Problème de satisfiabilité

Algorithme de Davis-Putnam-Logemann-Loveland(DPLL)

Question 1 : Donner le pseudocode de chaque algorithme approché.

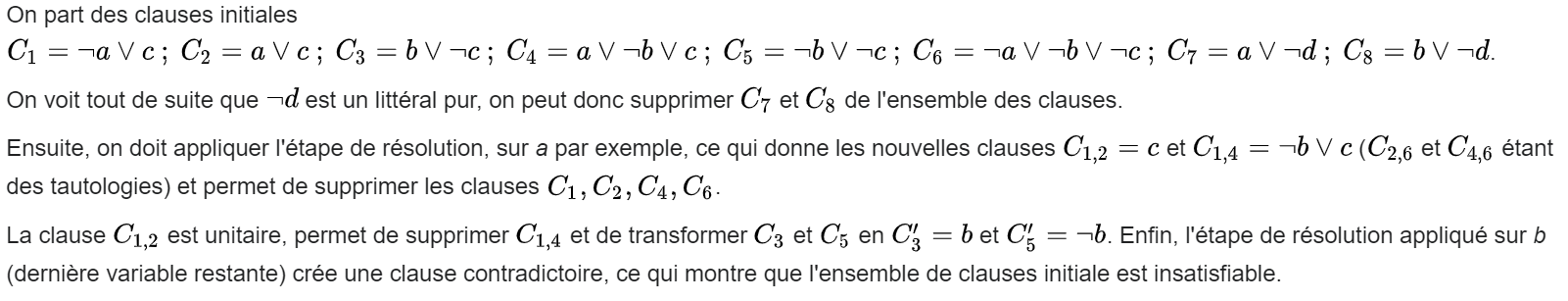
En calcul propositionnel, l'algorithme de Davis-Putnam est une méthode de détermination de la satisfiabilité d'une formule en forme normale conjonctive, c'est-à-dire une conjonction de clauses (disjonctions de littéraux). Il a été développé par Martin Davis et Hilary Putnam.

Cet algorithme est essentiellement une boucle sur l'application de trois règles, dont les deux premières se retrouvent dans l'algorithme DPLL. On observera que chaque règle transforme une formule de départ (disons f) en une autre formule f' (ayant moins de littéraux) avec la garantie que la non-satisfiabilité de f' implique la non-satisfiabilité de f. Ainsi l'algorithme termine (puisque le nombre de variables est fini) et par transitivité la non-satisfiabilité de la formule finale implique celle de la formule donnée en entrée.

Question 2 : Expliquer l'utilité des différents paramètres de l'algorithme.

Pour chaque clause unitaire (ne contenant qu'un littéral), on supprime les clauses comprenant ce littéral, et on enlève le littéral opposé des autres clauses. Par exemple, une clause x montre que x est vrai, on peut supprimer toutes les clauses de la forme A ∨ x et supprimer tous les ¬ x des autres clauses. Cette règle peut se répéter jusqu'à ce qu'il ne reste plus de clause unitaire.

Si une variable apparaît exclusivement sous la forme de littéral positif (ou de littéral négatif) dans l'ensemble de clauses, on peut supprimer toutes les clauses dans laquelle elle apparaît. Par exemple, si on ne trouve aucune clause de la forme ¬ x, toutes les clauses de la forme A ∨ x peuvent être supprimées. Là encore, cette règle peut se répéter et se combiner.

Question 3 : Comment peut-on appliquer chaque algorithme approché pour résoudre votre problème d'optimisation?