Aix-Marseille Université

Compléxité

Le Solveur Minisat

Autheur:
Michaël Gileta
Yohan Roux

Référent:
Kévin Perrot

Vendredi 13 Octobre

Exercice 1

Question a

(A OR B OR NOT(C) OR D)
AND
(NOT(B) OR C)
AND
(NOT (A) OR NOT (D))

Question b

Conversion format intermédiaire

$(1\|2\|-3\|4)$
&
$(-2\|3)$
&
$(-1\|-4)$

Conversion format Minisat

p cnf 4 3
1 2 -3 4 0
-2 3 0
-1 -4 0

Réponse Minisat

La formule est satisfaisable.

Question c

i.

$\Phi = (\neg t \to \neg s) \to (((b \vee t) \to s) \wedge ((r \wedge m) \to (b \vee a)) \wedge \neg r)$

1. $(t \vee \neg s) \equiv (\neg t \to \neg s)$
2. $(b \vee t) \to s \equiv (\neg b \wedge \neg t) \vee s \equiv (s \vee \neg b) \wedge (s \vee \neg t)$
3. $(r \wedge m) \to (b \vee a) \equiv (\neg r \vee \neg m) \vee (b \vee a) \equiv (\neg r \vee \neg m \vee b \vee a)$

$$\begin{aligned}
\Phi_2 &= (t \vee \neg s) \to ((s \vee \neg b) \wedge (s \vee \neg t) \wedge (\neg r \vee \neg m \vee b \vee a) \wedge \neg r) \\
&= (\neg t \wedge s) \vee ((s \vee \neg b) \wedge (s \vee \neg t) \wedge (\neg r \vee \neg m \vee b \vee a) \wedge \neg r) \\
&= (\neg t \vee s \vee \neg b) \wedge (\neg t \vee s) \wedge (\neg t \vee \neg r \vee \neg m \vee b \vee a) \wedge (\neg r \vee \neg t) \wedge (s \vee \neg b) \wedge (s \vee \neg t) \wedge (s \vee \neg r \vee \neg m \vee b \vee a) \wedge (\neg r \vee s) \\
&= (\neg t \vee s \vee \neg b) \wedge (\neg t \vee s) \wedge (\neg t \vee \neg r \vee \neg m \vee b \vee a) \wedge (\neg r \vee \neg t) \wedge (s \vee \neg b) \wedge (s \vee \neg r \vee \neg m \vee b \vee a) \wedge (s \vee \neg r)
\end{aligned}$$

ii.

(NOT(T) OR S OR NOT(B))
AND
(NOT(T) OR S)
AND
(NOT(T) OR NOT(R) OR NOT(M) OR B OR A)
AND
(NOT(R) OR NOT(T))
AND
(S OR NOT(B))
AND
(S OR NOT(R) OR NOT(M) OR B OR A)
AND
(S OR NOT(R))

iii.

p cnf 6 8
-1 2 -4 0
-1 2 0
-1-5 -6 4 3 0
-5 -1 0
2 -4 0 2 -5 -6 4 3 0
2 -5 0

Correspondance entre les numéros des variables et leur significations.
T = 1, S = 2, A = 3, B = 4, R = 5, M = 6
Lors de la première exécution, la formule est satisfaisable avec ce résultat : -1 -2 -3 -4 -5 -6 0
Ce qui correspond à l'affectation des valeurs correspondantes :
T = 0, S = 0, A = 0, B = 0, R = 0, M = 0

Pour avoir une autre solution il suffit d'ajouter le négatif de la solution trouvé en premier lieu en clause.
Nouvelle clause ajouté à la fin du fichier: 1 2 3 4 5 6 0
On obtient bien une autre solution: -1 2 -3 -4 -5 -6 0
T = 0, S = 1, A = 0, B = 0, R = 0, M = 0

Question d

Fonction Test
Entrée : un nombre binaire i et Φ une formule.
Sortie : SAT si Φ est Satisfaisable avec i comme modèle, INSAT sinon

Fonction IsTautology
Entrée : un nombre binaire i et Φ une formule.
Sortie : SAT si Φ est Satisfaisable avec i comme modèle, INSAT sinon

```
begin
    b = 2^NbVar − 1 for i : 0 → b do
        if Test(bytes(i), Φ) then
            Retourne FAUX
        else
        end
    end
    Retourne Vrai
end
```

Question e

i.

Un seul objet par tiroir :
$\rightarrow$ Soit $C_{i,j}$ l'objet i $(1 < i < n+1)$ dans le tiroir j $(1 < j < n)$
$\bigwedge_{i \to n+1} (\bigvee_{j \to n} C_{ij})$

Un tiroir ne peut avoir qu'un seul objet :
$\bigwedge_{i \to n+1} (\bigvee_{i \to n+1} (C_{ij} \bigwedge_{i \neq k}^{k \to n+1} \neg C_{kj})$

ii.

Test sur la valeur n=2, voir le fichier e.cnf

iii.

Si l'on fait croitre n, le nombre de clause augmente de façon exponentiel.

Exercice 2

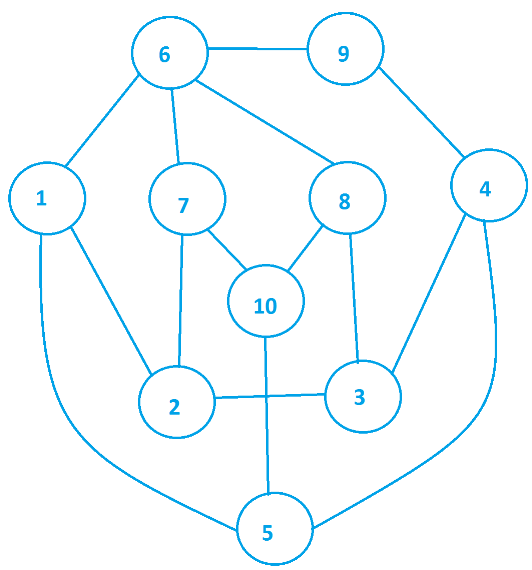
1. Soit une réduction polynomiale d'un problème 3-COL à SAT
   Un sommet peut prendre 3 valeurs possibles, une par couleur soit RGB (Red, Green, Blue).
   Ces valeurs sont évalué à 1 si le sommet a cette couleur, et à 0 sinon.
   Soit R1,G1,B1, le triplet de valeur pour le sommet 1.

   Pour chaque sommet, on obtient 4 clauses.
   (R1 ∨ G1 ∨ B1) ∧ (¬R1 ∨ ¬G1) ∧ (¬R1 ∨ ¬B1) ∧ (¬B1 ∨ ¬G1)
   Ces 4 clauses sont pour chaque sommet appartenant au graph.

   Ensuite en fonction des arrêtes du graph, plusieurs clauses se froment.
   Pour une arrête entre le sommet 1 et 2. On génére les 3 clauses suivantes.
   (¬R1 ∨ ¬R2) ∧ (¬G1 ∨ ¬G2) ∧ (¬B1 ∨ ¬B2)
   On génére donc ces 3 clauses pour chaque arrêtes du graphes.

   On obteint donc :
   $$\bigwedge_{i \to nbSommet} (Ri \vee Gi \vee Bi) \wedge (\neg Ri \vee \neg Gi) \wedge (\neg Ri \vee \neg Bi) \wedge (\neg Bi \vee \neg Gi)$$
   $$\bigwedge_{i,j \to nbSommet} (\neg Ri \vee \neg Rj) \wedge (\neg Gi \vee \neg Gj) \wedge (\neg Bi \vee \neg Bj)$$

2. Graph :



3. Voir programme joint

4. Voir programme joint

5. Voir programme joint

6. Quand on fait grandir la taille du graph, le nombre de clause vaut :
   4xNbSommets + 3xNbArretes, ce qui se fait rapidement.
   Mais le temps de résolution est de plus en plus long.