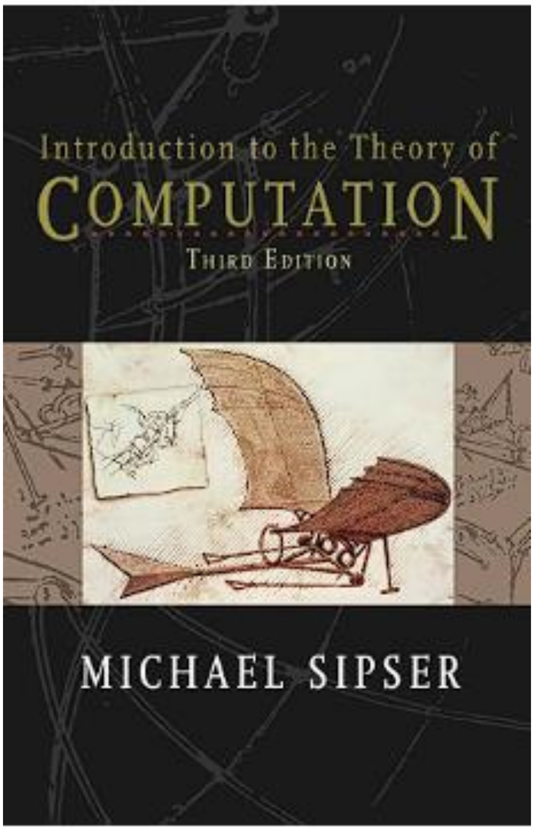
Memoria

Ejercicio Sipser 7.24



Asignatura: Algoritmia y Complejidad

Titulación: Grado en Ingeniería Informática

Curso: 2023-2024

Autor: Juan Francisco Sobrino Ramírez

Contenido

Enunciado

Una 2cnf-fórmula es un AND de cláusulas, dónde cada cláusula es un OR de al menos dos literales. Dado 2SAT = {<ϕ>| ϕ es una 2cnf-fórmula satisfacible}. Demuestra que 2SAT ∈ P.

Definiciones Previas

Clase P

La P en la clase P significa tiempo polinomial. Es el conjunto de problemas de decisión (problemas con respuesta "sí" o "no") que pueden ser resueltos por una máquina determinista en tiempo polinomial:

P=$\bigcup_k TIME(n^k)$

La solución a los problemas P es fácil de encontrar. P es a menudo una clase de problemas computacionales que son solucionables y tratables (los problemas se pueden resolver tanto en la teoría como en la práctica). Es invariante para los modelos de Computación.

Fundamentos lógicos

En este problema trataremos con fórmulas booleanas por ello debemos recordar los términos:

- **Literal:** aparición de variable booleana que podrá tomar los valores 1 (true) y 0 (false).
- **Fórmula booleana:** conjunto de literales relacionados con conectivas.
- **Conectivas:** relacionan las variables booleanas. Consideraremos tres: negación (¬), conjunción (∧) y disyunción (∨).
- **Cláusula:** fórmula booleana consistente en la disyunción de cero o más literales.
- **CNF – Fórmula Normal Conjuntiva:** una fórmula booleana está en CNF si es la conjunción de cero o más cláusulas.
- **Fórmula satisfacible:** una fórmula booleana es satisfacible si existe al menos una asignación de sus variables de entrada que la hace tomar el valor 1.
- **Grafo dirigido:** Tipo de grafo en el cual las aristas tienen un sentido definido.
- **Grafo dirigido fuertemente conexo:** Un grafo dirigido es llamado fuertemente conexo si para cada par de vértices u y v existe un camino de u hacia v y un camino de v hacia u.

Problema 2SAT

Sea SAT = {<f> / f es una fórmula booleana satisfactible}, 2SAT es un caso especial de SAT en forma normal conjuntiva con cláusulas de 2 literales. El problema de decisión de 2SAT puede enunciarse como:

Dado una fórmula de forma normal conjuntiva con 2 literales por cláusula, ¿es posible asignar tales valores a las variables para que CNF sea verdadero?

Esto se puede entender mejor con unos ejemplos:

- **Ejemplo Válido:** $\Phi_1 = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2)$

Para $x_1 = 1$, $x_2 = 1$: $\Phi_1 \in$ 2SAT

- **Ejemplo No Válido:** $\Phi_2 = \Phi_1 \wedge (\neg x_1 \vee \neg x_2)$

Para todas las combinaciones posibles: $\Phi_2 \notin$ 2SAT

En cuanto al campo de aplicación, el problema 2SAT es un caso especial del problema de satisfacción booleana (SAT) y tiene muchas aplicaciones en ciencias de la computación, como en la teoría de la complejidad computacional, en la verificación formal y en la planificación automática.

Por ejemplo, podemos destacar, su uso en la colocación sin conflictos de objetos geométricos y en la agrupación de datos.

Resolución

Enfoque

Primero necesitamos convertir el problema a una forma diferente, la llamada forma normal implicativa. Supongamos que la cláusula *(A V B)* es verdadera, podemos expresarla como una implicación:

- Si A = 0, B debe ser 1, es decir $(\neg A \rightarrow B)$
- Si B = 0, A debe ser 1, es decir $(\neg B \rightarrow A)$

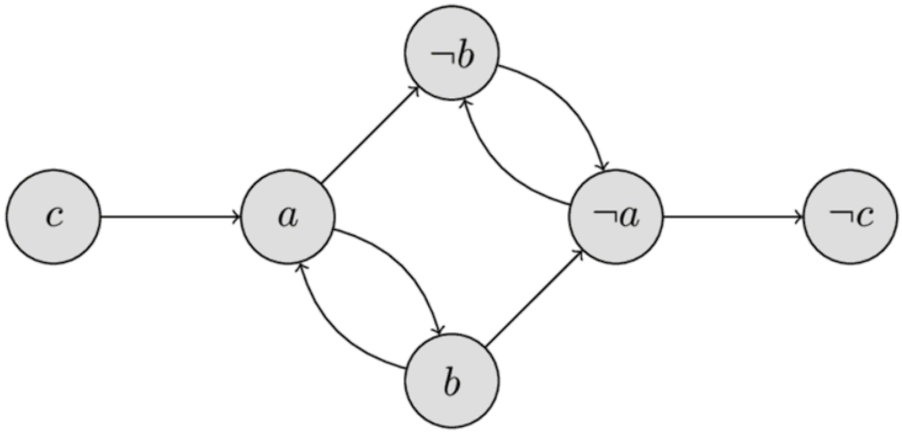
Es decir:

(A V B) es equivalente *$(\neg A \rightarrow B) \wedge (\neg B \rightarrow A)$*

Grafo de Implicación

Con esto podemos construir un grafo dirigido en el que para cada variable x_i, habrá dos nodos. Las aristas corresponderán a las implicaciones.

Por ejemplo, para la siguiente fórmula en forma normal conjuntiva ϕ= $(a \vee \neg b) \wedge (\neg a \vee b) \wedge (\neg a \vee \neg b) \wedge (a \vee \neg c)$, obtenemos este grafo:



Número de literales (l) = 2

Número de cláusulas (m) = 4

Número de nodos = 2*l = 2*3 = 6

Número de aristas = 2*m = 2*4 = 8

Si *x* es accesible desde ¬*x* y ¬*x* es accesible desde *x*, entonces el problema no tiene solución. Si *x* es verdadero entonces la implicación nos dice que ¬*x* también debería ser verdadero. Es decir, si se puede llegar a un vértice desde un segundo y al segundo se puede llegar desde el primero, entonces estos dos vértices están en el mismo componente fuertemente conectado.

Por tanto, para que 2-SAT tenga solución es necesario y suficiente que para cualquier variable *x*, los vértices *x* y ¬*x* estén en diferentes componentes fuertemente conexas (CFC) del grafo. Este criterio se puede verificar en *O(l+m)* tiempo encontrando todas las CFC con el algoritmo de Kosaraju o Tarjan.

También, podemos tratarlo cómo que ϕ es satisfactoria si G no contiene un ciclo que contenga tanto *xi* como ¬*xi* para cualquier *i*. A ese ciclo lo llamaremos ciclo de inconsistencia. Probar si G contiene un ciclo de inconsistencia se realiza fácilmente en tiempo polinómico con un algoritmo de marcado o un algoritmo de búsqueda en profundidad.

Construcción del algoritmo

En el primer paso, construimos el grafo de implicaciones y encontramos todos los componentes fuertemente conexos (CFC).

Clasificamos los CFC en orden topológico y asignamos valores a las variables basándonos en esta clasificación. Si el CFC de x es menor que el de $\neg x$, entonces x=*FALSE*, y x=*TRUE* de lo contrario. Esto se puede lograr con ayuda del algoritmo de Kosaraju en tiempo $O(l+m)$.

Se demuestra que esta asignación no conduce a contradicciones. Por ejemplo, si x = *TRUE*, se argumenta que no hay un camino desde x hasta ¬x en el grafo. Además, se demuestra que no puede existir una variable y tal que tanto y como $\neg y$ sean accesibles desde x, ya que esto llevaría a una contradicción (por la propiedad del gráfico de implicación $\neg x$ < es accesible desde ambos y e $\neg y$).

Conclusiones del algoritmo

Este algoritmo demuestra que 2SAT está en P porque todos los pasos del algoritmo se pueden realizar en tiempo polinomial.

En particular, la construcción del grafo y la búsqueda de CFC se pueden hacer en tiempo lineal en el tamaño de la fórmula 2SAT, y la verificación de la asignación y la asignación de valores de verdad también se pueden hacer en tiempo lineal.

Cómo hemos comentado anteriormente existen diversas formas de tratar este problema, similares entre ellas, pero la mostrada en este documento es la más comúnmente usada e intuitiva.

Cabe destacar que el problema 2SAT a diferencia del problema SAT general, que es NP-completo, el problema se puede resolver en tiempo polinomial por lo que una reducción polinómica entre ellos demostraría que P=NP.

Referencias

- https://people.seas.harvard.edu/~cs125/fall14/lec19.pdf
- https://www.geeksforgeeks.org/
- https://cstheory.stackexchange.com/questions/
- https://en.wikipedia.org/wiki/2-satisfiability
- https://math.stackexchange.com/questions/