1.5 - 2SAT

Algorisme per a comprovar si la fòrmula F és satisfactible.

1. Contruir el graf d'implicacions G_F de la instància d'entrada i trobar les seves components fortament connexes.

2. Mirar si alguna de les components fortament connexes conté una variable i la seva negació. Si és el cas, aleshores la fòrmula és insatisfactible i acabem. Altrament serà satisfactible.

Aquesta darrera propietat és suficient per decidir si la fòrmula F és satisfactible i la podem, doncs, utilitzar la seva comprovació per dissenyar l'algorisme que decideix la satisfactibilitat. Demostració:

- $\Rightarrow$ Si a una component connexa forta hi són x i $\bar{x}$, sabem que al graf d'implicacions G_F hi ha camins de x a $\bar{x}$, i de $\bar{x}$ a x. Tenint en compte que els arcs de G_F es correspon amb implicacions lògiques correctes, deduïm que $x \Rightarrow \bar{x}$ i $\bar{x} \Rightarrow x$; deduïm, doncs, una contradicció de F i, per tant, F no és satisfactible.

- $\Leftarrow$ Hem de veure que, si no hi ha cap component connexa forta de G_F que contingui tant un literal com la seva negació, aleshores la fòrmula F és satisfactible. Demostrarem el recíproc: suposem que la fòrmula F és no satisfactible; això implica que podem deduir una contradicció de la forma $x \iff \bar{x}$ per a alguna variable x. Per tant, x i $\bar{x}$ pertanyen a la mateixa component connexa forta.

En cas que la fòrmula sigui satisfactible, volem també calcular una de les assignacions a les variables que produeixi aquesta satisfactibilitat. L'algorisme serà el següent:

1. Construir el graf condensat del graf d'implicacions (el graf que té un vèrtex per cada component fortament connexa, i una aresta de la component i a la component j sempre que el graf original contingui una arc (u, v) tal que u pertany a la component connexa i i v pertany a la component connexa j.

2. Ordenar topològicament el graf condensat (noteu que el graf condensat és un graf dirigit acíclic i, per tant, es pot). Podem fer servir l'algorisme de Kosaraju o Tarjan, però ens va millor aquest últim perquè ja genera l'ordre topològic invers.

Una vegada tenim l'ordenacio topològica inversa, podem començar a assignar valors als literals. L'assignació de valors la farem de la següent manera: Per a cada component en l'ordre topològic invers, si les seves variables no tenen ja una assignació a TRUE, posar tots els literals a la component a TRUE. Això fa que a tots els literals complementaris d'aquestes a altres components, se'ls hi assigni FALSE. Quan un literal es posa a TRUE, tots els literals que són accessibles a partir d'ell via una cadena d'implicacions també es posaran a TRUE. De manera simètrica, quan un literal és assignat a FALSE, tots els literals que porten a ell via una cadena d'implicacions s'assignaran també a FALSE.

El cost d'aquest algorisme és $O(n+m)$, on $n = |V|$ i $m = |E|$ de G_F.