vee G)$ y $eval_I(\neg F)$ se definen en función de $eval_I(F)$ y $eval_I(G)$ de acuerdo con las tablas siguientes:

$\wedge$	0	1	$\perp$
0	0	0	$\perp$
1	0	1	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$

$\vee$	0	1	$\perp$
0	0	1	$\perp$
1	1	1	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$

$\neg$	0	1	$\perp$
	1	0	$\perp$

Como en lógica proposicional, diremos que una interpretación I satisface F si y sólo si $eval_I(F) = 1$.

Nótese que las tablas propuestas, en las que cualquier operación que tenga un argumento indefinido tiene resultado indefinido, no son las únicas que extienden de manera razonable la lógica proposicional. Por ejemplo, en una solución alternativa se puede considerar que $0 \wedge \perp = 0$, y similarmente $1 \vee \perp = 1$. De esta manera, las tablas serían:

$\wedge$	0	1	$\perp$
0	0	0	0
1	0	1	$\perp$
$\perp$	0	$\perp$	$\perp$

$\vee$	0	1	$\perp$
0	0	1	$\perp$
1	1	1	1
$\perp$	$\perp$	1	$\perp$

$\neg$	0	1	$\perp$
	1	0	$\perp$

También hay otra posibilidad. Por motivos de eficiencia, muchos lenguajes de programación (como C, C++, Java), implementan las operaciones lógicas de manera que sólo se evalúan los argumentos si es necesario, y se comienza la evaluación por la izquierda. Si se considera que $\perp$ denota la no terminación de un cálculo, entonces esto significa que $0 \wedge \perp = 0$, pero $\perp \wedge 0 = \perp$ (nótese que este $\wedge$ ya no es conmutativo). Las correspondientes tablas serían:

$\wedge$	0	1	$\perp$
0	0	0	0
1	0	1	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$

$\vee$	0	1	$\perp$
0	0	1	$\perp$
1	1	1	1
$\perp$	$\perp$	$\perp$	$\perp$

$\neg$	0	1	$\perp$
	1	0	$\perp$

■

35. (dificultad 2) Como el ejercicio anterior, pero considerando $I{:}\mathcal{P} \to [0 \ldots 1]$, es decir, la interpretación de un símbolo p es una probabilidad (un número real entre 0 y 1). En este caso, la evaluación de una fórmula F en una interpretación I puede dar algo (remotamente) parecido a la probabilidad de satisfacción de F en I. En la lógica que has definido, ¿la evaluación de F en una I determinada, y la de $F \wedge F$ en esa misma I dan el mismo resultado?

36. (dificultad 2) Tienes delante de tí tres cajas cerradas, dos de ellas vacías y una llena de oro. La tapa de cada caja contiene una afirmación respecto de su contenido. La caja A dice "El oro no está aquí", la caja B dice "El oro no está aquí" y la caja C dice "El oro está en la caja B". Si sabemos que una y sólo una las afirmaciones es cierta, qué caja contiene el oro?

37. (dificultad 3) Considera el siguiente fragmento de código, que devuelve un booleano:

```
int i;
bool a, b;
...
if (a and i>0)
  return b;
else if (a and i<=0)
  return false;
else if (a or b)
  return a;
else
  return (i>0);
```

Simplifícalo sustituyendo los valores de retorno por un solo valor de retorno que sea una expresión booleana en $i > 0$, a y b:

```
int i;
bool a, b;
return ...;
```

38. (dificultad 2) Demuestra que I satisface la fórmula $((\cdots(p_1 \leftrightarrow p_2) \leftrightarrow \cdots) \leftrightarrow p_n)$ (donde $n \geq 1$) si y sólo si asigna el valor 0 a un número par de símbolos de predicado de la fórmula.

39. (dificultad 3)

    a) La fórmula $(((p \to q) \to p) \to p)$ es una tautología?

    b) Si definimos recursivamente A_0 como $(p \to q)$ y A_{n+1} como $(A_n \to p)$, ¿para qué valores de n es A_n una tautología?

---

a) Para demostrar que la fórmula $(((p \to q) \to p) \to p)$ es una tautología, calculamos su tabla de verdad:

p	q	$(p \to q)$	$((p \to q) \to p)$	$(((p \to q) \to p) \to p)$
0	0	1	0	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

b) Antes de iniciar la demostración vamos a observar qué pasa con las fórmulas A_n. Tenemos que

$$\begin{array}{rcll} A_1 & \equiv & ((p \to q) \to p) & \text{[Definición de } A_1\text{]} \\ & \equiv & (\neg(\neg p \vee q) \vee p) & \text{[Definición de } \to\text{]} \\ & \equiv & ((p \wedge \neg q) \vee p) & \text{[de Morgan y doble negación]} \\ & \equiv & p & \text{[absorción]} \end{array}$$

A continuación nos fijamos que A_2 es una tautología:

$$\begin{array}{rcll} A_2 & \equiv & (A_1 \to p) & \text{[Definición de } A_2\text{]} \\ & \equiv & (p \to p) & \text{[Lema de sustitución]} \end{array}$$

Si continuamos, vemos que A_3 vuelve a ser equivalente a p:

$$\begin{array}{rcll} A_3 & \equiv & (A_2 \to p) & \text{[Definición de } A_3\text{]} \\ & \equiv & (\neg A_2 \vee p) & \text{[Definición de } \to\text{]} \\ & \equiv & p & [\neg A_2 \text{ es insatisfactible]} \end{array}$$

Estas manipulaciones nos ayuden a ver que los que tenemos que demostrar es que A_n ($n \geq 1$) es lógicamente equivalente a p si n es impar, y una tautología si n es par. No obstante, tales manipulaciones **NO** deberían aparecer, por ejemplo, en un examen. Su presencia en este documento se debe únicamente a motivos pedagógicos.

Haremos la demostración por inducción sobre n. El caso base es $n = 1$, para el que tenemos que demostrar que $A_1 \equiv p$. En efecto, observando la tabla de verdad del apartado anterior, vemos que esto es así.

Supongamos ahora que $n > 1$. Distinguimos dos casos:

- Si n es par, entonces tenemos que demostrar que A_n, o sea $\neg A_{n-1} \vee p$ es una tautología. Por hipótesis de inducción, tenemos que $A_{n-1} \equiv p$. Entonces, por el lema de sustitución, basta con demostrar que $(\neg p \vee p)$ es una tautología. Pero esto es así porque $(\neg p \vee p) \equiv p \vee \neg p$, y ya vimos en un ejercicio anterior que $p \vee \neg p$ es una tautología.
- Si n es impar, entonces tenemos que demostrar que $A_n \equiv p$. Como A_n es $(\neg A_{n-1} \vee p)$, basta con demostrar que $(\neg A_{n-1} \vee p) \equiv p$. Como por hipótesis de inducción A_{n-1} es una tautología, $\neg A_{n-1}$ es insatisfactible. Por la ley de insatisfactible 2, tenemos precisamente que $(\neg A_{n-1} \vee p) \equiv p$.

En particular, hemos demostrado que, para $n \geq 1$, A_n es una tautología si y sólo si n es par. Como, utilizando la tabla de verdad del apartado anterior, observamos que A_0 no es una tautología, podemos concluir que los únicos valores para los cuales A_n es una tautología son $n \geq 2$ y par.

■

40. (dificultad 3) Sea A^* la fórmula resultante de intercambiar en A los símbolos $\wedge, \vee$ y reemplazar cada símbolo de predicado por su negación. Demuestra que A^* es lógicamente equivalente a $\neg A$.

41. (dificultad 2) Dada una cadena de símbolos w, escribimos $pa(w)$ para denotar el número de paréntesis de abrir que hay en w. Similarmente, escribimos $pc(w)$ para los de cerrar. Demuestra que para toda fórmula F tenemos $pa(F) = pc(F)$.