Algoritmia y Complejidad

Ejercicio 24 de Sipser

Juan Carlos Alcausa Luque

Universidad de Málaga

Índice

1. Enunciado

Una 2cnf-fórmula es un AND de cláusulas, donde cada cláusula es un OR de a lo sumo dos literales. Dado $2SAT = \{\langle\phi\rangle \mid \phi$ es una 2cnf-fórmula$\}$ se pide demostrar que $2SAT \in P$.

2. Satisfacibilidad de 2-CNF

La cláusula $(x \vee y)$ es lógicamente equivalente a cada una de las expresiones $(\neg x \to y)$ y $(\neg y \to x)$. Representamos la fórmula 2-CNF ϕ sobre las variables $x_1, \ldots, x_m$ mediante un grafo dirigido G con $2m$ nodos etiquetados con los literales sobre estas variables. Para cada cláusula en ϕ, colocamos dos aristas en el grafo correspondientes a las dos implicaciones mencionadas. Adicionalmente, colocamos una arista de $\neg x$ a x si (x) es una cláusula unitaria y la arista inversa si $(\neg x)$ es una cláusula.

Teorema 1. *ϕ es satisfacible si y solo si G no contiene un ciclo que contenga tanto x_i como $\neg x_i$ para algún i.*

Llamaremos a tal ciclo un *ciclo de inconsistencia*. Verificar si G contiene un ciclo de inconsistencia se puede realizar fácilmente en tiempo polinómico con un algoritmo de marcado, o un algoritmo de búsqueda en profundidad.

Demostración. Demostramos que G contiene un ciclo de inconsistencia si y solo si no existe una asignación satisfactoria.

Comenzamos suponiendo, por reducción al absurdo, que G contiene un ciclo de inconsistencia. Como la secuencia de implicaciones en cualquier ciclo de inconsistencia produce la equivalencia lógica $x_i \leftrightarrow \neg x_i$ para algún i, llegamos a contradicción y por tanto, si G contiene un ciclo de inconsistencia, ϕ debe ser insatisfacible.

A continuación, demostramos el recíproco. Escribimos $x \xrightarrow{*} y$ si G contiene un camino del nodo x al nodo y. Debido a que G contiene las dos aristas designadas para cada cláusula en ϕ, tenemos $x \xrightarrow{*} y$ si y solo si $\neg y \xrightarrow{*} \neg x$.

Si G no contiene un ciclo de inconsistencia, construimos una asignación satisfactoria para ϕ de la siguiente manera:

1. Escogemos cualquier variable x_i. No podemos tener tanto $x_i \xrightarrow{*} \neg x_i$ como $\neg x_i \xrightarrow{*} x_i$ porque G no contiene un ciclo de inconsistencia.

2. Seleccionamos el literal x_i si $x_i \xrightarrow{*} \neg x_i$ es falso, y en otro caso seleccionamos $\neg x_i$.

3. Asignamos el literal seleccionado y todos los literales implicados (aquellos alcanzables a lo largo de caminos desde el nodo seleccionado) como Verdadero.

4. Nótese que nunca asignamos tanto x_j como $\neg x_j$ a Verdadero porque si $x_i \xrightarrow{*} x_j$ y $x_i \xrightarrow{*} \neg x_j$ entonces $\neg x_j \xrightarrow{*} \neg x_i$ y por lo tanto $x_i \xrightarrow{*} \neg x_i$, y no habríamos seleccionado el literal x_i (similarmente para $\neg x_i$).

5. Luego, eliminamos todos los nodos etiquetados con literales asignados o sus complementos de G, y repetimos este procedimiento hasta que todas las variables sean asignadas.

La asignación resultante satisface cada implicación y por lo tanto cada cláusula en ϕ. Así, ϕ es satisfacible.

$\square$

3. Conclusión

Este algoritmo permite verificar la satisfacibilidad de cualquier fórmula 2-CNF en tiempo polinómico, transformando el problema en la detección de ciclos de inconsistencia en un grafo dirigido. La implementación práctica puede realizarse mediante algoritmos estándar de búsqueda en grafos.