

Inteligencia Artificial - CC52A
Guía 1

1. Dado $\alpha, \beta \in L(P)$, demuestre que $\alpha \models \beta$ y $\beta \models \alpha$ si y sólo si $\models (\alpha \leftrightarrow \beta)$ (es decir, $\alpha \models \beta$ y $\beta \models \alpha$ si y sólo si $(\alpha \leftrightarrow \beta)$ es una tautología).
2. 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 $\Sigma \models \beta$.
3. 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 $\Sigma \cup \{\varphi\} \models \theta$.
4. 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 $\Sigma \cup \{\alpha\} \models \beta$?
5. 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 $\beta[\alpha_1/p]$ y $\beta[\alpha_2/p]$ también son equivalentes.

6. 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 $\neg\phi$.

7. 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?
8. 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.

9. 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.

10. ¿Es $\{\neg, \text{MAYORIA}\}$ funcionalmente completo?
 11. 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.

12. 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.

13. 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.

14. 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?
15. 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 $S = S'$.
16. Estudie detenidamente la demostración de la completitud débil del método de resolución para lógica proposicional.

En los siguientes ejercicios 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.

1. Encuentre un algoritmo eficiente que dada una fórmula φ en CNF construye una fórmula ψ en 3-CNF tal que φ es satisfacible si y sólo si ψ es satisfacible.
2. Encuentre un algoritmo eficiente que verifique si una fórmula en DNF es satisfacible.
3. Encuentre un algoritmo eficiente que verifique si una fórmula en CNF es una tautología.
4. 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.
5. 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.
6. 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.

7. 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 A_1 se tiene que $(a, b) \in A_1$ si y sólo si $(f(a), f(b)) \in A_2$. 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.
8. Un *clique* es un grafo G tal que para todo par de nodos u, v en G se tiene que existe un eje entre ellos.
9. Demuestre que para cada fórmula ϕ en CNF uno puede encontrar eficientemente un grafo G y un entero k , tal que ϕ es satisfacible si y sólo si G contiene un clique de tamaño k .

Sea P el siguiente conjunto de variables proposicionales:

$$\{p_{i,j} \mid i \in [1, 3], j \in [1, 2]\}.$$

Intuitivamente, la variable proposicional $p_{i,j}$, para cada $i \in [1, 3]$ y $j \in [1, 2]$, expresa que el elemento i está en la posición j .

- a) Construya una fórmula proposicional α en $L(P)$, tal que α **está en CNF**, y, además, α es satisfacible si y sólo si todo elemento $i \in [1, 3]$ está en al menos una posición $j \in [1, 2]$, y no existe un elemento $i \in [1, 3]$ que está en posiciones distintas $j_1, j_2 \in [1, 2]$.
- b) Construya una formula β en $L(P)$, tal que β **está en DNF**, y tal que β es satisfacible si y sólo si existen dos elementos distintos $i_1, i_2 \in [1, 3]$ que están en la misma posición $j \in [1, 2]$.
- c) Demuestre que $\alpha \models \beta$.
10. Dada una matriz C de 3×3 que contiene números entre 0 y 3, decimos que C es *completable* si es que existe una manera de reemplazar los números 0 por números entre 1 y 3 de tal forma que la suma de cada fila y de cada columna es la misma. Por ejemplo, la siguiente matriz es completable:

| | | |
|---|---|---|
| 2 | 0 | 0 |
| 0 | 2 | 0 |
| 0 | 0 | 3 |

puesto que podemos reemplazar los valores 0 por los siguientes valores:

| | | |
|---|---|---|
| 2 | 2 | 1 |
| 2 | 2 | 1 |
| 1 | 1 | 3 |

de manera tal que la suma de cada fila y de cada columna es 5. En cambio, la siguiente matriz no es completable:

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 3 | 0 | 0 |

Dada una matriz C de 3×3 , construya una fórmula φ en lógica proposicional tal que C es completible si y sólo si φ es satisfacible. En particular, φ tiene que ser construida de tal forma que cada valuación σ que satisface a φ represente una forma de completar C .

11. 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.

12. Sea C un conjunto de cláusulas proposicionales y m un literal, i.e. una variable proposicional o la negación de una variable proposicional.

Decimos que el conjunto de cláusulas C deriva la cláusula c usando k pasos de resolución proposicional, $k \geq 0$, si existe secuencia $c_0, c_1, \dots, c_k = c$ de cláusulas tal que para cada $0 \leq i \leq k$ se tiene lo siguiente:

- (a) $c_i \in C$; ó
- (b) existen $j, k < i$ tal que c_i se obtiene mediante resolución proposicional desde c_j y c_k .

Defina $C(m)$ como el siguiente conjunto de cláusulas:

$$C(m) = \{c \mid c \in C, m \notin c, \bar{m} \notin c\} \cup \{(c - \bar{m}) \mid c \in C, m \notin c, \bar{m} \in c\}.$$

Recuerde que para literal m , $\bar{\bar{m}} = \neg p$ si $m = p$ y $\bar{\bar{m}} = p$ si $m = \neg p$.

Demuestre lo siguiente:

- a) Si se puede derivar la cláusula c desde $C(m)$ usando k pasos de resolución proposicional, $k \geq 0$, entonces se puede derivar la cláusula c' desde C usando k pasos de resolución proposicional, donde c' es la cláusula c ó la cláusula $c \cup \{\bar{m}\}$.
 - b) Si $C(m)$ deriva la cláusula vacía \square usando k_1 pasos de resolución proposicional, $k_1 \geq 0$, y $C(\bar{m})$ deriva la cláusula vacía \square usando k_2 pasos de resolución proposicional, $k_2 \geq 0$, entonces C deriva la cláusula vacía \square usando a lo más $k_1 + k_2 + 1$ pasos de resolución proposicional.
13. Demuestre que existe un algoritmo que funciona en tiempo polinomial tal que, dada una fórmula ϕ de la lógica proposicional, construye una fórmula ψ en CNF tal que ϕ es satisfacible si y solo si ψ es satisfacible.