

Métodos Lógicos para Ciencia de la Computación - IIC2212
Guía 1

1. El conectivo lógico NOR es definido de la siguiente forma:

p	q	$p \text{ NOR } q$
0	0	1
0	1	0
1	0	0
1	1	0

Demuestre que NOR es funcionalmente completo.

2. Sea $\oplus$ el conectivo lógico binario definido como sigue: Para una valuación $\sigma : P \rightarrow \{0, 1\}$ se tiene que $\sigma(\alpha \oplus \beta) = 1$ si y sólo si $\sigma(\alpha) = 1 - \sigma(\beta)$.

¿Es el conjunto $\{\oplus, \leftrightarrow\}$ de conectivos lógicos funcionalmente completo?

3. El conectivo ternario MAYORIA es definido de la siguiente forma:

p	q	r	MAYORIA(p, q, r)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Demuestre que MAYORIA no es funcionalmente completo.

4. ¿Es $\{\neg, \text{MAYORIA}\}$ funcionalmente completo?

5. El conectivo unario $\perp$ es definido de la siguiente forma:

p	$\perp p$
0	0
1	0

Este conectivo usualmente se denota sin la letra proposicional porque su valor de verdad es siempre 0 (por ejemplo, denotamos $p \wedge (\perp q)$ como $p \wedge \perp$).

Demuestre que $\{\neg, \text{MAYORIA}, \perp\}$ es funcionalmente completo.

6. Formalice el siguiente argumento en el cálculo proposicional:

“Si Superman fuera capaz y deseara prevenir el mal, entonces lo haría. Si Superman fuera incapaz de prevenir el mal, entonces sería impotente, y si no deseara prevenir el mal, entonces sería malévolo. Si Superman existe, no es ni impotente ni malévolo. Superman no previene el mal. Entonces, Superman no existe.”

Demuestre usando resolución que Superman no existe.

7. Sea p una proposición atómica en P . Para cada $\alpha, \beta \in L(P)$ defina recursivamente la *fórmula obtenida desde α reemplazando p por β* , denotado por $\alpha[\beta/p]$, como sigue:

- $\alpha[\beta/p] = \begin{cases} \alpha & \text{si } \alpha \text{ es proposición atómica y } \alpha \neq p \\ \beta & \text{si } \alpha = p \end{cases}$
- $(\alpha_1 \vee \alpha_2)[\beta/p] = \alpha_1[\beta/p] \vee \alpha_2[\beta/p]$
- $(\neg\alpha)[\beta/p] = \neg\alpha[\beta/p]$

Demuestre que si α_1 y α_2 son equivalentes entonces para todo $\beta \in L(P)$ se tiene que $\alpha_1[\beta/p]$ y $\alpha_2[\beta/p]$ también son equivalentes.

8. Defina recursivamente el *dual* de una fórmula ϕ , denotado por ϕ^* , como sigue:

- $p^* = \neg p$, si $p \in P$;
- $(\alpha \vee \beta)^* = \alpha^* \wedge \beta^*$;
- $(\alpha \wedge \beta)^* = \alpha^* \vee \beta^*$;
- $(\neg\alpha)^* = \neg\alpha^*$.

Demuestre que para todo ϕ en $L(P)$ se tiene que ϕ^* es equivalente a $\neg\phi$.

9. Decimos que una fórmula φ está en 3-CNF si φ está en CNF y cada una de sus cláusulas contiene a lo más tres literales. Por ejemplo, $(p \vee q \vee \neg r) \wedge (\neg p \vee s)$ está en 3-CNF mientras que $(p \vee \neg q \vee \neg r \vee s)$ no está en 3-CNF.

Demuestre que existen fórmulas que no son equivalentes a ninguna fórmula en 3-CNF.

10. Decimos que una fórmula φ está en k -CNF ($k \geq 2$) si φ está en CNF y cada una de sus cláusulas contiene a lo más k literales. ¿Existe algún valor de k para el cual toda fórmula es equivalente a una fórmula en k -CNF?

11. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$, demuestre que $\Sigma \models \alpha \rightarrow \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$.

12. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$, demuestre que si α es una tautología, entonces $\Sigma \cup \{\alpha\} \models \beta$ si y sólo si $\Sigma \models \beta$.

13. Dado $\Sigma \subseteq L(P)$ y $\varphi, \psi, \theta \in L(P)$, demuestre que si $\varphi \rightarrow \psi$ es una tautología, entonces $\Sigma \cup \{\varphi, \psi\} \models \theta$ si y sólo si $\Sigma \cup \{\varphi\} \models \theta$.

14. Sea $P = \{p_0, p_1, \dots\}$. Defina una sucesión de fórmulas ϕ_i , para $i \geq 0$, por medio de la siguiente definición inductiva:

$$\phi_0 \equiv p_0 \quad \phi_{i+1} \equiv (\phi_i \leftrightarrow p_{i+1}).$$

Demuestre que para todo $i \geq 0$ se tiene que, si $\sigma : P \rightarrow \{0, 1\}$ es una valuación cualquiera, entonces $\sigma(\phi_i) = 1$ si y sólo si $|\{p_j \mid \sigma(p_j) = 0, j \leq i\}|$ es par.

15. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$ tal que α y $\Sigma \cup \{\beta\}$ no tienen variables proposicionales en común. ¿Es cierto que $\Sigma \models \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$?
16. Sea P un conjunto de variables proposicionales, y asuma que H es el siguiente conjunto de fórmulas en $L(P)$:

$$\{p \mid p \in P\} \cup \{(p_1 \wedge \dots \wedge p_n) \rightarrow p \mid \{p_1, \dots, p_n, p\} \subseteq P\}.$$

Sea $\Sigma \subseteq H$. Demuestre que no existe una variable proposicional q en P tal que $\Sigma \models \neg q$.

17. Considere una valuación $\sigma : P \rightarrow \{0, 1\}$. Demuestre que el conjunto $S = \{\phi \mid \phi \in L(P), \sigma(\phi) = 1\}$ satisface lo siguiente:

Para todo conjunto S' de fórmulas en $L(P)$ se tiene que si $S \subseteq S'$ y S' es satisfacible, entonces $S = S'$.

18. Sea $\{\phi_1, \phi_2, \dots\}$ un conjunto (no necesariamente finito) de fórmulas en $L(P)$. Asuma que para cada valuación $\sigma : P \rightarrow \{0, 1\}$ se tiene que existe un entero i tal que $\sigma(\phi_i) = 1$.

Asuma que $\{\alpha_1, \alpha_2, \dots\}$ es un conjunto de fórmulas en $L(P)$ tal que para cada $i \geq 1$ se tiene que α_i está en CNF y α_i es equivalente a $\neg\phi_i$.

Use el teorema de compacidad para demostrar que existe un entero j tal que

$$\{\alpha_1, \alpha_2, \dots, \alpha_j\} \vdash \square.$$

19. Sea P un conjunto de variables proposicionales, y asuma que $\dashv$ es una relación binaria sobre $L(P) \times 2^{L(P)}$. Es decir, $\dashv$ es un subconjunto del conjunto $\{(\phi, \Sigma) \mid \phi \in L(P) \text{ y } \Sigma \subseteq L(P)\}$.

Asuma que para todo $(\phi, \Sigma) \in L(P) \times 2^{L(P)}$ se tiene que si $(\phi, \Sigma) \in \dashv$ entonces $\Sigma \models \phi$.

Sea $\Phi = \{\phi_0, \phi_1, \dots\}$ un conjunto satisfacible de fórmulas en $L(P)$, tal que para todo conjunto Φ' de fórmulas en $L(P)$ se tiene que si $\Phi \subseteq \Phi'$ y Φ' es satisfacible, entonces $\Phi = \Phi'$.

Sea ψ una fórmula en $L(P)$ tal que $(\psi, \Phi) \in \dashv$. Demuestre que $\psi \in \Phi$.

20. Sea $\Sigma \subseteq L(P)$ un conjunto de fórmulas finitamente satisfacible. Demuestre que para todo $p \in P$ se tiene que $\Sigma \cup \{p\}$ es finitamente satisfacible o $\Sigma \cup \{\neg p\}$ es finitamente satisfacible.
21. Sea P un conjunto de variables proposicionales. Definimos la *lógica infinitaria* sobre P , denotada por $L_\infty(P)$, como el menor conjunto que satisface las siguientes reglas:

- Si $\Sigma \subseteq L(P)$ entonces $(\bigwedge \Sigma) \in L_\infty(P)$.
- Si $\phi \in L_\infty(P)$ entonces $(\neg\phi) \in L_\infty(P)$.
- Si $\phi, \psi \in L_\infty(P)$ entonces $(\phi \vee \psi) \in L_\infty(P)$.

Por ejemplo, si $P = \{p_i \mid i \in \mathbb{N}\}$ y $\Sigma = \{(\neg p_i) \mid p_i \in P\}$, entonces $\bigwedge \Sigma$ y $(\neg \bigwedge \Sigma)$ son fórmulas en $L_\infty(P)$. La primera fórmula dice que todas las variables proposicionales son falsas, mientras la segunda dice que al menos una variable proposicional es verdadera.

Definimos formalmente la semántica de $L_\infty(P)$ como sigue. Sea $\sigma : P \rightarrow \{0, 1\}$ una valuación y ϕ una fórmula en $L_\infty(P)$:

- Si $\phi = (\bigwedge \Sigma)$ y $\Sigma \subseteq L(P)$, entonces $\sigma(\phi) = 1$ si y sólo si para cada $\psi \in \Sigma$ se tiene que $\sigma(\psi) = 1$.
- Si $\phi = (\neg \alpha)$, entonces $\sigma(\phi) = 1$ si y sólo si $\sigma(\alpha) = 0$.
- Si $\phi = (\alpha \vee \beta)$, entonces $\sigma(\phi) = 1$ si y sólo si $\sigma(\alpha) = 1$ o $\sigma(\beta) = 1$.

¿Es el teorema de compacidad válido para $L_\infty(P)$? Es decir, dado $\Gamma \subseteq L_\infty(P)$, ¿es cierto que si todo subconjunto finito de Γ es satisfacible entonces Γ es satisfacible?

22. En este problema se pide al alumno demostrar que si junto con las reglas de resolución y factorización podemos usar tautologías en cualquier parte de una demostración, entonces el sistema resultante es completo. Vale decir, si C es una cláusula y Σ es un conjunto de cláusulas tal que $\Sigma \models C$, entonces $\Sigma \vdash C$.

Formalmente, una demostración por resolución de la cláusula C desde el conjunto de cláusulas Σ es una secuencia $C_1, \dots, C_n$ tal que $C_n = C$ y para cada $1 \leq i \leq n$ se tiene que:

- $C_i \in \Sigma$, o
- C_i es una tautología, o
- existe $j < i$ tal que C_i es obtenido desde C_j usando regla de factorización, o
- existen $j, k < i$ tal que C_i es obtenido desde C_j y C_k usando regla de resolución.

Como antes, si existe demostración por resolución de C desde Σ escribimos $\Sigma \vdash C$.

Note que si C es tautología entonces trivialmente se tiene que $\Sigma \vdash C$. Por tanto, para demostrar que el sistema es completo le basta con demostrar lo siguiente:

- a) Si Σ es inconsistente entonces $\Sigma \vdash C$ para toda cláusula C .
- b) Si Σ es consistente, C no es tautología y $\Sigma \models C$, entonces $\Sigma \vdash C$.

En los ejercicios 12 - 20 decimos que un algoritmo es eficiente si el número de pasos ejecutado por el algoritmo es n^c cuando la entrada tiene largo n , donde c es una constante. Por ejemplo, un algoritmo que funciona en tiempo n^2 es eficiente mientras que un algoritmo que funciona en tiempo 2^n no lo es.

- 12. Encuentre un algoritmo eficiente que dada una fórmula φ en CNF construya una fórmula ψ en 3-CNF tal que φ es satisfacible si y sólo si ψ es satisfacible.
- 13. Encuentre un algoritmo eficiente que verifique si una fórmula en DNF es satisfacible.
- 14. Encuentre un algoritmo eficiente que verifique si una fórmula en CNF es una tautología.

15. Un grafo G es una tupla (N, A) , donde N es un conjunto de nodos y $A \subseteq N \times N$ es un conjunto de arcos. Un grafo es no dirigido si cada vez que $(a, b) \in A$ se tiene que $(b, a) \in A$.

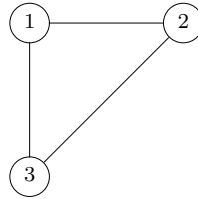
Un grafo no dirigido $G = (N, A)$ es 3-coloreable si existe una asignación de colores para los nodos tal que nodos adyacentes reciben colores distintos. Formalmente, G es 3-coloreable si existe una función $f : N \rightarrow \{\text{blanco}, \text{azul}, \text{rojo}\}$ tal que para cada $(a, b) \in A$ se tiene que $f(a) \neq f(b)$.

Demuestre que el problema de 3-coloración puede ser reducido eficientemente al problema de satisfacibilidad. Vale decir, encuentre un algoritmo eficiente que dado un grafo G construye una fórmula φ tal que G es 3-coloreable si y sólo si φ es satisfacible. Estime el número de pasos de su algoritmo cuando el grafo G tiene n nodos y m arcos.

16. ¿Qué pasa en el problema anterior si también consideramos algoritmos no eficientes? ¿Cuáles son las fórmulas de menor largo que podríamos usar en la reducción?
17. Un grafo no dirigido $G = (N, A)$ es k -coloreable ($k \geq 2$) si existe una función $f : N \rightarrow \{1, 2, \dots, k\}$ tal que para cada $(a, b) \in A$ se tiene que $f(a) \neq f(b)$.

Demuestre que el problema de k -coloración puede ser reducido eficientemente al problema de satisfacibilidad. Estime el número de pasos de su algoritmo para un grafo con n nodos y m arcos.

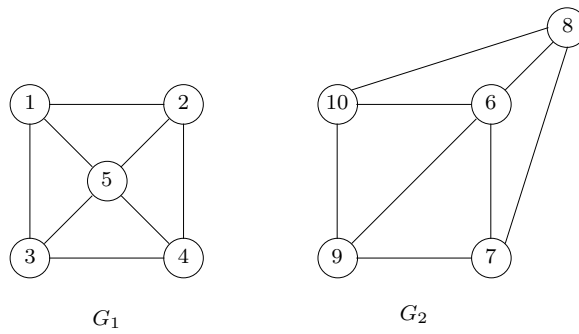
18. Usando la fórmula construida en el problema anterior y resolución, demuestre que el siguiente grafo no es 2-coloreable:



19. Suponga que existe un algoritmo eficiente para el problema de satisfacibilidad, es decir, un algoritmo que dada una fórmula φ con n letras proposicionales, en n^c pasos retorna 1 si la fórmula es satisfacible y 0 en caso contrario, donde c es una constante fija.

Encuentre un algoritmo eficiente que dada una fórmula φ retorna una valuación σ tal que $\sigma(\varphi) = 1$ si φ es satisfacible, y retorna 0 en caso contrario.

20. Decimos que dos grafos $G_1 = (N_1, A_1)$ y $G_2 = (N_2, A_2)$ son *isomorfos* si existe una biyección $f : N_1 \rightarrow N_2$ tal que para todo a y b en N_1 se tiene que $(a, b) \in A_1$ si y sólo si $(f(a), f(b)) \in A_2$. Por ejemplo, los siguientes grafos son isomorfos:



Encuentre un algoritmo eficiente que dados dos grafos G_1 y G_2 construye una fórmula φ tal que G_1 y G_2 son isomorfos si y sólo si φ es satisfacible. Estime el número de pasos de su algoritmo cuando G_1 tiene n_1 nodos y m_1 arcos, y G_2 tiene n_2 nodos y m_2 arcos.

21. Encuentre un método que no use tablas de verdad para demostrar que una fórmula es una tautología. Use su método para demostrar que $(p \leftrightarrow q) \leftrightarrow (\neg p \leftrightarrow \neg q)$ es una tautología.
22. El *principio de los cajones* establece que si $n+1$ objetos son distribuidos en n cajones, entonces al menos habrá un cajón con más de un objeto. Demuestre el principio para $n = 2$ usando cálculo proposicional y resolución.