

## 0.1. Resumen

La *coloración de mapas* es un problema clásico en la *Teoría de Grafos*, donde cada país se modela como un vértice y las fronteras entre países como aristas. El *Teorema de los Cuatro Colores* establece que cualquier mapa plano puede colorearse con cuatro colores sin que dos regiones adyacentes compartan el mismo color.

En este artículo, exploramos la generalización del problema de coloración de mapas al caso de la Tierra y la Luna, conocido como el **Earth Moon Problem**, propuesto por Ringel. Este problema busca determinar el número mínimo de colores necesarios para colorear un mapa donde cada país en la Tierra y su colonia lunar deben recibir el mismo color, respetando la restricción de que las regiones adyacentes en cualquiera de los dos cuerpos celestes deben tener colores distintos.

Nuestro principal aporte es demostrar que el problema de la 3-*coloración* de la Tierra-Luna es *NP-completo*, mediante una reducción desde 3-*SAT*, lo que implica que no existe un algoritmo eficiente para resolverlo en general (suponiendo  $P \neq NP$ ). Además, complementamos demostraciones previas que aparecían incompletas en la literatura y modelamos el problema como un *problema de satisfacción de restricciones* (CSP), lo que permite un análisis más profundo de su complejidad computacional.

Este trabajo no solo aporta una nueva demostración de que el problema de coloración de la Tierra-Luna con 3 colores es NP-completo, sino que también abre la puerta a futuros estudios sobre su dificultad para diferentes números de colores.

Por último, describir el problema de coloración de la Tierra-Luna a través de grafos, un caso abierto en la coloración de grafos que extiende el problema de la coloración de mapas planos. En términos de grafos, esto se puede reformular como la búsqueda del **número cromático máximo** de un grafo  $G$  que es la unión de dos grafos planares (sobre el mismo conjunto de vértices). Se demuestra mediante inducción que  $G$  es 12-coloreable, como observó Heawood. Ringel conjeturó que el Problema de la Tierra-Luna era 8-coloreable pero Sulanke reportó un ejemplo que requiere 9 colores, aún no se conoce si existen configuraciones que requieran 10, 11 o 12 colores.

## 0.2. Introducción

La coloración de mapas de la tierra es uno de los problemas de optimización más estudiados en la Teoría de Grafos. El problema de la coloración de un mapa consiste en asignar un color a cada vértice (país) de tal manera que dos vértices conectados por una arista obtengan colores diferentes, minimizando el número total de colores utilizados.

**Definición 0.1.** Una **coloración** (o **coloración propia**) de vértices en grafos es una asignación de colores a los vértices de un grafo  $G$ . De tal manera que cualquiera vértices adyacentes tienen distintos colores. Es decir, se busca una función

$$\varphi : V \longrightarrow Colores = \{1, 2, \dots, k\}$$

de tal forma que si  $uv \in E$  entonces  $\varphi(u) \neq \varphi(v)$ .

- Una coloración usando exactamente  $s$  colores se llama  **$s$ -coloración**.
- Un grafo es  **$k$ -coloreable** si existe una  $s$ -coloración de  $G$  para algún entero  $s \leq k$ .

Ahora, la siguiente afirmación no es verdadera:

”Es posible asignar uno de cuatro colores a cada país en cualquier mapa, de manera que ningún par de países que compartan una frontera tengan el mismo color.” [Hutchinson, 1993]

Sin embargo, este no es el enunciado del famoso *Teorema de los Cuatro Colores*, que fue demostrado hace más de 30 años utilizando numerosas comprobaciones por computadora.

La figura 1 muestra un pequeño ejemplo de un mapa que necesita cinco colores si cada par de países adyacentes debe recibir colores diferentes. La característica importante es que un país (#5) es desconectado y está compuesto por dos regiones.

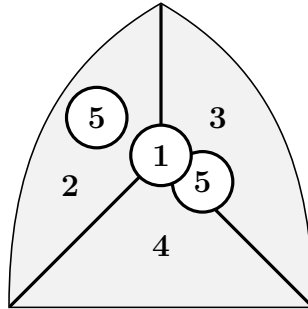


Figura 1: Un mapa plano que necesita cinco colores. Mapa tomado de [Hutchinson, 1993].

Los mapas que incluyen países desconectados son posibles; un ejemplo es el mapa de América del Norte. Esto plantea la siguiente pregunta: ¿Es posible colorear el mapa actual del mundo con cuatro colores de manera que todas las partes de cada país reciban el mismo color y que dos países diferentes con un arco fronterizo en común no reciban el mismo color? Para dar una respuesta afirmativa a dicha pregunta nos apoyaremos en el siguiente teorema.

**Teorema de los Cuatro Colores:** Este teorema establece que cualquier mapa dibujado en el plano puede colorearse con solo cuatro colores, de tal manera que cada par de regiones conectadas que comparten un borde reciban colores diferentes.

Escrito en términos de grafos:

**Teorema 0.2.** [Kempe, 1879]

*Cualquier grafo planar simple  $G$  es 4-coloreable.*

Este resultado fue propuesto por primera vez en 1852 por Francis Guthrie. Una demostración fue publicada en 1879 por Kempe, pero 11 años después, Heawood encontró un error en la prueba. Finalmente, en 1976, el teorema fue demostrado por Appel y Haken, aunque de una manera inusual. Brevemente, la demostración de Appel y Haken consiste en mostrar que cada mapa plano contiene solo una de una lista de al menos 1,800 configuraciones, y que cada configuración admite una reducción, permitiendo una demostración por inducción. Aunque los números involucrados son inusualmente grandes, la parte más inusual de la prueba es que se utilizaron aproximadamente 1,200 horas de tiempo de computadora para generar la lista de 1,800 configuraciones y verificar que las coloraciones de estas admitían la reducción necesaria [Hutchinson, 1993].

El mapa de la figura 1 también podría representar límites políticos en los que el área #5 represente, por ejemplo, un imperio. Cuando Heawood encontró tanto un error en el argumento de Kempe como descubrió que no podía corregirlo, inventó generalizaciones sobre la coloración de mapas que, hasta cierto punto, pudo resolver. Su investigación dio origen al campo de la *Teoría de Grafos Topológicos* tal como se estudia hoy en día. Primero investigó el problema de la coloración de imperios: si los mapas están formados por países unidos en imperios, ¿cuántos colores se necesitan para colorear tales mapas, siempre que todos los países en un imperio reciban el mismo color y que los imperios con una frontera común reciban colores diferentes?

Ringel sugirió una variación del problema de la coloración de imperios. Supongamos que la Luna fuera colonizada y quisiéramos colorear un mapa de la tierra y la luna con el mínimo número de colores de tal forma que:

1. Las regiones adyacentes en la tierra o en la luna reciban colores diferentes, y
2. Un país en la tierra y su colonia lunar reciban el mismo color.

A este problema se le conoce como el **Problema de la Tierra-Luna**. Este problema ha estado abierto durante más de 20 años.

Por ultimo se añade un teorema y un corolario que usaremos más adelante:

**Teorema 0.3.** *Sea  $G$  un grafo de orden  $n$  y tamaño  $m$ , cuyo conjunto de vértices es  $V = \{v_1, v_2, \dots, v_n\}$ . Entonces se satisface que;*

$$\sum_{i=1}^n \deg(v_i) = 2m.$$

**Corolario 0.4.** *Si  $G$  es un grafo planar de orden  $n \geq 3$  y tamaño  $m$ , entonces se cumple la siguiente desigualdad:*

$$m \leq 3n - 6.$$

### 0.3. Complejidad computacional de problemas que involucran coloración en grafos.

Comenzaremos con algunas definiciones antes de pasar a los resultados principales. Tanto en las matemáticas como en la teoría de la informática se han estudiado diferentes aspectos de los *problemas de decisión*. Algunos de estos estudios se enfocan en determinar si existen *algoritmos* eficientes para resolverlos y en qué medida algunos problemas de decisión son más difíciles de resolver que otros. Al analizar estos problemas, encontramos que algunos presentan mayor complejidad que otros. Un problema de decisión importante dentro de la teoría de grafos, que será el principal objeto de estudio en este capítulo, es el llamado *3-coloración de grafos planares*. Este problema consiste en:

*Determinar si dado cualquier grafo planar no dirigido  $G$ , es posible colorear los vértices de  $G$  con tres colores, digamos (Azul, Rojo y Verde) de tal manera que ningún par de vértices adyacentes tenga el mismo color.*

El problema de determinar si es posible colorear un grafo planar con 3 colores se convierte en un problema de decisión con el siguiente formato:

**Problema:** 3-coloración de grafos planares  $G$ .

**Entrada:** Un grafo planar no dirigido. ¿Es 3-coloreable?

**Salida:** Sí o No.

Sí, nos dice que el grafo es 3-coloreable; No, nos dice que no lo es.

La segunda pregunta que nos hacemos en este caso es:

*¿Qué tan difícil es este problema desde el punto de vista computacional?*

Para la elaboración de esta sección se consultaron los libros [McEliece, 1978] [Arora and Barak, 2009] [Lau, 2006] y los artículos [Demain et al., 2020] [Tsiatas, 2008].

**Definición 0.5.** Un **problema de decisión** es aquel que admite únicamente dos posibles respuestas: “Sí” o “No”, para cualquier entrada.

Para resolver problemas de decisión y otros tipos de problemas matemáticos, a menudo recurrimos a algoritmos. Estos son fundamentales para la solución de problemas tanto en computación como en matemáticas. Su definición es la siguiente:

**Definición 0.6.** Un **algoritmo** es un conjunto finito de instrucciones matemáticas bien definidas, que, cuando se ejecutan correctamente, produce un resultado específico a partir de una entrada dada.

Los algoritmos suelen ser vistos como una secuencia de instrucciones matemáticas para resolver problemas.

Un problema de decisión que puede ser resuelto a través de un algoritmo en un tiempo finito se llama **decidible**.

Entender los algoritmos es fundamental para resolver problemas computacionales. Sin embargo, no solo es importante encontrar una solución, sino también evaluar cuán eficiente es el algoritmo que utilizamos. Aquí es donde entra en juego el concepto de *complejidad* de un algoritmo, que a continuación definimos.

**Definición 0.7.** La **complejidad** de un algoritmo se refiere al tiempo y la memoria requeridos para ejecutar el algoritmo en función del tamaño de la entrada.

Para cuantificar la complejidad, utilizamos la notación **Big-O**, denotada como  $O(g(n))$ , que describe el crecimiento de una función  $f$  en términos del tamaño de la entrada. La comprensión de la notación Big-O es esencial para evaluar y comparar algoritmos en términos de su eficiencia.

**Definición 0.8.** (Big-O) Sean  $f, g : \mathbb{N} \rightarrow \mathbb{R}$  dos funciones definidas en los números enteros positivos,  $\mathbb{N}$ . Decimos que  $g$  es una **cota superior asintótica** para  $f$  si existe un número real  $C \in \mathbb{R}$  y un entero positivo  $n_0 \in \mathbb{N}$  tal que:

$$|f(n)| \leq C|g(n)| \quad \text{para todo número entero positivo } n \geq n_0.$$

En tal caso, se escribe

$$f = O(g) \text{ o } f \in O(g).$$

La cota superior asintótica  $g(n)$  se elige típicamente de manera que sea lo más simple y pequeña posible. Decimos que  $f(n)$  tiene un crecimiento **lineal, cuadrático, cúbico o polinomial** en  $n$  si  $f(n)$  pertenece a  $O(n)$ ,  $O(n^2)$ ,  $O(n^3)$  o  $O(n^k)$  para algún  $k \in \mathbb{N}$ , respectivamente. Cuando hablamos de la *eficiencia* de un algoritmo, nos referimos al tiempo que este tarda en ejecutarse en función del tamaño de la entrada. *Los algoritmos con tiempo de ejecución polinomial*, como la suma o multiplicación de números, se consideran **eficientes**. Por otro lado, si un algoritmo necesita iterar sobre cada instancia de un conjunto con  $2^n$  elementos, *su complejidad es exponencial en  $n$* . Un problema se considera **difícil** si no existe un algoritmo eficiente, es decir, de tiempo polinomial, que pueda resolverlo.

Existen clases distintas de problemas; aquellos cuyos algoritmos tienen tiempo de ejecución polinomial, o no.

## 0.4. Clases de problemas

A continuación definimos las clases de problemas más usados, para preparar esta sección nos basamos en las notas del curso *Design and Analysis of Algorithms* del Profesor Erick Demaine [Demaine et al., 2020].

**Definición 0.9.**

- Un algoritmo es polinomial si para algún  $k$ , su tiempo de ejecución sobre las entradas de tamaño  $n$  es  $O(n^k)$ . La clase de **problemas tractables**, denotada por **P**, comprende los problemas de decisión que se pueden resolver mediante un algoritmo de tiempo polinomial. Estos problemas se consideran eficientes.
- La clase de problemas **NP**, denominada **tiempo polinomial no determinista** es la clase de problemas de decisión en los que se permite adivinar y *verificar* su solución en tiempo polinomial. El hecho de

que una solución pueda adivinarse a partir de muchas opciones polinomiales en tiempo constante se le conoce como no-determinista.

**Definición 0.10.** Dentro de la clase de problemas NP hay una clase de problemas denominados **NP-completos**, que en términos generales son considerados difíciles de resolver y si existe un algoritmo que pueda resolver un problema NP-completo en tiempo polinomial, entonces también puede resolver cualquier otro problema NP en tiempo polinomial. En otras palabras, un problema  $X$  es NP-completo si  $X \in \text{NP-hard}$  (véase definición abajo).

En la práctica, verificar si una prueba es válida parece ser más sencillo que encontrar la solución a un problema. Sin embargo, en el campo de la informática, persiste una pregunta fundamental sin resolver: ¿Es la clase NP estrictamente más grande que la clase P? Esta pregunta es una de las incógnitas más importantes en la teoría de algoritmos.

**Definición 0.11.** Una **reducción** del problema  $A$  al problema  $B$  en tiempo polinomial, denotada por  $A \leq_p B$ , es una transformación tal que existe un algoritmo que convierte las entradas del problema  $A$  en entradas equivalentes del problema  $B$  en tiempo polinomial. Equivalente significa que para cualquier entrada, el problema  $A$  y el problema  $B$  producen la misma respuesta (sí o no). Sea  $A \leq_p B$ :

- Si  $A \in \text{NP-hard}$ , entonces  $B \in \text{NP-hard}$ .
- Si  $B \in P$ , entonces  $A \in P$ .
- Si  $B \in \text{NP}$ , entonces  $A \in \text{NP}$ .

**Definición 0.12.** Un problema  $X$  es **NP-hard** o **NP-duro** si cada problema  $Y \in \text{NP}$  se reduce a  $X$ . Es decir, un problema es NP-hard si es al menos tan difícil como todos los problemas en NP.

Con lo anterior podemos definir formalmente los problemas *NP-completos*.

**Definición 0.13.** Un problema  $X$  es **NP-completo** si  $X \in \text{NP}$  y  $X$  es NP-hard.

Un ejemplo muy famoso de problemas NP-completos es el problema denominado 3SAT, con el fin de hacer este trabajo auto contenido y definir formalmente un problema 3SAT introduciremos las siguientes definiciones.

**Definición 0.14.** Una **fórmula booleana** es una expresión formada con las variables  $u_1, \dots, u_n$  con  $u_i \in \{0, 1\}$  para  $i \in \{1, 2, \dots, n\}$ , junto con los operadores lógicos **y** ( $\wedge$ ), **no** ( $\neg$ ) y **o** ( $\vee$ ). Sea  $\varphi$  una fórmula booleana y  $z \in \{0, 1\}^n$ , entonces  $\varphi(z)$  denota el valor de  $\varphi$  cuando a las variables de  $\varphi$  se les asignan los valores  $z$ . Una fórmula  $\varphi$  es **satisfacible** si existe alguna asignación  $z$  tal que  $\varphi(z)$  sea verdadera.

Una fórmula booleana está en **forma CNF** (Forma Normal Conjuntiva) si es una conjunción (y) de disyunciones (o) de variables o sus negaciones. Es decir, una fórmula CNF tiene la forma

$$\bigwedge_i \left( \bigvee_j v_{ij} \right),$$

donde cada  $v_{ij}$  es un literal que puede ser la variable  $u_k$  o su negación  $\neg u_k$ . Los términos  $v_{ij}$  se llaman **literales** y las expresiones  $\left( \bigvee_j v_{ij} \right)$  se llaman **cláusulas**. Un **kCNF** es una fórmula CNF en la cual todas las cláusulas contienen a lo sumo  $k$  literales.

Con esta definición podemos dar paso a una definición formal de un tipo de fórmula booleana denominada 3SAT, para después demostrar que el problema de decidir si una fórmula booleana que está en 3CNF es o no válida se reduce a un problema de 3-coloración en grafos en tiempo polinomial y viceversa. Así que son equivalentemente difíciles y concluiremos que el problema de 3-coloración en grafos es NP-completo pues 3SAT lo es.

**Definición 0.15. 3SAT:** El problema de decisión 3SAT se puede plantear como sigue. Dada una fórmula booleana de la forma:

$$(x_1 \vee x_3 \vee x_6) \wedge (\overline{x_2} \vee x_3 \vee x_7) \wedge \dots,$$

¿Existe una asignación de variables verdadero (1) y falso (0), tal que toda la fórmula evalúe a verdadero (1)?

Fue probado que el problema 3SAT resulta ser NP-completo por Cook en 1971 [Arora and Barak, 2009]. A continuación, enunciamos el teorema y definimos formalmente el problema:

**Teorema 0.16.** (Teorema de Cook-Levin [Arora and Barak, 2009]) *Sea SAT el lenguaje de todas las fórmulas CNF satisfacibles y 3SAT el lenguaje de todas las fórmulas 3CNF satisfacibles. Entonces:*

1. SAT es NP-completo.
2. 3SAT es NP-completo.

En el problema de decisión de 3-coloración, se nos da un grafo  $G$  y nos preguntamos si existe una forma de colorear los vértices de dicho grafo utilizando tres colores, a saber: rojo, verde o amarillo, de tal manera que ningún par de vértices adyacentes comparta el mismo color.

El objetivo de esta sección es demostrar una reducción en tiempo polinomial del problema 3SAT al problema de 3-coloración. Es decir, demostraremos el siguiente teorema:

**Teorema 0.17.**  $3SAT \leq_p 3\text{-Coloración}$ .

Antes de demostrar este teorema, esbozaremos la estrategia de la prueba. Dada una entrada del problema 3SAT (una fórmula booleana), debemos construir un grafo que será 3-coloreable si y solamente si la fórmula booleana es satisfacible. La idea de la siguiente prueba fue tomada de [Tsias, 2008].

*Demostración.* La demostración se ejecutará en varios pasos.

- **Paso 1:** Comenzamos construyendo un grafo que contiene 3 vértices etiquetados como  $T$ ,  $F$ , y  $S$ , conectados formando un triángulo. Coloreamos estos vértices con tres colores diferentes. Sin pérdida de generalidad, podemos colorearlos como se muestra en la figura 2:

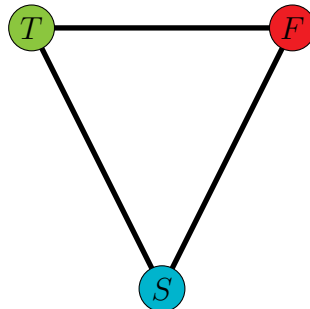


Figura 2: Primer paso

En esta construcción, el color verde indica los valores verdaderos, el color rojo indica los valores falsos y el color de  $S$  no representa ningún valor.

- **Paso 2:** En nuestra fórmula, tenemos variables y negaciones de variables, conocidas como literales. Debemos asignar valores a los literales, por lo que basta asignar valores a su variable correspondiente. Para asegurar esto, hacemos lo siguiente:

Para cada variable  $x_i$  y su negación  $\overline{x_i}$ , creamos dos vértices en nuestro grafo y conectamos estos vértices al vértice  $S$ , como se muestra en la figura 3.

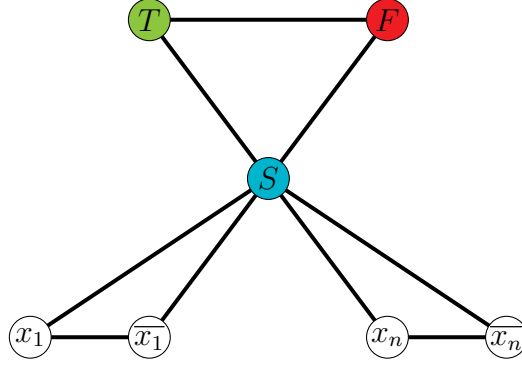


Figura 3: Segundo paso

Esto asegura que los literales  $x_i$  y  $\overline{x_i}$  no reciban el mismo color que el vértice  $S$ , ya que están conectadas a  $S$  y, por lo tanto, deben ser coloreadas de rojo o verde. Para garantizar que  $x_i$  y  $\overline{x_i}$  no reciban el mismo color (es decir, que si  $x_i$  es verdadero, entonces  $\overline{x_i}$  sea falso, y viceversa), conectamos el vértice  $x_i$  con el vértice  $\overline{x_i}$ , como se muestra en la figura 3.

De esta manera, si coloreamos el grafo de una forma válida, obtendremos valores de verdad para las variables y sus negaciones que son consistentes.

- **Paso 3:** Finalmente, necesitamos representar cada cláusula de la fórmula booleana a través de un grafo. Para ello, representaremos el operador booleano  $OR$  mediante una construcción que llamaremos “gadget” (véase figura 4). Este gadget es 3-coloreable si y solo si al menos una de las literales en la cláusula es asignada o coloreada con  $T$  (verdadero), como veremos a continuación.

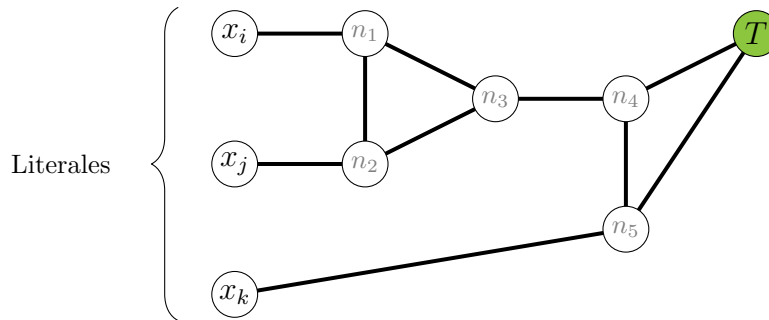


Figura 4: Tercer paso

En el gadget de la figura 4, si todas las literales se colorean como Falso (rojo), el vértice  $n_5$  debe ser coloreado de azul, ya que está conectado a  $T$  (verde) y a  $x_k$  (rojo). A continuación, el vértice  $n_4$  debe ser coloreado de rojo, ya que está conectado a  $T$  (verde) y a  $n_5$  (azul), como se muestra en la figura 5:

Esto implica que ni el vértice  $n_3$ , ni  $n_1$ , ni  $n_2$  pueden ser coloreados de rojo. Por lo tanto, solo pueden tener los colores verde o azul. Sin embargo, al tratarse de tres vértices y solo dos colores disponibles,

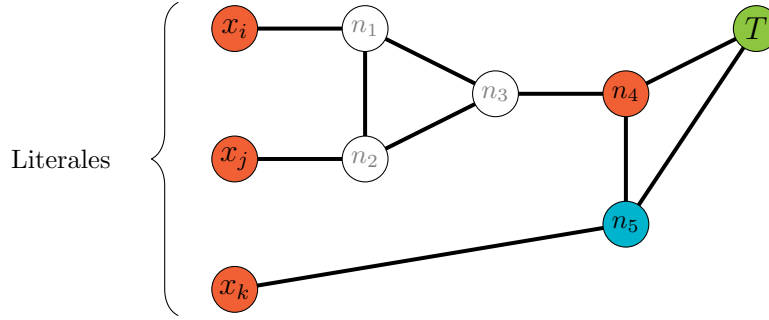


Figura 5: Coloración del tercer paso

necesariamente habrá un par de vértices que compartan el mismo color. Como estos tres vértices están conectados entre sí, se produciría una contradicción, ya que dos vértices conectados tendrían el mismo color. De este modo, si las tres literales son falsas, es imposible colorear el gadget.

- **Paso 4:** El paso final consiste en intersectar los gadgets, es decir, se crea un gadget por cada cláusula y se busca empatar esos gadgets. Aunque el resultado es grande, está bien estructurado, como se muestra en la figura 6. De este modo, el grafo resultante es 3-coloreable si y solo si la fórmula booleana se satisface.

En el caso de que la fórmula booleana representada por la figura no se satisfaga, es decir, si  $x_1$ ,  $x_2$ , y  $\overline{x_3}$  son coloreadas de rojo, entonces al aplicar el proceso descrito en el paso 3, se llega a la conclusión de que es imposible colorear el grafo.

- **Paso 5:** Para demostrar que esta reducción se realiza en tiempo polinomial, supongamos que la fórmula tiene  $n$  variables y  $m$  cláusulas. Entonces, el número de vértices y aristas se calcula de la siguiente manera:

**Vértices:**

- 3 vértices para el triángulo formado por los vértices  $T$ ,  $F$  y  $S$  (figura 2).
- 2 vértices por cada variable: se requieren  $2n$  vértices extra (figura 3).
- 5 vértices por cada cláusula: se requieren  $5m$  vértices extra (figura 4).

**Aristas:**

- 3 aristas para el triángulo formado por los vértices  $T$ ,  $F$  y  $S$  (figura 2).
- Aristas entre pares  $(x_i, \overline{x_i})$  opuestos a las literales:  $n$  aristas (figura 3).
- Aristas de las literales al vértice  $S$ : Se tienen  $2n$  aristas (figura 3).
- 10 aristas por cláusula (gadget): Se tienen  $10m$  aristas (figura 4).

De este modo, el grafo generado tiene  $2n + 5m + 3$  vértices y  $3n + 10m + 3$  aristas. La construcción del grafo se realiza en un número de pasos que es polinomial respecto a  $n$  y  $m$ , garantizando así que la reducción se lleva a cabo en tiempo polinomial.

Hasta ahora hemos probado que un problema 3SAT se reduce a un problema de 3-coloración en grafos en general. Como 3SAT es un problema NP-completo por la reducción la 3-coloración en grafos también lo es.

□



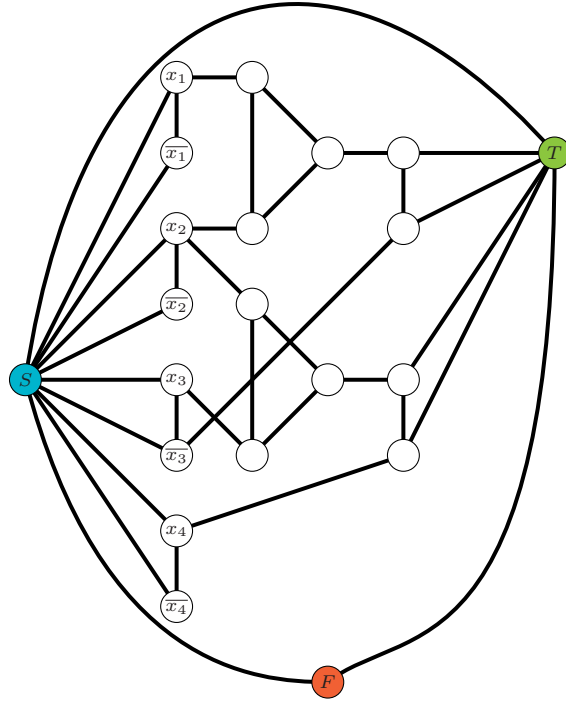


Figura 6: Reducción completa de la fórmula 3SAT  
 $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee x_4)$  a una 3-coloración

Otra forma de demostrar que los problemas de 3-coloración en grafos son NP es modelar este problema como un problema de satisfacción de restricciones, dado que estos últimos son NP. En general, aunque en 2017 se demostró que, en el caso de que  $P \neq NP$  [Zhuk, 2020], los problemas de satisfacción de restricciones pertenecen a la clase de problemas en los que se satisface la dicotomía; es decir, son P o son NP.

## 0.5. Problemas de Satisfacción de Restricciones.

Los Problemas de Satisfacción de Restricciones (CSP, por sus siglas en inglés) constituyen un área fundamental en la teoría de la computación. Estos se definen como conjuntos de objetos que deben satisfacer una serie de restricciones o limitaciones.

A continuación, los definiremos formalmente:

**Definición 0.18.** Un **Problema de Satisfacción de Restricciones CSP** se define como una triplete  $\langle X, D, C \rangle$ , donde:

- $X = \{x_1, x_2, x_3, \dots\}$  es un **conjunto de variables**, que puede o no ser infinito.
- $D = \{d_1, d_2, d_3, \dots\}$  es el **conjunto de valores** que pueden tomar las variables o dominio.
- $C = \{c_1, c_2, \dots, c_k\}$  es un **conjunto finito de restricciones**, donde cada restricción  $c_i = \langle t_i, R_i \rangle$  para  $i \in \{1, 2, \dots, k\}$  consiste de una  $n_i$ -**tupla de variables**  $t_i = (x_{i1}, x_{i2}, \dots, x_{in_i})$  y una **relación  $n_i$ -aria**  $R_i$  sobre  $D$ .

**Definición 0.19.** Una **evaluación** de las variables es una función

$$\varphi : X \rightarrow D.$$

Decimos que una evaluación  $\varphi$  **satisface** una restricción  $c_i = \langle t_i, R_i \rangle$  con  $t_i = (x_{i1}, x_{i2}, \dots, x_{in_i})$  si la tupla de valores  $(\varphi(x_{i1}), \varphi(x_{i2}), \dots, \varphi(x_{in_i}))$  pertenece a la relación  $R_i$ .

Una **solución del CSP** es una evaluación que satisface todas las restricciones del problema.

Muchos problemas prácticos pueden modelarse como problemas de satisfacción de restricciones. Un ejemplo clásico de esto es el *problema de satisfacibilidad booleana (SAT)* visto en la sección anterior. En esencia, este problema implica encontrar asignaciones de valores a variables que hagan que una fórmula lógica proposicional sea verdadera. A continuación mostraremos como un problema SAT se modela como un problema CSP. El siguiente ejemplo se construyó tomando como base el ejemplo del libro [Lau, 2006].

**Ejemplo 0.20.** El **problema de satisfacibilidad booleana (SAT)** consiste en determinar si existen valores de ceros y unos para las variables de una fórmula lógica proposicional como la siguiente, que hagan verdadera la fórmula. Consideremos la siguiente fórmula:

$$t(x_1, x_2, x_3, x_4) := (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (\overline{x_3} \vee \overline{x_4} \vee x_1) \wedge (\overline{x_3} \vee \overline{x_2} \vee x_4) \wedge (\overline{x_1} \vee \overline{x_3}).$$

El problema consiste en asignar valores de 0 o 1 para  $a_1, a_2, a_3, a_4$  de tal manera que al evaluar la fórmula  $t$ , esta resulte verdadera, es decir,  $t(a_1, a_2, a_3, a_4) = 1$ . Este tipo de problemas se puede expresar como un CSP, en específico, de la siguiente manera:

- Sea  $V = \{x_1, x_2, x_3, x_4\}$ .
- Sea  $D = \{0, 1\}$ .
- Sea  $C$  el conjunto de restricciones definido por:

$$C := \{c_1, c_2, c_3, c_4, c_5\}$$

donde:

- $c_1 := ((x_1, x_2, x_3, x_4), D^4 \setminus \{(0, 0, 0, 0)\})$ .
- $c_2 := ((x_1, x_2, x_3), D^3 \setminus \{(1, 1, 0)\})$ .
- $c_3 := ((x_3, x_4, x_1), D^3 \setminus \{(1, 1, 0)\})$ .
- $c_4 := ((x_3, x_2, x_4), D^3 \setminus \{(1, 1, 0)\})$ .
- $c_5 := ((x_1, x_3), D^2 \setminus \{(1, 1)\})$ .

Por lo tanto, la solución del ejemplo anterior consiste en encontrar las asignaciones de valores  $\varphi : X \rightarrow D$  que satisfagan todas las restricciones del problema. En este caso, las soluciones son las siguientes:

	$x_1$	$x_2$	$x_3$	$x_4$
$\varphi_1$	0	0	0	1
$\varphi_2$	0	0	1	0
$\varphi_3$	0	1	0	0
$\varphi_4$	0	1	0	1
$\varphi_5$	1	0	0	0
$\varphi_6$	1	0	0	1

Los CSP abarcan una amplia gama de problemas, desde el famoso Problema de las Ocho Reinas hasta el desafiante Teorema de los Cuatro Colores en la coloración de mapas. Numerosos juegos y rompecabezas, como por ejemplo el Sudoku, pueden modelarse como problemas de satisfacción de restricciones.

A continuación, formularemos el problema de coloración de mapas como un problema CSP, donde la tarea consiste en determinar si un mapa dado puede ser coloreado con tres colores distintos de manera que ningún par de regiones adyacentes tenga el mismo color.

**Definición 0.21.** El **Problema de Coloración** de un grafo (véase teorema 0.1). Sea un grafo  $G = (V, E)$  donde  $V = \{v_1, v_2, \dots, v_n\}$  es el conjunto de vértices y  $E = \{e_1, e_2, \dots, e_m\}$  es el conjunto de aristas. El problema consiste en determinar si es posible colorear los vértices de  $G$  con  $k$  colores de tal manera que vértices adyacentes tengan colores diferentes.

El problema anterior lo formularemos como un CSP, donde:

- $X = \{v_1, v_2, \dots, v_n\}$  es el conjunto de vértices de  $G$ .
- $D = \{d_1, d_2, \dots, d_k\}$  es el conjunto de colores que pueden asignarse a los vértices.
- $C = \{c_1, c_2, \dots, c_m\}$  es el conjunto de restricciones, donde cada restricción  $c_i = \langle e_i, \neq_D \rangle$  para  $i \in \{1, 2, \dots, m\}$ , con  $e_i = (u_i, v_i)$  representando las aristas de  $G$  y  $\neq_D$  la relación de desigualdad sobre  $D$ , definida como:

$$\neq_D = \{(d_i, d_j) \in D^2 \mid d_i \neq d_j\}.$$

Por lo tanto, una solución al problema de colorear el mapa consiste en encontrar una función

$$\varphi : X \longrightarrow D$$

que asigne un color a cada vértice en  $v_i \in X$  con  $i \in \{1, 2, \dots, n\}$  tal que

$$\text{para todo } e_i = (u_i, v_i) \in E \text{ entonces } (\varphi(u_i), \varphi(v_i)) \in \neq.$$

Lo anterior equivale a ver si

para todo  $e_i = (u_i, v_i) \in E$  entonces  $\varphi(u_i) \neq \varphi(v_i)$ .

Finalizaremos este capítulo modelando un problema de 3-coloración como un problema 3SAT.

**Ejemplo 0.22. Modelando un problema de 3-coloración de grafos como un problema 3SAT.**

Sea un grafo  $G = (V, E)$ . Formularemos este problema como un 3SAT de la siguiente manera:

- Las variables del conjunto  $u \in \{u_{1,i}, u_{2,i}, \dots, u_{n,i}\}$  con  $i \in \{1, 2, 3\}$  representan los colores asignados a los vértices en  $G$ . Hay 3 variables por cada vértice en  $G$ , una por cada color posible.
- El dominio  $D = \{0, 1\}$  indica los valores que puede tomar cada variable  $u$ , donde 1 significa que el color es asignado y 0 que no lo es.
- Para cada vértice  $v_j \in V$  con  $j \in \{1, 2, \dots, n\}$ , asignamos las siguientes restricciones:

- Cada vértice debe ser coloreado con al menos un color, es decir, se debe satisfacer la siguiente fórmula:

$$u_{j,1} \vee u_{j,2} \vee u_{j,3}$$

para esta restricción se necesita una sola cláusula por cada vértice, es decir  $n$  cláusulas en total para el grafo.

- En nuestro problema de coloración no se pueden asignar dos colores al mismo tiempo al mismo vértice y esto se logra con la siguiente expresión lógica:

$$(\overline{u_{j,1}} \vee \overline{u_{j,2}}) \wedge (\overline{u_{j,1}} \vee \overline{u_{j,3}}) \wedge (\overline{u_{j,2}} \vee \overline{u_{j,3}})$$

para esta restricción se necesitan  $3 \times n$  cláusulas.

- Cada par de vértices adyacentes son de diferente color. Para todo  $e_j = (v_j, v_k) \in E$  se tiene la restricción:

$$(\overline{u_{j,1}} \vee \overline{u_{k,1}}) \wedge (\overline{u_{j,2}} \vee \overline{u_{k,2}}) \wedge (\overline{u_{j,3}} \vee \overline{u_{k,3}})$$

para esta restricción se necesitan  $3 \times m$  cláusulas

Así, el problema de 3-coloración se reduce a encontrar una asignación a las variables  $z \in \{0, 1\}^{3n}$  que haga que la fórmula booleana descrita por la intersección de todas las cláusulas anteriores sea satisfacible.

Con este ejemplo, observamos que para cualquier grafo  $G = (V, E)$  se puede escribir como un problema 3SAT utilizando un total de  $n + 3n + 3m = 4n + 3m$  restricciones. De este modo hay una reducción de un problema de coloración en grafos a un problema 3SAT en tiempo polinomial.

Para demostrar que el problema de decidir si un grafo planar se puede colorear con 3 colores es NP-completo, se realiza una reducción en tiempo polinomial desde el problema de colorear cualquier grafo con 3 colores. Es decir, se demuestra el siguiente teorema:

**Teorema 0.23.** *El problema de decidir si un grafo es 3-coloreable se reduce en tiempo polinomial al problema de decidir si un grafo planar es 3-coloreable.*

$$\text{Grafo 3-coloreable} \leq_p \text{Grafo planar 3-coloreable}$$

La estrategia consiste en comenzar con un grafo no planar que tenga cruces, y reemplazar cada cruce por un *gadget planar*: un subgrafo de 11 vértices que puede ser coloreado con 3 colores. Este proceso se realiza para cada cruce, transformando así el grafo original en un grafo planar que es 3-coloreable si, y solo si, el grafo original no planar también lo es. El lector interesado en los detalles de la reducción puede consultar [Gorrie, 2014].

A partir del teorema anterior y de la reducción del problema 3-SAT al problema de 3-coloración, obtenemos que:

$$3\text{-SAT} \leq_p \text{Grafo planar 3-coloreable}.$$

Y, dado que 3-SAT es NP-completo, concluimos que el problema de 3-coloración en grafos planares también es NP-completo.

## 0.6. El Problema de la Tierra-Luna (Earth-Moon Problem).

El problema de la Tierra-Luna es un problema que permanece abierto dentro del ámbito de la coloración de grafos, planteado por Gerhard Ringel en 1959 [Hutchinson, 1993]. Como vimos en la introducción este problema es una extensión del problema de la coloración de mapas planos, los que discutimos en el capítulo 2, cuya solución se obtiene a través del Teorema de los Cuatro Colores (véase teorema 0.2).

De manera intuitiva, el problema puede enunciarse de la siguiente forma:

*¿Cuántos colores se necesitan para colorear los mapas políticos de la tierra y la luna en un futuro hipotético, donde cada país de la tierra tiene una colonia en la luna que debe recibir el mismo color?*



Figura 7: Territorio de países en tierra y luna.

En la figura 7 se muestra un ejemplo de una coloración para la tierra y la luna. Nótese que cada país debe ser coloreado con el mismo color tanto en la tierra como en la luna, aunque la disposición geográfica es diferente en ambos cuerpos celestes.

La pregunta anterior se puede formular de la siguiente manera: ¿Cuál es el número mínimo de colores necesarios para colorear un conjunto de países, de tal forma que no haya dos países que compartan una frontera común y estén coloreados con el mismo color, bajo la condición de que cada país consiste en una región en la tierra y una región en la luna?

En términos de teoría de grafos, esta pregunta se puede reformular como sigue:

¿Cuál es el número cromático máximo de un grafo  $G$  que es la unión de dos grafos planares (sobre el mismo conjunto de vértices)?

Nos gustaría establecer algunas cotas sobre el número de colores necesarios. Comenzaremos con un ejemplo dado por Thom Sulanke en 1974, que refuta la conjetura de Ringel, la cual postulaba que 8 colores serían suficientes para resolver la pregunta.

Antes de continuar, es importante señalar que para la preparación de este capítulo se consultaron los artículos [Ringel, 2012], [Hutchinson, 1993] y [Gonthier et al., 2008]. La mayoría de las demostraciones aquí expuestas se completaron o se hizo una demostración alternativa.

**Ejemplo 0.24.** Supongamos que tenemos la coloración del mapa en “la tierra” donde cada letra  $\{A, B, C, D, E, F, P, Q, R, S, T\}$  representa un país diferente.

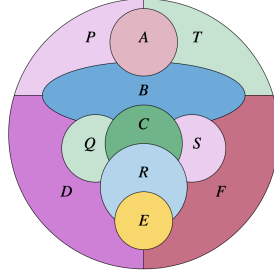


Figura 8: Coloración en “la tierra”.

El grafo correspondiente  $G_1$  (figura 9) para este mapa tiene 11 vértices (uno por cada país) y 26 aristas (una por cada colindancia), cumpliendo la desigualdad de Euler para grafos planos,  $m = 26 < 3 \times 11 - 6$ , donde  $m$  es el número de aristas y 11 es el número de vértices. Así,

$$G_1 = (V_1, E_1)$$

queda definido por

$$V_1 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

$$E_1 = \left\{ \begin{array}{l} AT, AB, AP, BC, BD, BF, BT, BP, BS, BQ, \\ CQ, CS, CR, DP, DQ, DR, DE, DF, ER, EF, FT, \\ FS, FR, PT, QR, RS \end{array} \right\}$$

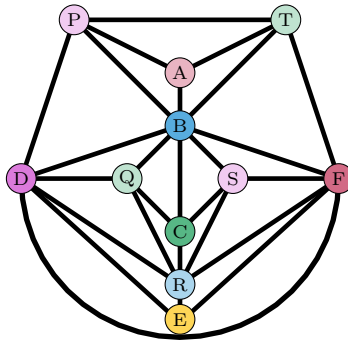


Figura 9: Grafo de mapa en “la tierra”.  $G_1$

El grafo  $G_1$  es isomorfo al grafo de la figura 10, que a simple vista no parece ser plano; sin embargo, como se observa en la figura 9, sí posee una representación plana. Más adelante usaremos este grafo para simplicidad visual.

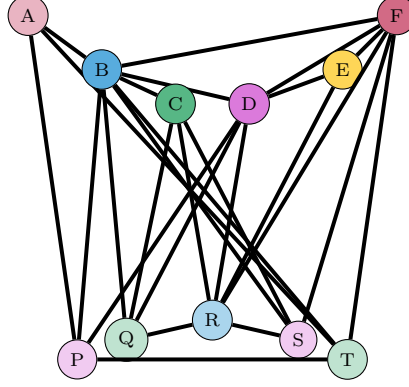


Figura 10: Grafo de mapa en “la tierra”.

A continuación, analizamos el mapa en la luna. Supongamos que los 11 países del mapa terrestre se mantienen en la luna, pero en este caso, cada país está dividido en regiones diferentes a las del mapa terrestre. Esta variación en la división regional altera las colindancias entre las regiones en la luna, en comparación con las del mapa de la tierra.

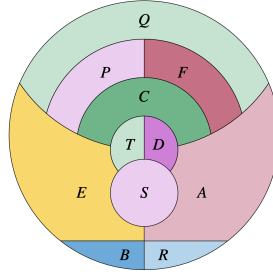


Figura 11: Coloración en “la luna”.

Denotemos por  $G_2$  el grafo que representa el mapa lunar (figura 12). En  $G_2$  las aristas indican las nuevas colindancias entre las regiones lunares.

El grafo  $G_2$  correspondiente al mapa lunar tiene 11 vértices (uno por cada región) y 24 aristas. Así,  $G_2 = (V_2, E_2)$  queda definido por

$$V_2 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

$$E_2 = \left\{ \begin{array}{l} AC, AQ, AF, AD, AS, AE, AR, \\ BR, BE, CF, CD, CT, CE, CP, \\ DS, DT, ET, ES, EQ, EP, \\ FQ, FP, PQ, ST \end{array} \right\}$$

Reordenando los vértices alfabéticamente para facilitar el análisis, obtenemos el grafo de la figura 13.

Para garantizar que las colindancias sean consistentes entre los dos mapas, debemos considerar la unión de los grafos  $G_1$  y  $G_2$ . En donde el conjunto de vértices  $\{A, B, C, D, E, F, P, Q, R, S, T\}$  es el mismo, pero el de aristas será la unión de ellos. Al grafo resultante, que representa la combinación de las regiones de ambos mapas, le llamamos  $G_3 = (V_3, E_3)$  donde:

$$V_3 = V_1 = V_2 = \{A, B, C, D, E, F, P, Q, R, S, T\}$$

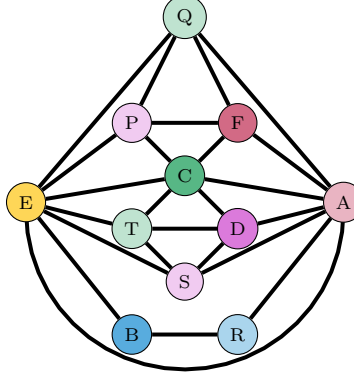


Figura 12: Grafo de mapa en “la luna”.  $G_2$

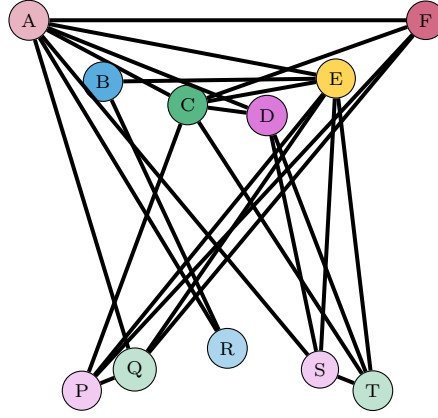


Figura 13: Grafo de mapa en “la luna”.

$$E_3 = E_1 \cup E_2 = \left\{ \begin{array}{l} AC, AQ, AF, AD, AS, AE, AR, BR, BE, CF, \\ CD, CT, CE, CP, DS, DT, ET, ES, EQ, EP, \\ FQ, FP, PQ, ST, AT, AB, AP, BD, BF, BT, \\ BP, BS, BQ, CQ, CS, CR, DP, DQ, DR, DE, \\ ER, EF, FT, FS, FR, PT, QR, RS \end{array} \right\}$$

El grafo final  $G_3$  combina las coloraciones de las regiones de la Tierra y la Luna, respetando las adyacencias especificadas en ambos mapas. La pregunta clave es: ¿cuál es el número mínimo de colores necesarios para colorear este grafo de manera que no haya dos regiones adyacentes con el mismo color?

En este caso, hemos determinado que se requieren al menos 9 colores, y este número es óptimo. Esto se debe a que el subgrafo formado por los vértices  $A$  a  $F$  y las aristas que los conectan es un grafo completo  $K_6$ , donde cada vértice está conectado con todos los demás. Por lo tanto, se necesitan al menos 6 colores para colorear  $K_6$ .

Por otro lado, el subgrafo formado por los vértices  $P$  a  $T$  es un ciclo  $C_5$ , que también requiere colores distintos de los usados en  $K_6$ . Para colorear un ciclo  $C_5$ , se necesitan al menos 3 colores adicionales. Dado que  $G_3$  es la unión de  $K_6$  y  $C_5$ , la cota mínima de colores necesarios es 9. Además, este número es suficiente, como se ha demostrado en el ejemplo, donde se logra una coloración correcta con 9 colores.

Como se puede observar, el Problema de la Tierra-Luna se puede formular como un grafo  $G = (V, E)$  cuyo conjunto de vértices es el mismo, pero cuyo conjunto de aristas se puede dividir para formar dos grafos planares  $G_1$  y  $G_2$ . A la operación inversa, es decir, construir un grafo de Tierra-Luna la llamaremos *unión*.



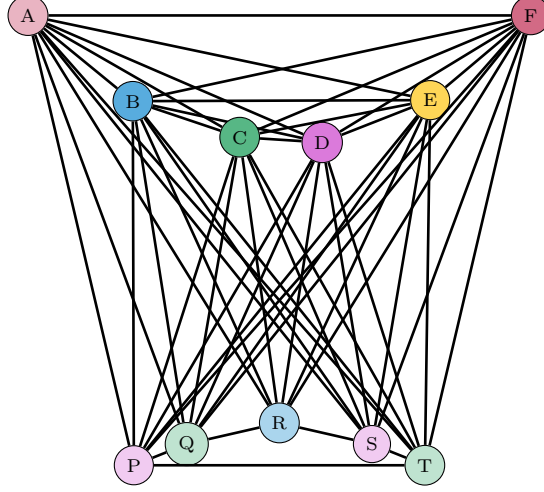


Figura 14: Grafo Tierra-Luna.  $G_3$

**Definición 0.25.** Sean  $G_1 = (V, E_1)$  y  $G_2 = (V, E_2)$  dos grafos planos definidos sobre el mismo conjunto de vértices  $V$ , definimos la **unión** de estos grafos como el grafo  $G = (V, E)$ , donde  $E = E_1 \cup E_2$ . Es decir, el grafo unión mantiene el mismo conjunto de vértices y su conjunto de aristas corresponde a la unión de las aristas de  $G_1$  y  $G_2$ .

Por el corolario 0.4, sabemos que un grafo planar con  $n \geq 3$  vértices tiene a lo sumo  $m \leq 3n - 6$  aristas. De este modo, el Problema de la Tierra-Luna tendrá como máximo  $3n - 6$  aristas en un grafo plano  $G_1$  y  $3n - 6$  aristas en el grafo plano  $G_2$ .

Por lo tanto, tenemos el siguiente corolario.

**Corolario 0.26.** Si  $G$  es un grafo con  $n \geq 3$  vértices que es la unión de dos grafos planares, entonces  $G$  tendrá como máximo  $|E| \leq 2(3n - 6) = 6n - 12$  aristas.

**Corolario 0.27.** Si  $G$  es un grafo que es la unión de dos grafos planares, existe al menos un vértice con grado no mayor a 11.

*Demostración.* Si  $n \leq 2$  el resultado es trivial, ya que todos los vértices tendrán grado no mayor a 11. Supongamos que  $n \geq 3$ .

El resultado se obtendrá por contradicción. Supongamos, que tenemos un grafo  $G = (V, E)$  unión de dos planares de orden  $n$  y tamaño  $m$  en el cual todos sus vértices tienen un grado mayor o igual a 12.

Usando el corolario anterior (0.26) y el teorema 0.3, podemos afirmar que:

$$\sum_{i=1}^n \deg(v_i) = 2m \leq 2(6n - 12) = 12n - 24.$$

Sin embargo, bajo nuestra hipótesis de que todos los vértices tienen un grado mayor o igual a 12, y aplicando nuevamente el teorema 0.3, obtenemos:

$$12n \leq \sum_{i=1}^n \deg(v_i) = 2m.$$

Combinando las dos desigualdades anteriores obtenemos:

$$12n \leq 2m \leq 12n - 24.$$

Esta última desigualdad es una contradicción. Por lo tanto, llegamos a la conclusión de que debe existir al menos un vértice en  $G$  con un grado a lo más de 11.  $\square$

**Teorema 0.28.** [Hutchinson, 1993] *El grafo  $G$  del Problema de la Tierra-Luna es 12-coloreable.*

*Demostración.* Probaremos este teorema usando inducción matemática sobre el orden de  $G$ . Sea  $n$  el orden de  $G$ .

*Base de inducción:* Supongamos que  $n \leq 12$ . Si  $G$  tiene  $n \leq 12$  vértices, entonces existe una 12-coloración para  $G$ , ya que, basta con asignar un color distinto a cada vértice, y por construcción, no habrá dos vértices adyacentes con el mismo color. Por lo tanto, el resultado se cumple para  $n \leq 12$ .

*Hipótesis de inducción:* Supongamos que todo grafo planar de orden  $n$  es 12-coloreable.

*Paso de inducción:* Por demostrar que cualquier grafo  $G = (V, E)$  de orden  $n + 1$  es 12-coloreable. Por el colorario 0.27 sabemos que en cualquier grafo que es la unión de dos grafos planares existe al menos un vértice  $v$  con  $\deg(v) \leq 11$ . Sea  $G' = (V', E)$  el subgrafo de  $G$ , donde  $V' = V \setminus \{v\}$  y  $E'$  incluye todas las aristas que no son incidentes con  $v$ . En otras palabras,  $G'$  es el grafo resultante de “eliminar”  $v$  y todas las aristas que lo conectan. Entonces, el orden de  $G'$  es  $n$ , y por la hipótesis de inducción, obtenemos que  $G'$  es 12-coloreable. De lo anterior se sigue que existe una  $s$ -coloración para  $G'$  con  $s \leq 12$ . Utilizamos esta 12-coloración para  $G$  con los colores  $\{c_1, c_2, \dots, c_{12}\}$ , y solo falta asignar un color a  $v$ . Sabemos que  $v$  tiene a lo más 11 aristas incidentes a él. Supongamos sin pérdida de generalidad que estas aristas están conectadas a lo más a 11 vértices diferentes  $\{v_1, v_2, \dots, v_{11}\}$ . Asignamos el color no utilizado para  $\{v_1, v_2, \dots, v_{11}\}$  a  $v$ . De esta manera, se conservan las características de la 12-coloración en  $G$ , y  $G$  de orden  $n + 1$  es 12-coloreable.  $\square$

Hasta aquí, el ejemplo 14 demuestra que se requieren al menos 9 colores para resolver el Problema de la Tierra-Luna. Por otro lado, el teorema 0.28 establece que es posible utilizar hasta 12 colores. Esto plantea la pregunta: ¿es factible lograr una coloración con menos de 12 colores? Por lo que sabríamos que la respuesta al mínimo número de colores necesarios se encuentra en el conjunto:

$$\{9, 10, 11, 12\}$$

Dejando abiertas las posibilidades de exploración en torno a la existencia de configuraciones que requieran 10 u 11 colores. Esta incertidumbre resalta la complejidad del problema y nos surge la pregunta de cual es la complejidad computacional del problema de  $n$ -coloración de la Tierra-Luna con  $3 \leq n \leq 11$ .

Para concluir este capítulo, modelaremos el Problema de la Tierra-Luna mostrado en la figura 7 como un Problema de Satisfacción de Restricciones (CSP).

Con el objetivo de analizar su complejidad computacional, en la siguiente sección presentaremos la modelación de el Problema de la Tierra-Luna como un CSP, con el propósito de demostrar que es un problema de complejidad NP-hard.

Hasta el momento, hemos demostrado que el problema de coloración del modelo Tierra-Luna presenta desafíos significativos, con un mínimo de 9 colores necesarios y la posibilidad de utilizar hasta 12. Este análisis nos lleva a considerar la complejidad intrínseca de la coloración de grafos y abre la puerta a la exploración de la complejidad computacional para la  $n$ -coloración de el Problema de la Tierra-Luna con  $3 \leq n \leq 8$ . Debra Boutin, Ellen Gethner y Thom Sulanke en [Boutin et al., 2008] proporcionaron un catálogo de 40 nuevos ejemplos (configuraciones) de el Problema de la Tierra-Luna que pueden ser coloreados con 9 colores. Además, presentan una nueva metodología para construir configuraciones adicionales, lo que permite crear una familia infinita de ejemplos de problemas de la Tierra-Luna que son 9-coloreables.

Debido a que el Problema de la Tierra-Luna puede ser modelado como un problema de satisfacción de restricciones (CSP), sabemos que este problema tiene una complejidad en NP-hard. Sin embargo, para  $n \geq 12$ , el teorema 0.28 establece que el problema de decidir si un grafo que representa un problema de la Tierra-Luna es  $n$ -coloreable siempre tiene una solución afirmativa. Esto implica que dicho problema puede resolverse en tiempo polinomial. Como se demostró en el capítulo 3, el problema de la 3-coloración en la Tierra es NP-completo, ya que se redujo del problema 3-SAT a la 3-coloración de grafos. Por lo tanto, el Problema de la Tierra-Luna con 3 colores también será NP-completo, dado que en la tierra y la luna lo es.

Para concluir este capítulo, construiremos una reducción en tiempo polinomial del problema 3SAT al problema de 3-coloración de la Tierra-Luna. De esta manera, probaremos de forma independiente, y sin hacer uso del hecho de que el problema de 3-coloración en la Tierra es NP-completo, que el problema de 3-coloración de la Tierra-Luna también es NP-completo.

**Teorema 0.29.**  $3SAT \leq_p \text{Problema de la Tierra} - \text{Luna}$ .

Antes de demostrar este teorema, esbozaremos la estrategia de la prueba. Dada una entrada del problema 3SAT (una fórmula booleana), debemos construir un mapa en la tierra y un mapa en la luna que será 3-coloreable si y solamente si la fórmula booleana es satisfacible.

*Demostración.* La demostración se ejecutará en varios pasos. Sea una entrada del problema 3SAT con  $n$  variables y  $m$  cláusulas.

- **Paso 1:** Empezamos construyendo un mapa de la tierra que incluye tres países adyacentes, etiquetados como  $T$ ,  $F$ , y  $S$ . Aquí,  $T$  es un cuadrado,  $F$  es un  $(m + 2)$ -ágono, y  $S$  es  $(n + 2)$ -ágono. Asignamos a cada país un color distinto utilizando tres colores diferentes. Sin pérdida de generalidad, los colores se distribuyen como se muestra en la figura 15:

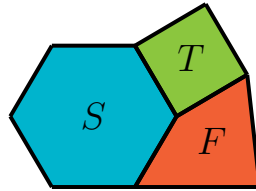


Figura 15: Primer paso.

En esta construcción, el color verde representa los valores verdaderos, el rojo representa los valores falsos y el color de  $S$  no indica ningún valor específico.

- **Paso 2:** En nuestra fórmula, se incluyen variables y sus negaciones, denominadas literales. Asignamos valores a cada literal de manera que si una literal  $x_i$  es verdadera su negación  $\overline{x_i}$  sea falsa. Para garantizar esta condición, procedemos de la siguiente forma:

Para cada literal  $x_i$  y su negación  $\overline{x_i}$ , agregamos dos países (cuadrados) en nuestro mapa de la tierra. Estos países se conectan entre sí y a uno de los lados disponibles del país  $S$  (disponibles  $n$  lados, uno para cada variable y su negación), como se ilustra en la figura 16.

Esto asegura que las literales  $x_i$  y  $\overline{x_i}$  no reciban el mismo color que el país  $S$ , ya que sus países están conectadas a  $S$  y, por lo tanto, deben ser coloreadas de rojo o verde. Para garantizar que  $x_i$  y  $\overline{x_i}$  no reciban el mismo color (es decir, que si  $x_i$  es verdadero, entonces  $\overline{x_i}$  sea falso, y viceversa), conectamos el país  $x_i$  con el país  $\overline{x_i}$ , como se muestra en la figura 16.

De esta manera, si coloreamos el mapa de una forma válida, obtendremos valores de verdad para las variables y sus negaciones que son consistentes.

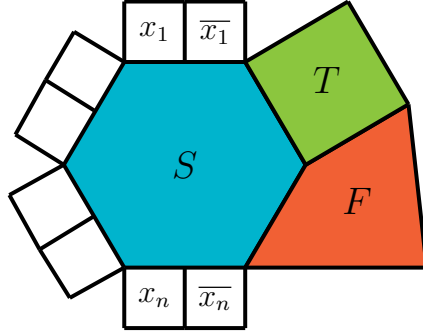


Figura 16: Segundo paso.

- **Paso 3:** Finalmente, debemos representar cada cláusula de la fórmula booleana en el mapa. Para esto, utilizaremos una construcción denominada “gadget” para simular el operador booleano OR (véase la figura 17). Este gadget es 3-coloreable si y solo si al menos uno de los países representado por las literales en la cláusula se asigna o colorea con  $T$  (verdadero), como se explicará a continuación.

El gadget que construiremos es análogo al descrito en el paso 3 de la demostración del Teorema 0.17, con la diferencia de que en esta ocasión se conecta al país que representa el valor *Falso*. Una parte del gadget se sitúa en la “tierra” y consiste en varios países conectados a uno de los lados del país correspondiente a  $F$ . Dado que el polígono que representa a  $F$  tiene  $m + 2$  lados, podemos utilizar un lado distinto para cada gadget asociado a cada cláusula de la fórmula booleana.

Cada uno de estos gadgets incluye también países en la “luna”, los cuales representan los literales de la cláusula correspondiente. Para ello, cada literal se modela mediante un polígono en la luna: específicamente, utilizamos un  $m$ -ágono para cada literal, o un triángulo en caso de que  $m < 3$ . Esta construcción permite representar adecuadamente la estructura lógica de cada cláusula dentro del grafo, respetando las restricciones de adyacencia y colorabilidad necesarias para la reducción.

Estos países de las literales en la luna no serán colindantes entre sí. En este modelo, para cada cláusula, adicional se agregarán algunos de los países del gadget en uno de los lados de los  $m$ -ágonos que representan las literales utilizadas en esa cláusula.

En el gadget mostrado en la figura 17 que se representa con los grafos de la figura 17 y la figura 18, si todos los países asociados a literales se colorean como Falsos (rojo), ninguno de los países  $n_1$ ,  $n_5$ ,  $n_6$  o  $n_2$  puede ser coloreado de rojo, ya que colindan con los países correspondientes a las literales y  $F$ . Además, dado que en la tierra el país  $n_5$  colinda con  $n_6$ , si  $n_5$  es verde,  $n_6$  debe ser azul, y viceversa. Esto implica que el país  $n_4$  debe ser coloreado de rojo, ya que colinda con  $n_5$  (verde/azul) y con  $n_6$  (azul/verde).

**Grafo en la tierra**

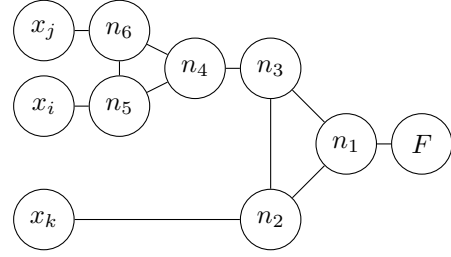
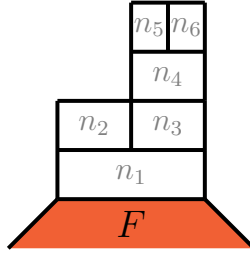


Figura 17: Tercer paso (tierra).

**Grafo en la luna**

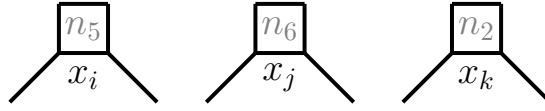


Figura 18: Tercer paso (luna).

Por lo tanto, el país  $n_3$  debe ser coloreado de verde o azul, dado que está conectado a  $n_4$  (rojo). Por lo tanto, los países  $n_2$ ,  $n_1$  y  $n_3$  solo pueden tener los colores verde o azul. Sin embargo, al tratarse de tres países y solo dos colores disponibles, necesariamente habrá un par de vértices que compartan el mismo color. Como estos tres países están conectados entre sí, se produciría una contradicción, ya que dos países conectados tendrían el mismo color. De este modo, si los países de las tres literales son falsas, es imposible colorear el gadget. Adicionalmente, es sencillo ver que si alguno es verde, se puede colorear el gadget.

**Paso 4:** El paso final consiste en intersectar los gadgets, de manera que no colinden con los países ya existentes, es decir, se crean los mapas por cada clausal y se busca empatar esos gadgets. Aunque el resultado es grande, está bien estructurado, como se muestra en la figura 20. De este modo, el mapa resultante es 3-coloreable si y solo si la fórmula booleana se satisface.

En el caso de que la fórmula booleana representada por la figura no se satisfaga, es decir, si  $x_1$ ,  $x_2$ , y  $\overline{x_3}$  son coloreadas de rojo, entonces al aplicar el proceso descrito en el paso 3, se llega a la conclusión de que es imposible colorear el mapa.

- **Paso 6:** Para demostrar que esta reducción se realiza en tiempo polinomial, supongamos que la fórmula tiene  $n$  literales y  $m$  cláusulas. Entonces, el número de países se calcula de la siguiente manera:

**Países en la tierra:**

- 3 países para  $T$ ,  $F$  y  $S$  (figura 15).
- 2 países por cada variable: se requieren  $2n$  países extra (figura 16).
- 6 países por cada cláusula: se requieren  $6m$  países extra (figura 18).

**Países en la luna:**

- 1 país por cada literal: se requieren a lo más  $2n$  países extra (figura 18).
- 3 países conectados a las literales por cada clausal: se requieren  $3m$  países adicionales (figura 18).

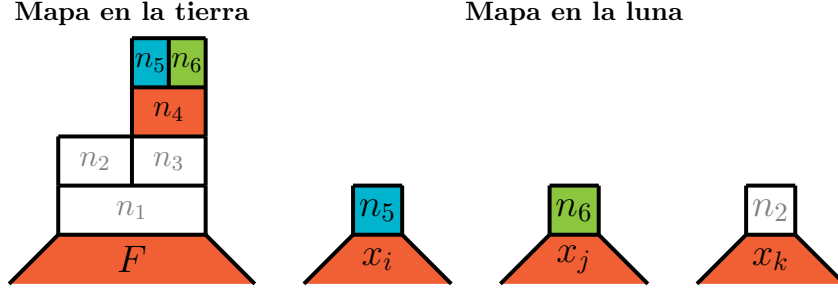


Figura 19: Tercer paso (coloración).

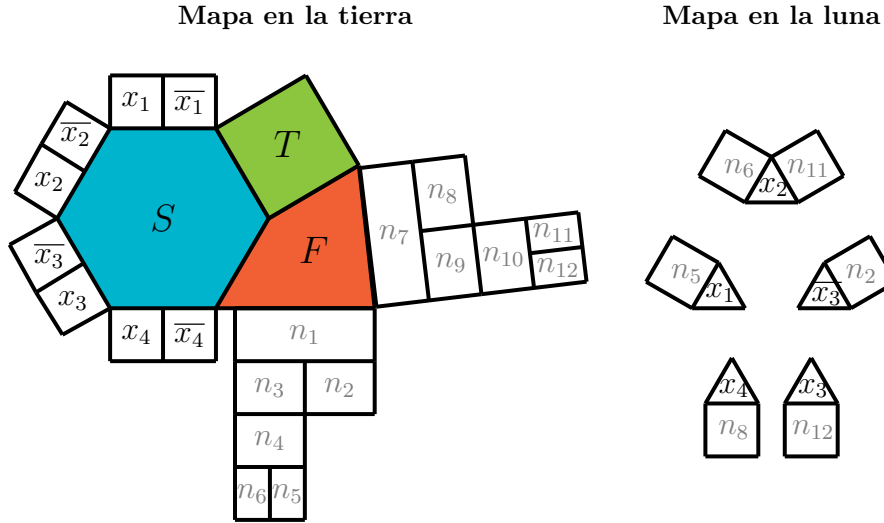


Figura 20: Reducción completa de la fórmula 3SAT  
 $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee x_3 \vee x_4)$  a una 3-coloración.

De este modo, los mapas generados tienen en total  $4n + 9m + 3$  países.

Hasta ahora, hemos demostrado que un problema 3SAT puede reducirse al Problema de la Tierra-Luna de forma general. Dado que 3SAT es un problema NP-completo, la reducción implica que el Problema de la Tierra-Luna también es NP-completo.  $\square$

En este trabajo, se ha presentado la reducción del problema 3SAT al problema de 3-coloración de la Tierra-Luna, demostrando que decidir si un problema de Tierra-Luna puede ser coloreado con 3 colores distintos es NP-completo. En [Behrisch et al., 2025], se continúa el desarrollo de nuevos gadgets y la realización de reducciones del 3SAT a la coloración de el Problema de la Tierra-Luna para los casos de  $n = 4$ ,  $n = 5$  y  $n = 6$ . Hasta la fecha, se ha probado que la coloración de un mapa de la Tierra-Luna con 4, 5 y 6 colores también es un problema NP-completo. Se planea publicar dichos resultados en [Behrisch et al., 2025].

# Bibliografía

- [Arora and Barak, 2009] Arora, S. and Barak, B. (2009). *Computational complexity: a modern approach*. Cambridge University Press.
- [Behrisch et al., 2025] Behrisch, M., Calderón, A., and Vargas, E. (2025). On the complexity of the k-coloring for the earth-moon map problem. *En proceso*.
- [Boutin et al., 2008] Boutin, D. L., Gethner, E., and Sulanke, T. (2008). Thickness-two graphs part one: New nine-critical graphs, permuted layer graphs, and catlin’s graphs. *Journal of Graph Theory*, 57(3):198–214.
- [Demain et al., 2020] Demain, E., Ku, J., and Solomon, J. (2020). Lecture 19: Complexity. *MIT OpenCourse*.
- [Gonthier et al., 2008] Gonthier, G. et al. (2008). Formal proof—the four-color theorem. *Notices of the AMS*, 55(11):1382–1393.
- [Gorrie, 2014] Gorrie, D. (2014). Scribe notes: Planar graphs and space complexity. <https://people.eecs.berkeley.edu/~gorrie/cs170/notes/planar.pdf>. Lecture notes.
- [Hutchinson, 1993] Hutchinson, J. P. (1993). Coloring ordinary maps, maps of empires, and maps of the moon. *Mathematics Magazine*, 66(4):211–226.
- [Kempe, 1879] Kempe, A. B. (1879). On the geographical problem of the four colours. *American journal of mathematics*, 2(3):193–200.
- [Lau, 2006] Lau, D. (2006). *Function algebras on finite sets: Basic course on many-valued logic and clone theory*. Springer Science & Business Media.
- [McEliece, 1978] McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116.
- [Ringel, 2012] Ringel, G. (2012). *Map color theorem*, volume 209. Springer Science & Business Media.
- [Tsiatas, 2008] Tsiatas, A. (2008). Phase transitions in boolean satisfiability and graph coloring. *Department of Computer Science, Cornell University*.
- [Zhuk, 2020] Zhuk, D. (2020). A proof of the csp dichotomy conjecture. *Journal of the ACM (JACM)*, 67(5):1–78.