

1.3. Equivalencia, expresividad y satisfacibilidad

En esta sección recordaremos y veremos algunas propiedades útiles de la lógica proposicional.

1.3.1. Equivalencias

Comenzaremos recordando la noción de equivalencia entre fórmulas proposicionales. Intuitivamente, dos fórmulas proposicionales son equivalentes si tienen el mismo contenido semántico, es decir, si para cualquier valuación de sus variables, ambas fórmulas evalúan al mismo valor de verdad.

Definición. Dos fórmulas proposicionales bien formadas α y β son *equivalentes* si para toda valuación v de sus variables se cumple $\hat{v}(\alpha) = \hat{v}(\beta)$. Denotaremos esto como $\alpha \equiv \beta$.

Notemos que podemos describir la equivalencia de varias maneras: $\alpha \equiv \beta$ si $\alpha \leftrightarrow \beta$ es una tautología o si $\alpha \models \beta$ y $\beta \models \alpha$.

Recordaremos algunas equivalencias clásicas de la lógica proposicional. Usaremos el símbolo $\top$ para denotar una tautología y recordemos que usamos el símbolo $\perp$ para denotar una contradicción.

Proposición (Equivalencias clásicas). Sean α, β, γ fórmulas bien formadas. Tenemos las siguientes equivalencias:

- $\alpha \wedge \top \equiv \alpha$ (Identidad del $\wedge$)
- $\alpha \vee \perp \equiv \alpha$ (Identidad del $\vee$)
- $\alpha \wedge \perp \equiv \perp$ (Anulación del $\wedge$)
- $\alpha \vee \top \equiv \top$ (Anulación del $\vee$)
- $\alpha \wedge \beta \equiv \beta \wedge \alpha$ (Conmutatividad del $\wedge$)
- $\alpha \vee \beta \equiv \beta \vee \alpha$ (Conmutatividad del $\vee$)
- $(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ (Asociatividad del $\wedge$)
- $(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ (Asociatividad del $\vee$)
- $\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ (Distributividad del $\wedge$ sobre el $\vee$)
- $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (Distributividad del $\vee$ sobre el $\wedge$)
- $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ (De Morgan)
- $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ (De Morgan)
- $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$ (Caracterización de la implicancia)

Notemos que cada equivalencia nos da dos reglas de inferencia adicionales, una para cada dirección de la equivalencia.

Ejemplo. La equivalencia de De Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ nos da las siguientes reglas de inferencia:

- Si hemos demostrado $\neg(\alpha \wedge \beta)$, entonces podemos concluir $\neg\alpha \vee \neg\beta$.
- Si hemos demostrado $\neg\alpha \vee \neg\beta$, entonces podemos concluir $\neg(\alpha \wedge \beta)$.

Ejemplo. Demostremos $\{\neg(p \wedge q), p\} \vdash \neg q$ usando la equivalencia de De Morgan como regla de inferencia.

1. $\neg(p \wedge q)$ (Premisa)
2. p (Premisa)
3. $\neg p \vee \neg q$ (De Morgan, aplicada a 1)
4. $\neg p$ (Supuesto temporal: caso 1 de la disyunción)
5. $p \wedge \neg p$ ($\wedge$ -introducción, utilizando 2 y 4)
6. $\perp$ ($\perp$ -introducción, en 5)
7. $\neg q$ ($\perp$ -eliminación, en 6)
8. $\neg q$ (Supuesto temporal: caso 2 de la disyunción)
9. $\neg q$ ($\vee$ -eliminación, utilizando 3 y los bloques 4–7 y 8)

1.3.2. Funciones booleanas y expresividad

La lógica proposicional tiene una íntima relación con las *funciones booleanas*. La **potencia cartesiana A^n denota el conjunto de todas las n -tuplas de elementos de A** . Una función booleana es una función $f: \{0, 1\}^n \rightarrow \{0, 1\}$, es decir, una función que toma n bits de entrada y devuelve un bit de salida. Por ejemplo, la función $\wedge: \{0, 1\}^2 \rightarrow \{0, 1\}$ que representa la conjunción es una función booleana. De hecho, cada fórmula proposicional con n variables proposicionales puede ser vista como una función booleana de n variables. Por ejemplo, la fórmula $(p \wedge q) \vee r$ puede ser vista como la función booleana $f(p, q, r) = (p \wedge q) \vee r$.

¿Pueden todas las funciones booleanas ser representadas por una fórmula proposicional? Notemos que cada función booleana puede ser representada por una tabla de verdad, esto es simplemente listar todas las combinaciones de entrada y su correspondiente salida.

Ejemplo. Consideremos la función booleana $f: \{0, 1\}^3 \rightarrow \{0, 1\}$ dada por la siguiente tabla de verdad:

p	q	r	$f(p, q, r)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

¿Qué fórmula proposicional representa esta función booleana? Notemos que esta función booleana devuelve 1 para las siguientes combinaciones de entrada: $(0, 0, 1)$, $(1, 0, 0)$ y $(1, 1, 0)$. Es decir, $f(p, q, r) = 1$ si y solo si $(p, q, r) \in \{(0, 0, 1), (1, 0, 0), (1, 1, 0)\}$. Por lo tanto, podemos representar esta función booleana de la siguiente manera:

$$f(p, q, r) = (\neg p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (p \wedge q \wedge \neg r).$$

La fórmula que resultó del ejemplo anterior tiene una forma especial: es una disyunción de conjunciones de variables (o sus negaciones). Esto se conoce como la *forma normal disyuntiva* de una fórmula proposicional.¹

Definición. Una fórmula proposicional está en *forma normal disyuntiva* si es una disyunción de conjunciones de variables proposicionales o sus negaciones.

Podemos generalizar el razonamiento del ejemplo anterior para obtener lo siguiente.

Teorema. Toda función booleana $f: \{0, 1\}^n \rightarrow \{0, 1\}$ puede ser representada por una fórmula proposicional en forma normal disyuntiva.

Demostración. Sea $f: \{0, 1\}^n \rightarrow \{0, 1\}$ una función booleana con $n \geq 1$ entradas $p_1, \dots, p_n$. Para cada $a = (a_1, \dots, a_n) \in \{0, 1\}^n$ tal que $f(a) = 1$, definamos

$$\ell_i(a) = \begin{cases} p_i & \text{si } a_i = 1, \\ \neg p_i & \text{si } a_i = 0, \end{cases} \quad C_a = \ell_1(a) \wedge \ell_2(a) \wedge \dots \wedge \ell_n(a).$$

Notemos que C_a evalúa a 1 bajo una valuación v si y solo si

$$(v(p_1), \dots, v(p_n)) = a.$$

- **Caso 1:** f no es la función constante 0. Luego, existe al menos un $a \in \{0, 1\}^n$ tal que $f(a) = 1$. Sean $a^{(1)}, \dots, a^{(k)}$ todos los elementos de $\{0, 1\}^n$ tales que $f(a^{(j)}) = 1$. Consideremos la fórmula en forma normal disyuntiva

$$\phi = C_{a^{(1)}} \vee C_{a^{(2)}} \vee \dots \vee C_{a^{(k)}}.$$

Veamos que ϕ representa a f . Sea v una valuación de las variables proposicionales $p_1, \dots, p_n$. Si $f(v(p_1), \dots, v(p_n)) = 1$, entonces $(v(p_1), \dots, v(p_n)) = a^{(j)}$ para algún $j \in \{1, \dots, k\}$, y por lo tanto $\hat{v}(C_{a^{(j)}}) = 1$, con lo que $\hat{v}(\phi) = 1$. Por otro lado, si $f(v(p_1), \dots, v(p_n)) = 0$, entonces $(v(p_1), \dots, v(p_n)) \notin \{a^{(1)}, \dots, a^{(k)}\}$, y por lo tanto $\hat{v}(C_{a^{(j)}}) = 0$ para todo $j \in \{1, \dots, k\}$, con lo que $\hat{v}(\phi) = 0$.

- **Caso 2:** f es la función constante 0. En este caso, la fórmula $p_1 \wedge \neg p_1$ representa a f y está en forma normal disyuntiva. □

Notemos que la fórmula obtenida a partir de la tabla de verdad no es única: por ejemplo, podemos aplicar equivalencias para obtener fórmulas equivalentes. A pesar de esto, concluimos algo interesante: toda función booleana puede ser representada usando únicamente los operadores $\wedge$, $\vee$ y $\neg$.

¹(Otra forma normal común es la *forma normal conjuntiva*, que es una conjunción de disyunciones.)

Definición. Un conjunto de operadores es *funcionalmente completo* si pueden ser usados para representar cualquier función booleana.

En este lenguaje, tenemos lo siguiente.

Corolario. El conjunto de operadores $\{\wedge, \vee, \neg\}$ es funcionalmente completo.

Notemos que existen otros conjuntos de operadores que también son funcionalmente completos. Para ver esto, basta con mostrar que estos operadores “simulan” los operadores de otro conjunto funcionalmente completo.

Ejemplo. El conjunto de operadores $\{\wedge, \neg\}$ es funcionalmente completo. ¿Por qué? Notemos que podemos simular el operador $\vee$ usando únicamente $\wedge$ y $\neg$:

$$\alpha \vee \beta \equiv \neg(\neg\alpha \wedge \neg\beta).$$

Por lo tanto, podemos representar cualquier función booleana utilizando únicamente $\wedge$ y $\neg$: primero representamos la función booleana usando $\wedge$, $\vee$ y $\neg$, y luego reemplazamos cada ocurrencia de $\vee$ por su equivalente en términos de $\wedge$ y $\neg$.

1.3.3. Satisfacibilidad

Una de las propiedades más interesantes de las fórmulas proposicionales es la de satisfacibilidad.

Definición. Una fórmula proposicional α es *satisfacible* si existe una valuación v de sus variables tal que $\hat{v}(\alpha) = 1$.

En otras palabras, una fórmula es satisfacible si no es una contradicción.

Ejemplo. La siguiente fórmula es satisfacible:

$$(p \vee q) \wedge (\neg p \vee r) \wedge (\neg q \vee \neg r).$$

Para ver esto, basta con considerar la valuación $v(p) = 0$, $v(q) = 1$ y $v(r) = 0$, con lo que la fórmula evalúa a 1.

Por otro lado, la siguiente fórmula no es satisfacible:

$$(p \vee q) \wedge (\neg p \vee r) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r).$$

Lo que podemos ver via su tabla de verdad

p	q	r	Fórmula
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

La satisfacibilidad es una de las propiedades más importantes de las fórmulas proposicionales. Por ejemplo, la satisfacibilidad es uno de los problemas centrales de la teoría de la complejidad computacional, pues es un problema canónico que creemos es difícil de resolver, pero fácil de certificar.

A pesar de lo anterior, en la actualidad existen SAT-solvers bastante eficientes que pueden resolver problemas de satisfacibilidad con bastantes variables proposicionales de manera razonable. Dichos programas utilizan una variedad de técnicas para determinar si una fórmula es satisfacible o no, reportando una valuación que hace que la fórmula sea satisfacible en caso de que lo sea. Esto es bastante útil para aplicaciones, pues la flexibilidad de la lógica proposicional hace que podamos modelar una gran variedad de problemas en términos de fórmulas proposicionales, y luego resolverlos utilizando SAT-solvers.

Ejemplo. Supongamos que tenemos que programar tres exámenes en dos franjas horarias: mañana y tarde. Tenemos tres ramos: lógica, computación y matemáticas. Tenemos las siguientes restricciones:

- Cada examen debe ser programado en exactamente una franja horaria.
- Los exámenes de lógica y computación no pueden ser programados en la misma franja horaria.
- El examen de matemáticas debe ir antes que el de computación.

Modelemos este problema via la lógica proposicional. Consideremos las siguientes variables proposicionales:

- L_m : El examen de lógica es programado en la mañana.
- L_t : El examen de lógica es programado en la tarde.
- C_m : El examen de computación es programado en la mañana.
- C_t : El examen de computación es programado en la tarde.
- M_m : El examen de matemáticas es programado en la mañana.
- M_t : El examen de matemáticas es programado en la tarde.

La gran gracia de modelar via lógica proposicional es que podemos modelar cada restricción por separado, y luego combinarlas via $\wedge$ para obtener una fórmula proposicional que modela el problema completo. Luego, podemos traducir las restricciones a fórmulas proposicionales de la siguiente manera:

- Cada examen debe ser programado en exactamente una franja horaria:
 - $L_m \vee L_t$ (El examen de lógica debe ser programado en al menos una franja horaria)
 - $\neg(L_m \wedge L_t)$ (El examen de lógica no puede ser programado en ambas franjas horarias)
 - $C_m \vee C_t$ (El examen de computación debe ser programado en al menos una franja horaria)
 - $\neg(C_m \wedge C_t)$ (El examen de computación no puede ser programado en ambas franjas horarias)

- $M_m \vee M_t$ (El examen de matemáticas debe ser programado en al menos una franja horaria)
- $\neg(M_m \wedge M_t)$ (El examen de matemáticas no puede ser programado en ambas franjas horarias)
- Los exámenes de lógica y computación no pueden ser programados en la misma franja horaria:
 - $\neg(L_m \wedge C_m)$ (Los exámenes de lógica y computación no pueden ser programados en la mañana)
 - $\neg(L_t \wedge C_t)$ (Los exámenes de lógica y computación no pueden ser programados en la tarde)
- El examen de matemáticas debe ir antes que el de computación: $C_t \rightarrow M_m$ (Si el examen de computación es programado en la tarde, entonces el examen de matemáticas debe ser programado en la mañana)

Podemos combinar todas estas fórmulas proposicionales utilizando $\wedge$ para obtener una fórmula proposicional que modela el problema completo:

$$\begin{aligned}\phi = & (L_m \vee L_t) \wedge \neg(L_m \wedge L_t) \wedge (C_m \vee C_t) \wedge \neg(C_m \wedge C_t) \\ & \wedge (M_m \vee M_t) \wedge \neg(M_m \wedge M_t) \wedge \neg(L_m \wedge C_m) \\ & \wedge \neg(L_t \wedge C_t) \wedge (C_t \rightarrow M_m).\end{aligned}$$

Si usamos ϕ como entrada de un SAT-solver, este nos diría que ϕ es satisfacible, y nos daría la siguiente valuación que hace que ϕ sea satisfacible:

- $v(L_m) = 1$ (El examen de lógica es programado en la mañana)
- $v(L_t) = 0$ (El examen de lógica no es programado en la tarde)
- $v(C_m) = 0$ (El examen de computación no es programado en la mañana)
- $v(C_t) = 1$ (El examen de computación es programado en la tarde)
- $v(M_m) = 1$ (El examen de matemáticas es programado en la mañana)
- $v(M_t) = 0$ (El examen de matemáticas no es programado en la tarde)

Ejemplo. Veamos un segundo ejemplo. Supongamos que tenemos un conjunto de n estaciones de transmisión $E = \{1, 2, \dots, n\}$. Cada estación de transmisión puede transmitir en una de tres frecuencias: baja, media y alta. Queremos asignar una frecuencia a cada estación de transmisión de tal manera que no haya interferencia entre estaciones de transmisión cercanas. La interferencia ocurre cuando dos estaciones de transmisión a menos de 10 km de distancia transmiten en la misma frecuencia. Supongamos que conocemos los pares de estación cercanos dados como una lista C . ¿Es posible asignar frecuencias a las estaciones de transmisión de tal manera que no haya interferencia entre estaciones cercanas?

Introduzcamos las siguientes variables proposicionales:

- $e_{i,b}$: La estación de transmisión e_i transmite en la frecuencia baja.

- $e_{i,m}$: La estación de transmisión e_i transmite en la frecuencia media.
- $e_{i,a}$: La estación de transmisión e_i transmite en la frecuencia alta.

La primera restricción a considerar es que cada estación de transmisión debe transmitir en exactamente una frecuencia: **Para una familia finita de fórmulas, $\bigwedge$ y $\bigvee$ abreviarán, respectivamente, su conjunción y su disyunción.**

- $e_{i,b} \vee e_{i,m} \vee e_{i,a}$ (La estación de transmisión e_i debe transmitir en al menos una frecuencia) Luego, esta restricción se puede expresar como:

$$\Phi_{\text{una}} = \bigwedge_{i=1}^n (e_{i,b} \vee e_{i,m} \vee e_{i,a})$$

- $\neg(e_{i,b} \wedge e_{i,m}) \wedge \neg(e_{i,b} \wedge e_{i,a}) \wedge \neg(e_{i,m} \wedge e_{i,a})$ (La estación de transmisión e_i no puede transmitir en más de una frecuencia) Tenemos, entonces:

$$\Phi_{\text{no-más}} = \bigwedge_{i=1}^n (\neg(e_{i,b} \wedge e_{i,m}) \wedge \neg(e_{i,b} \wedge e_{i,a}) \wedge \neg(e_{i,m} \wedge e_{i,a}))$$

La segunda restricción es que no haya interferencia entre estaciones cercanas:

- Para cada par de estaciones cercanas $(e_i, e_j) \in C$, debemos tener la restricción:

$$\neg(e_{i,b} \wedge e_{j,b}) \wedge \neg(e_{i,m} \wedge e_{j,m}) \wedge \neg(e_{i,a} \wedge e_{j,a})$$

Es decir, tenemos la fórmula:

$$\Phi_{\text{interferencia}} = \bigwedge_{(i,j) \in C} (\neg(e_{i,b} \wedge e_{j,b}) \wedge \neg(e_{i,m} \wedge e_{j,m}) \wedge \neg(e_{i,a} \wedge e_{j,a}).)$$

Podemos combinar todas estas fórmulas proposicionales utilizando $\wedge$ para obtener una fórmula proposicional que modela el problema completo.

$$\Phi = \Phi_{\text{una}} \wedge \Phi_{\text{no-más}} \wedge \Phi_{\text{interferencia}}$$

Luego, podemos usar un SAT-solver para determinar si esta fórmula proposicional es satisfacible o no, lo que nos dirá si es posible asignar frecuencias a las estaciones de transmisión de tal manera que no haya interferencia entre estaciones cercanas.

Ejemplo. Otra cosa que podemos hacer vía SAT-solvers es diseñar un solver de Sudoku. El [Sudoku](#) es un juego clásico en el que, dada una cuadrícula de 9×9 con algunos números del 1 al 9 ya escritos, se busca llenar el resto con números del 1 al 9 de tal manera que cada fila, cada columna y cada subcuadrícula de 3×3 contenga todos los números sin repetir ninguno.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Figura 1.1: Un puzzle de Sudoku a la izquierda y su solución a la derecha.

Para diseñar un solver de Sudoku mediante un SAT-solver, describamos primero el tablero usando conjuntos. Sea $D = \{1, \dots, 9\}$ el conjunto de dígitos y sea $Q = D \times D$ el conjunto de celdas. Para $i, j \in D$ y $a, b \in \{0, 1, 2\}$, definamos

$$F_i = \{(i, j) : j \in D\}, \quad C_j = \{(i, j) : i \in D\},$$

$$B_{a,b} = \{(3a + r, 3b + s) : r, s \in \{1, 2, 3\}\}.$$

Estos son, respectivamente, los conjuntos de celdas de la fila i , la columna j y la subcuadrícula (a, b) de 3×3 . Sea además $P \subseteq Q$ el conjunto de celdas que contienen una pista y sea $g: P \rightarrow D$ la función que indica el dígito escrito en cada una.

Para cada $c \in Q$ y cada $k \in D$, introducimos la variable proposicional

$$x_{c,k} : \text{ la celda } c \text{ contiene el dígito } k.$$

Usaremos $\bigwedge_{t \in T} \phi_t$ y $\bigvee_{t \in T} \phi_t$ como abreviaturas para la conjunción y disyunción, respectivamente, de una familia finita de fórmulas, con las convenciones $\bigwedge \emptyset = \top$ y $\bigvee \emptyset = \perp$. Si X es un conjunto finito no vacío de variables proposicionales, la fórmula

$$\text{Uno}(X) = \left(\bigvee_{p \in X} p \right) \wedge \left(\bigwedge_{\substack{Y \subseteq X \\ |Y|=2}} \neg \left(\bigwedge_{p \in Y} p \right) \right)$$

expresa que exactamente una variable de X es verdadera.

Podemos traducir las reglas del Sudoku a fórmulas proposicionales de la siguiente manera:

- Cada celda debe contener exactamente un número del 1 al 9:

$$\Phi_{\text{cel}} = \bigwedge_{c \in Q} \text{Uno}(\{x_{c,k} : k \in D\}).$$

- Cada número del 1 al 9 debe aparecer exactamente una vez en cada fila:

$$\Phi_{\text{fil}} = \bigwedge_{i \in D} \bigwedge_{k \in D} \text{Uno}(\{x_{c,k} : c \in F_i\}).$$

- Cada número del 1 al 9 debe aparecer exactamente una vez en cada columna:

$$\Phi_{\text{col}} = \bigwedge_{j \in D} \bigwedge_{k \in D} \text{Uno}(\{x_{c,k} : c \in C_j\}).$$

- Cada número del 1 al 9 debe aparecer exactamente una vez en cada subcuadrícula de 3×3 :

$$\Phi_{\text{sub}} = \bigwedge_{a \in \{0,1,2\}} \bigwedge_{b \in \{0,1,2\}} \bigwedge_{k \in D} \text{Uno}(\{x_{c,k} : c \in B_{a,b}\}).$$

- Las pistas iniciales deben respetarse:

$$\Phi_{\text{pistas}} = \bigwedge_{c \in P} x_{c,g(c)}.$$

Por lo tanto,

$$\Phi_S = \Phi_{\text{cel}} \wedge \Phi_{\text{fil}} \wedge \Phi_{\text{col}} \wedge \Phi_{\text{sub}} \wedge \Phi_{\text{pistas}}$$

es satisfacible si y sólo si el Sudoku tiene solución. En efecto, desde un modelo de Φ_S recuperamos la solución escribiendo en cada celda c el único dígito k para el cual $x_{c,k}$ es verdadero; recíprocamente, toda solución del Sudoku induce un modelo de Φ_S .

1.3.4. Ejercicios

1. Resuelva los siguientes problemas.

- Demuestre que $(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$.
- El operador **NAND**, denotado por $\uparrow$, es tal que $\alpha \uparrow \beta$ es verdadero exactamente cuando “ α y β no son ambos verdaderos”. Demuestre que $\{\uparrow\}$ es funcionalmente completo.
- Considere la función booleana dada por la tabla

p	q	r	$f(p, q, r)$
0	0	0	1
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	0

Construya una fórmula proposicional que represente a f .

- Se debe formar un equipo eligiendo integrantes entre Ana, Bruno, Carla y Diego, sujeto a las siguientes restricciones:
 - al menos uno entre Ana y Bruno debe ser elegido;
 - si Carla es elegida, entonces Diego también debe serlo;
 - Bruno y Carla no pueden ser elegidos simultáneamente;

- exactamente uno entre Ana y Diego debe ser elegido.

Introduzca variables proposicionales y construya una fórmula Ψ cuyas valuaciones que evalúan a 1 correspondan exactamente a los equipos permitidos.

- (e) Un *literal* es una variable proposicional o la negación de una variable proposicional. Una fórmula está en *forma normal conjuntiva* si es una conjunción de disyunciones de literales. Demuestre que toda fórmula proposicional ϕ es equivalente a una fórmula en forma normal conjuntiva.

2. El objetivo de esta pregunta es diseñar un solver de *Akari* via SAT-solvers. Puede probar el juego en dailyakari.com.

Un tablero de Akari es una cuadrícula rectangular formada por celdas blancas y negras; algunas celdas negras contienen un número entre 0 y 4. Las lámparas se colocan únicamente en celdas blancas. Cada lámpara ilumina su propia celda y las celdas blancas que encuentra en las cuatro direcciones cardinales, hasta llegar al borde del tablero o a una celda negra. Una solución debe cumplir las siguientes reglas:

- toda celda blanca debe estar iluminada;
- dos lámparas no pueden iluminarse entre sí;
- si una celda negra contiene el número k , entonces exactamente k de sus 4 vecinas deben contener una lámpara.

Las celdas negras sin número no imponen una condición sobre sus vecinas, pero note que bloquean el paso de la luz.

Identifique cada celda con sus coordenadas (i, j) . Sea W el conjunto de celdas blancas y sea N el conjunto de celdas negras numeradas; para cada $b \in N$, denote por $n_b \in \{0, 1, 2, 3, 4\}$ el número escrito en b . Diremos que dos celdas blancas distintas u y w se *ven* si pertenecen a una misma fila o columna y todas las celdas estrictamente entre ellas son blancas. Para cada $w \in W$ y cada $b = (i, j) \in N$, defina, respectivamente,

$$V(w) = \{w\} \cup \{u \in W : u \text{ se ve con } w\},$$

$$A(b) = \{(i', j') \in W : |i - i'| + |j - j'| = 1\}.$$

Así, $V(w)$ contiene las celdas desde las cuales una lámpara iluminaría a w , mientras que $A(b)$ contiene las vecinas ortogonales blancas de b .

- (a) ¿Qué variables proposicionales usaría para modelar el problema de Akari?
- (b) Sea X un conjunto finito de variables proposicionales y sea $k \geq 0$. Construya una fórmula $\text{Exact}_k(X)$ que sea verdadera exactamente cuando k variables de X son verdaderas. Puede usar las convenciones $\bigwedge \emptyset = \top$ y $\bigvee \emptyset = \perp$.
- (c) Para cada celda negra numerada b , construya una fórmula que exprese que exactamente n_b de sus vecinas contienen una lámpara.
- (d) Para cada celda blanca w , construya una fórmula que exprese que w está iluminada.
- (e) Construya una fórmula que impida colocar lámparas en dos celdas blancas que se ven entre sí.

- (f) Construya una fórmula Φ_A que es satisfacible si y sólo si el tablero de Akari tiene solución. ¿Cómo usaría un SAT-solver para encontrar una solución al tablero de Akari?

3. El objetivo de esta pregunta es demostrar una versión proposicional del *teorema de interpolación de Craig*. Sean α y β fórmulas tales que $\alpha \models \beta$, y suponga que comparten al menos una variable. Denote por

$$P = \{p_1, \dots, p_m\}, \quad Q = \{q_1, \dots, q_k\}, \quad R = \{r_1, \dots, r_\ell\}$$

los conjuntos de variables que aparecen solamente en α , que aparecen en ambas fórmulas y que aparecen solamente en β , respectivamente.

Una fórmula γ es una *interpolación* de α y β si todas sus variables pertenecen a Q y además $\alpha \models \gamma$ y $\gamma \models \beta$.

- (a) Definimos $f: \{0, 1\}^k \rightarrow \{0, 1\}$ como $f(x) = 1$ si y solo si existe una valuación w de $P \cup Q$ tal que

$$w(q_i) = x_i \quad \text{para todo } i \in \{1, \dots, k\} \quad \text{y} \quad \hat{w}(\alpha) = 1.$$

Muestre que existe una fórmula γ con variables en Q tal que $\hat{v}(\gamma) = f(v(q_1), \dots, v(q_k))$ para toda valuación v de Q .

- (b) Demuestre que $\alpha \models \gamma$.
(c) Demuestre que $\gamma \models \beta$. Concluya que γ es una interpolación de α y β .
(d) Muestre que γ es la consecuencia lógica más fuerte de α que puede expresarse usando solamente las variables compartidas; es decir, si δ es cualquier fórmula construida únicamente con variables de Q y $\alpha \models \delta$, entonces $\gamma \models \delta$.
(e) Aplique la construcción anterior a

$$\alpha = (p \vee q) \wedge (\neg p \vee r), \quad \beta = q \vee r \vee s.$$

Compruebe primero que $\alpha \models \beta$ y luego construya una interpolación que use solamente las variables q y r .