



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**ILUMINACIÓN DE POLÍGONOS ORTOGONALES CON REFLECTORES DE
180° GIRATORIOS COLOCADOS EN ARISTAS**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)

PRESENTA:
ERICK SOLÍS VILLARREAL

DIRECTOR DE TESIS
DR. JORGE URRUTIA GALICIA
INSTITUTO DE MATEMÁTICAS UNAM

CODIRECTOR DE TESIS
DR. CARLOS BRUNO VELARDE VELÁZQUEZ
IIMAS-UNAM

MÉXICO, D. F. JUNIO 2014



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

La presente investigación es un trabajo en el que participaron varias personas, directa o indirectamente. Agradezco al Dr. Jorge Urrutia por haber confiado en mi persona, por la paciencia y por la dirección de este trabajo. A mi cotutor el Dr. Carlos Velarde por la lectura, corrección y la inigualable ayuda que me ofreció durante todo este periodo. Al Dr. David Flores por sus comentarios, aportaciones y sus atinadas correcciones.

Gracias también a mis compañeros que me apoyaron y me permitieron alterar sus vidas; Carmen, March, Israel, Joel, Josue, Victor, Moradito, Varytec, entre muchos otros.

A mi mamá y mi papá, María del Consuelo y Enrique Solís, por el invaluable apoyo que me han brindado a lo largo de mi vida. A mis dos hermanos, Quique y Pepe, por todo el apoyo y sus palabras de aliento.

A mis otros hermanos, Omar y Ajelandro, por acompañarme en mi travesía. Al Sartuño por permitirme tener aquellas charlas desestresantes. A mi compañera Ivonne por tenerme la paciencia suficiente y sus constantes comentarios "¿A qué hora vas a acabar la tesis?" Gracias a todos

Índice general

Agradecimientos	3
Índice de figuras	9
1. Introducción	3
2. Teoría de Gráficas	5
2.1. Introducción	5
2.2. Definiciones básicas	5
3. El Problema de la Galería de Arte	9
3.1. Introducción	9
3.2. Definición del problema	9
3.3. Vigilancia de polígonos simples	9
3.4. Iluminación de polígonos con reflectores	13
3.4.1. Introducción	13
3.4.2. Iluminación con reflectores	14
3.4.3. Iluminación de estrados	15
3.5. Iluminación de polígonos ortogonales	16
3.5.1. Iluminación de polígonos ortogonales con reflectores	21
3.5.2. Partición de polígonos ortogonales en piezas con forma de L	24
3.5.3. Algoritmo para partir polígonos ortogonales en piezas con forma de L	29
3.6. La dificultad del problema	36

3.7. Iluminación con radares	39
4. Teoría de la complejidad computacional	45
4.1. Introducción	45
4.2. Problemas de decisión	45
4.3. SAT	46
4.4. Lenguajes y esquemas de codificación	47
4.5. Algoritmos de tiempo polinomial	48
4.6. Introducción a las máquinas de Turing	48
4.6.1. Máquinas de Turing	49
4.6.2. Máquinas de Turing deterministas y problemas \mathcal{P}	49
4.6.3. Máquina de Turing no determinística y problemas \mathcal{NP}	51
4.6.4. Reducción polinomial	54
4.6.5. Problemas \mathcal{NP} -Completos. Teorema de Cook	57
4.6.6. Problemas \mathcal{NP} -Duros.	59
5. Reducción	61
5.1. Introducción	61
5.2. Complejidad computacional del problema	61
5.2.1. El patrón de una literal	62
5.2.2. Pseudo literal y Pseudo literal auxiliar	66
5.2.3. Patrón de una cláusula	66
5.2.4. Patrón de una variable	69
5.2.5. Construcción completa	70
6. Vigilancia de Polígonos Ortogonales con Radares	75
6.1. Introducción	75
6.2. Vigilancia de polígonos ortogonales con radares	75
6.2.1. Polígono ortogonal en forma de L	75
6.2.2. Polígono ortogonal	76

ÍNDICE GENERAL

7

Bibliografía

79

Índice de figuras

2.1. Gráfica con 8 vértices y 10 aristas.	6
2.2. Una gráfica plana G y su gráfica plana dual G^*	7
3.1. El polígono queda vigilado	11
3.2. Polígono denominado Peine.	11
3.3. Todo polígono simple tiene algún vértice convexo	12
3.4. Una triangulación T de un polígono simple	13
3.5. El caso en que x y y están en aristas diferentes de P	15
3.6. El problema de iluminación de estrados.	16
3.7. Imposibilidad de agrupar triángulos para obtener cuadriláteros convexos.	17
3.8. La nariz en e de un polígono 1-ortogonal.	18
3.9. Buscando los vértices de Q' , si e'' alcanzó una arista completa.	19
3.10. Buscando los vértices de Q' , desplazando h , caso 1.	19
3.11. Buscando los vértices de Q' , caso 2.	20
3.12. Polígono ortogonal cuadrilaterizado 4-coloreado.	20
3.13. Polígono ortogonal denominado peine.	21
3.14. Clasificación para vértices cóncavos y convexos de un polígono P	22
3.15. Reflectores colocados con la regla SI.	23
3.16. Familia de polígono con $12 + 8k$ vértices que requieren $4 + 3k$ reflectores.	23
3.17. Un corte de P y sus vértices cóncavos.	25
3.18. Polígono sin corte impar y polígono sin cortes horizontales	26
3.19. La <i>partición-H</i> de P y su <i>gráfica-H</i>	27

3.20. Una región hundida S siempre permite un corte impar.	27
3.21. Un polígono ortogonal en espiral.	29
3.22. La adición de un vértice cóncavo.	31
3.23. Recorriendo el borde para calcular paridad	32
3.24. Los tres tipos de vértices enumerados	32
3.26. Cómo afecta los cortes impares la paridad	34
3.27. Polígono ortogonal denominado histograma.	34
3.28. Si un polígono no tiene un corte impar, en dos histogramas	35
3.29. Se siempre es visible	36
3.30. Orientación inicial de los radares en L	40
3.31. El dual del plano	40
3.32. (a) $0 \leq t \leq \pi$. (b) $\pi < t < 2\pi$	42
3.33. Configuración inicial para 3 radares en posición colineal.	42
3.34. Siempre se mantiene cubierto el plano	43
4.1. Máquina de Turing determinista.	49
4.2. Máquina de Turing no determinista.	53
4.3. Probable descripción de \mathcal{NP}	58
4.4. Probable descripción de \mathcal{NP} -Duro, si $\mathcal{P} \neq \mathcal{NP}$	60
5.1. El patrón de una literal.	63
5.2. Prueba de que 2 radares son insuficientes	64
5.3. Un patrón de literal iluminado, si $y_1 = F$	65
5.4. Pseudo literal.	66
5.5. Pseudo literal auxiliar.	66
5.6. Colocando los cinco vértices y ocultandolos de Z	67
5.7. Triángulos de sombra	68
5.8. Sincronizando cuatro radares en configuración fuerte	69
5.9. El patrón de una variable.	70
5.10. Iluminación parcial del patrón de una variable, si $x_1 = V$	70

<i>ÍNDICE DE FIGURAS</i>	11
5.11. El polígono completo.	71
5.12. Alineación de las espinas dependiendo de l	72
6.1. Polígono ortogonal con forma de L, con aristas a, b, c, d, e, f	76
6.2. Segmento capaz de visualizar todo el polígono en forma de L.	77

Capítulo 1

Introducción

Imagine la siguiente situación: Se encuentra recorriendo un museo, en donde se exponen pinturas, esculturas y gran variedad de obras de arte. Usted logra visualizar que el lugar se encuentra lleno de guardias, pero se da cuenta que existe un área en particular que no es vigilada por ningún guardia, y que podría fácilmente hurtar una pieza. Una pregunta lógica podría ser, habiendo tantos guardias ¿cuántos guardias son necesarios y cómo ubicarlos eficientemente, de tal forma que vigilen de manera permanente todas las obras de arte?

Esa pregunta, en realidad, es uno de los problemas más antiguos que estudia la geometría computacional. Este problema es conocido como “*El Problema de la Galería de Arte*”.

Victor Klee planteó, en los años 70 del siglo pasado, el problema de determinar el número de guardias suficientes para cubrir el interior de una galería de arte de n paredes.

Vasek Chvátal demostró en un Teorema conocido como “*El Teorema de la Galería de Arte de Chvátal*” (o en ocasiones “*El Teorema de los Guardias*”), que $\lfloor \frac{n}{3} \rfloor$ guardias son ocasionalmente necesarios y siempre suficientes para cubrir un polígono simple de n vértices (Chvátal 1975).

En los futuros años se realizaron muchas investigaciones sobre galerías de arte, por lo que para este trabajo se realizó una selección muy breve de estas investigaciones, de tal forma que se muestre algunas de las variantes del problema de galerías de arte, así como las herramientas básicas para sus pruebas.

Esta tesis se organiza de la siguiente forma:

En el capítulo 2 se presentan conceptos y propiedades básicas de teoría de gráficas para el entendimiento de los problemas de la galería de arte.

En el capítulo 3 se da una introducción a algunos problemas sobre galerías de arte, que sirven de base para el desarrollo del presente trabajo.

En el capítulo 4 se exponen una serie de conceptos básicos sobre teoría de la complejidad, que son un preámbulo teórico necesarios para el presente trabajo.

En el capítulo 5 se presenta uno de los resultados de esta investigación, en el que se concluye que encontrar el mínimo número de radares en vértices para vigilar un polígono simple sin hoyos, es \mathcal{NP} -duro.

En el capítulo 6 se presenta un algoritmo para ubicar la posición de los radares.

Capítulo 2

Teoría de Gráficas

2.1. Introducción

La teoría de gráficas es una herramienta muy importante para las ciencias computacionales, ya que puede modelar varios fenómenos del mundo real y es fundamental para la comprensión de las estructuras de datos, por mencionar algunas de sus cualidades. En nuestro caso, para el presente trabajo es una herramienta fundamental pero para poder hacer uso de ella primero debemos definir algunos conceptos.

2.2. Definiciones básicas

Una gráfica G es una tupla (V, E) que consiste de un conjunto $V(G)$ de vértices o nodos y un conjunto $E(G)$, ajeno de $V(G)$, de aristas junto con una función f_G que asocia a cada arista con un par de vértices de G . Por ejemplo, si se tiene una arista e y un par de vértices u, v , tales que $f_G(e) = \{u, v\}$, entonces entendemos que e es la arista que une a los vértices u y v , y que u y v son los extremos de e . Por simplicidad, usaremos uv para referirnos a una arista que une a los vértices $\{u, v\}$.

También podemos ver a una gráfica de manera más visual, es decir, podemos dibujar a los vértices como puntos y a las aristas como a una líneas y cada una conecta a dos puntos, véase la figura 2.1. Por medio de esta representación es mucho más fácil comprender algunas de sus propiedades.

A continuación presentaremos algunas definiciones importantes:

Definición 1 *Cuando los extremos de una arista son el mismo vértice, entonces la arista recibe el nombre de lazo.*

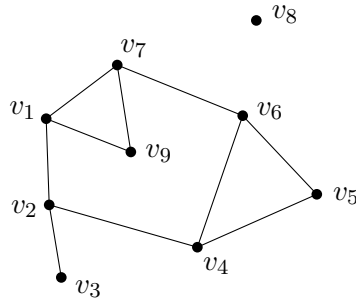


Figura 2.1: Gráfica con 8 vértices y 10 aristas.

Definición 2 Si la arista no es un lazo entonces le llamaremos enlace.

Definición 3 Si dos enlaces diferentes tienen los mismos extremos, entonces les llamaremos aristas paralelas.

Definición 4 Una gráfica es simple si no contiene lazos ni aristas paralelas.

Definición 5 El grado de un vértice es el número de aristas que inciden en un vértice v de una gráfica G .

Definición 6 Sea G una gráfica. Una gráfica H es una subgráfica de G , si H es una gráfica tal que $V(H) \subseteq V(G)$ y $E(H) \subseteq E(G)$.

Definición 7 Un camino en una gráfica G , es una sucesión finita de vértices v_1, v_2, \dots, v_n y aristas e_1, e_2, \dots, e_m de G , tal que sus vértices pueden ser ordenados en una secuencia lineal de tal forma que un par de vértices son adyacentes son consecutivos en la secuencia, es decir, $v_1, e_1, v_2, e_2, \dots, e_m, v_n$.

Definición 8 La longitud de un camino es definida como el número de aristas que lo conforman.

Definición 9 Un uv -camino en una gráfica G , es un camino que comienza en el vértice u y termina en el vértice v .

Definición 10 Un uv -paseo en una gráfica, es un uv -camino que no repite aristas.

Definición 11 Una uv -trayectoria, es un uv -camino tal que no repite vértices.

Definición 12 *Un ciclo es un uv -camino que no repite aristas ni vértices y los vértices u y v son el mismo.*

Definición 13 *Dos vértices u y v en una gráfica G , están conectados si $u = v$ o si $u \neq v$ y existe una uv -trayectoria en G .*

Definición 14 *Una gráfica G es conexa, si para todo par de vértices de G están conectados. En caso contrario se dice que G es desconexa o no conexa.*

Definición 15 *Una gráfica es plana si puede dibujarse en el plano de tal forma que las líneas que representan sus aristas no se intersecan entre ellas (excepto en sus extremos), es decir, dos aristas tienen un punto común si y sólo si son adyacentes.*

Definición 16 *Una gráfica plana divide al plano en diferentes regiones, a la cerradura de cada una de estas regiones se le llama cara de la gráfica, teniendo siempre una cara no acotada llamada cara exterior. El resto de caras son llamadas caras interiores.*

Definición 17 *Sea G una gráfica plana, se puede definir a otra gráfica G^* a partir de G de la siguiente forma: por cada cara f de G se tiene un vértice f^* en G^* , y por cada arista e de G , se consideran las caras f y g , incidentes en e , entonces G^* tendrá la arista e^* que conecta a los vértices f^* y g^* . Decimos que G^* es la gráfica dual de G . Podemos ver una gráfica dual en la figura 2.2*

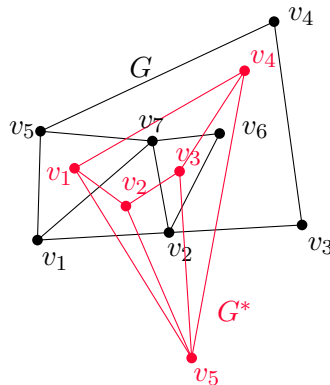


Figura 2.2: Una gráfica plana G y su gráfica plana dual G^*

Definición 18 *Una gráfica geométrica, es una gráfica dibujada en el plano, de tal forma que sus vértices son puntos, y una arista uv es el segmento uv .*

Capítulo 3

El Problema de la Galería de Arte

3.1. Introducción

Tal como la situación planteada en el capítulo de introducción, el problema de vigilar galerías de arte se ha tratado en el área de la geometría computacional. La interpretación o representación geométrica que se le ha dado es representar la galería mediante un polígono simple P y a los guardias como puntos al interior de P . A través de esta representación se han presentado varios resultados, entonces para poder analizar estos resultados necesitaremos primero definir de manera más concreta el problema.

3.2. Definición del problema

Un polígono simple P es definido como una colección de n vértices v_1, v_2, \dots, v_n y n aristas $v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1$, tal que las aristas no adyacentes no se intersecan. Un guardia es definido como un punto en el polígono. Entonces el problema consiste en encontrar el mínimo conjunto de guardias G , tal que cualquier punto $p \in P$ sea visto por al menos un guardia $g \in G$. Decimos que g ve, cubre o vigila a p si el segmento \overline{gp} se encuentra contenido en P .

3.3. Vigilancia de polígonos simples

La demostración que a continuación se enunciará fue dada por S. Fisk [2] en 1975. Esta es muy elegante y menos compleja que la que dio Chvátal originalmente.

Teorema 1 $\lfloor \frac{n}{3} \rfloor$ guardias son siempre suficientes y a veces necesarios para vigilar un polígono simple P de n vértices.

Observación 1 Un guardia situado en cualquier vértice de un triángulo vigila completamente dicho triángulo.

Prueba. Consideremos un polígono simple P de n vértices.

Primer paso (Triangulación)

Triangulemos¹ P , es decir, descomponemos P en triángulos cuya unión es P , con interiores disjuntos y cuyos vértices son vértices de P .

Segundo paso (Coloración)

Sea T nuestra triangulación. Colorearemos los vértices de los triángulos de T utilizando tres colores. En esta 3-coloración de T cada triángulo tiene un vértice de cada color.

Tercer paso (Colocación de guardias)

Elegimos un color de vértice, rojo para ejemplificar. Cada triángulo de T tiene un vértice rojo. Si colocamos un guardia en cada vértice rojo, todos los triángulos quedarán vigilados y, por tanto todo el polígono.

El polígono tiene n vértices y disponemos de 3 colores. Por tanto, alguno de los tres colores se utiliza en a lo más $\lfloor \frac{n}{3} \rfloor$ vértices, (esto es una aplicación inmediata del principio del palomar: Si cada color se utilizara en más de $\lfloor \frac{n}{3} \rfloor$ vértices, sumando los vértices de cada color tendríamos más de n vértices). Basta con colocar los guardias en los vértices con el color menos utilizado para garantizar que $\lfloor \frac{n}{3} \rfloor$ guardias son siempre suficientes para vigilar cualquier polígono de n vértices. Esto se puede observar de manera gráfica en la figura 3.1.

Cuarto paso (Necesidad de los $\lfloor \frac{n}{3} \rfloor$ guardias)

Para comprobar que este número de guardias es a veces necesario, basta considerar el polígono llamado “peine” mostrado en la figura 3.2 con $n = 3k$ vértices, donde k es el número de dientes. Es fácil observar que para vigilar este polígono se necesitan al menos k guardias, uno por cada diente del peine.

Para completar la demostración se debe responder a las preguntas ¿todo polígono simple admite una triangulación? y ¿toda triangulación es 3-coloreable?

Lema 1 Todo polígono se puede triangular.

Prueba. Por inducción sobre el número de vértices del polígono. Sea P un polígono simple con n vértices.

¹Se le llama triangulación, a la gráfica geométrica formada por los vértices y las aristas de P junto con las diagonales o aristas de los triángulos.

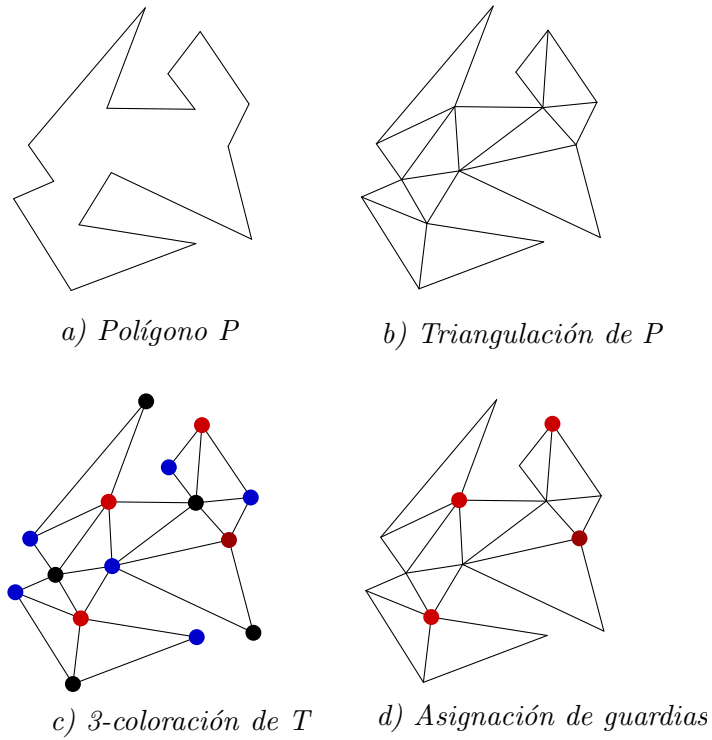


Figura 3.1: El polígono completo queda vigilado por los guardias colocados en los vértices de color rojo, que no exceden la tercera parte del total de vértices de P .

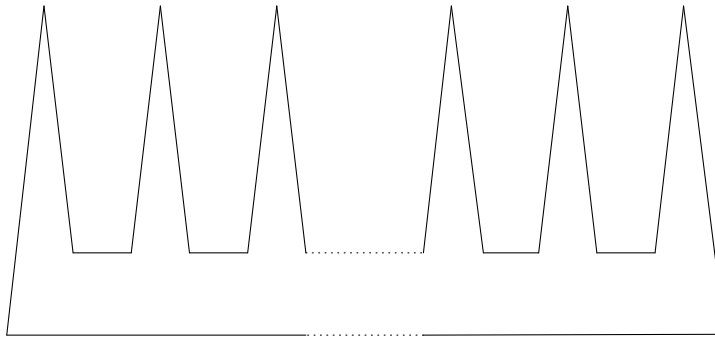


Figura 3.2: Polígono denominado Peine.

- (1) Si $n = 3$, el polígono es un triángulo.
- (2) H.I. Asumimos que nuestro lema es cierto para toda $n > 3$.
- (3) Ahora probaremos cuando nuestro polígono tiene $n + 1$ vértices, siempre podemos

trazar una diagonal que descompone al polígono P en P_1 y P_2 , tales que el número de vértices de ambos es menor a $n+1$. Por hipótesis de inducción P_1 y P_2 se pueden triangular. \square

Lema 2 *Todo polígono de n vértices, $n \geq 4$, admite una diagonal interna.*

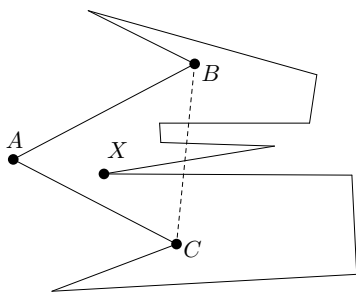


Figura 3.3

Prueba. En primer lugar observemos que todo polígono simple tiene algún vértice convexo (observe la figura 3.3). Sea P un polígono, A un vértice convexo de P y sean B, C los vértices adyacentes a A . Si el segmento \overline{BC} está contenido en el polígono será la diagonal buscada. Si no es así, en el triángulo formado por los vértices ABC habrá vértices de P . Tomamos el vértice dentro del triángulo ABC más alejado de la recta BC , sea X ese vértice. De esta forma, AX estará contenido en P y es la diagonal buscada. \square

Lema 3 *Cualquier triangulación de un polígono simple es un grafo plano 3-coloreable.*

Prueba. Lo demostraremos por inducción sobre el número de triángulos de T . Si T es un triángulo es evidente que es 3-coloreable. Si T tiene $n + 1$ triángulos, debe tener al menos un vértice v de grado 2. La gráfica $T - v$ es una triangulación y tiene n' triángulos, entonces, por hipótesis de inducción, es 3-coloreable. Pintando a v del color distinto a los colores de sus 2 vecinos en T se obtiene una 3-coloración de T . \square

Únicamente nos falta probar que:

Lema 4 *Toda triangulación tiene al menos un vértice de grado 2.*

Prueba. Sea T' la gráfica dual de T (figura 3.4), aquella que tiene como vértices a las caras de los triángulos de T y por aristas las diagonales que comparten dos triángulos. Como T es una gráfica plana (por construcción), entonces T tiene $n - 2$ triángulos, por la característica de Euler, entonces T' tiene $n - 2$ vértices. Como una diagonal en T sólo se comparte con 2 triángulos se pueden hacer dos observaciones: tiene $(n - 2) - 1$ diagonales en T y $n - 3$ aristas en T' . Por último, podemos observar que ningún triángulo se encuentra aislado, por lo tanto, existe un camino para todo v en T' . Con todo lo anterior podemos concluir que, como T' tiene $n - 2$ vértices, $n - 3$ aristas y T' es conexo, entonces T' es un árbol. Ahora es fácil notar que si se quita una diagonal en T o una arista en T' , T' queda no conexa esto es porque es un árbol y como T' siempre tiene al menos un vértice de grado 1, entonces T siempre tiene un vértice de grado 2. \blacksquare

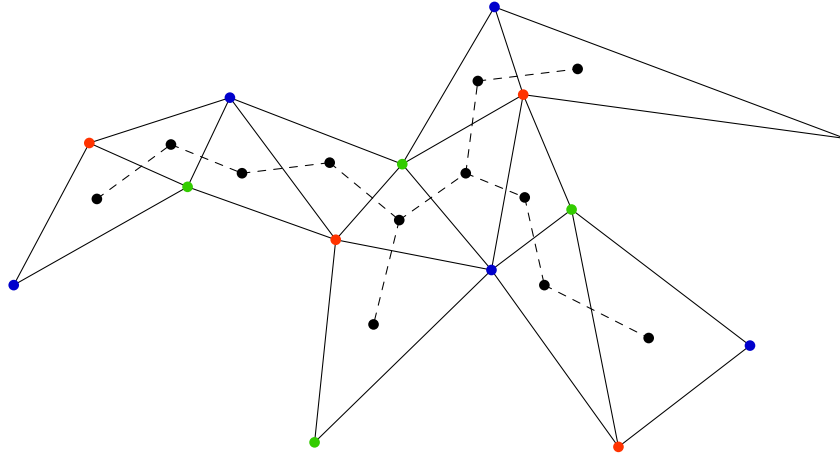


Figura 3.4: Una triangulación T de un polígono simple, junto a una representación planar de su gráfica dual T' y su 3-coloración.

Con esta última demostración queda claro que, $\lfloor \frac{n}{3} \rfloor$ guardias son siempre suficientes y a veces necesarios para vigilar un polígono simple P de n vértices.

Desde que Klee formuló aquella pregunta, originó un importante número de investigaciones relacionadas. Afortunadamente para muchos, en 1987 O'Rourke publicó “*Art Gallery: Theorems and Algorithms*” [10] recopilando todos los resultados obtenidos hasta ese momento en el campo de la iluminación de polígonos². Los avances y nuevas líneas abiertas desde entonces pueden consultarse en los trabajos recopilatorios de T. C. Shermer [12] y J. Urrutia [13], publicados en 1992 y en el 2000, respectivamente.

En las siguientes secciones se presentan algunas investigaciones relacionadas con el presente trabajo.

3.4. Iluminación de polígonos con reflectores

3.4.1. Introducción

Ahora imaginemos que queremos iluminar un piso completo de alguna casa u oficina, ¿cuántos focos serían suficientes para iluminarlo? Geométricamente podemos representar la habitación como un polígono P y a los focos como puntos en el interior de P . Decimos que un punto p de P queda iluminado si y sólo si existe un foco f , tal que el segmento \overline{pf} se encuentra totalmente contenido en P . Podemos observar que el problema es equivalente

²El concepto de iluminación será introducido en la siguiente sección.

al problema de la galería de arte y he aquí el por qué a los problemas de vigilancia de galerías de arte se les refiera también como problemas de iluminación. De aquí en adelante se utilizarán los términos vigilar o iluminar de manera indistinta.

3.4.2. Iluminación con reflectores

Hasta ahora hemos visto que los guardias o focos tienen un ángulo de iluminación de 360° pero, ¿qué pasaría si se restringe el ángulo de visión? Bueno, pues esta pregunta fue planteada por J. Urrutia [13, Capítulo 5]. Llamaremos reflector a esta fuente de luz. Para el resto de esta tesis el reflector tendrá las siguientes características: Un reflector r_i será una fuente de luz localizada en un punto p de un plano llamado este ápice; r_i ilumina únicamente dentro de un ángulo de iluminación α_i , y puede ser rotado al rededor de p .

Teorema 2 *Dado un polígono convexo P y tres reflectores r_1, r_2, r_3 con ángulo de iluminación $\alpha_1, \alpha_2, \alpha_3$ respectivamente, tales que $\alpha_1 + \alpha_2 + \alpha_3 = \pi$, siempre es posible iluminar P posicionando los reflectores en exactamente tres vértices diferentes.*

Prueba. Claramente el resultado es cierto para P si este tiene 3 vértices. Ahora consideraremos cualquier polígono convexo P con al menos 4 vértices y supongamos que $\alpha_1 \leq \alpha_2 \leq \alpha_3$, con, $\alpha_2 \leq \frac{\pi}{2}$. Ya que P tiene al menos 4 vértices, uno de sus ángulos internos es mayor o igual que $\frac{\pi}{2}$. Sea T un triángulo con ángulos internos α_1, α_2 y α_3 tal que:

1. El vértice de T de tamaño α_2 se encuentre colocado sobre un vértice $v \in P$ de ángulo interior mayor o igual a α_2 .
2. Los otros vértices de T se encuentren colocados sobre 2 puntos x, y en la frontera de P (excluyendo los vértices).

Se pueden observar 2 casos:

1. Supongamos que tanto x como y pertenecen a la misma arista de P , sea e esta arista. Es fácil ver que los ángulos formados por u, v, x y v, y, w , donde u y w son los vértices adyacentes a e , son de tamaño menor o igual que α_1 y α_3 respectivamente e iluminan completamente P .
2. Supongamos que tanto x como y pertenecen a diferentes aristas de P , digamos e_x y e_y ; ver figura 3.5.

Coloquemos un reflector r_2 con un ángulo de α_2 sobre v de manera que ilumine a T . Sea C el círculo que circunscribe a T . Al menos un vértice de e_x y uno de e_y no están contenidos en el interior de C . Sean u y w estos vértices. Se distinguen dos casos:

- a) $u \neq w$. Coloquemos un reflector r_1 sobre u iluminando la zona angular determinada por v, u, x y otro r_3 sobre w iluminando la zona angular determinada por v, w, y . Como u y w no están en el interior de C , los tamaños de las zonas angulares de r_1 y r_3 son a lo más α_1 y α_3 respectivamente. P queda completamente iluminado.
- b) $u = w$. Sea T' el triángulo determinado por el segmento \overline{xy} , y las tangentes a C en x y y . El ángulo generado en el vértice z de T' que no está sobre C es $\pi - 2\alpha_2$. Notemos que z pertenece al interior del triángulo determinado por x, y y u , por lo que el ángulo de dicho triángulo en u es menor que $\pi - 2\alpha_2$. Como $\alpha_1 \leq \alpha_2 \leq \alpha_3$, $\pi - 2\alpha_2 \leq (\pi - (\alpha_1 + \alpha_2)) = \alpha_3$. Por tanto colocando un reflector de tamaño a lo más α_3 en u , P queda iluminado. ■

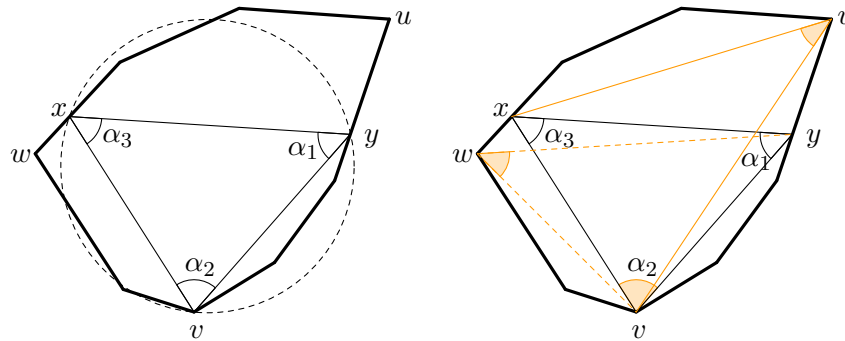


Figura 3.5: El caso en que x y y están en aristas diferentes de P .

3.4.3. Iluminación de estrados

Pensemos en el siguiente escenario: Vamos a ir a una obra de teatro y una vez llegado a él, observamos que en el estrado se encuentran numerosos reflectores, estos reflectores pueden iluminar el estrado. Sin embargo no lo pueden iluminar completamente, usted se puede preguntar ¿cuántos reflectores serían necesarios para iluminarlo completamente? Este interesante problema fue planteado por J. Urrutia [13, Capítulo 5] en 1992. Formalmente, representamos al estrado como un segmento de recta L . Tenemos una colección finita de n reflectores r_1, \dots, r_n que iluminan en una zona angular de tamaño α_i , $i = 1, \dots, n$ respectivamente. Dichos reflectores se encuentran fijos y pueden rotar sobre sus ejes. La pregunta a responder es la siguiente: ¿Será posible rotar r_1, \dots, r_n de tal manera que L quede completamente iluminado? Véase la figura 3.6.

El problema de iluminación de estrados se vuelve interesante dado que la intuición indica que debería tener una solución simple, sin embargo, H. Ito, H. Uehara y M. Yoko-

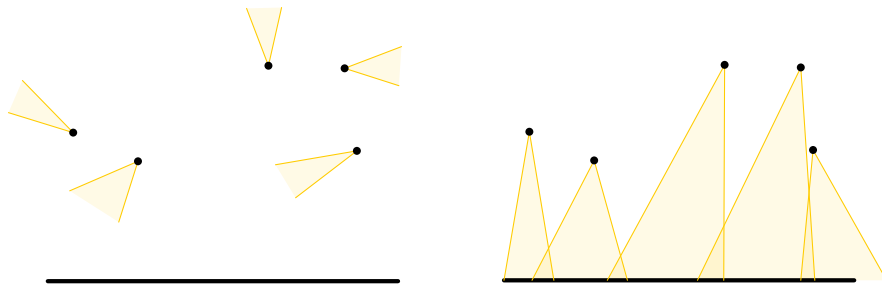


Figura 3.6: El problema de iluminación de estrados.

yama, prueban que el problema de iluminación de estrados es un problema NP-completo [6].

J. Czyzowicz, E. Rivera-Campo y J. Urrutia trabajaron en otra variante del problema de iluminación de estrados [8], donde se busca minimizar el ángulo de iluminación de los reflectores. Los autores lo plantean de la siguiente manera: Sea l un segmento de recta sobre el eje x y P_n un conjunto de n puntos con coordenada y positiva. Lo que se pretende buscar es encontrar un conjunto de reflectores $R = r_1, \dots, r_m$, tales que sus ápices se encuentren en alguno de los puntos de P_n , de tal manera que R ilumine l y que además $\alpha_1 + \dots + \alpha_m$ se minimice.

A este problema se le conoce como “*iluminación óptima de estrados*”, y el algoritmo presentado lo resuelve en $O(n \log)$.

3.5. Iluminación de polígonos ortogonales

Uno se podría preguntar ¿quién monta una exhibición de arte en un museo que tenga una forma de polígono simple tan arbitraria? Bueno pues, supongo que esta pregunta originó el estudio del siguiente problema, ya que la mayoría de construcciones que conocemos mantienen una estructura ortogonal, uno podría pensar que este tipo de polígonos (ortogonales) tienen especial interés, ya que se ven como un mejor modelo para aplicaciones. Además, desde el punto de vista matemático, su inherente estructura nos permite obtener resultados elegantes e interesantes.

Un polígono P es ortogonal, si cualquier arista de P es paralela a alguno de dos ejes ordenados, y además cada ángulo interno o externo entre dos aristas es $\frac{\pi}{2}$ ó $\frac{3\pi}{2}$.

Un polígono ortogonal puede ser vigilado con $\frac{n}{4}$ guardias, este resultado fue dado por J. Kahn, M. Klawe y D. Keitman en [9].

Para probar el resultado de J. Kahn, M. Klawe y D. Kleitman, primero debemos probar que todo polígono ortogonal tiene una cuadrilaterización convexa. Intuitivamente

esta prueba debería ser sencilla, dado que ya se sabe que cualquier polígono se puede triangular, entonces basta con agrupar triángulos, pero el agrupar triángulos no sirve en este caso como se puede observar en la figura 3.7:

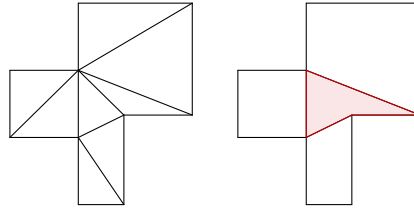


Figura 3.7: Imposibilidad de agrupar triángulos para obtener cuadriláteros convexos.

Definamos que significa cuadrilaterizar. Una cuadrilaterización convexa de un polígono ortogonal P , es una partición de P en un conjunto de cuadriláteros convexos con interiores mutuamente ajenos, de tal forma que las aristas de los cuadriláteros formados son aristas de P , o bien diagonales de P .

Teorema 3 *Cualquier polígono 1-ortogonal es convexamente cuadrilaterizable.*

Ahora debemos definir que es un polígono 1-ortogonal. Un polígono P es 1-ortogonal si cumple las siguientes propiedades:

1. Todas las aristas de P , con la posible excepción de una arista distinguida e , que llamaremos arista inclinada, son paralelas al eje x o y .
2. Con la posible excepción de e , las aristas se alternan entre horizontales y verticales.
3. Todos los ángulos internos son menores o iguales a $\frac{3\pi}{2}$.
4. El interior de la nariz de la arista e no contiene vértices de P .
5. P tiene un número par de aristas.

Notemos que un polígono ortogonal es un polígono 1-ortogonal. En la definición no se prohíbe que e pueda ser paralela al eje x o y . Si P es ortogonal, entonces e puede ser cualquier arista de P .

La nariz en e forma un triángulo T , que se encuentra al interior de P , con un lado horizontal, otro vertical, que no son aristas de P , y e como hipotenusa; ver figura 3.8.

Si e es una arista horizontal o vertical, entonces la nariz es vacía.

Prueba. Si P tiene 4 aristas entonces P es ortogonal y el teorema es cierto. Ahora demostraremos si P tiene más de 4 aristas, siempre existe un cuadrilátero convexo Q' , el cual si

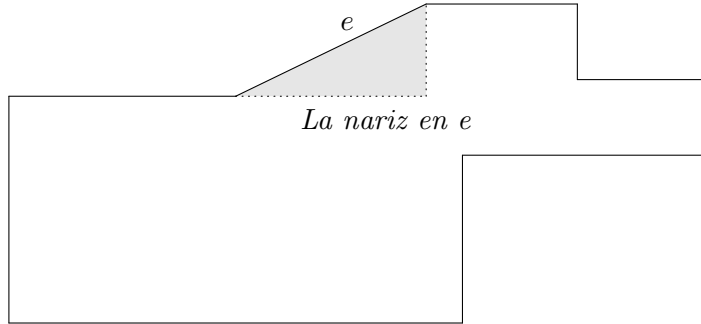


Figura 3.8: La nariz en e de un polígono 1-ortogonal.

se elimina, divide a P en subpolígonos 1-ortogonales más pequeños, e inductivamente el teorema será cierto.

Asumimos s.p.g. (sin pérdida de generalidad) que la inclinación de e está entre 0 y $\frac{\pi}{2}$ y que los vértices de e se encuentran etiquetados como u y v , tal que u es el vértice con menor ordenada. Sea w el tercer vértice de la nariz T de e .

Ya que P tiene un número par de aristas, y todos los ángulos internos son de tamaño a lo más $\frac{3\pi}{2}$, y s.p.g. ambas aristas adyacentes a e son horizontales. Sea e' la arista horizontal de P que precede a e en antihorario. A continuación definiremos los 4 vértices de Q' . Los primeros 2 vértices de Q' son u y v . Sea e'' el cateto vw . Desplazamos a e'' a la derecha hasta que alcance un vértice de P . Si e'' toca a una arista entera de P , entonces los vértices x y y incidentes a esta arista, junto con u y v son los vértices de Q' . Véase la figura 3.9. Vamos a suponer que e'' alcanzó a un sólo vértice x de P ; x es el tercer vértice de Q' .

Consideramos ahora el segmento de línea horizontal semi-abierta h , que une a u con la línea vertical e'' desplazada; h no contiene a u , pero sí contiene al vértice incidente de la derecha. Desplazamos a h hacia abajo hasta que alcance a un vértice de P , o bien, toque una arista horizontal f de P . En el primer caso, sea y el vértice extremo derecho de la arista que alcanzó h . El cuarto vértice de Q' es y . Este caso es ilustrado en la figura 3.10.

Consideraremos ahora el caso cuando h alcanza una arista f' de P . Esto sólo puede pasar si x es el segundo vértice incidente de e' . Este caso está ilustrado en la figura 3.11. En este caso, consideraremos el segmento de línea vertical semi-abierta h' que une a x con f' , que es cerrada en la parte inferior y abierta en la superior. Desplazamos a h' hacia la derecha hasta que alcance un vértice de P . Si toca mas de un vértice, se elige a y , como el vértice con menor ordenada, ese sería el último vértice de Q' .

Ahora es fácil de verificar que P se puede dividir en 2 o 3 polígonos 1-ortogonales. ■

Ahora si podemos demostrar el resultado de J. Kahn, M. Klawe y D. Kleitman:

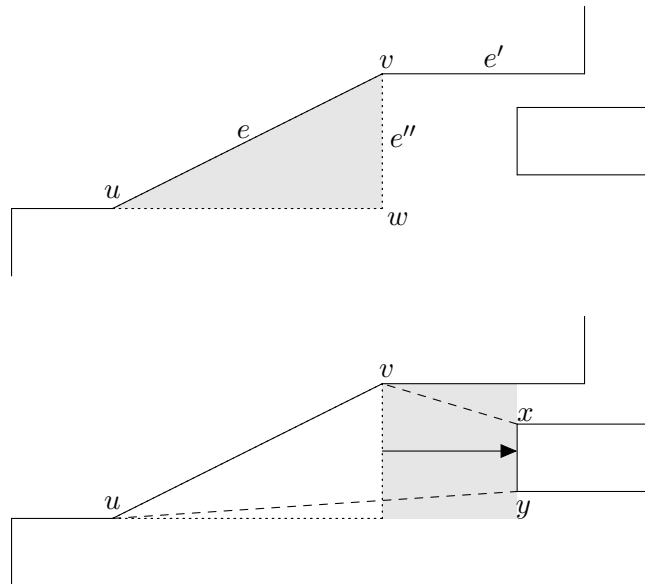


Figura 3.9: Buscando los vértices de Q' , si e'' alcanzó una arista completa.

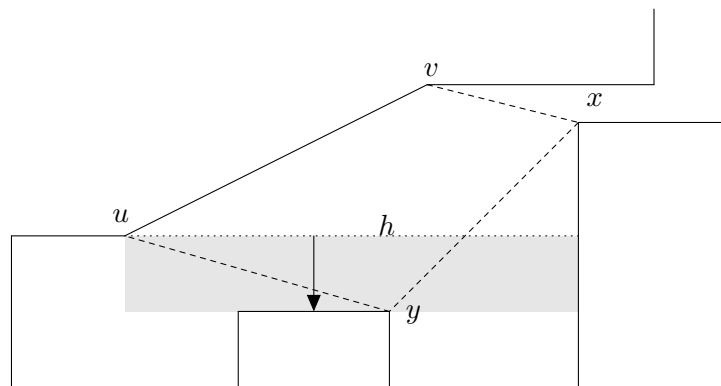


Figura 3.10: Buscando los vértices de Q' , desplazando h , caso 1.

Teorema 4 *Cualquier polígono ortogonal con n vértices puede vigilarse con a lo más $\frac{n}{4}$ guardias.*

Esta prueba es bastante similar a la demostración de Fisk [2] así que la dejamos en manos del lector, véase figura 3.12. Esta 4-coloración de P , divide en 4 clases cromáticas, por lo que la cardinalidad de una de estas es a lo más $\lfloor \frac{n}{4} \rfloor$.

Para mostrar que $\lfloor \frac{n}{4} \rfloor$ son ocasionalmente necesarios, considere el polígono de peine ortogonal P_m con $n = 4m$ vértices presentado en la figura 3.13. Para vigilar P_m necesitamos

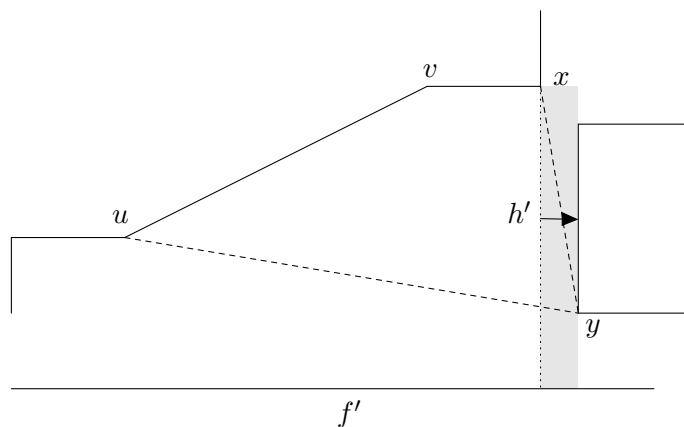


Figura 3.11: Buscando los vértices de Q' , caso 2.

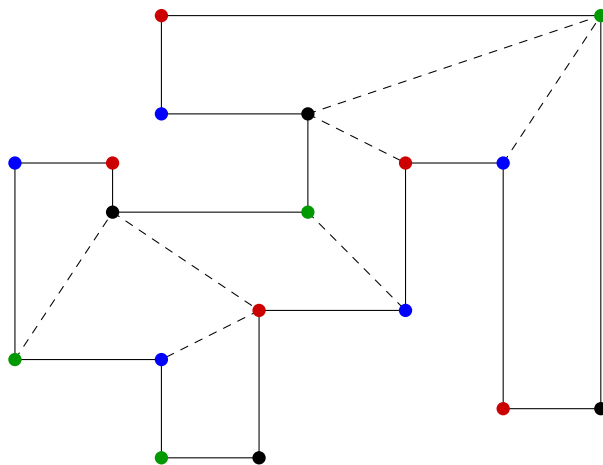


Figura 3.12: Polígono ortogonal cuadrilaterizado 4-coloreado.

m guardias.

Esta prueba promovió el estudio de descomposición de polígonos ortogonales en cuadriláteros convexos. Sack [11] dio un algoritmo que obtiene dicha división en tiempo $O(n)$.

H. Edelsbrunner, J. O'Rourke y E. Welzl [5] presentaron un algoritmo que en tiempo $O(n)$ obtiene un conjunto de $\lfloor \frac{n}{4} \rfloor$ puntos guardia que vigilan un polígono ortogonal dado; su algoritmo se basa en una descomposición del polígono en subpolígonos con forma de "L".

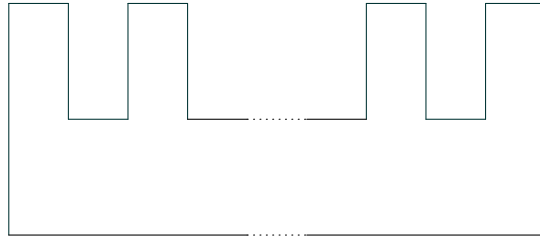


Figura 3.13: Polígono ortogonal denominado peine.

3.5.1. Iluminación de polígonos ortogonales con reflectores

El esfuerzo y dedicación de estudiar polígonos ortogonales iluminados con reflectores formó el siguiente resultado que no es tan evidente y su demostración es bastante sencilla y elegante. Abello, Estivill-Castro, Shermer y Urrutia [7], estudiaron este resultado:

Tenemos que considerar la siguiente clasificación aristas y de vértices, para lograrlo tenemos que colocarnos a interior del polígono ortogonal:

- Aristas superiores: Son las aristas superiores, es decir, las que usted vería si mira hacia el norte dentro del polígono.
- Aristas inferiores: Son las aristas inferiores, es decir, las que usted vería si mira hacia el sur dentro del polígono.
- Aristas derechas: Son las aristas derechas, es decir, las que usted vería si mira hacia el este dentro del polígono.
- Aristas izquierdas: Son las aristas izquierdas, es decir, las que usted vería si mira hacia el oeste dentro del polígono.
- Vértices superior izquierdos (SI): Son los vértices cuyas aristas incidentes son aristas superiores y aristas izquierdas.
- Vértices superior derechos (SD): Son los vértices cuyas aristas incidentes son aristas superiores y aristas derechas.
- Vértices inferior izquierdos (II): Son los vértices cuyas aristas incidentes son aristas inferiores y aristas izquierdas.
- Vértices inferior derechas (ID): Son el conjunto de vértices cuyas aristas incidentes son aristas inferiores y aristas derechas.

Podemos observar que un polígono ortogonal tiene 2 clases de vértices: cóncavos y convexos (véase la figura 3.14).

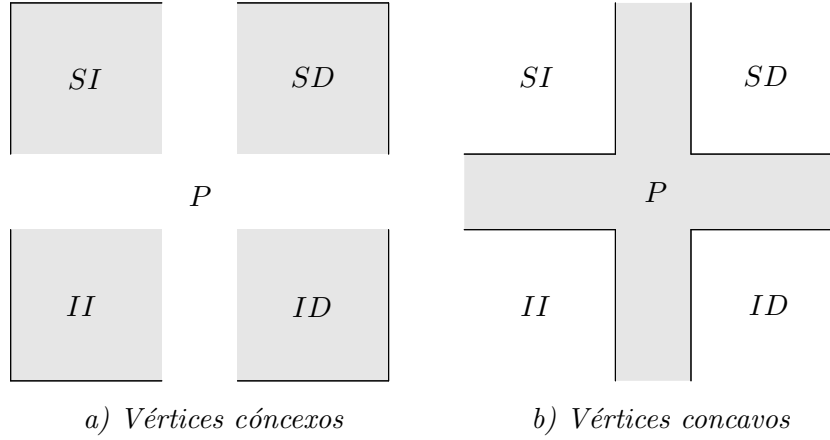


Figura 3.14: Clasificación para vértices cóncavos y convexos de un polígono P .

Teorema 5 *Cualquier polígono ortogonal P con n vértices son suficientes y ocasionalmente necesarios $\lfloor \frac{3n-4}{8} \rfloor$ reflectores ortogonales.*

Un reflector de tamaño angular $\frac{\pi}{2}$ se dice que es ortogonal.

Denotaremos como regla SI si colocamos un reflector en cada vértice SI. Sucede lo mismo para denotar a las reglas SD, II, ID cada una con sus respectivos vértices.

Prueba. Primero probaremos que al colocar un reflector ortogonal en cada elemento de algún conjunto de vértices; SI, SD, II ó ID, iluminan P . s.p.g. tomaremos la regla SI, iluminando el área entre $\frac{3\pi}{2}$ y 2π .

Lema 5 *Sea P un polígono ortogonal, entonces el interior de P puede ser iluminado por cualquier regla o conjunto de vértices SI, SD, II ó ID.*

Sea p cualquier punto en P . Considere el segmento de línea horizontal h más grande totalmente contenido en P y con vértice derecho en p . Sea e la arista vertical, la cual contiene el vértice izquierdo de h . Deslizamos h hacia arriba hasta alcanzar una arista superior de P o el vértice superior x de e . En el segundo caso, existe un reflector en x que ilumina p . Supongamos que h alcanza una arista superior f de P . Sea y el vértice izquierdo de f , claramente $y \in SI$ por lo que hay un reflector en y que ilumina p ; ver figura 3.15. \square

Si usamos los cuatro conjuntos y colocamos reflectores en cada uno de sus elementos, entonces colocaremos exactamente dos reflectores en cada vértice cóncavo y uno en cada vértice convexo. Si P tiene r vértices cóncavos c vértices convexos, entonces el número

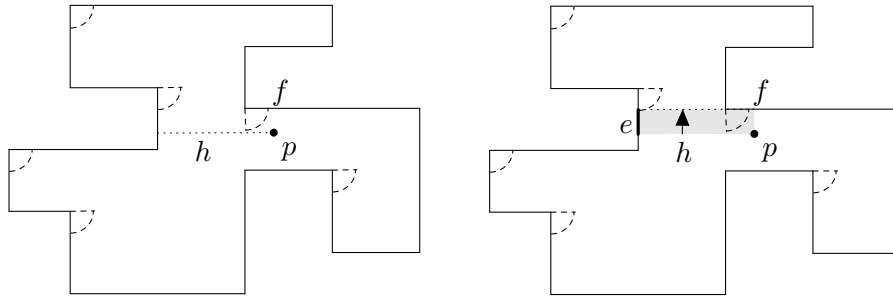


Figura 3.15: Reflectores colocados con la regla SI.

total m de reflectores usados por los cuatro conjuntos es $2r + c$. Sabemos que $r = \frac{n-4}{2}$ y $c = \frac{n+4}{2}$, por lo que:

$$m = 2r + (n - r) = 2\frac{n-4}{2} + \frac{n+4}{2} = \frac{3n-4}{2}$$

Claramente $\frac{3n-4}{2}$ son los reflectores utilizados por las 4 reglas (SI, SD, II, ID), y la cardinalidad de uno de los conjuntos es a lo más $\frac{3n-4}{8}$.

Para mostrar que la cota es justa, consideremos la siguiente familia de polígonos que tiene $12 + 8k$ vértices y se necesitan $4 + 3k$ reflectores para iluminarla; ver figura 3.16. Podemos observar que $\frac{3(12+8k)-4}{8} = 4 + 3k$ reflectores son requeridos para iluminar esta familia. ■

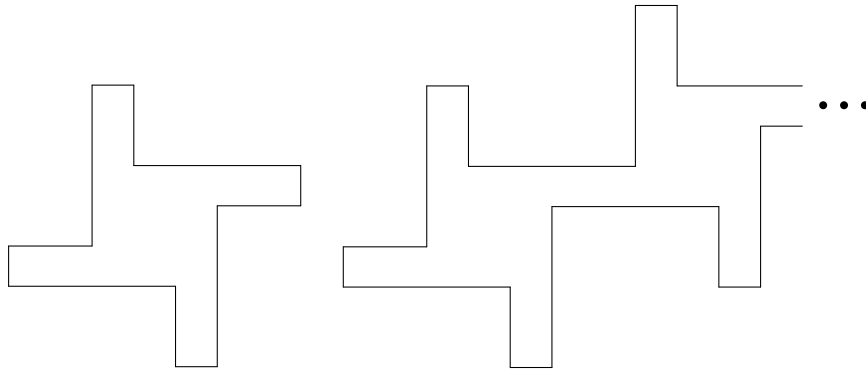


Figura 3.16: Familia de polígono con $12 + 8k$ vértices que requieren $4 + 3k$ reflectores.

En 1986, E. Györy [4] obtuvo otra prueba del teorema 4. En esta prueba él descompone el polígono ortogonal en a lo más $\lfloor \frac{n}{4} \rfloor$ polígonos en forma de “L” y cada uno de dichos polígonos pueden vigilarse con solo un guardia. En la siguiente sección se reproducirá esta demostración.

3.5.2. Partición de polígonos ortogonales en piezas con forma de L

Una prueba diferente del teorema [4] se obtuvo por J. O'Rourke [10]. Su enfoque lo hace a través de polígonos ortogonales en forma de L, notemos que tienen seis vértices. Tenga en cuenta que cualquier polígono en forma de L puede ser vigilado por un solo guardia. La idea principal de O'Rourke es dividir un polígono ortogonal en piezas en forma de L, cada una de las cuales puede ser vigilado con un guardia.

O'Rourke reformula su resultado en términos de r , el número de vértices cóncavos de un polígono ortogonal, en lugar de n , el número total de vértices. Esta reformulación se justifica porque existe una relación fija entre r y n :

Lema 6 *En un polígono ortogonal de n vértices, r son cóncavos, es decir, $n = 2r + 4$.*

Prueba. Sea c el número de vértices cuyo ángulo interno es $\frac{\pi}{2}$; claramente, $n = c + r$. Dado que la suma de los ángulos internos de un polígono simple es $(n - 2)\pi$, y dado que cada ángulo en un vértice cóncavo es de $\frac{3\pi}{2}$, entonces:

$$(n - 2)\pi = c\left(\frac{\pi}{2}\right) + r\left(\frac{3\pi}{2}\right)$$

Si despejamos c y lo sustituimos en $n = c + r$ nos da como resultado $n = 2r + 4$. \square

Como $\lfloor \frac{n}{4} \rfloor = \lfloor \frac{(2r+4)}{4} \rfloor = \lfloor \frac{r}{2} \rfloor + 1$, entonces el teorema 4 se puede establecer de la siguiente manera:

Teorema 6 $\lfloor \frac{r}{2} \rfloor + 1$ guardias son necesarios y suficientes para vigilar el interior de un polígono ortogonal P con r vértices cóncavos.

Prueba. Esta prueba se realiza por inducción sobre el r . Si $r = 0$ el resultado es claramente verdadero. Surgen 2 casos:

1. Hay un segmento de recta l horizontal o vertical que conecta dos vértices cóncavos de P tal que el interior de l está totalmente contenido en el interior de P .
2. No existe tal segmento de línea horizontal o vertical. En este caso, decimos que P está en posición general.

Sea l un segmento de línea horizontal como en el caso (1). Entonces cuando l divide a P en dos subpolígonos ortogonales P_1 y P_2 con r_1 y r_2 vértices cóncavos, respectivamente, tales que $r_1 + r_2 = r - 2$. Por inducción, podemos vigilarlos con $\lfloor \frac{r_1}{2} \rfloor + 1$ y $\lfloor \frac{r_2}{2} \rfloor + 1$ guardias respectivamente. Pero como $(\lfloor \frac{r_1}{2} \rfloor + 1) + (\lfloor \frac{r_2}{2} \rfloor + 1) \leq \lfloor \frac{r}{2} \rfloor + 1$, entonces el teorema se cumple.

Algunas terminologías y resultados necesitan ser probados en (2). Un “corte” de un polígono ortogonal es una extensión de una de las aristas incidentes a un vértice cóncavo de P hacia el interior de P hasta que alcanza el límite de P , véase la figura 3.17. Notemos que un corte a través de un vértice cóncavo u “resuelve” ó elimina a u , ya que éste deja de ser un vértice cóncavo para cualquiera de los 2 subpolígonos generados (por ejemplo como para P_1 ó P_2) cuando P es dividida por el corte. Un corte es llamado corte impar, si al menos P_1 ó P_2 tiene un número impar de vértices cóncavos.

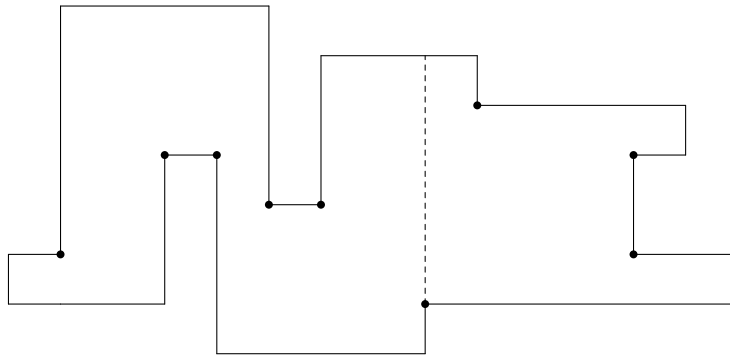


Figura 3.17: Un corte de P y sus vértices cóncavos.

Para terminar la prueba de nuestro resultado, utilizaremos el siguiente resultado, que se demostrará después de que termine nuestra prueba: *Cada polígono ortogonal en posición general tiene un corte impar.*

Sea P un polígono ortogonal en posición general, y consideremos que un corte impar lo divide en dos subpolígonos P_1 y P_2 . Supongamos s.p.g. que P_1 tiene un número impar de vértices cóncavos. Sea r_1 y r_2 el número de vértices cóncavos de P_1 y P_2 , respectivamente. A continuación por inducción, podemos vigilar a P_1 y P_2 con $(\lfloor \frac{r_1}{2} \rfloor + 1)$ y $(\lfloor \frac{r_2}{2} \rfloor + 1)$ vértices guardia³ respectivamente. Tenga en cuenta que $r_1 + r_2 = r - 1$, y como r_1 es impar se sigue por un simple caso de análisis que $(\lfloor \frac{r_1}{2} \rfloor + 1) + (\lfloor \frac{r_2}{2} \rfloor + 1) \leq \lfloor \frac{r}{2} \rfloor + 1$. Como se puede observar el teorema se cumple. ■

Para terminar la demostración del teorema 6, ahora demostramos que cada polígono ortogonal en posición general tiene un corte impar.

Notemos primero que la condición de posición general es esencial. El polígono en la figura 3.18 (a) no tiene ningún corte impar. Además, un corte horizontal no siempre existe. El polígono en la figura 3.18 (b) no tiene ningún corte impar horizontal. Esto complica un poco las cosas. Dividimos cortes horizontales y cortes verticales, los cuales denotaremos como *cortes-H* y *cortes-V*, respectivamente.

Observe que si el número de vértices cóncavos de P es par, entonces el número de

³Son aquellos que pueden vigilar los 360° .

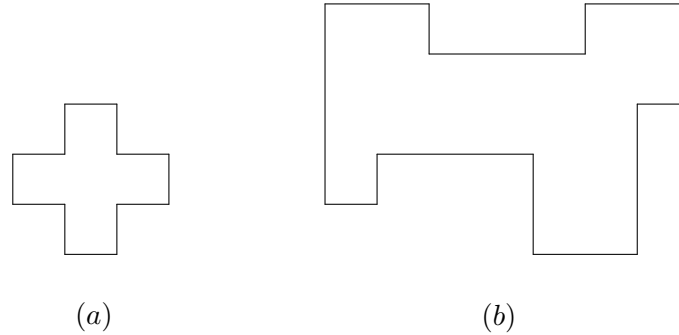


Figura 3.18: Un polígono ortogonal sin ningún corte impar, y un polígono ortogonal sin cortes horizontales.

vértices de P_1 más los de P_2 es igual a $r - 1$, el cual es impar, y por lo tanto P_1 ó P_2 debe tener un número impar de vértices cóncavos. Supongamos entonces que P tiene un número impar de vértices cóncavos.

Un vértice cóncavo v de P lo llamaremos *aislado- H* si el otro vértice incidente de la arista horizontal de v que lo contiene es convexo. Una arista horizontal e de P se llamará una *arista- H* si los dos vértices de e son cóncavos.

La división de P la llamaremos *partición- H* y la obtendremos de la siguiente manera: Extendemos todas las *aristas- H* de P en ambos extremos hasta que toquen el borde de P ; ver figura 3.19. La *gráfica- H* de P es la gráfica de adyacencia de las regiones de la partición inducida en P por todas sus *aristas- H* orientadas de la siguiente manera: Si dos regiones A y B de la partición son adyacentes y están separadas por la extensión de una *arista- H* en la frontera de A , orientemos la arista que las une de A hacia B ; ver figura 3.19.

Ahora clasificaremos a los vértices de la *gráfica- H* de P de la siguiente manera:

Nombre	Grado total	Aristas entrantes	Aristas salientes
Vértice hoja	1	1	0
Vértice rama	3	1	2
Vértice origen	2 ó 4	0	2 ó 4
Vértice hundido	2	2	0

Lema 7 Si la *gráfica- H* de P contiene un vértice hundido, entonces tiene un corte impar.

Prueba. Sea S la región que corresponde a un vértice hundido de P , y sean C_1 y C_2 la región superior e inferior, respectivamente de un *corte- H* de S . Sea k el número de vértices *aislados- H* que contiene S , y sea u el número total de vértices cóncavos en la porción del

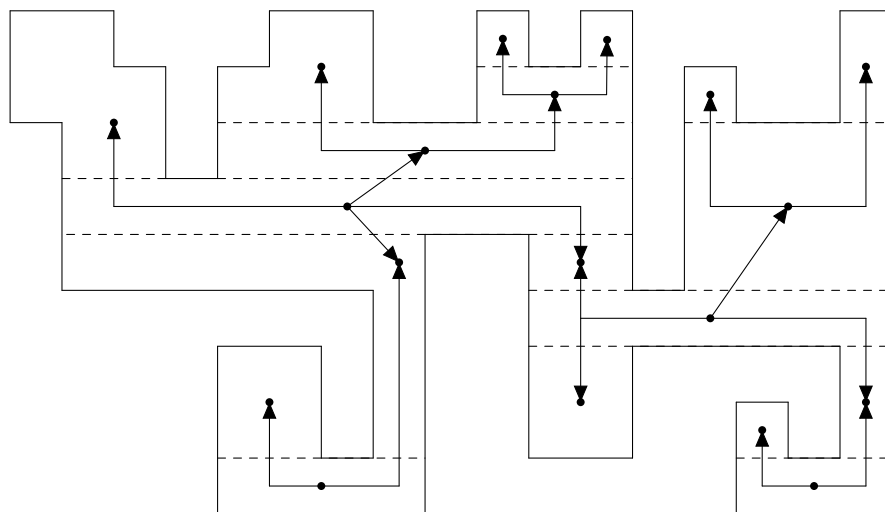


Figura 3.19: La *partición-H* de P y su *gráfica-H*.

polígono original, por encima de C_1 (sin incluir el vértice que forma a C_1), véase la figura 3.20.

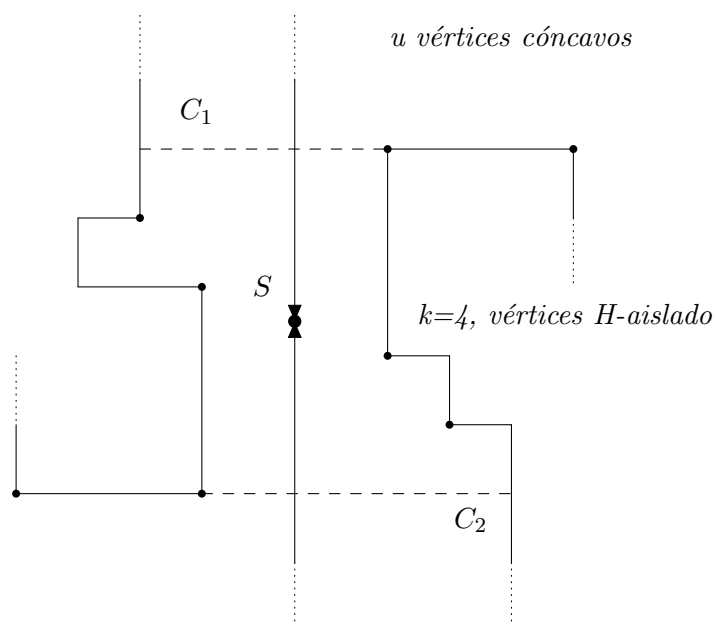


Figura 3.20: Una región hundida S siempre permite un corte impar.

Si u es impar entonces C_1 es un corte impar. Si u es par, un corte en el vértice *aislado-H* más elevado de la región S (si $k > 0$) o en C_2 (si $k = 0$) es un corte impar y el lema se cumple. \square

Ahora definiremos los siguiente, llamaremos región hoja, rama, origen o hundida si es la región correspondiente a un vértice hoja, rama, origen o hundido, respectivamente. A continuación probaremos lo siguiente:

Lema 8 *Si P no contiene cortes-impares, su gráfica- H contiene un solo vértice origen, P tiene exactamente un vértice aislado- H , y se encuentra localizado en la región origen de su gráfica- H .*

Prueba. Observe que si la gráfica- H de P contiene dos vértices origen, entonces la ruta que los unen deben contener necesariamente un vértice hundido. Sin embargo, ya que P no contiene cortes-impares, su gráfica- H no tiene vértices hundidos. La gráfica- H de P contiene un vértice origen único. Puesto que la gráfica- H de P es un árbol con un solo vértice origen, cada vértice de la gráfica- H , a excepción de su vértice origen, tiene una sola arista de entrada. Sea K una de estas regiones de la partición- H de P . P no admite ningún corte impar, ya que el corte correspondiente a la arista entrante de K es un corte par. Supongamos s.p.g. que esta arista es una arista inferior de K . Entonces si K contiene vértices *aislados-H*, el más bajo de ellos genera un corte par, lo que es una contradicción.

Sin embargo, ya que P contiene un número impar de vértices cóncavos (o cualquier corte- H que pudiera ser un corte impar) debe contener al menos un vértice *aislado-H*, y debe ser un vértice de la región de origen de la partición- H de P . Por otra parte, notemos que si la región de origen contiene más de un vértice *aislado-H*, uno de ellos genera un corte impar, lo cual es una contradicción, por lo que el lema se cumple. \square

Podemos darnos cuenta de que todos los resultados probados para cortes- H también sirven para los cortes- V , y en particular se tiene que si P no admite cortes-impares- V , tiene exactamente un vértice *aislado-V* situado en la región origen de la gráfica- V de P .

Ahora demostraremos:

Lema 9 *Cualquier polígono ortogonal P en posición general con un número $r \geq 3$ de vértices cóncavos admite un corte impar.*

Prueba. Supongamos que P no admite cortes-impares, entonces tiene exactamente un vértice *aislado-H* u y un vértice *aislado-V* v . Además estos vértices se encuentran en las regiones origen de la gráfica- H y la gráfica- V de P . Cualquier otro vértice cóncavo de P pertenece a una *arista-H* y *arista-V*. Por lo tanto todos los vértices cóncavos de P se encuentran en una cadena de vértices cóncavos que se inicia, por ejemplo, en u y termina en v , ver figura 3.21. Esto implica que tanto u como v se encuentran en las regiones de las hojas de la particiones- H y particiones- V de P , lo cual es una contradicción. \square

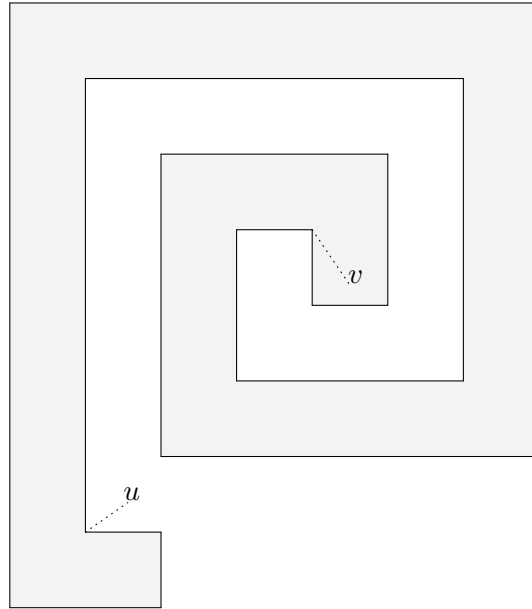


Figura 3.21: Un polígono ortogonal en espiral.

Esto concluye nuestra prueba del teorema 6. ■

3.5.3. Algoritmo para partir polígonos ortogonales en piezas con forma de L

La prueba descrita en la sección anterior no se traduce directamente en un algoritmo, debido a la dificultad de encontrar cortes-impares. Sin embargo, ahora mostramos que, con la adición de algunas ideas nuevas, el algoritmo puede ser implementado y ejecutado en tiempo $O(n \ln n)$. La aplicación del nuevo algoritmo de trapezoidalización reduce el tiempo a $O(n \ln \ln n)$. Vamos a suponer que todo polígono se encuentra en posición general; este es el caso más difícil, como se mostró en la sección anterior.

En primer lugar el polígono es preprocesado para detectar qué cortes horizontales son impares. Entonces se particiona en cada corte impar horizontal. Finalmente, los guardias se colocan en cada una de las piezas resultantes. Esto logrará la cobertura con $\lfloor \frac{r}{2} \rfloor + 1$ guardias.

Se debe poner atención en dos puntos. El primero, que es más fácil algorítmicamente hacer un corte impar en un polígono con un número impar r de vértices cóncavos que en uno con un número par. La razón, es que no hay actualización de los cálculos preprocesados que son necesarios para cada corte cuando el número de vértices cóncavos es impar. Por lo tanto damos el paso, contrario a la intuición, de introducir un nuevo vértice cóncavo en

el polígono si r es par; la cota $\lfloor \frac{r}{2} \rfloor + 1$ claramente no es afectada.

El segundo punto se produce cuando todos los cortes-impares horizontales han sido colocados, y sólo quedan cortes-impares verticales. Como se mencionó anteriormente, el polígono debe tener una estructura restringida, de lo contrario el costo computacional sería mucho.

Estas afirmaciones se justifican a continuación.

El algoritmo para localizar $\lfloor \frac{r}{2} \rfloor + 1$ guardias en un polígono ortogonal de n vértices, r de los cuales son cóncavos, consta de seis pasos distintos:

1. Si r es par, a continuación, añadir un vértice cóncavo adicional.
2. Realizar un barrido del plano para encontrar todos los cortes horizontales.
3. Recorrer el borde del polígono una vez, etiquetando los cortes con su paridad.
4. Realizar la partición del polígono en cada corte impar horizontal.
5. Para cada pieza resultante, colocar un guardia en cada vértice cóncavo.
6. Retirar el vértice extra, si se introdujo en el paso 1.

Cada uno de estos pasos se describirá en detalle y justificadamente.

Añadir el vértice cóncavo: $O(n)$

Si r es par, entonces $\lfloor \frac{(r+1)}{2} \rfloor + 1 = \lfloor \frac{r}{2} \rfloor + 1$, por lo que la adición de un vértice cóncavo está justificada. La razón para ello se ha aludido más arriba, y será extendida más adelante. El vértice cóncavo adicional se puede añadir en tiempo lineal como siguiente. Elija un vértice convexo arbitrio $v = (x, y)$. Encontrar la separación más pequeña no nula en su arista horizontal y vertical Δx y Δy de las coordenadas x y y de cada uno de los vértices, buscar estas coordenadas se puede hacer en tiempo $O(n)$.

Eliminar el vértice v y reemplazarlo con otros tres como se ilustra en la figura 3.22. Esto no interfiere con la visibilidad, se mostrará más adelante.

Barrer el plano encontrando cortes horizontales: $O(n \ln n)$ o $O(n \ln \ln n)$

Cada vértice cóncavo determina un corte horizontal único. El objetivo de este paso es encontrar cada corte horizontal, para insertar un nuevo vértice “artificial” en una lista circular de vértices que representan el polígono al final de cada corte, una lista doblemente enlazada que une cada vértice cóncavo asociado con su vértice artificial. Esta estructura de datos sirve como entrada al paso 3 del algoritmo. Las ubicaciones de los vértices artificiales se encuentran por un barrido horizontal de una línea de arriba a abajo en el polígono. Este es un barrido plano estándar, y es casi idéntico al algoritmo de partición monótona de Lee

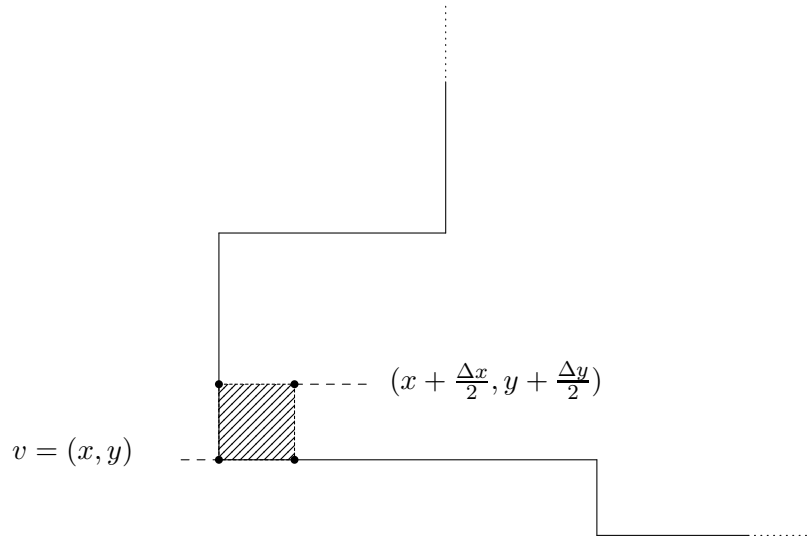


Figura 3.22: La adición de un vértice cóncavo.

y Preparata. Las aristas verticales del polígono están ordenadas por su máxima ordenada. En cada posición de la línea de barrido H , una estructura de datos S contiene todas las aristas verticales encontradas por H , organizadas de izquierda a derecha. Cuando H se mueve hacia abajo y encuentra un vértice cóncavo v , una arista vertical interseca el corte horizontal de v y v se encontrará disponible en S como adyacente a la arista vertical cuya parte superior o inferior es v (ver figura 3.23). Después de calcular las coordenadas del vértice artificial correspondiente, una arista vertical se conecta o se elimina de la estructura de datos S dependiendo de si v es la parte superior o la parte inferior de la arista vertical. La estructura de datos puede insertar y eliminar en tiempo $O(\ln n)$, lo que nos lleva a $O(n \ln n)$ de tiempo total. Dado que estos cortes horizontales son precisamente los que el algoritmo de trapezoidalización de Tarj y Van Wyk encuentra, el uso de ese algoritmo en lugar de un barrido plano reduce la complejidad de tiempo de éste paso a $O(n \ln \ln n)$. Como este es el único paso que podría ser super-lineal, todo el algoritmo es $O(n \ln \ln n)$.

Recorrer el borde para calcular paridad de los cortes: $O(n)$

El siguiente paso es determinar cuáles cortes horizontales son cortes impares, es decir, que haya un número impar de vértices cóncavos a un lado u otro. A medida que se conoce el número total de vértices cóncavos r , encontrar el número a un lado de un corte determina el número para el otro lado. El número a un lado de cada corte se puede encontrar en un único recorrido del borde del polígono.

Podemos distinguir tres tipos de vértices: convexos, cóncavos y artificiales. Comenzamos en un vértice arbitrario, e inicializamos un contador a cero, y procedemos en sentido antihorario alrededor del polígono. Si el siguiente vértice es convexo, no se hace nada, y si

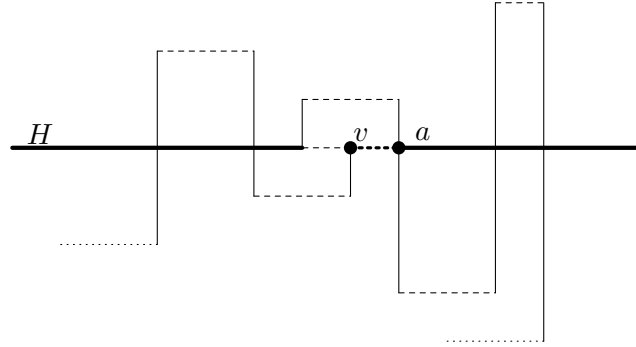


Figura 3.23: Cuando la línea de barrido H alcanza un vértice cóncavo v , el vértice artificial a en su corte se encuentra en una arista vertical adyacente a una arista de v en S .

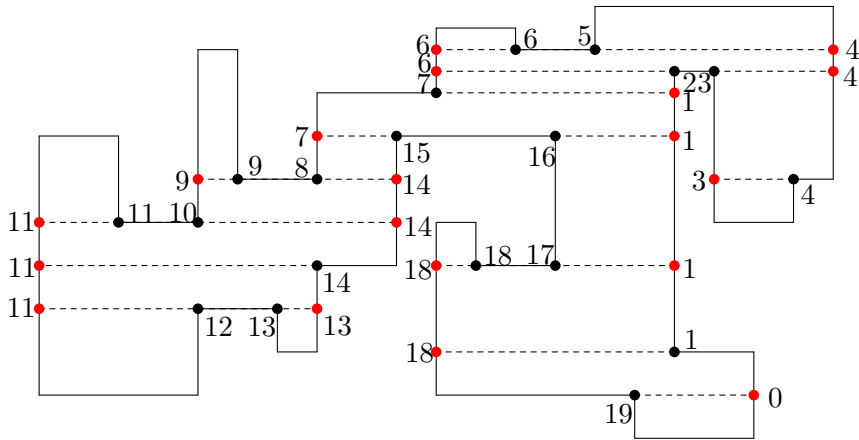


Figura 3.24: Los vértices cóncavos son los puntos negros, los artificiales son los puntos rojos y fueron enumerados en sentido antihorario, incrementando la etiqueta en cada vértice cóncavo.

el siguiente vértice es artificial, etiqueta con el valor del contador, y si el siguiente vértice es cóncavo, incrementa el valor del contador y se etiqueta el vértice con este nuevo valor (ver figura 3.24). Tan pronto como dos vértices extremos de un corte se encuentren etiquetados, el número de vértices cóncavos k para un lado se determina por la diferencia entre las dos etiquetas, cuando el vértice artificial de un corte se haya encontrado en segundo lugar entonces, la diferencia de las dos etiquetas es k , por otro lado, si el vértice cóncavo es el segundo en ser encontrado entonces, k es la diferencia de las etiquetas menos 1 (véase la figura 3.25).

En realidad no es necesario calcular el número de vértices cóncavos a cada lado, ya

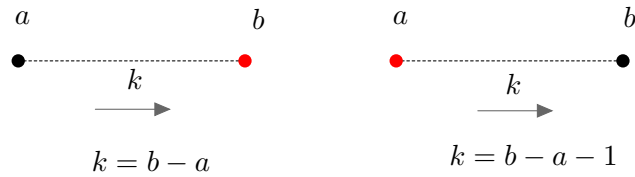


Figura 3.25: Las etiquetas a y b asignadas a dos vértices del mismo corte, es determinada por el número de vértices cóncavos k para un lado de P , por su diferencia o su diferencia menos 1.

que sólo se necesita la paridad. Así, cada vértice cóncavo y artificial puede ser etiquetado como par o impar durante el recorrido, y la paridad de los cortes queda determinado por la modificación directa de las reglas anteriores.

Cortar en cada corte impar horizontal: $O(n)$

El objetivo del cuarto paso es cortar el polígono en cada par de corte horizontal; las piezas restantes serán fáciles de cubrir con los guardias. El único problema de este paso es que, después de que se hace un corte, las paridades calculadas en el paso anterior pueden no ser las correctas para los cortes en las dos piezas. Por ejemplo, la figura 3.26 muestra cómo un corte en un corte impar puede, o bien, no afectar la paridad de otros cortes, o podría cambiar su paridad, dependiendo de si el número impar de vértices cóncavos está dentro de la parte que incluye el corte o no. Sin embargo, tenga en cuenta que las situaciones de la figura 3.26 sólo pueden surgir cuando el número total de vértices cóncavos r es par. Cuando r es impar, entonces sólo hay un tipo de corte impar: un corte que tiene un número impar de vértices cóncavos a cada lado. El cortar el polígono en un corte impar no afecta la paridad de los cortes de cada mitad (ver la figura 3.26).

Desde el paso 1 del algoritmo, se garantiza que r es impar, todos los cortes impares horizontales se pueden hacer sin ningún tipo de actualización de las paridades calculadas en el paso 3. Al término de esta división, nos quedamos con varios polígonos ortogonales, en los que todos tienen cortes pares horizontales.

Colocación de Guardias: $O(n)$

Consideremos ahora un polígono ortogonal sin cortes impares horizontales. Si no tiene vértices cóncavos, entonces es un rectángulo y se puede cubrir con un solo guardia colocándolo en cualquier lugar del interior. Esto satisface la cota $\lfloor \frac{r}{2} \rfloor + 1$ del teorema.

Si hay más de un vértice cóncavo, en el lema 9 se establece la existencia de cortes impares verticales. Sin embargo, encontrar tales cortes haría necesaria la repetición de los cuatro pasos anteriores del algoritmo, y en última instancia puede conducir a cambiar a cortes horizontales. Tal oscilación y el cómputo que implica podría ser muy caro. Afor-

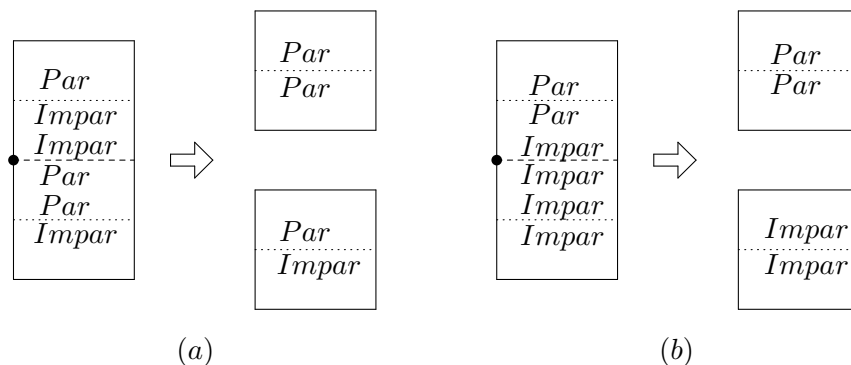


Figura 3.26: (a) Si se hace un corte en un corte impar puede cambiar la paridad de otros cortes, si el número de vértices cóncavos es par. (b) Si el número de vértices cóncavos es impar, entonces hacer un corte en un corte impar no afecta la paridad de otros cortes.

tunadamente, los polígonos sin cortes-impares horizontales tienen una estructura especial que hace que la colocación de guardias sea fácil, pero antes de verla daremos la definición de *histograma* vertical.

Un polígono ortogonal Q es llamado *histograma* vertical, si tiene una arista horizontal (llamada base de Q), tal que cada punto de Q es visible desde algún punto de su base; Véase figura 3.27.

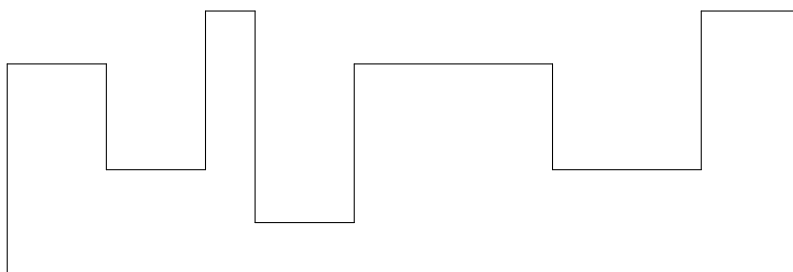


Figura 3.27: Polígono ortogonal denominado histograma.

Se comprobó en la sección anterior que un polígono con solamente cortes pares horizontales debe parecerse al de la figura 3.28: está formado por dos *histogramas* que se encuentran unidos por sus bases. Esta restricción estructural se manifestó de una manera diferente, pero en términos equivalentes en el lema 8. Un polígono que no tiene cortes impares horizontales debe tener un vértice cóncavo cuyo corte horizontal parte al polígono en dos histogramas verticales (como el que se aprecia en la figura 3.28). Dicho polígono claramente debe tener un número impar de vértices cóncavos.

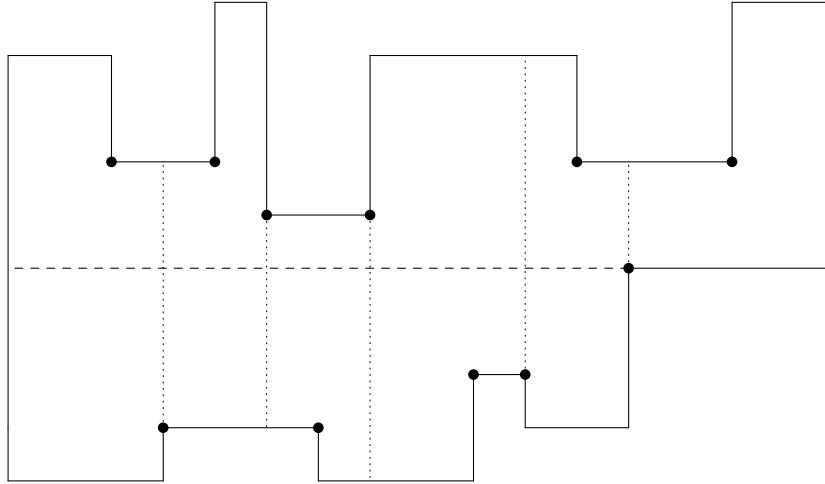


Figura 3.28: Polígonos sin un corte impar horizontal, tienen un corte horizontal que particiona a P en dos histogramas verticales, los cuales después se pueden cortar verticalmente en piezas con forma de L.

Los guardias colocados en cada vértice cóncavo de izquierda a derecha necesariamente cubrirán a P . Para ver esto, deje que los vértices cóncavos sean v_1, v_2, \dots, v_k en forma ordenada de menor a mayor con respecto a su coordenada x , k es impar. Haga un corte vertical en v_2, v_4, \dots, v_k . El polígono ha sido ahora dividido en $\lfloor \frac{k}{2} \rfloor + 1$ en forma de L, cada uno con un vértice cóncavo (ver figura 3.28). Coloque los guardias en $v_1, v_3, \dots, v_{2\lfloor \frac{k}{2} \rfloor + 1}$. Esto claramente cubre el polígono con $\lfloor \frac{k}{2} \rfloor + 1$ guardias, logrando la cota del teorema.

La clasificación de los vértices cóncavos se puede lograr en tiempo lineal por la fusión de las dos cadenas de histogramas, que necesariamente ya se encuentran ordenados. Una vez que el orden es determinado, ningún corte vertical debe hacerse en realidad: los guardias se asignan sólo a todos los vértice cóncavos. Tenga en cuenta que después de que el paso 5 se ha completado, el polígono se ha dividido en piezas implícitamente en forma de L, uno de estos puede ser modificado para hacer un rectángulo en el paso final, esto se discute a continuación.

Remover el vértice extra: $O(n)$

Si se introdujo un vértice extra mediante la eliminación de un c “chip” en una esquina convexa del polígono en el paso 1 del algoritmo (véase la figura 3.29), entonces, devolver el polígono a su forma original no requerirá más guardias, ni siquiera mover las ubicaciones de los guardias ya existentes. Consideremos dos casos. Si el vértice introducido v se le asignó un guardia en el paso 5, entonces, retiramos el chip y regresamos la esquina a su forma original (convexa), y colocar el guardia en la esquina. Es evidente que la región en

forma de L anteriormente cubierto por el guardia de v es ahora un rectángulo y cubierta por el guardia en la esquina. Si v no tiene asignado un guardia, entonces, se hizo un corte en ese vértice, ya sea un corte horizontal en el paso 3 o un corte vertical (implícito) en el paso 4. En cualquiera de los dos casos, una de las dos aristas incidentes a v es una arista completa de una región en forma de L que tiene un guardia g en ella (véase la figura 3.29). Es evidente que el guardia g ve a c en cualquiera de los 2 casos.

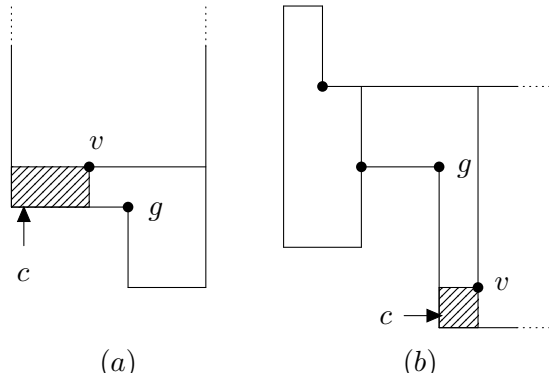


Figura 3.29: La región c del chip es visible para algún guardia g , sin importar que el corte de v se realice en la horizontal (a) o en la vertical (b).

3.6. La dificultad del problema

Existe una gran cantidad de restricciones que pueden ser impuestas a los guardias usados para vigilar la galería. A continuación se mencionan los principales tipos de guardias o fuentes de luz:

- Guardias en los vértices: Comúnmente llamados guardias vértice, en este caso los guardias se localizan en los vértices del polígono.
- Guardias en puntos: También llamados guardias punto. Estos son colocados en cualquier punto dentro o en la frontera del polígono.
- Guardias en las aristas: Son llamados guardias aristas. A los guardias se les permite moverse sobre la arista y un punto se considera vigilado si es visible desde algún punto de la ruta o arista.
- Guardias móviles: En esta variante a los guardias se les permite desplazarse sobre segmentos de línea contenidos en el polígono.

- Reflectores: Un reflector es una fuente de luz la cual ilumina con un ángulo restringido α y que puede ser rotado sobre su ápice. El ápice de los reflectores puede ser colocado en los vértices o en puntos de cualquier parte del polígono.

Los guardias son usados para vigilar las galerías de arte. Como ya se había explicado, estas galerías pueden ser expresadas geoméricamente como polígonos. A continuación se muestran los principales objetivos geoméricos o tipos de galerías usadas:

- Galerías normales: Este tipo de galerías tienen forma de polígonos simples.
- Galerías tradicionales: En esta variante la galería está formada por un conjunto de habitaciones rectangulares dentro de un edificio rectangular.
- Galerías ortogonales: En este tipo de galerías se restringe a que las aristas sean paralelas a alguno de los ejes coordenados.
- Galerías con huecos: Para esta variante, se permite que la galería contenga otros polígonos, llamados huecos, cuyo interior no se considera parte del interior de la galería.
- Familia de conjuntos de convexos: En esta variante la galería está formada por un conjunto de polígonos convexos.

A continuación se presentan algunos resultados para estas variantes:

Nombre	G. punto	G. vértice	G. arista	G. móviles	Reflectores
Galerías normales	-	$\lfloor \frac{n}{3} \rfloor$	-	-	-
Galerías tradicionales ortogonales	-	$\lfloor \frac{n}{2} \rfloor$	-	-	-
Familia de conjuntos de convexos	$4n - 7$	-	-	-	-
Polígonos ortogonales	-	$\lfloor \frac{n}{4} \rfloor$	-	-	$\lfloor \frac{3n+4}{8} \rfloor$
Galerías normales con hoyos	$\lfloor \frac{n+h}{3} \rfloor$	$\lfloor \frac{n+2h}{3} \rfloor$	-	-	-
Galería ortogonal con hoyos	$\lfloor \frac{n}{4} \rfloor$	$\lfloor \frac{n+h}{4} \rfloor$	$\lfloor \frac{n}{4} \rfloor$	$\lfloor \frac{n}{4} \rfloor$	$\lfloor \frac{3n+4(h-1)}{8} \rfloor$

Sin embargo, hasta ahora no existe ningún resultado que coloque el mínimo número de guardias, es decir, estos resultados son en ocasiones necesarios y siempre suficientes. Pero esto ¿a qué se debe? Recordemos que en este capítulo, se demostró que la prueba de la galería de arte del teorema de Fisk puede cubrir un polígono de n vértices con $\lfloor \frac{n}{3} \rfloor$ guardias en tiempo $O(n \ln n)$. Aunque $\lfloor \frac{n}{3} \rfloor$ es necesario en algunos casos, a menudo estos son muchos más guardias de los necesarios para cubrir un polígono en particular. Por ejemplo, los polígonos convexos sólo requieren un guardia, pero el algoritmo de Avis y

Toussaint seguiría colocando $\lfloor \frac{n}{3} \rfloor$ guardias. A continuación se muestran algunos problemas que son, hasta ahora, intratables o problemas \mathcal{NP} .

La tabla que se muestra a continuación mostrada divide a los problemas en: problemas de vigilancia por cobertura y por descomposición. Los problemas de vigilancia de galerías de descomposición son aquellos en los que se trata de descomponer el polígono en el menor número de subpolígonos para después vigilar cada subpolígono. Los problemas de vigilancia de galerías por cobertura son aquellos que buscan el mínimo número de guardias para cubrir el polígono.

	Por cobertura		Por partición	
	c/Steiner	s/Steiner	c/Steiner	s/Steiner
Polígonos simples	\mathcal{NP} -duro	\mathcal{NP} -completo	?	$O(n^7 \log n)$
Polígonos con hoyos	\mathcal{NP} -duro	\mathcal{NP} -completo	\mathcal{NP} -duro	\mathcal{NP} -completo
Polígonos ortogonales monótonos		$O(n^2)$		$O(n)$
Cuadrícula 2D		$O(n^{2.5})$?
Cuadrícula 3D		\mathcal{NP} -completo		\mathcal{NP} -completo

Pero ¿qué es un problema \mathcal{NP} -completo?, para entender esto debemos aprender algunos conceptos sobre la teoría de la complejidad, estos bases serán dados en el siguiente capítulo.

3.7. Iluminación con radares

En esta sección incrementaremos el grado de dificultad al problema de vigilancia de galerías de arte, se presentarán algunos resultados dados por E. Kranakis, F. MacQuarie y J. Urrutia [1] quienes se dedicaron a estudiar reflectores giratorios que llamaremos radares.

En esa investigación se propone y estudia el problema de la cobertura con radares, un nuevo problema relativo a sensores rotatorios para la cobertura ininterrumpida de una región en el plano. Más concretamente si suponemos que tenemos n radares con idéntico rango y haz de luz se planea responder a, ¿cuál es la orientación inicial de estos radares, el ángulo mínimo y el rango de iluminación necesario para que una recta dada (finita o infinita) o el dominio de un plano esté cubierto por los radares todo el tiempo?

El problema lo podemos definir de la siguiente manera. Llamaremos radar a un reflector con una apertura de ángulo fijo variable que gira a velocidad constante. Supongamos que nos dan una región plana finita o infinita y que tenemos n radares modelados como antenas direccionales con rangos dados idénticos y diferentes haces de luz. Nosotros proveeremos un algoritmo que determinará la orientación inicial de los radares a fin de garantizar que ningún punto dentro del dominio se deje de cubrir. Además, también estamos interesados en los algoritmos para lograr cubrir el dominio con radares óptimos (ángulo/rango) y llevar a cabo esta tarea de monitoreo.

Para poder proseguir se necesita ofrecer alguna terminología básica, definiciones y la notación necesaria. A lo largo de la sección supondremos que tenemos n radares idénticos. Cada radar consta de una antena direccional giratoria con un rango (también llamada de radio) $r > 0$, ancho del haz de luz (también llamado ángulo) $0 \leq \phi \leq 2\pi$ que gira alrededor de su ápice (que es en una posición fija) con una velocidad constante en sentido horario. Todos los radares giran en la misma dirección y a la misma velocidad. Las antenas se colocan en alguna orientación inicial (esto determinado por un algoritmo) que depende de la localización particular de cada radar en relación con el resto de los radares. El área de cobertura de un radar en el tiempo t es el sector circular de radio r y ángulo ϕ determinado por el radar durante su rotación en el tiempo t . Un punto en una región plana R dada, se dice que se encuentra cubierto en el tiempo t si está dentro del rango de al menos uno de los n radares en el tiempo t . Se estudia el problema de la cobertura de R con un conjunto de radares que tienen la misma velocidad, sentido de rotación, ángulo ϕ y rango r .

Definición 19 *Ángulos $\Phi_r(P, R)$ y $\Phi(P, R)$. Sea P un conjunto de puntos en el plano y R una región plana. Sea $\Phi_r(P, R)$ el ínfimo sobre todos los ángulos $\phi \leq 2\pi$ tales que si los radares con ángulo ϕ y rango r se encuentran en los puntos $p \in P$ entonces, existen orientaciones iniciales de cada radar de tal manera que toda la región R se cubra en todo momento bajo la continua rotación de las antenas direccionales de los radares. Para el caso de reflectores con rango infinito usamos la notación $\Phi(P, R)$.*

Teorema 7 Para cualquier conjunto de puntos P con $n \geq 2$ reflectores en una línea L se tiene que $\Phi(P, L) = \frac{3\pi}{n}$.

Prueba. s.p.g., se supone que la línea L a cubrir es horizontal. Sea $P = p_0, p_1, \dots, p_{n-1}$ el conjunto de n radares en la línea L , tal que $p_i < p_{i+1}$ con respecto a su coordenada x , para $i = 0, 1, \dots, n-2$,

En primer lugar se prueba que un ángulo de $\frac{3\pi}{n}$ es siempre suficiente. Deje que la orientación inicial del radar p_i sea $F_{p_i}(\frac{i \cdot 3\pi}{n}; 0)$, para $i = 0, 1, \dots, n-1$, véase la figura 3.30.

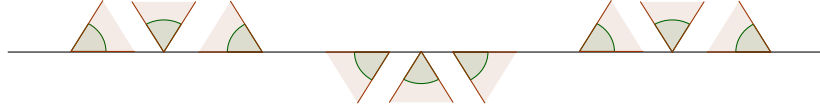


Figura 3.30: Orientación inicial de los radares en L .

Definimos el dual de un plano de la siguiente manera: cada radar i es un sector de un círculo unitario C delimitado por $\frac{i \cdot 3\pi}{n}$ y $\frac{(i+1) \cdot 3\pi}{n}$, en el tiempo t , la línea L se representa como un segmento de línea dirigida \vec{L} de tal manera que \vec{L} cruza el centro de C y la cabeza de \vec{L} forma un ángulo t con la horizontal; véase la figura 3.31 (a).

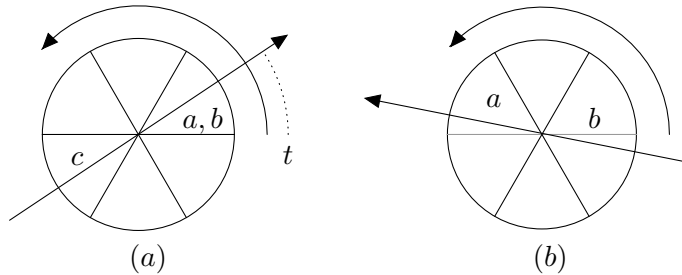


Figura 3.31: (a) Orientación en t . (b) Si $\Phi(n, L) < \frac{3\pi}{n}$, L no se encuentra en todo momento cubierto.

En el plano dual, los radares están estáticos mientras que la línea L rota en todo momento. La orientación \vec{L} de L preserva las rotaciones de los radares en el plano original. La punta de \vec{L} representa el ∞ y la contra-punta representa al $-\infty$ del plano original.

Dado que la suma de los ángulos es 3π , el sector circular de $[0, \pi)$ de C en el plano dual es siempre cubierto por dos conjuntos $S_1, S_2 \subseteq P$ de radares, mientras que el sector circular $[\pi, 2\pi)$ de C en el plano dual está cubierto por un conjunto $S_3 \subseteq P$ de radares. Observe que cada radar de S_3 está entre S_1 y S_2 en el plano original. Sean $a \in S_1, b \in S_2$ y $c \in S_3$ los radares que cubren un segmento de \vec{L} en el momento t en el plano dual. Si a y b cubren la punta de \vec{L} , c cubre la contra-punta. Por lo tanto, L está totalmente cubierto

por c y b en el plano original. Del mismo modo, si a y b cubren la contra-punta de \vec{L} , c cubre la punta. Por lo tanto, L está totalmente cubierta por a y c en el plano original.

Ahora demostraremos que un ángulo de $\frac{3\pi}{n}$ siempre es necesario. Supongamos por contradicción que la suma de los ángulos es menos de 3π . Por lo tanto, existe un tiempo t cuando sólo dos radares, por ejemplo a y b , cubren un segmento de \vec{L} en el plano dual como se muestra en la figura 3.31 (b). Supongamos que a cubre la contra-punta y b cubre la punta de \vec{L} en el plano dual. Entonces, L está totalmente cubierta en el plano original. Sin embargo, en el momento $t + \pi$, a cubre la punta y el b cubre la contra-punta de \vec{L} en el plano dual. Por lo tanto, el segmento de línea \overline{ab} de L en el plano original no está cubierto. Esto contradice la suposición. Esto completa la prueba del teorema. ■

Podemos observar que si L es finito, entonces es suficiente usar un rango igual a la longitud de L . Así, obtenemos el siguiente corolario del teorema 7 cuando L es finito.

Corolario 1 *Para cualquier conjunto de puntos P con $n \geq 2$ radares en una línea L con una longitud r tenemos que $\Phi_r(P, L) = \frac{3\pi}{n}$.*

Lema 10 *Sean a y b dos radares de ángulo ϕ en una línea horizontal. Supongamos que las orientaciones iniciales de las antenas de a , b son $F_a(\pi - \phi; 0)$, $F_b(\pi; 0)$, respectivamente. Además, asumimos que la coordenada x de a es menor que la coordenada x de b . Si $0 \leq t \leq \pi$, la intersección de los radares cubre un sector de 2ϕ . Si $\pi < t < 2\pi$, entonces dejan un corredor sin cubrir.*

Prueba. Sean l_a, l_b, r_a y r_b los rayos izquierdos y derechos que definen los haces de los radares en a y b respectivamente. Sea h la horizontal. En el momento t , $\angle(r_a, h) = \pi - t - \phi$, $\angle(l_a, h) = \pi - t$, $\angle(r_b, h) = \pi - t$ y $\angle(l_b, h) = \pi - t + \phi$. Observe que $\angle(r_a, r_b) = \phi$ y $\angle(l_a, l_b) = \phi$. Por lo tanto, cuando $\pi < t < 2\pi$, los rayos de r_a y l_b no se intersectan, es decir, $F_a(\pi; t) \cap F_b(\pi - \phi; t) = 0$ ya que la coordenada x de a es menor que la coordenada x de b y se forma un pasillo sin cubrir; vea la figura 3.32 (a). Sin embargo, cuando $0 \leq t \leq \pi$, r_a intersecta l_b ya que la coordenada x de a es menor que la coordenada x de b . Considere el punto de intersección x entre r_a y l_b ; vea la figura 3.32 (b). No es difícil ver que l_a y l_b determinan una cobertura incidente a x con un ángulo de 2ϕ . □

Teorema 8 *Sea P el plano entero. Tenemos que $\Phi(3, P) = \pi$.*

Prueba. Sea p, q, r tres radares en el plano. Si los radares son colineales entonces la configuración inicial se representa en la figura 3.33, se puede observar por el teorema 7 que la configuración es correcta.

Supongamos s.p.g., que los tres radares no están en la posición colineal. Además podemos suponer que el segmento \overline{pr} de línea es horizontal y q es superior a p y r . Sea C la

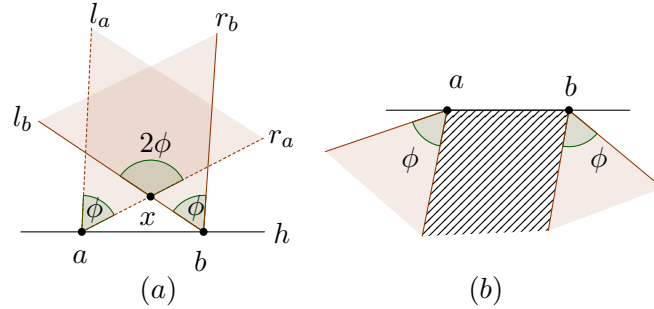


Figura 3.32: (a) $0 \leq t \leq \pi$. (b) $\pi < t < 2\pi$.

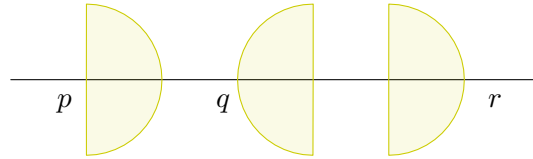


Figura 3.33: Configuración inicial para 3 radares en posición colineal.

circunferencia circunscrita de p, q, r . La orientación de p es $F_p(l; 0)$, donde l es la tangente de C en p, q como $F_q(\pi + \angle(qpr); 0)$ y r como $F_r(0; 0)$ como se muestra en la figura 3.34 (a). Considere cualquier punto a en la circunferencia de C de pqr . Observe que el ángulo que cada radar forma con a es igual a un arco; véase la figura 3.34 (b). Por lo tanto, se intersectan en a . Se puede observar que cuando a está en el arco pr , qr deja una parte no cubierta o un cono de sombra, sin embargo, p cubre el cono de sombra. Cuando está en el arco rq , los roles cambian a p, q y r , respectivamente, y cuando a se encuentra en el arco qp , los roles cambian para p, r y q respectivamente. Esto prueba el límite superior si los puntos no son colineales.

Supongamos ahora que p, q, r son colineales. s.p.g. suponemos que están en una línea horizontal y la coordenada x de q es mayor que la coordenada x de p y esta a su vez es menor que la coordenada x de la r . Orientamos a p, q, r como $F_p(0, 0)$, $F_q(\pi, 0)$ y $F_r(0, 0)$. Por el lema 10, p y q cubren el plano en el tiempo $t < \pi$ q y r cubren el plano en el momento $\pi \leq t < 2\pi$.

Para probar que la cota es justa, asumiremos por contradicción que $\Phi(3, P) = \pi - \epsilon$. Supongamos que en el instante t los radares cubren el plano. Por lo tanto, existe un punto a en el área que está cubriendo p en el cual se intersectan q y r como se muestra en la figura 3.34 (c). Sin embargo, a no está cubierto en el tiempo $t + \pi$ ya que $\Phi(3, P) = \pi - \epsilon$.

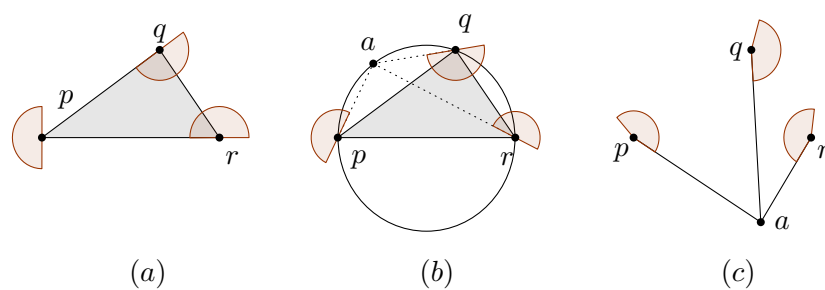


Figura 3.34: (a) Tres radares cubriendo el plano. (b) Tres radares cubriendo el plano. (c) El plano no siempre se mantiene cubierto si $\Phi(3, P) = \pi - \epsilon$.

Capítulo 4

Teoría de la complejidad computacional

4.1. Introducción

En el capítulo anterior mencionamos que el problema de vigilancia de galerías de arte es un problema NP, para poder comprender que es un problema de esta clase, primero debemos entender algunos conceptos básicos sobre la teoría de la complejidad.

4.2. Problemas de decisión

¿Qué un problema de decisión? Es uno en donde las respuestas posibles son “sí” o “no”. Un ejemplo típico es la pregunta: ¿Es primo un número entero dado? Y una instancia de este problema sería: ¿Es primo el número 17? Para definirlo formalmente primero debemos saber a qué llamaremos problema:

Definición 20 *Un problema será una expresión que: (1) presente una lista de parámetros, detallando sus características; (2) enuncie una propiedad que deban cumplir dichos parámetros. Una instancia del problema se obtendrá especificando valores particulares sobre todos los parámetros. Dada una instancia, el problema en sí consistirá en decidir si los parámetros del caso cumplen o no con la propiedad dada por (2).*

Una vez definido lo que es un problema podemos formalizar ¿qué es un problema de decisión?

Definición 21 *Diremos que un problema Π es de decisión si el conjunto de todas sus instancias, D_{Π} , contiene un subconjunto distinguido Y_{Π} , de instancias que satisfagan la*

propiedad mencionada en la definición anterior (a las que llamaremos instancias positivas), de manera tal que el problema consiste en, dada una instancia I de Π , decidir si pertenece o no a Y_Π .

4.3. SAT

Un ejemplo clásico, y como veremos más adelante fundamental, de problema de decisión es el siguiente:

La satisfacibilidad (por abreviación SAT) es un problema de decisión de lógica booleana y los términos utilizados para describirlo son definidos a continuación:

Sea $X = \{x_1, x_2, \dots, x_n\}$ el conjunto de variables booleanas. Una asignación de verdad para X es una función $f : X \rightarrow \{V, F\}$. Si $f(x_i) = V$ decimos que x_i es verdadera, de lo contrario, si $f(x_i) = F$ decimos que x_i es falsa. Si x es una variable en X , entonces x y \bar{x} son literales de X . La literal x es verdadera según f si y sólo si la variable x es verdadera según f , y la literal \bar{x} es verdadera si y sólo si la variable x es falsa.

Una cláusula conforme con X es un conjunto de literales de X , tal como $\{x_1, \bar{x}_3, x_8\}$. Esto representa la disyunción de las literales y es satisfacible por una asignación de verdad si y sólo si al menos uno de sus miembros es verdadero bajo esa asignación, la cláusula se verá satisfecha por f a menos que $f(x_1) = F, f(x_3) = T$ y $f(x_8) = F$. Una colección C de cláusulas dada por X es satisfacible si y sólo si, existe alguna asignación para X que simultáneamente satisfaga todas las cláusulas en C . Tal asignación de verdad, es llamada una asignación de verdad satisfactoria para C . El problema de satisfacibilidad se especifica de la siguiente manera:

Instancia: Un conjunto X de variables y una colección C de cláusulas sobre X . Pregunta: ¿Existe una asignación de verdad satisfactoria para C ?

Por ejemplo para $X = \{x_1, x_2, x_3, x_4\}$:

$$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee \bar{x}_4)$$

admite una asignación de verdad satisfactoria dando valores: $x_1 = V, x_2 = V, x_4 = V$ y $x_3 = F$.

Por otro lado, esta otra expresión:

$$(\bar{x}_1) \wedge (x_1 \vee x_2) \wedge (\bar{x}_2 \vee x_3) \wedge (\bar{x}_3 \vee x_4) \wedge (x_1 \vee \bar{x}_4)$$

no admite ninguna asignación. Es imposible se satisfacerla. Para verificarlo, intentamos fabricar una asignación que haga verdadera a la expresión: la primera cláusula nos fuerza

a definir $x_1 = F$. Y esto, a su vez, nos obliga a definir sucesivamente $x_2 = V, x_3 = V$ y $x_4 = V$. Pero bajo esta asignación, la última cláusula resulta falsa, y por lo tanto también resulta falsa la expresión.

4.4. Lenguajes y esquemas de codificación

Dado un conjunto de símbolos Σ , nuestro alfabeto, denotaremos por Σ^* al conjunto de todas las posibles cadenas (palabras) que puedan formarse con los elementos de Σ , incluyendo la cadena vacía. Por ejemplo si, los símbolos admitidos son sólo el 0 y el 1, las posibles palabras serán entonces todas las cadenas de ceros y unos. Estas cadenas pueden representar números expresados en binario, o caracteres, según alguna tabla de codificación como la ASCII.

Un *lenguaje* sobre Σ será cualquier subconjunto \mathcal{L} de Σ^* . Observemos que cualquier información o programa que se cargue en una computadora es un lenguaje en $\{0, 1\}^*$.

Un *esquema de codificación* sirve para relacionar los problemas de decisión y los lenguajes. Un esquema de codificación e para un problema Π provee una forma de describir cada instancia de Π a cadenas de símbolos sobre algún alfabeto Σ . De esta manera el problema Π y el esquema de codificación e para Π , particiona a Σ^* en tres clases de cadenas: aquellas instancias que no se encuentran codificadas para Π , aquellas instancias que se encuentran codificadas para Π , las cuales son instancias positivas, y aquellas instancias que se encuentran codificadas para Π , las cuales son instancias negativas. La segunda clase de cadenas es el lenguaje que se asocia con Π y e , llamaremos *lenguaje asociado al problema* Π , al conjunto de las representaciones de las instancias positivas de Π .

$$\mathcal{L}(\Pi, e) = \{x \in \Sigma^* : \Sigma \text{ es el alfabeto usado por } e \text{ y } x \text{ se codifica según } e \text{ a una instancia } I \in Y_{\Pi}\}$$

Puesto que la función de un esquema de codificación, es especificar la forma en la que se deben interpretar los datos y las instrucciones (tengamos en cuenta que en el idioma básico de una computadora toda la información está dada por cadenas de unos y ceros), el tamaño de una instancia, es decir el espacio que ocuparán los valores de los parámetros que la definen, dependerá del esquema de codificación que se utilice. Podemos asumir, sin embargo, que bajo cualquier esquema razonable el tamaño de una instancia se mantendrá controlado. Más específicamente, si según los criterios de un esquema e_1 , una instancia I ocupa n bytes, bajo las especificaciones de otro esquema e_2 , I no debería ocupar más que n bytes. Esto último puede formalizarse de la siguiente manera: si el tamaño de I bajo e_1 es n , entonces existirá un polinomial p de manera que el tamaño de I bajo cualquier otro esquema razonable e_2 sea menor o igual que $p(n)$. El concepto de esquema razonable es algo esquivo. De hecho, no conocemos una definición precisa.

Sin embargo, convendremos en que un esquema razonable es aquel que, por un lado se limita a retener la información esencial de una instancia, sin agregar datos superfluos y, además, lo hace de manera que posibilite la decodificación de la información almacenada, es decir: que para cada componente de una instancia admite un algoritmo polinomial que la interpreta. Puede verse un ejemplo detallado en [3], página 21.

Si admitimos esto último, la eficiencia de un algoritmo, en términos de polinomialidad o exponencialidad, será independiente del esquema utilizado. De este modo, el lenguaje asociado a un problema Π puede pensarse como dependiendo exclusivamente del problema, y no del esquema. Cada problema quedará entonces asociado directamente a un lenguaje y en lugar de $\mathcal{L}(\Pi, e)$ se escribirá simplemente $\mathcal{L}(\Pi)$.

4.5. Algoritmos de tiempo polinomial

Diferentes algoritmos poseen una gran variedad de diferentes funciones de complejidad de tiempo. Y la caracterización de cuales de ellos son lo suficientemente eficientes y la de los que no son tan eficientes, siempre dependerá de la situación en cuestión. Sin embargo, los científicos computacionales reconocen una simple distinción que ofrece una considerable comprensión de este tema. A continuación se presenta esta distinción entre algoritmos de tiempo polinomial y algoritmos de tiempo exponencial.

Una función $f(n)$ es $O(g(n))$ cuando existe una constante c tal que $|f(n)| \leq c \cdot |g(n)|$ para todos los valores de $n \geq n_0$. Se dice que un algoritmo es de tiempo polinomial si cuya función de complejidad de tiempo es $O(p(n))$ para alguna función polinomial p , donde n es usada para denotar la longitud de la entrada. Cualquier algoritmo cuya función de complejidad de tiempo no pueda ser acotada, es llamado un algoritmo de tiempo exponencial (aunque hay que señalar que esta definición incluyen ciertas funciones de complejidad de tiempo no polinomial, como $n^{\log n}$, las cuales no son normalmente consideradas como funciones exponenciales).

4.6. Introducción a las máquinas de Turing

La teoría de la complejidad computacional es la parte de la teoría de la computación, que aborda el estudio formal de los recursos requeridos durante el cálculo para resolver un problema (algorítmico) dado, es decir, estudia la manera de clasificar los problemas de acuerdo a la dificultad inherente de resolverlos. De esta manera se definen varias clases de complejidad (\mathcal{P} , \mathcal{NP} , ...) pudiéndose entonces clasificar a los algoritmos según sus características.

En la siguiente sección introduciremos el concepto de lo que es una máquina de Turing, que está muy relacionada con los conceptos de complejidad y computabilidad.

4.6.1. Máquinas de Turing

Alan Turing desarrolló un modelo matemático simple conocido con el nombre de máquina de Turing; este modelo matemático ha servido como base para poder expresar en términos sencillos lo que un mecanismo de computación puede o no hacer. Es importante entonces revisar el funcionamiento de este modelo matemático que parece ser sencillo y a la vez muy poderoso.

Se definen diferentes máquinas de Turing de acuerdo a la manera en que trabajan, existen máquinas de Turing deterministas y no deterministas, cada una de éstas está orientada a resolver un tipo especial de problemas.

4.6.2. Máquinas de Turing deterministas y problemas \mathcal{P}

El modelo que utilizamos para formalizar la noción de computadora es el de una máquina de Turing determinística (MTD).

Esta máquina de Turing consta de tres partes, esquematizadas en la figura 4.1:

- Una lista finita de instrucciones (algoritmo) y un controlador de estados (que nos indica que instrucción se está ejecutando).
- Una cinta semi-infinita (o infinita hacia un lado) dividida en celdas.
- Una cabeza o lector/grabador de las celdas.

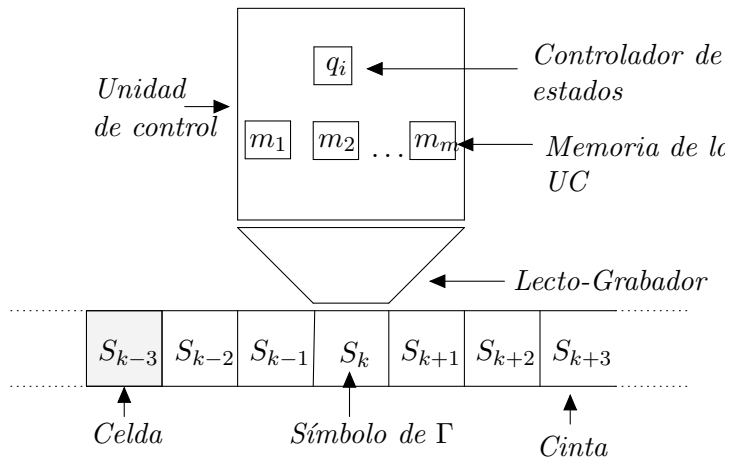


Figura 4.1: Máquina de Turing determinista.

Un programa o algoritmo para una máquina de Turing determinista M queda dado de la siguiente manera: $M = (Q, \Sigma, \Gamma, \delta, q_0, b, F)$, donde:

- Q : es un conjunto finito de estados; cada uno de los elementos indica un estado en el que puede estar el control finito.
- Γ : es un conjunto finito de símbolos de cinta permitidos.
- b : un símbolo $b \in \Gamma$, que representa un espacio en blanco.
- q_0 : un estado inicial $q_0 \in Q$, es el estado distinguido en el que arranca la máquina de Turing.
- Σ : un alfabeto de entrada $\Sigma \subseteq \Gamma$. No contiene a b . Tales símbolos son usados en la codificación de la cadena de entrada o *input*.
- F : es un conjunto finito de estados $F \subseteq Q$, los estados finales o de paro de la máquina de Turing.
- δ : es la función de transición $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{Izq, Der\}$, donde $\{Izq, Der\}$ denotan la dirección del movimiento de la cabeza lectora en la cinta, pudiendo ser ésta respectivamente hacia la izquierda o a la derecha. Esta función representa las distintas instrucciones del programa. La asignación $\delta(q, \alpha) = (q', \beta, \Delta)$, donde $q, q' \in Q$ y $\alpha, \beta \in \Gamma$, indica que si el controlador de estados se encuentra en q y el lector-grabador apuntando a un casillero con el símbolo α , entonces el programa deberá: borrar α para escribir en su lugar el símbolo β , moverse sobre la cinta una celda hacia la derecha o hacia la izquierda según sea $\Delta = Izq$ o $\Delta = Der$, y finalmente ubicar el controlador de estados en q' . Cuando el controlador de estados se encuentra en algún estado final el programa puede terminar.

Para nosotros la entrada del programa será una instancia de un problema de decisión, entonces tenemos dos únicos estados terminales de F : q_Y y q_N . Inicialmente, la cinta tendrá el símbolo blanco b en todas sus celdas. El primer paso de todo programa será cargar la cadena de símbolos que representen a la instancia dada en alguna parte de la cinta. Luego el lector-grabador se ubicará en una celda inicial y el controlador de estados en q_0 . A partir de allí se comportará según las indicaciones de la función δ . Si una instancia I queda representada por la cadena $x \in \Sigma^*$, diremos que el tamaño de x (o, eventualmente, de I) es n si x tiene n símbolos y lo denotaremos como $|x| = n$.

Dado un programa M sobre una máquina de Turing determinística, y $x \in \Sigma^*$, diremos que M acepta a x si, al ser ingresado x como entrada, M alcanza el estado q_Y . El *lenguaje reconocido* por M , $\mathcal{L}_M \subset \Sigma^*$, será el formado por todas las cadenas x que son aceptadas por M :

$$\mathcal{L}_M = \{x \in \Sigma^* : M \text{ acepta } x\}$$

El tiempo usado en el cálculo de un programa M sobre una MTD con una entrada x , es el número de pasos que ocurren en ese cálculo hasta que se alcanza un estado terminal. Para un programa M sobre una MTD que alcanza un estado final para todas sus entradas $x \in \Sigma^*$, su función de complejidad de tiempo $T_M : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ está dada por:

$$T_M(n) = \max \left\{ \begin{array}{l} m : \text{Existe una } x \in \Sigma^*, \text{ con } |x| = n \text{ tal que el} \\ \text{cálculo de } M \text{ para una entrada } x \text{ toma } m \text{ pasos} \end{array} \right\}$$

Un programa M de este tipo se dice que es un programa sobre una MTD de tiempo polinomial si existe un polinomio p tal que, para toda $n \in \mathbb{Z}^+$, $T_M(n) \leq p(n)$.

Ahora daremos una definición de la clase de lenguajes que será considerada como la clase \mathcal{P} :

$$\mathcal{P} = \{\mathcal{L} : \text{existe un programa } M \text{ sobre una MTD de tiempo polinomial tal que } \mathcal{L} = \mathcal{L}_M\}$$

Diremos que un problema Π pertenece a la clase de complejidad \mathcal{P} si existe algún esquema de codificación e tal que el lenguaje asociado a Π , $\mathcal{L}(\Pi, e)$ pertenece a \mathcal{P} . Observemos que como $\mathcal{L}(\Pi, e) = \mathcal{L}(\Pi)$, esto nos permite independizarnos del esquema. Puesto que los tamaños de una instancia sobre distintos esquemas razonables están vinculados polinomialmente, si para algún esquema “razonable” e , $\mathcal{L}(\Pi, e) \in \mathcal{P}$, entonces $\mathcal{L}(\Pi, e') \in \mathcal{P}$ para cualquier otro esquema “razonable” e' .

Por su definición, la clase \mathcal{P} corresponde a los problemas de decisión para los cuales existen algoritmos polinomiales. El objetivo de la siguiente sección es definir una clase análoga para problemas a los que no se conocen algoritmos polinomiales que los resuelvan.

En otras palabras un problema es polinomial, o bien, pertenece a la clase \mathcal{P} si existe un algoritmo determinístico polinomial que lo resuelva.

4.6.3. Máquina de Turing no determinística y problemas \mathcal{NP}

Consideremos el problema SAT antes definido. Una instancia I de SAT será una cadena de cláusulas sobre el conjunto de variables $\{x_1, \dots, x_k\}$. Hasta ahora, no se conoce ningún algoritmo polinomial capaz de decidir (para cualquier I) si existe alguna asignación valores para las variables que satisfaga I . Observemos, sin embargo, que, dada una asignación de valores, es muy sencillo verificar si ésta hace la expresión verdadera o falsa: basta reemplazar cada variable por el valor correspondiente y realizar las operaciones binarias necesarias. En particular, si se tiene una instancia positiva I de SAT y una asignación de valores que hace verdadera la expresión dada por I , existe un algoritmo polinomial que verifica este hecho, corroborando que I es realmente positiva.

Esta propiedad de verificabilidad en tiempo polinomial podría generalizarse del siguiente modo: dado un problema Π y una instancia $I \in Y_{\Pi}$, existe una estructura E (una asignación de valores en SAT, un ciclo en TSP (*Traveling Salesman Problem*), etc.), y un algoritmo polinomial que, a partir de E , corrobora que efectivamente $I \in Y_{\Pi}$. Gran cantidad de problemas de decisión cumplen con esta propiedad. La clase \mathcal{NP} será definida a través de una formalización de esta idea. Para llevar a cabo esta formulación definiremos más formalmente la máquina de Turing no determinística (MTND).

Dada una entrada x , podemos pensar en la máquina de Turing no determinista M como la ejecución de todas las secuencias de movimientos en paralelo hasta que la instancia sea positiva o no haya más movimientos posibles. Es decir, después de i movimientos podemos pensar que existen un número de “copias” de M . Cada copia representa una instancia en la que M podría estar después de i movimientos. En el movimiento $(i + 1)$ una C se replica asimismo j veces, es decir se hacen j copias de C , si la máquina de Turing tiene j elecciones para el siguiente movimiento.

De esta manera, las posibles secuencias de movimientos que M puede hacer con la entrada x pueden ser organizadas en un árbol de instancias, en donde cada camino desde la raíz hasta una hoja representa una secuencia de movimientos posibles. Entonces un estructura E es la secuencia de movimientos más pequeña que termina en satisfacer a la instancia. Si existe E , entonces M realizó $|E|$ movimientos, M se detiene y acepta la entrada x . El tiempo que tomó en procesar a x es la longitud de E . Si para la entrada x no existe una E , entonces M rechaza a x y el tiempo de procesamiento de x se deja indefinido.

A menudo es conveniente pensar que M “adivina” solamente la secuencia de movimientos E y después verifica que E ciertamente satisfaga la instancia.

Una máquina de Turing no determinística es una construcción formal similar a la de la máquina de Turing determinística. Tiene, como ésta última, una cinta semi-infinita dividida en celdas, un dispositivo controlador de estados y un lecto-grabador asociado a él. A éstos elementos se agregan, como se observa en la figura 4.2:

1. Un dispositivo adivinador cuya función es “adivinar” la estructura E que hace positiva a una instancia dada; y
2. Un nuevo lecto-grabador, asociado al dispositivo de adivinación.

Un programa M sobre una MTND constará de los mismos elementos que formaban un programa en una máquina determinística. Como antes, el primer paso del programa será cargar los datos de una instancia I en las primeras celdas de la cinta. Ahora bien, una vez ingresada la instancia, el programa pasará por dos fases sucesivas claramente distinguidas:

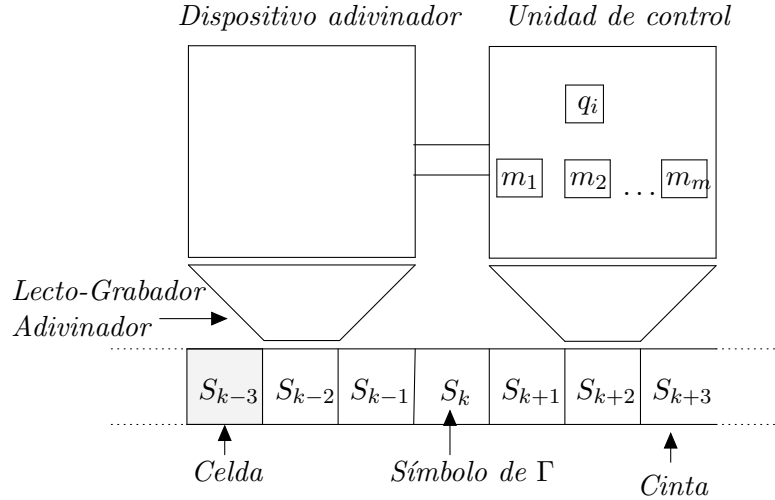


Figura 4.2: Máquina de Turing no determinista.

1. Aquí el controlador de estados y su lecto-grabador permanecen inactivos, mientras entra en juego el dispositivo de adivinación. Éste último adivina una estructura E que su correspondiente lecto-grabador escribe en la cinta. Una vez finalizado este proceso, el adivinador y su lecto-grabador quedan inactivos.
2. Esta fase es idéntica a un programa sobre una máquina determinística, cuya función es verificar, a partir de la estructura E adivinada en la fase anterior y guardada en la cinta, que la instancia almacenada este en Y_{Π} .

Diremos que un programa M sobre una MTND acepta a una cadena $x \in \Sigma^*$ si para alguna de las cadenas en Γ^* susceptibles de ser adivinadas por el dispositivo de adivinación, M alcanza en un tiempo finito el estado q_Y al serle ingresada x como entrada. Nuevamente, el lenguaje reconocido por M es:

$$\mathcal{L}_M = \{x \in \Sigma^* : M \text{ acepta } x\}$$

Definiremos el tiempo que necesita M para aceptar x como el mínimo número de operaciones que debe realizar la fase determinística de M antes de llegar a q_Y , tomado sobre todas las posibles estructuras que, al ser adivinadas en la primera fase, llevan a este resultado positivo. Consecuentemente, la *función de complejidad* de M vendrá dada por:

$$\mathcal{T}_M(n) = \max \left\{ \{1\}; \left\{ m : \exists x \in \mathcal{L}_M, |x| = n \begin{array}{l} \text{el tiempo necesario} \\ \text{para que } M \text{ acepte a } x \text{ es } m \end{array} \right\} \right\}$$

Notemos que la función de complejidad de tiempo para M depende únicamente en el número de pasos que ocurren en los cálculos para comprobar que la instancia es positiva

y que por convención $\mathcal{T}_M(n)$ se establece igual a 1 cuando no hay entradas de longitud n que sean aceptadas por M .

De acuerdo con esta definición diremos que un programa M es polinomial sobre una máquina de Turing no determinística si existe un polinomio p tal que $\mathcal{T}_M = \mathcal{O}(p(n))$.

Observemos que, dado un problema Π , la existencia de un programa M polinomial sobre una MTND que lo resuelva es equivalente a que Π satisfaga la propiedad de verificabilidad polinomial de la que hablamos arriba. De este modo, de manera análoga a lo hecho en la sección anterior, definiremos:

$$\mathcal{NP} = \{\mathcal{L} : \exists \text{ un programa polinomial } M \text{ sobre una MTND tal que } \mathcal{L}_M = \mathcal{L}\}$$

Nuevamente, aprovechando la asociación entre problemas y lenguajes antes establecida, nos permitiremos un abuso de notación y diremos que un problema Π pertenece a \mathcal{NP} , si $\mathcal{L}(\Pi) \in \mathcal{NP}$.

Con estas definiciones es inmediato que $\mathcal{P} \subseteq \mathcal{NP}$. En efecto, si existe un programa M sobre una MTD que resuelve un problema Π en tiempo polinomial, entonces puede construirse un programa sobre una MTND que verifique que una instancia dada de Π es positiva. Para ello basta con poner M como la segunda parte del programa, ignorando el resultado de la fase adivinatoria. No queda para nada clara, la inclusión opuesta. De hecho, como dijimos anteriormente, se cree que algunos problemas, como SAT y TSP, son intrínsecamente intratables. La traducción de esta aseveración al lenguaje de clases que hemos introducido es ni más ni menos que: $\mathcal{P} \not\subseteq \mathcal{NP}$. Esta es una de las conjeturas abiertas más importantes de la matemática moderna.

4.6.4. Reducción polinomial

El concepto de reducción polinomial formaliza la idea de que un problema Π_1 es más difícil que otro problema Π_2 ; o, mejor dicho: Π_1 es al menos tan difícil como Π_2 . Puesto que nuestra noción formal de algoritmo es la de máquina de Turing determinista, y que ésta opera sobre lenguajes y no sobre problemas, la definición de reducción polinomial se dará primero para aquellos. Posteriormente, la asociación entre lenguajes y problemas nos permitirá aplicarla directamente a éstos últimos.

Definición 22 Diremos que un lenguaje $\mathcal{L}_1 \subseteq \Sigma_1^*$ se reduce polinomialmente a otro lenguaje $\mathcal{L}_2 \subseteq \Sigma_2^*$, si existe una función $f : \Sigma_1^* \rightarrow \Sigma_2^*$, computable por una máquina de Turing determinística en tiempo polinomial, de manera que para toda cadena $x \in \Sigma_1^*$, $x \in \mathcal{L}_1$ si y sólo si la cadena $f(x) \in \mathcal{L}_2$.

Si existe una reducción polinomial lo denotaremos como $\mathcal{L}_1 \leq_p \mathcal{L}_2$ y se lee \mathcal{L}_1 se reduce a \mathcal{L}_2 .

El significado de reducción polinomial es dado por el siguiente lema:

Lema 11 *Si $\mathcal{L}_1 \leq_p \mathcal{L}_2$ entonces $\mathcal{L}_2 \in \mathcal{P}$ implica $\mathcal{L}_1 \in \mathcal{P}$.*

Prueba. Supongamos que $\mathcal{L}_1 \subseteq \Sigma_1^*$ y $\mathcal{L}_2 \subseteq \Sigma_2^*$. Si $\mathcal{L}_1 \in \mathcal{P}$, podemos construir una MTD M_2 que reconoce a \mathcal{L}_2 en tiempo polinomial. Llamemos p_2 al polinomio que acota el tiempo que necesita M_2 para efectuar este reconocimiento. Así mismo, sean M_f el programa polinomial que calcula la función $f : \Sigma_1^* \rightarrow \Sigma_2^*$, y p_f el polinomio que acota el tiempo de este cálculo. Por otra parte, $|f(x)| \leq p_f(|x|)$.

Ahora bien, el programa M_1 definido sobre Σ_1^* que consiste en aplicar primero f mediante M_f y luego M_2 es un programa que reconoce a \mathcal{L}_1 , puesto que, por definición de reducción polinomial, $x \in \mathcal{L}_1 \Leftrightarrow f(x) \in \mathcal{L}_2$. Además, el tiempo que necesita M_1 para reconocer una cadena $x \in \mathcal{L}_1$ es menor o igual que $p_f(|x|) + p_2(p_f(|x|))$, con lo cual existe un polinomio $(p_f + p_2 \circ p_f)$ que acota la complejidad de M_1 , es decir: $\mathcal{L}_1 \in \mathcal{P}$. \square

Corolario 2 *La composición de funciones computables polinomialmente en una MTD es polinomial.*

Nuestra intención es que \leq_p nos permita comparar problemas para poder afirmar que un problema es tan o más difícil que otro. Es razonable esperar, entonces, que \leq_p defina un orden parcial sobre \mathcal{NP} . Podemos ver que $\mathcal{L} \leq_p \mathcal{L}, \forall \mathcal{L} \in \mathcal{NP}$, por lo cual \leq_p resulta reflexiva. El siguiente lema demuestra que es también transitiva.

Lema 12 *Si $\mathcal{L}_1 \leq_p \mathcal{L}_2$ y $\mathcal{L}_2 \leq_p \mathcal{L}_3$, entonces $\mathcal{L}_1 \leq_p \mathcal{L}_3$.*

Prueba. Sean Σ_1, Σ_2 y Σ_3 los alfabetos sobre los que se definen $\mathcal{L}_1, \mathcal{L}_2$ y \mathcal{L}_3 respectivamente. Como $\mathcal{L}_1 \leq_p \mathcal{L}_2$ existe una transformación polinomial $f_1 : \Sigma_1^* \rightarrow \Sigma_2^*$. Análogamente, existe una $f_2 : \Sigma_2^* \rightarrow \Sigma_3^*$. Definimos $f : \Sigma_1^* \rightarrow \Sigma_3^*$, $f = f_2 \circ f_1$. Dado que $x \in \Sigma_1$ tenemos que: $x \in \mathcal{L}_1 \Leftrightarrow f_1(x) \in \mathcal{L}_2 \Leftrightarrow f(x) = f_2(f_1(x)) \in \mathcal{L}_3$. Por otra parte, por el lema 11 f resulta polinomial. \square

Si Π_1 y Π_2 son problemas de decisión, con esquemas de codificación asociados e_1 y e_2 , podemos escribir $\Pi_1 \leq_p \Pi_2$ (con sus respectivos esquemas de codificación) siempre que exista una transformación polinomial de $\mathcal{L}[\Pi_1, e_1]$ a $\mathcal{L}[\Pi_2, e_2]$. Se omitirán los esquemas de codificación asumiendo que se están utilizando esquemas de codificación razonables. De esta manera, podemos considerar una reducción polinomial para el problema de decisión Π_1 , a el problema de decisión Π_2 como una función $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$ que satisface las siguientes dos condiciones:

1. f es computable por un algoritmo de tiempo polinomial; y

2. para todas las $I \in D_{\Pi_1}$, $I \in Y_{\Pi_1}$ si y sólo si $f(I) \in Y_{\Pi_2}$.

Veamos en un ejemplo cómo demostrar en la práctica una reducción polinomial. Para ello definamos:

3-SAT: Designamos con este nombre al caso particular del problema SAT en que todas las cláusulas involucradas en la expresión, contienen exactamente 3 literales.

Proposición 1 $SAT \leq_p 3\text{-SAT}$

Prueba. Consideremos una instancia I de SAT dada por la expresión:

$$\xi = \prod_{i=1}^N C_i$$

donde las C_i , para $i = 1, \dots, N$ son cláusulas sobre el conjunto de variables $X = \{x_1, \dots, x_n\}$. Para demostrar la proposición deberemos mostrar una instancia I' de 3-SAT dada por una expresión ζ sobre un conjunto de variables X' , que pueda calcularse en tiempo polinomial a partir de ξ y que cumpla con la propiedad de que ξ sea satisfecha por una asignación de valores para las variables de X si y sólo si lo propio sucede con ζ sobre X' .

La demostración es constructiva: examinaremos las cláusulas de ξ y definiremos para cada una de ellas una secuencia ζ_i de cláusulas de tres literales. La concatenación de las ζ_i formará ζ . Para referirnos a las literales utilizamos la letra λ . Dada una C_i puede darse cualquiera de los siguientes casos:

1. Si C_i tiene exactamente tres literales, no hay nada que hacer. Definimos simplemente $\zeta_i = C_i$.
2. Si C_i tiene más de tres literales: $C_i = (\lambda_1 \vee \lambda_2 \vee \dots \vee \lambda_k), k > 3$, definimos nuevas variables y_1, y_2, \dots, y_{k-3} y:

$$\zeta_i = (\lambda_1 \vee \lambda_2 \vee y_1) \wedge (\overline{y_1} \vee \lambda_3 \vee y_2) \wedge (\overline{y_2} \vee \lambda_4 \vee y_3) \wedge \dots \wedge (\overline{y_{k-3}} \vee \lambda_{k-1} \vee \lambda_k)$$

Queda claro que C_i es verdadera si y sólo si ζ_i lo es: una asignación de valores que haga a C_i verdadera, deberá hacer alguno de las $\lambda_{i_j} = V$; sustituyendo ese valor en ζ_i pueden tomarse valores para las nuevas variables de modo que ζ_i sea verdadera. Inversamente, si ζ_i es verdadera, alguna de las x_{i_j} deberá ser V , con lo cual C_i es verdadera. Por otro lado, observemos que $k \leq n$ (si $k > n$, C_i resulta trivialmente verdadera y puede descartarse), de modo que la cantidad de variables auxiliares y y, por lo tanto, la longitud de ξ'_i queda acotada por un polinomio en el número de variables n .

3. Si C_i involucra menos de 3 literales entonces:

- si $C_i = (\lambda)$ definiremos variables auxiliares z, w, u, v y:

$$\zeta_i = (\lambda \vee z \vee w) \wedge (\bar{z} \vee u \vee v) \wedge (\bar{z} \vee \bar{u} \vee v) \wedge (\bar{z} \vee u \vee \bar{v}) \wedge (\bar{z} \vee \bar{u} \vee \bar{v}) \wedge (\bar{w} \vee u \vee v) \wedge (\bar{w} \vee \bar{u} \vee v) \wedge (\bar{w} \vee u \vee \bar{v}) \wedge (\bar{w} \vee \bar{u} \vee \bar{v})$$

- si $C_i = (\lambda_1 \vee \lambda_2)$ definimos variables auxiliares z, u, v y:

$$\zeta_i = (\lambda \vee z \vee w) \wedge (\bar{z} \vee u \vee v) \wedge (\bar{z} \vee \bar{u} \vee v) \wedge (\bar{z} \vee u \vee \bar{v}) \wedge (\bar{z} \vee \bar{u} \vee \bar{v})$$

La construcción de las cláusulas artificiales fuerza a z y w (o a z , en el segundo caso) a ser falsas, con lo cual ζ_i será verdadera si y sólo si C_i es verdadera. Nuevamente, la longitud de ζ_i está acotada por un polinomio en n .

Finalmente:

$$\zeta = \prod_{i=1}^N \zeta_i$$

define una instancia I' de 3-SAT, sobre el conjunto de variables X' formado por X y las variables auxiliares que hayan sido necesarias, que resulta positiva si y sólo si I era una instancia que satisface SAT.

4.6.5. Problemas \mathcal{NP} -Completo. Teorema de Cook

La reducción polinomial nos permite definir en \mathcal{NP} la relación \sim dada por:

$$\mathcal{L}_1 \sim \mathcal{L}_2 \Leftrightarrow \mathcal{L}_1 \leq_p \mathcal{L}_2 \text{ y } \mathcal{L}_2 \leq_p \mathcal{L}_1$$

De este modo \sim es simétrica por definición, mientras que hereda la reflexividad y la transitividad de \leq_p (ver lema 12). La relación \sim define entonces una relación de equivalencia en \mathcal{NP} . Diremos que \mathcal{L}_1 es *polinomialmente equivalente* a \mathcal{L}_2 si $\mathcal{L}_1 \sim \mathcal{L}_2$. Como siempre, extenderemos nuestra definición diciendo que Π_1 es polinomialmente equivalente a Π_2 si $\mathcal{L}(\Pi_1) \sim \mathcal{L}(\Pi_2)$.

Observemos que si dos problemas son polinomialmente equivalentes la existencia de un algoritmo eficiente que resuelva cualquiera de ellos implica automáticamente la existencia de un algoritmo eficiente para el otro.

Definición 23 *Un lenguaje $\mathcal{L} \in \mathcal{NP}$ es \mathcal{NP} -Completo cuando para todo otro lenguaje $\mathcal{L}' \in \mathcal{NP}$ se tiene que $\mathcal{L}' \leq_p \mathcal{L}$. Más informalmente, $\Pi \in \mathcal{NP}$ es \mathcal{NP} -Completo si para todo otro problema Π' en \mathcal{NP} se tiene que $\Pi' \leq_p \Pi$.*

Dado que la reducción polinomial nos da una medida comparativa de la dificultad de los problemas, los problemas \mathcal{NP} -Completos serán, por definición, los problemas más difíciles en \mathcal{NP} . La clase de los problemas \mathcal{NP} -Completos es, la clase de los problemas considerados intratables. Al ser todos polinomialmente equivalentes entre sí, la resolución mediante un algoritmo eficiente de cualquiera de ellos traería la inmediata resolución de todos los problemas \mathcal{NP} . Es decir, si para algún $\Pi \in \mathcal{NP}$ -Completo, se tuviese que $\Pi \in \mathcal{P}$, entonces resultaría que $\mathcal{P} = \mathcal{NP}$. Como hemos remarcado anteriormente, la resistencia que han mostrado algunos problemas ha llevado a pensar que $\mathcal{P} \neq \mathcal{NP}$. Esta sospecha obliga a suponer que la geografía de la clase \mathcal{NP} es como la que se describe en la figura 4.3.

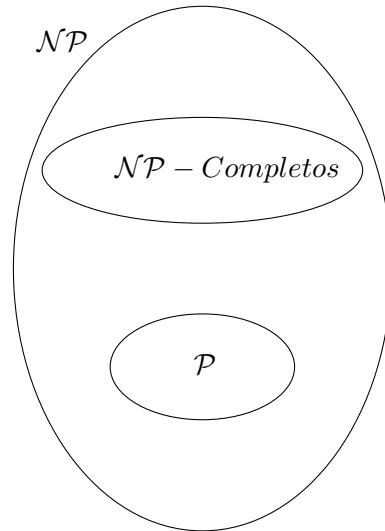


Figura 4.3: Probable descripción de \mathcal{NP} .

La dificultad que han representado los problemas \mathcal{NP} -Completos (e.g.: SAT, TSP, etc.) en comparación con aquellos que se sabe positivamente que pertenecen a \mathcal{P} , sumada a la equivalencia que acabamos de exponer entre todos los \mathcal{NP} -Completos obliga a abrigar pocas esperanzas en cuanto a la posibilidad de hallar algoritmos eficientes para los problemas más difíciles. Sin embargo, mientras no se conozca una demostración de que $\mathcal{P} \neq \mathcal{NP}$ (o, eventualmente, de lo contrario), no quedará más remedio que navegar sobre la conjetura, corriendo el riesgo de considerar “intratables” a problemas que no lo son. Tendremos, entonces, resultados del tipo: si $\Pi \neq \mathcal{NP}$, entonces Π es intratable.

Para poder concluir¹ que un problema Π es intratable deberemos demostrar que Π es \mathcal{NP} -Completo. Si nos atuviésemos estrictamente a la definición, deberíamos, para ello, ver que todo problema \mathcal{NP} se reduce polinomialmente a Π . Haciendo uso de la equivalencia

¹Es decir, conjeturar pensando que $\mathcal{P} \neq \mathcal{NP}$

entre los \mathcal{NP} -Completo, el siguiente lema nos permitirá sortear esta dificultad técnica en todos los casos salvo en uno, el primero.

Lema 13 Sean $\mathcal{L}_1, \mathcal{L}_2 \in \mathcal{NP}$. Si \mathcal{L}_1 es \mathcal{NP} -Completo y $\mathcal{L}_1 \leq_p \mathcal{L}_2$, entonces \mathcal{L}_2 es \mathcal{NP} -Completo.

Prueba. Sea $\mathcal{L} \in \mathcal{NP}$. Como \mathcal{L}_1 es \mathcal{NP} -Completo tenemos que $\mathcal{L} \leq_p \mathcal{L}_1$. Luego, por la transitividad de \leq_p , $\mathcal{L} \leq_p \mathcal{L}_2$. Y esto es cierto para cualquier $\mathcal{L} \in \mathcal{NP}$, con lo cual, \mathcal{L}_2 es \mathcal{NP} -Completo. \square

Entonces, para demostrar que un problema Π es \mathcal{NP} -Completo deberemos ver:

1. Que Π es \mathcal{NP} y,
2. Que hay algún $\Pi' \in \mathcal{NP}$ -Completo tal que $\Pi' \leq_p \Pi$.

Obviamente, este tipo de demostración necesita contar con algún Π a priori y, hasta el momento, no tenemos ninguno. El teorema de Cook demuestra que SAT es \mathcal{NP} -Completo, y que, por lo tanto, esta clase es no vacía.

4.6.6. Problemas \mathcal{NP} -Duros.

Para esta sección daremos un breve resumen un poco más informal para tener un mejor panorama y una mejor comprensión de la clase \mathcal{NP} .

Sea Π un problema de decisión entonces:

- $\Pi \in \mathcal{P}$ si: Π puede ser resuelto por una Máquina de Turing Determinista en tiempo polinomial.
- $\Pi \in \mathcal{NP}$ si: Π puede ser resuelto por una Máquina de Turing No Determinista en tiempo polinomial.
- $\Pi \in \mathcal{NP} - \text{Completo}$ si: $\Pi \in \mathcal{NP}$ y $\Pi' \in \mathcal{NP}$ -Completo tal que $\Pi' \leq_p \Pi$.

Ahora bien, para definir un problema \mathcal{NP} -Duro debemos tomar en cuenta los siguientes puntos.

Intuitivamente hay problemas que son aún más difíciles que los problemas $\mathcal{NP} - \text{Completo}$. Y a estos problemas los definiremos de la siguiente manera:

Definición 24 Un lenguaje $\mathcal{L} \in \mathcal{NP}$ es \mathcal{NP} -Duro cuando para todo otro lenguaje $\mathcal{L}' \in \mathcal{NP} - \text{completo}$ se tiene que $\mathcal{L}' \leq_p \mathcal{L}$. Por otro lado si Π es un problema de decisión y $\Pi \in \mathcal{NP}$, entonces Π es un problema \mathcal{NP} -Duro si para todo $\mathcal{L}' \in \mathcal{NP}$ -completo se tiene que $\Pi' \leq_p \Pi$

Debemos aclarar que los problemas \mathcal{NP} -Duro no son necesariamente problemas de decisión. Podemos ver el esquema en la figura 4.4.

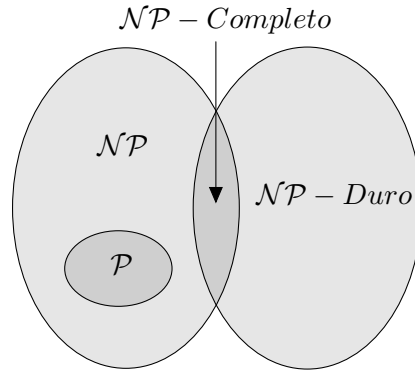


Figura 4.4: Probable descripción de \mathcal{NP} -Duro, si $\mathcal{P} \neq \mathcal{NP}$.

Capítulo 5

Reducción

5.1. Introducción

La prueba de vigilancia de galerías de arte usando reflectores giratorios de ángulo π presentada en la tesis de Israel Aldana Galván puede cubrir un polígono de n vértices con n guardias en tiempo $\mathcal{O}(n)$. Aunque n es necesario en algunos casos, es claro que son muchos más guardias de los necesarios para cubrir un polígono en particular. Por ejemplo, los polígonos convexos sólo requieren tres guardias por el teorema 8, pero el algoritmo de Israel Aldana Galván seguiría colocando n guardias. Es natural, entonces, buscar la colocación del número mínimo de guardias que cubren un polígono dado.

Nosotros daremos en este capítulo la complejidad computacional del problema de encontrar el mínimo número de reflectores giratorios de tamaño π para vigilar un polígono de n aristas y se muestra que este problema es fundamentalmente intratable, dado a que es un problema \mathcal{NP} -Duro.

5.2. Complejidad computacional del problema

Antes de empezar introduciremos la terminología que necesitamos para las secciones posteriores al presente trabajo.

Para nosotros un reflector giratorio será denominado como radar y un reflector giratorio de tamaño π lo denotaremos como radar- π . Si un radar- π se encuentra localizado en un vértice a , nos referiremos a él como el radar R_a .

Ahora bien podemos empezar la explicación de la prueba. Esta prueba muestra que un problema de decisión (3-SAT) es posible reducirlo a un problema de encontrar k guardias para vigilar un polígono. Para esta prueba necesitaremos crear un polígono simple. La

prueba está restringida a colocar los guardias (radares) en los vértices del polígono. Para la construcción de este polígono simple necesitaremos un conjunto de patrones: un patrón de literal, un patrón de cláusula y un patrón de variable. Los patrones de literal aparecen en los patrones de cláusulas y la consistencia de las literales estará dada por el patrón de variable (se explicará más adelante por qué se necesita esta consistencia).

Estos patrones los podemos ver como estructuras individuales que después acoplaremos con el objetivo de formar el polígono final.

Antes de explicar como se construyen los patrones daremos un resumen del 3-SAT. Recordemos que el 3-SAT está dado sobre un conjunto de variables booleanas $X = \{x_1, x_2, \dots, x_n\}$ junto con sus variables negadas, cláusulas, que son disyunciones de 3 literales de X por cláusula y el problema consiste en dada una instancia o fórmula del 3-SAT sobre X , que será un conjunto de cláusulas vinculadas por conjunciones, saber si existe una asignación de valores de verdad dada a las variables de manera que la instancia resulte verdadera.

Entonces si una fórmula tiene m cláusulas, el polígono que construiremos necesitará:

- $3m$ patrones de literal, uno por cada ocurrencia de literal en la fórmula.
- m patrones de cláusula, uno por cada cláusula.
- n patrones de variable, uno por cada variable. El patrón de variable tendrá como objetivo:
 - La representación de las asignaciones de valores a las variables.
 - Y garantizar que el valor asignado a cada variable x_i sea el que se use en cada ocurrencia de la variable en la fórmula.
- El polígono final será la representación de la instancia del 3-SAT.

5.2.1. El patrón de una literal

El patrón de una literal se muestra en la figura 5.1. Cuenta con 3 puntos distinguidos: p_l que únicamente es visible para 4 vértices a, g, f, v , otro punto distinguido es p_1 que únicamente es visible para los vértices a, b, c y por último p_2 que únicamente es visible para los vértices d, e, g . El polígono será creado de tal manera que ningún otro vértice pueda ver a p_l, p_1 ó p_2 . Como las etiquetas sugieren, al vértice v le será asignado un radar cuando el valor de verdad de la literal sea verdadero de lo contrario se le asignará a f cuando sea falso.

Pueden surgir algunas preguntas: ¿Por qué usar este diseño para una literal? y ¿por qué los puntos distinguidos tienen que ser vistos únicamente por algunos vértices? Recordemos que se dice que un punto es vigilado si en todo momento algún radar se encuentra

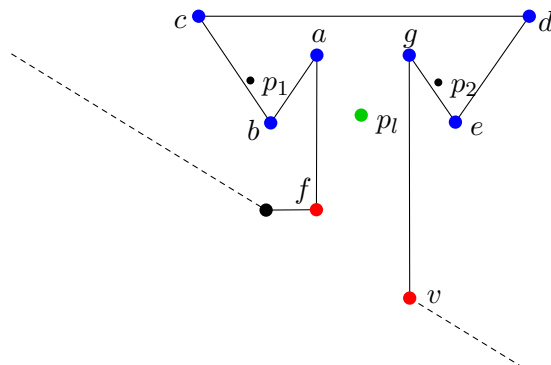


Figura 5.1: El patrón de una literal.

vigilándolo. Ahora bien, para responder a estas preguntas necesitaremos primero algo que nos garantice que para vigilar a un punto en todo momento se necesitan al menos 3 radares- π . Si tomamos esta última afirmación como verdadera, entonces podemos observar que para vigilar en todo instante a p_1 se necesitan al menos 3 radares y los únicos vértices que pueden ver a p_1 son a, b , y c . Esto implica que debemos colocar un radar en a, b , y c para que p_1 esté vigilado. Lo mismo sucede para p_2 . Por último, para vigilar al punto distinguido p_l también se necesitan 3 radares, pero como podemos observar, los vértices a y g forzosamente deben tener un radar, entonces bastará con colocar un radar en v ó f . Por consiguiente, este diseño nos sirve para que los patrones de literal tengan la característica de obligar a tener un radar en el vértice v o f .

Para que la observación anterior sea verdadera necesitamos hacer dos aclaraciones:

- Que un punto necesite al menos 3 radares para ser vigilado en todo momento y
- Que p_l pueda ser vigilado utilizando a R_a, R_g y $R_{v/f}$, dado que R_a y R_g también vigilan a otros puntos distinguidos.

Sabemos de antemano que al menos existe una configuración inicial de radares, dada por el teorema 8 del capítulo 1 (a la que nos referiremos de aquí en adelante como *configuración fuerte*), que garantiza que un punto p es vigilado en todo momento usando 3 radares- π , pero ¿se podrá iluminar alguna parte considerable del plano en todo momento con 2 radares? Para responder daremos antes una observación.

Observación 2 Sean a' y b' dos puntos distintos en el plano y sean l'_a y l'_b dos rectas que giran sobre dichos puntos, respectivamente, con la misma velocidad angular. Entonces, si l'_a y l'_b no son paralelas, en cada instante su intersección es un punto, mismo que describe una circunferencia que pasa por a' y b' . En caso de que l'_a y l'_b sean paralelas la intersección será definida por la recta $a'b'$, únicamente esta intersección se dará en los instantes en los que las rectas se enciman.

Lema 14 Sea P el plano. Sean l_a y l_b las rectas correspondientes a los radares- π R_a y R_b que giran con igual velocidad y sentido. Sea $p \in P$ un punto distinto a los de la intersección de l_a y l_b al rotar (es decir, p no pertenece al circuncírculo o a la recta ab , según sea el caso). Entonces p no es permanentemente vigilado por los radares y, además existe una vecindad de p tal que permanece en la sombra durante un intervalo de tiempo positivo.

Prueba. Caso en que l_a y l_b no son paralelas:

Sea x el punto de intersección de l_a y l_b . Dado que l_a y l_b se encuentran girando, entonces siempre encontraremos en algún momento un ángulo de sombra que barre la parte exterior, interior o ambas del circuncírculo (obsérvese la figura 5.2). No importando como se acomoden los radares siempre se tendrá una situación como los casos (a), (b), (c) y (d), únicamente se debe tomar en cuenta que si se presenta la intersección como en el caso (a) y (b) entonces los casos (c) y (d) ocurrirán π tiempo después en el arco contrario y viceversa.

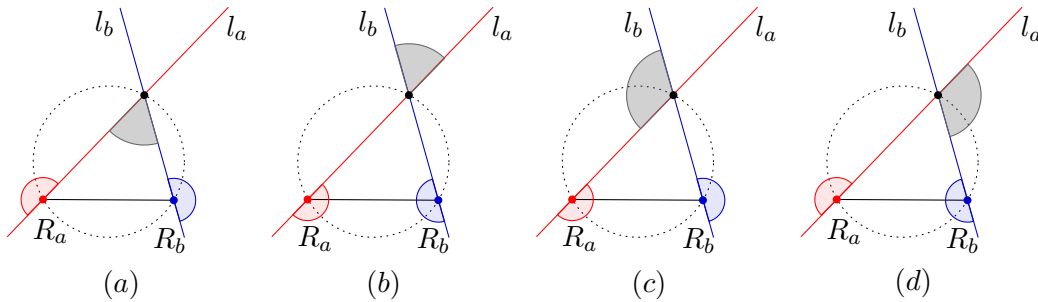


Figura 5.2: (a) El ángulo de sombra cuando R_a y R_b no cubren al segmento ab , barre la parte superior (del interior) del circuncírculo y todo el semiplano inferior, conforme x recorre el arco superior de a hacia b . (b) Durante la siguiente media revolución de los radares (ambos necesariamente cubriendo a ab), el ángulo de sombra barre la región del semiplano superior comprendida en el exterior del circuncírculo. (c) y (d) ilustran las otras dos maneras en que se puede presentar el cono de sombra, se dan cuando un radar cubre a ab pero el otro no. La observación crucial es que si en el arco superior se presentan los dos primeros casos, entonces en el arco inferior se tendrán los otros dos (y viceversa, desde luego).

Supongamos que los radares se encuentran como en los casos (a) y (b) (si no es así, véase de cabeza la figura, el arco inferior se volverá el superior y presentará dichos casos). Sea $p \in P$ un punto que no pertenece a la circunferencia. Por lo dicho en el pie de la figura 5.2, p pertenece a la región barrida por el ángulo de sombra, por lo tanto no es permanentemente vigilado.

La segunda parte de la conclusión es consecuencia de la primera y del hecho que la zona de sombra es una región abierta de P (un ángulo abierto en el caso en que l_a y l_b no

son paralelas; un semiplano abierto, un par de semiplanos abiertos o una banda abierta, si las rectas son paralelas). Entonces, si p no es vigilado en el instante t , necesariamente está en la región de sombra y, como esta es abierta, por definición (de conjunto abierto) existe $\epsilon > 0$ tal que la vecindad circular de radio ϵ centrada en p queda contenida en dicha región. Ahora bien, escogiendo ϵ suficientemente pequeña, digamos la mitad de la menor de las distancias que la circunferencia de la vecindad aludida guarda, respectivamente, con las rectas l_a y l_b . Entonces, como los radares giran con velocidad uniforme, necesariamente deberá transcurrir un tiempo positivo. \square

Ahora bien, ya sabemos que se necesitan al menos 3 radares para cubrir un punto, lo que necesitamos ahora es saber si se pueden sincronizar los radares, para lo cual necesitamos la siguiente definición.

Definición 25 Diremos que un radar se encuentra sincronizado, si pertenece a alguna configuración de otros dos radares con el propósito de vigilar a un punto.

Podemos observar que R_a se encuentra sincronizado con R_b y R_c , lo mismo pasa con R_g , se encuentra sincronizado con R_d y R_e . Pero también podemos ver que R_a y R_g no pertenecen a la misma sincronización, y esto nos permite sincronizarlos para vigilar a p ya sea con el grupo de radares en: a, g, v ó bien a, g, f .

Como podemos observar en la figura 5.3 el patrón de literal queda iluminado.

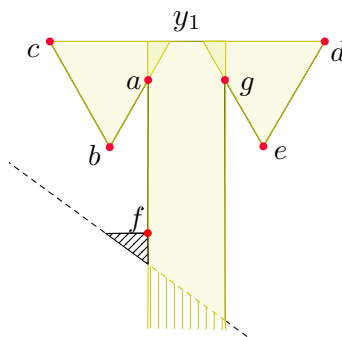


Figura 5.3: Un patrón de literal iluminado, si $y_1 = F$.

Tengamos en cuenta que para probar que el problema de vigilancia de galerías de arte usando radares es un problema \mathcal{NP} -Duro únicamente necesitamos 2 observaciones: la primera es que para vigilar un punto p en todo momento, necesitamos que al menos 3 vértices puedan ver a p para colocar los radares en dichos vértices y que exista al menos una configuración para esos radares que cumpla con vigilar a p en todo momento.

Por el teorema 8 siempre que 3 radares se encuentren en una configuración fuerte, cualquier punto p que sea visible por estos 3 radares quedará completamente vigilado. Entonces los grupos de radares $\{R_a, R_b, R_c\}$, $\{R_d, R_e, R_g\}$ y $\{R_a, R_g, R_{f/v}\}$ iluminarán

por completo el patrón de literal, con excepción de un pequeño triángulo que se muestra en la figura 5.3, pero de éste nos encargaremos más adelante.

5.2.2. Pseudo literal y Pseudo literal auxiliar

Para poder explicar los patrones de una cláusula y de una variable es necesario introducir dos estructuras. La primera es el pseudo literal, este tiene la misma estructura y propiedades de un patrón de una literal, con la diferencia de que no es un patrón de literal y llamaremos a sus puntos distinguidos p_3, p_4 y p_{aux1} , así como aux_a y aux_b a los vértices etiquetados f y v , respectivamente (véase la figura 5.4).

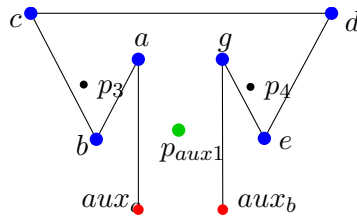


Figura 5.4: Pseudo literal.

Las dos únicas diferencias entre una pseudo literal y un pseudo literal auxiliar es que llamaremos a sus puntos distinguidos p_5, p_6, p_{aux2} y que el grupo de radares que pueden ver a p_{aux2} son R_a, R_g y un tercero R_y ajeno a la pseudo literal auxiliar (véase la figura 5.5).

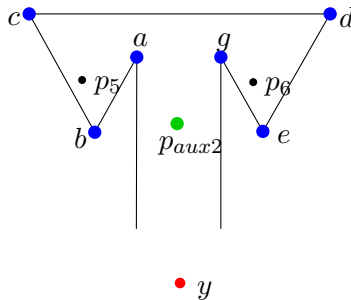


Figura 5.5: Pseudo literal auxiliar.

5.2.3. Patrón de una cláusula

El patrón de una cláusula se conforma de tres patrones de literal y dos pseudo literales. Un patrón de cláusula debe tener las siguientes características:

- Tiene una zona Z que únicamente puede ser visible para los vértices v_1, v_2, v_3, aux_1 y aux_2 .
- Al menos debe de haber un radar en algún vértice v para un patrón de literal con el objetivo de que el patrón de una cláusula sea verdadero, es decir que si al menos un patrón de literal tiene un radar en algún vértice v , el patrón de cláusula debe estar completamente iluminado.
- aux_1 y aux_2 deben ver a todos los f_i/v_i , para $i = 1, 2, 3$.
- Los vértices f_i, v_i, aux_1, aux_2 , para $i = 1, 2, 3$, deben estar colocados sobre la misma circunferencia de un círculo.

A continuación se puede observar gráficamente en la figura 5.6 como se construye un patrón de cláusula.

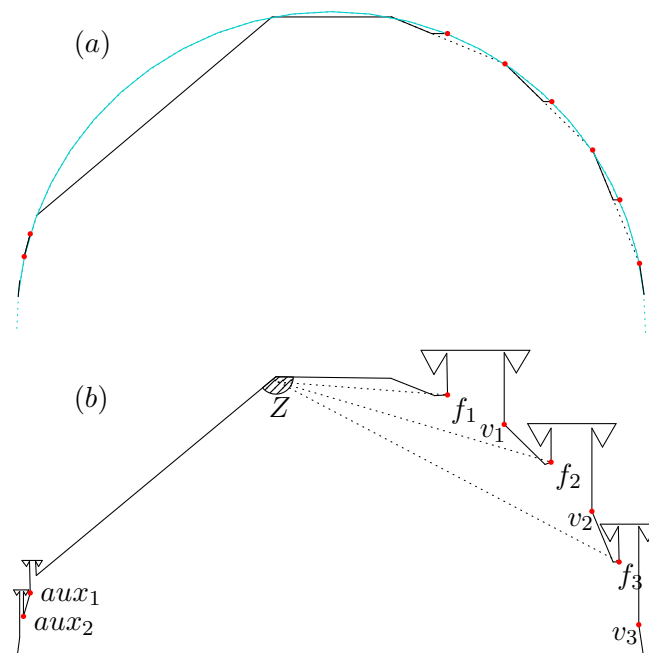


Figura 5.6: (a) Colocando los vértices en una circunferencia. (b) La zona Z es visible sólo para los vértices v_1, v_2, v_3, aux_1 , y aux_2 .

En la figura 5.6 (a) podemos observar que los vértices pueden ser colocados en la circunferencia. Note que se deja una pequeña ceja en tres de estos vértices. Y en la figura 5.6 (b) podemos ver que para los vértices v_1, v_2, v_3, aux_1 y aux_2 es visible la zona Z , y para f_1, f_2 y f_3 no es visible por la ceja creada. Por otro lado aux_1 y aux_2 pueden visualizar a f_i/v_i , para $i = 1, 2, 3$, con solo colocarlos por debajo de la ordenada de f_3 .

Tenemos claro que los tres patrones de literales y las dos pseudo literales quedan iluminadas completamente por construcción. Para vigilar la zona Z se necesita que al menos haya un radar en v_i y basta con sincronizar a R_{aux_1}, R_{aux_2} con cualquier R_{v_i} (véase la figura 5.7) y con esto se iluminaría casi todo el patrón de cláusula, excepto por los triángulos de sombra que forman las cejas de los patrones de literales (observemos que si R_{aux_1} y R_{aux_2} se sincronizan con R_{v_3} , entonces los triángulos de sombra de las tres literales se encontrarían iluminados, sin embargo esto no nos ayuda porque no podemos suponer que siempre habrá un radar en v_3).

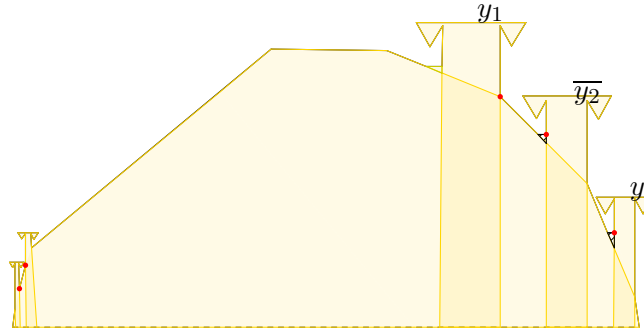


Figura 5.7: El patrón de una cláusula iluminada, con excepción de los triángulos de sombra formados por las cejas de los patrones de literal, si $y_1 = V, y_2 = V$ y $y_3 = F$.

Ahora bien, para iluminar los triángulos se necesita que al menos 3 radares vean a cada triángulo y ya que cada patrón de literal forzosamente debe tener un radar en f_i/v_i , para $i = 1, 2, 3$, entonces basta con agrupar a $\{R_{aux_1}, R_{aux_2}, R_{f_1/v_1}\}$, $\{R_{aux_1}, R_{aux_2}, R_{f_2/v_2}\}$ y $\{R_{aux_1}, R_{aux_2}, R_{f_3/v_3}\}$ para iluminar los 3 triángulos de sombra, respectivamente. Pero ¿es posible establecer alguna sincronización que nos permita esto?

Lema 15 Sean R_a, R_b y R_c radares que se encuentran en configuración fuerte. R_a y R_b se pueden sincronizar con cualquier número n de radares, siempre y cuando los n radares se encuentren sobre la circunferencia que pasa por a, b y c y pertenezcan al mismo arco de a, b y c .

Prueba. Si este lema es cierto, dado que los radares giran, esto implica que la siguiente suposición es cierta: S.P.G. R_a, R_b, R_c se encuentran en configuración fuerte en el tiempo 0, entonces en algún tiempo t la intersección del haz de luz de R_a y el haz de luz de R_b estará sobre otro R_{c_1} , tal que R_a, R_b, R_{c_1} se encuentren en configuración fuerte (véase la figura 5.8). Esto sólo es posible si el ángulo formado por acb es el mismo que el ángulo ac_1b . Lo anterior es cierto por la propiedad de *igualdad de ángulos inscritos*, que dice que si dos o más ángulos inscritos abarcan el mismo arco, éstos miden lo mismo. Tengamos en cuenta que los radares giran a la misma velocidad con el mismo sentido y por esta razón

cuando la intersección de los haces de R_a y R_b pase por c_1 , basta con colocar el radar de c_1 tangente al círculo, quedando así en configuración fuerte R_a, R_b, R_{c_1} . \square

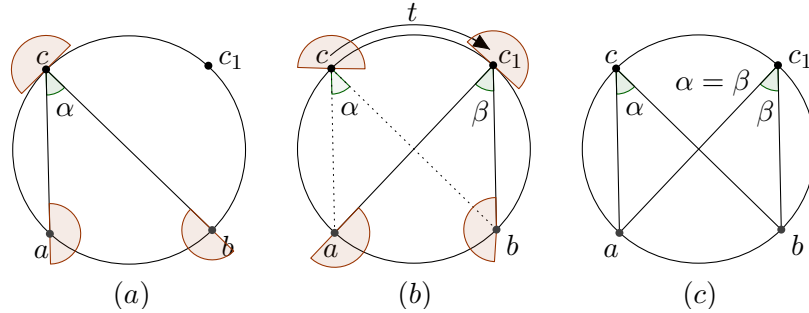


Figura 5.8: (a) Radares en a, b en configuración fuerte con radar en c colocado en la circunferencia del círculo, en el tiempo 0. (b) Radares en los vértices a, b, c_1 en configuración fuerte, en el tiempo t . (c) Igualdad de ángulos inscritos.

5.2.4. Patrón de una variable

Como habíamos mencionado antes, el patrón de una variable sirve para representar la asignación de valores de verdad a las variables y para modelar la sustitución de dichos valores en la fórmula, relacionando consistentemente cada patrón de variable x_i y \bar{x}_i que se usen en los distintos patrones de cláusula. El patrón de una variable se ilustra en la figura 5.9. El patrón de una variable x_i tiene las siguientes características:

- Está formado por 2 “pozos” con un vértice etiquetado F en el pozo izquierdo y otro vértice etiquetado V en el pozo derecho.
- Cada pozo cuenta con r_i “espinas” (pseudo literales auxiliares), donde r_i es el número de literales en las que la variable x_i participa.
- Cada espina se encuentra alineada con su respectivo vértice F/V de su pozo.
- Contiene una pseudo literal auxiliar que se usa para contener a un punto distinguido p_v que sólo es visible para los vértices a, g, F, V .
- Cada espina estará alineada con un vértice f_{a_i}/v_{a_i} de los patrones de cláusula (más adelante se verán las condiciones para alinearlas).

Para cada variable x_i , para $i = 1, \dots, n$ se requiere esta estructura. A la asignación $x_i = V$ le corresponde la colocación de un radar en el vértice V y si $x_i = F$ el radar se coloca en el vértice F . En esta sección sólo mostraremos como se vigilan algunos puntos

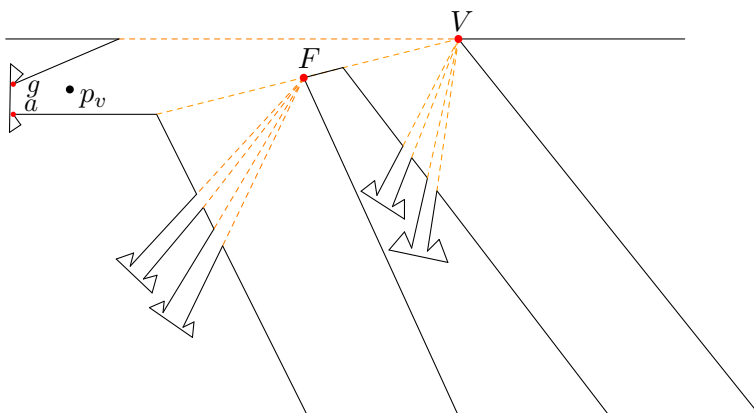
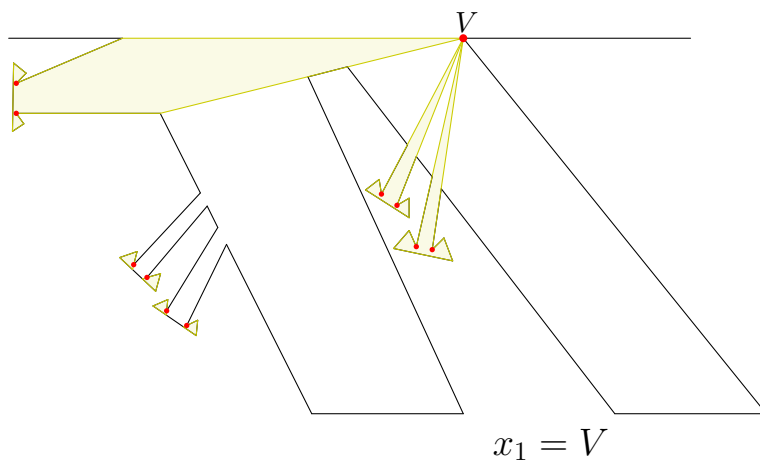


Figura 5.9: El patrón de una variable.

distinguidos del patrón de una variable, ya que necesitamos de todo el polígono para vigilarlo completamente. En la figura 5.10 se muestra como se ilumina un patrón de una variable si $x_1 = V$.

Figura 5.10: Iluminación parcial del patrón de una variable, si $x_1 = V$.

En la figura 5.10 se puede ver que la pseudo literal auxiliar que contiene a p_v queda iluminado colocando un radar en V , como existe R_V las demás espigas del pozo de V , pueden usarlo para iluminarse también. Esto es posible por que cada grupo de radares únicamente comparten a R_V .

5.2.5. Construcción completa

Hasta ahora hemos explicado:

- El patrón de una literal.
- El patrón de una cláusula.
- Y cómo representar la asignación de valores a las variables.

A continuación se explicará la forma de colocar los patrones de cláusula en relación a los patrones de variable, esto con el fin de mantener la consistencia entre la asignación anterior y los valores de las literales. Esto se puede ver como la representación de la sustitución de valores de las variables en las respectivas ocurrencias de estas en la fórmula booleana.

En la figura 5.11 se puede observar el polígono completo.

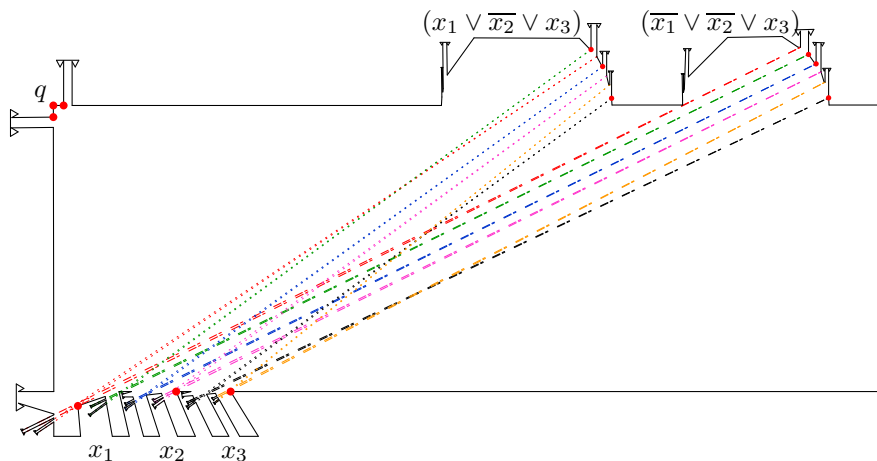


Figura 5.11: El polígono completo.

La estructura del polígono cuenta con 2 pseudo literales que se encuentran a un lado del vértice etiquetado q . Recordemos que cada pseudo literal cuenta con 2 vértices, aux_a y aux_b (véase figura 5.4). Denotemos por q_{aux_a} a uno de los vértices de una de las pseudo literales y por q_{aux_b} a uno de los vértices de la otra pseudo literal. Los vértices q , q_{aux_a} y q_{aux_b} pueden ver todo punto dentro de cualquier pozo (sin incluir las espinas). Entonces colocando radares en q , q_{aux_a} y q_{aux_b} se puede iluminar todos los pozos y a la región que une a los patrones de cláusulas.

Para que el polígono pueda quedar completamente iluminado, falta mencionar las condiciones para alinear las espinas con sus respectivos vértices f_{a_i}/v_{a_i} de los patrones de literales.

Sea $X = \{x_1, \dots, x_n\}$ el conjunto de los patrones de variables y $C = \{c_1, \dots, c_m\}$ el conjunto de los patrones de cláusulas. Sea $x_i \in X$, para $i = 1, \dots, n$ el patrón de una variable con sus distinguidos vértices V y F , sea $pozo_V$ el pozo derecho y $pozo_F$ el pozo

izquierdo de x_i , sea $c_j \in C$, para $i = 1, \dots, m$ y sea l un patrón de literal con sus vértices distinguidos v y f , $l \in c_j$, (tomemos cuenta que l puede tomar valores $\{x_i, \overline{x_i}\}$). Entonces tenemos que:

- Una espina en el pozo_F de x_i se encuentra alineada con F y v si $l = x_i$ (ver figura 5.12 (a)).
- Una espina en el pozo_F de x_i se encuentra alineada con F y f si $l = \overline{x_i}$ (ver figura 5.12 (b)).
- Una espina en el pozo_V de x_i se encuentra alineada con V y f si $l = x_i$ (ver figura 5.12 (a)).
- Una espina en el pozo_V de x_i se encuentra alineada con V y v si $l = \overline{x_i}$ (ver figura 5.12 (b)).

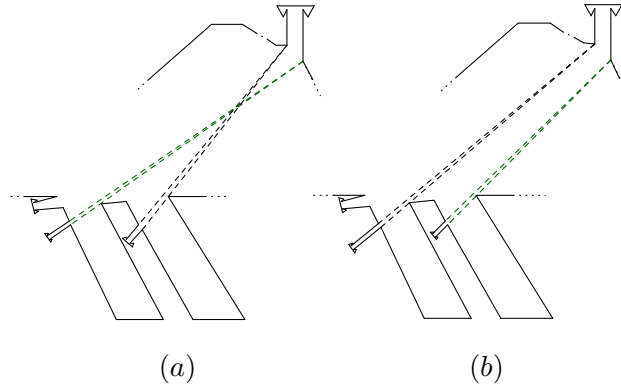


Figura 5.12: (a) Alineación para las espinas si $l = x_i$. (b) Alineación para las espinas si $l = \overline{x_i}$

La consecuencia de estas alineaciones es que si un radar es colocado en un F , entonces se iluminan todas las espinas del pozo_V , y si un radar es colocado en V , entonces se iluminan todas las espinas del pozo_F . Como ya se mencionó anteriormente, se necesita un radar en F ó V para iluminar el punto distinguido p_v asociado a x_i . Debido a que todas las espinas del pozo_F están alineadas con F y a sus respectivos vértices v , son los vértices definidos en la figura 5.1, para todos los patrones de literales x_i , y con F y a sus respectivos vértices f , para todos los patrones de literales $\overline{x_i}$, todas las espinas del pozo_F serán iluminadas si los radares son colocados en v y en f . Esto significa que si a las literales que involucran la variable x_i le son asignados valores de verdad consistentes con $x_i = V$, entonces un radar en F no es necesario para cubrir las espinas del pozo_F , pero sí un radar en el vértice V . Simétricamente, si $x_i = F$, entonces un radar en V de x_i no es necesario para cubrir las espinas del pozo_V , pero sí uno en el vértice F .

Ahora recordemos cuántos radares son necesarios para iluminar cada parte del polígono, para después sumarlos:

- El patrón de una literal necesita 7 radares.
- Una pseudo literal necesita 7 radares.
- Una pseudo literal auxiliar necesita 6 radares, ya que el séptimo no pertenece al pseudo-literal auxiliar.
- El patrón de una cláusula necesita 3 literales y 2 pseudo literales, por lo tanto en total se usan 35 radares.
- El patrón de una variable x_i necesita $2r_i + 1$ pseudo literales auxiliares y un radar F ó V , que suman un total de $6(2r_i + 1) + 1 = 12r_i + 7$ radares.
- El polígono completo tiene m patrones de cláusulas, n patrones de variables, 2 pseudo literales y 1 radar q , que nos dan un total de $m + n + 15$ radares. Entonces el número total de radares requeridos para iluminar el polígono lo determina la siguiente fórmula:

$$\begin{aligned}
 35m + (12r_1 + 7) + \dots + (12r_n + 7) + 15 &= 35m + 12(r_1 + \dots + r_n) + 7n + 15 \\
 &= 35m + 12(3m) + 7n + 15 \\
 &= 71m + 7n + 15
 \end{aligned}$$

Teorema 9 *Un conjunto dado de cláusulas es satisfacible si y sólo si el polígono construido puede ser vigilado con $K = 71m + 7n + 15$ radares.*

Prueba. Observemos que un punto dentro del polígono al menos necesita ser visto por tres vértices, para en ellos colocar tres radares y así pueda ser vigilado en todo momento. El patrón de una literal necesita al menos 7 radares para poder ser vigilado completamente, ya que se necesitan 3 radares para vigilar por completo a su punto distinguido p_1 y necesita de otros 3 radares para mantener vigilado a p_2 , una vez colocados estos 6 radares p_l puede ser vigilado colocando un radar más en v ó f , ya que dos radares de los que vigilan a p_1 y p_2 son capaces de ver a p_l . De manera análoga sucede para una pseudo literal. Una pseudo literal auxiliar requiere 3 radares para vigilar a p_5 y otros 3 radares para vigilar a p_6 , y esto es suficiente para mantenerla vigilada, ya que p_{aux2} es visible para los radares en a y g vigilan a p_5 y p_6 (véase figura 5.5), y, el tercer radar es ajeno a la estructura de la pseudo literal auxiliar. Teniendo esto en cuenta, entonces:

\Rightarrow) Si el conjunto de cláusulas es satisfacible, entonces existe una asignación de verdad para las variables tal que cada cláusula es verdadera. Colocando un radar en los vértices v o f apropiados de los patrones de literal en cada cláusula, necesariamente se cubre la

cláusula, debido a que un radar será asignado en al menos un vértice v y así agruparlo con los radares en aux_1 y aux_2 . Por el argumento presentado anteriormente, colocando un radar en el vértice V de un patrón de una variable para $x = V$, o en F si $x = F$, se cubren por completo todas las espinas (pseudo literales auxiliares) al igual que el punto distinguido p_v del patrón de variable para x . Finalmente, R_q junto con R_{qaux_a} y R_{qaux_b} cubren todos los pozos de todos los patrones de variables. De esta manera se logra iluminar el polígono con K radares.

⇐) Supongamos que el polígono se puede vigilar con K radares. Las pseudo literales en ambos costados del vértice q necesitan 7 radares cada una para poder ser vigiladas por completo, un radar debe ser colocado en q , para poder agruparlo con R_{qaux_a} y R_{qaux_b} , de otra manera K no es suficiente. Los $71m + 7n$ radares restantes son necesitados para cubrir $31m + 3n$ puntos distinguidos. Cada patrón de cláusula está formado por 3 patrones de literales y 2 pseudo literales, cada 1 de estas 5 estructuras necesitan 7 radares para poder ser vigiladas, y el patrón de una cláusula será vigilado por $R_{v_{i_m}}$ y $R_{f_{i_m}}$ sólo si $R_{aux_{1m}}$ y $R_{aux_{2m}}$ se pueden agrupar con al menos un $R_{v_{i_m}}$. Lo cual implica que se satisface la cláusula. Los $36m + 7n$ radares restantes serán ocupados por los patrones de variables. Cada patrón de una variable consta de $2r + 1$ pseudo literales auxiliares, de los cuales un patrón de variable será cubierto por 6 radares junto con R_V ó R_F , r pseudo literales auxiliares serán cubiertas cada una por 6 radares junto con R_V ó R_F , los r pseudo literales auxiliares restantes serán cubiertos sólo si todas las literales que utilizan esa variable se asignan consistentemente, como se observó antes. Note que una pseudo literal auxiliar necesita 6 radares para ser vigilada y un séptimo ajeno a la pseudo literal auxiliar, que en este caso el séptimo radar será colocado en V o F para cubrir $r + 1$ pseudo literales auxiliares y para las r restantes el séptimo radar serán los R_v y R_f de los patrones de cláusulas si estos son asignados consistentemente. Pero entonces la colocación de radares determina una asignación de verdad consistente que satisface la instancia dada de 3-SAT. ■

Dado lo anterior, entonces podemos decir que el problema de vigilancia de galerías de arte con radares (en vértices para polígonos simples) es al menos tan difícil que el 3-SAT, y es por esto que se da el siguiente corolario:

Corolario 3 *El problema de encontrar el mínimo número de radares en vértices para vigilar un polígono simple sin hoyos, es \mathcal{NP} -duro.*

Capítulo 6

Vigilancia de Polígonos Ortogonales con Radares

6.1. Introducción

Pediremos por última ocasión en el presente trabajo que echemos a andar nuestra imaginación y pensar en la siguiente situación: Estamos en una galería de arte ortogonal y observamos que no hay guardias, sin embargo, el sistema de vigilancia está dado por cámaras de vigilancia que giran 360° en su mismo eje, y su campo de visibilidad es de 180° grados (π radianes). Haciendo cálculos rápidos, podemos hacer dos observaciones: que las cámaras que vigilan no son suficientes y hay cámaras que se encuentran vigilando zonas ya vigiladas por otras cámaras. Entonces nos preguntamos si existe algún algoritmo que nos ubique y minimice el número de cámaras.

En el 2011 E. Kranakis, F. MacQuarie, O. Morales y J. Urrutia nos ofrecieron un resultado [1], para vigilar un plano con reflectores giratorios (que por convención hemos llamado radares). Y gracias a J. O'Rourke, aparte de presentarnos una prueba diferente para el teorema [4], nos ofreció un algoritmo para particionar un polígono ortogonal en subpolígonos con forma de L [10]. Estas dos investigaciones nos brindan herramientas suficientes para trabajar en un algoritmo, vigilar con radares- π a un polígono ortogonal.

6.2. Vigilancia de polígonos ortogonales con radares

6.2.1. Polígono ortogonal en forma de L

Antes de ver cómo se puede vigilar cualquier polígono ortogonal, primero veremos cómo podemos vigilar un polígono ortogonal con forma de L (polígono-L por brevedad).

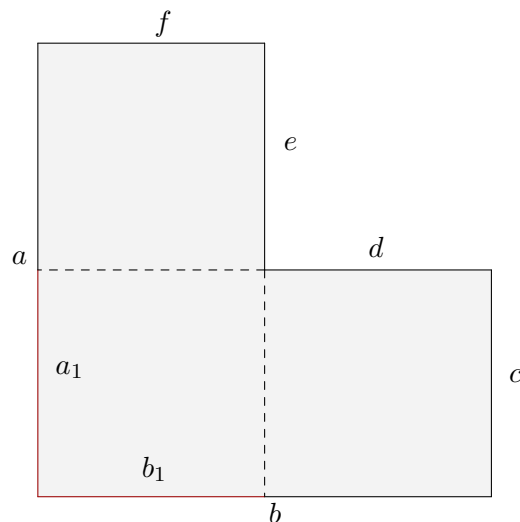


Figura 6.1: Polígono ortogonal con forma de L, con aristas a, b, c, d, e, f .

Como podemos observar en la figura 6.1, la arista a tiene un segmento de línea (a_1) para el cual le es visible todo el polígono, lo mismo sucede con la arista b y el segmento b_1 . Podemos observar que estos segmentos se encuentran unidos por el vértice que une a las aristas a y b , a este vértice le llamaremos vértice distinguido.

Entonces, por el teorema 7, si colocamos 3 radares- π colineales sobre a_1 o b_1 , podemos mantener vigilado el polígono en todo momento.

6.2.2. Polígono ortogonal

Sea P un polígono ortogonal, entonces podemos dar el siguiente algoritmo:

1. Particionamos a P con el algoritmo de partición de polígonos ortogonales en polígonos-L.
2. Buscamos en cada polígono-L obtenido del paso anterior un segmento de arista desde la cual sea visible todo ese polígono.
3. Colocamos 3 radares en el segmento encontrado del paso anterior.

En realidad es bastante sencillo este algoritmo. Ahora veremos su complejidad.

Particionar el polígono $O(n \ln n)$

Este paso, como vimos en el capítulo 1, puede ser ejecutado en tiempo $O(n \ln n)$.

Buscar el segmento $O(n)$

Este paso no es complicado, ya que el polígono en forma de L tiene un vértice cóncavo, y lo único que tenemos que hacer es extender las aristas incidentes en este, después inicializar un contador en cero y recorrer el polígono incrementando el contador hasta que sea igual a 3. El segmento buscado se encuentra entre el vértice 3 y 4 (figura).

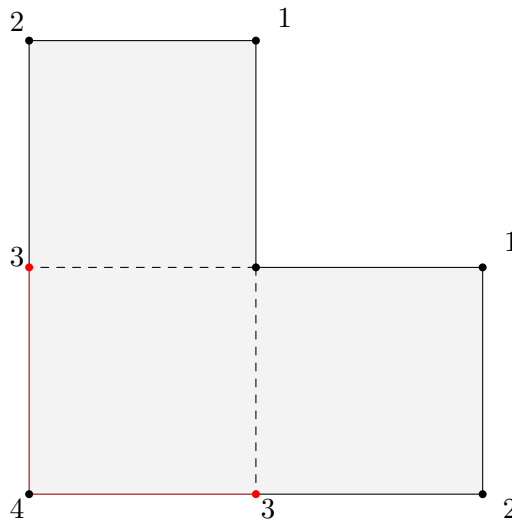


Figura 6.2: Segmento capaz de visualizar todo el polígono en forma de L.

Entonces lo único que tenemos que hacer es encontrar el vértice cóncavo, lo podemos hacer recorriendo el polígono en tiempo $O(6)$ (porque tiene 6 vértices) y después buscar el segmento entre el vértice 3 y 4 se puede hacer en $O(3)$. Esto se hace por cada polígono en forma de L, como hay $\frac{r}{2} + 1$ polígonos, donde r es el número de vértices cóncavos, y a lo más hay $n/2$ vértices cóncavos, entonces este paso lo podemos ejecutar en tiempo $O(n)$.

Colocar los radares $O(n)$

En este paso lo único que se tiene que hacer es colocar los radares en la posición correcta para vigilar al polígono. Está claro que este paso lo tenemos que hacer por cada polígono en forma de L y se puede ejecutar en $O(n)$. Observemos que este algoritmo puede colocar vértices al interior del polígono¹.

Este algoritmo se puede ejecutar en $O(n \ln n) + O(n) + O(n)$, y coloca $3(\lfloor \frac{r}{2} \rfloor + 1)$ radares.

Tenemos que $n = 2r + 4$, entonces:

¹Los puntos al interior del polígono llevan por nombre vértices *Steiner*

$$r = \frac{n - 4}{2}$$

Substituyendo a r nos queda que el número de radares colocados para vigilar a un polígono ortogonal con radares- π sobre las aristas es:

$$3 \left(\left\lfloor \frac{n - 4}{4} \right\rfloor + 1 \right)$$

Bibliografía

- [1] O. Morales E. Kranakis, F. MacQuarie and J. Urrutia. Uninterrupted coverage of a planar region with rotating directional antennae. (2011).
- [2] S. Fisk. A short proof of chvátal’s watchman theorem. (1978).
- [3] D. S. Garey, M. R. y Johnson. *Computers and Intractability - A guide to the Theory of NP Completness*. Freeman and Co, (1979).
- [4] E. Györy. A simple proof of the rectilinear art gallery theorem. pages 7:452–454, (1986).
- [5] J. O’Rourke y E. Welzl H. Edelsbrunner. Stationing guards in rectilinear art galleries. (1984).
- [6] H. Uehara y M. Yokoyama H. Ito. Np-completeness of stage illumination problem. pages pp. 88–92, (1998).
- [7] T. Shermer J. Abello, V. Estivill and J. Urrutia. Illuminating with orthogonal floodlights. (1995).
- [8] E. Rivera-Campo J. Czyzowicz and J. Urrutia. Optimal floodlight illumination of stages. (1993).
- [9] M. Klawe y D. Kleitman J. Kahn. Traditional galleries require fewer watchmen. (1983).
- [10] J. O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, (1987).
- [11] J. R. Sack. Rectilinear computational geometry. (TR-SCS-54-1984.).
- [12] T. C. Shermer. *Recent results in art galleries*. Proc. IEEE, 80(9):1384-1399, (1992).
- [13] J. Urrutia. *Art gallery and illumination problems*. Sack and J. Urrutia editors, North-Holland, (2000).