Complexité des Algorithmes et des Problèmes

Application pratique du problème SAT

Contrôle d'accès à une salle serveur

Dr Moustapha Diaby – ESATIC

Exercice : contrôle d'accès à une salle serveur

Une entreprise possède une salle contenant des serveurs critiques. Pour contrôler l'accès à cette salle, le système de sécurité utilise trois mécanismes :

- un **badge d'accès** ;
- une **empreinte biométrique** ;
- une **autorisation administrateur**.

On définit les variables booléennes suivantes :

$$B = \begin{cases} 1 & \text{si le badge est valide,} \\ 0 & \text{sinon,} \end{cases}$$

$$E = \begin{cases} 1 & \text{si l'empreinte biométrique est validée,} \\ 0 & \text{sinon,} \end{cases}$$

et

$$A = \begin{cases} 1 & \text{si l'autorisation administrateur est accordée,} \\ 0 & \text{sinon.} \end{cases}$$

Ainsi, la configuration

$$(B, E, A) = (1, 1, 0)$$

signifie que le technicien possède un badge valide et une empreinte biométrique reconnue, mais ne dispose pas d'une autorisation administrateur.

Règles de sécurité

L'entreprise impose les règles suivantes :

Règle 1. Le badge ou l'autorisation administrateur doit être valide.

Règle 2. Si le badge est utilisé, alors l'empreinte biométrique doit être validée.

Règle 3. Si l'empreinte biométrique n'est pas validée, alors une autorisation administrateur est obligatoire.

Règle 4. L'autorisation administrateur et l'empreinte biométrique ne doivent pas être utilisées simultanément.

Rappel

Le problème consiste à répondre à la question suivante :

Existe-t-il une configuration respectant toutes les règles ?

Il s'agit d'un problème de décision de type SAT.

Travail demandé

1. **Modélisation des règles**
   Traduire chacune des quatre règles de sécurité en une expression logique utilisant :

   $$\wedge, \qquad \vee, \qquad \neg.$$

2. **Transformation des implications**
   La deuxième règle peut s'écrire :

   $$B \Rightarrow E.$$

   Transformer cette implication en une clause utilisant uniquement $\vee$ et $\neg$.
   Faire de même pour la troisième règle :

   $$\neg E \Rightarrow A.$$

3. **Construction de la formule SAT**
   Construire la formule booléenne F représentant simultanément les quatre règles de sécurité.
4. **Premier technicien**
   Un premier technicien possède la configuration :

   $$(B, E, A) = (1, 1, 0).$$

   Vérifier chacune des quatre contraintes.
   Peut-il accéder à la salle serveur ?
5. **Deuxième technicien**
   Un deuxième technicien possède la configuration :

   $$(B, E, A) = (1, 0, 0).$$

   Vérifier chacune des contraintes.
   Peut-il accéder à la salle serveur ?
6. **Recherche de toutes les configurations**
   Comme il existe trois variables booléennes, le nombre total de configurations possibles est :

   $$2^3 = 8.$$

   Compléter le tableau suivant :

B	E	A	$B \vee A$	$\neg B \vee E$	$E \vee A$	$\neg A \vee \neg E$	F
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Déterminer toutes les configurations satisfaisantes.

En déduire la réponse au problème :

$$\boxed{\text{SAT}(F) = ?}$$

7. Nouvelle exigence

Le responsable informatique souhaite maintenant savoir s'il existe une configuration permettant d'accéder à la salle **sans utiliser l'autorisation administrateur**.

Cette nouvelle contrainte s'écrit :

$$A = 0.$$

Exprimer cette contrainte sous forme logique et construire la nouvelle formule :

$$F' = F \wedge \neg A.$$

La formule F' est-elle satisfiable ?

Si oui, donner une configuration satisfaisante.

8. Explosion combinatoire

Supposons maintenant que le système utilise n mécanismes de sécurité indépendants pouvant chacun être activés ou désactivés.

a) Combien existe-t-il de configurations possibles ?

b) Calculer ce nombre pour $n = 10$.

c) Calculer ce nombre pour $n = 20$.

d) Calculer ce nombre pour $n = 50$.

e) Quelle est la complexité d'une recherche exhaustive examinant toutes les configurations ?

9. Lien avec la classe NP

Supposons que le système comporte maintenant plusieurs centaines de mécanismes et de règles de sécurité.

Comparer les deux opérations suivantes :

— rechercher une configuration respectant toutes les règles ;

— vérifier qu'une configuration donnée respecte toutes les règles.

Expliquer pourquoi une configuration satisfaisante constitue un **certificat**.

En déduire pourquoi :

$$\boxed{\text{SAT} \in NP.}$$

Correction

1. **Modélisation des règles**
   La première règle :

   Le badge ou l'autorisation administrateur doit être valide.

   devient :

   $$\boxed{C_1 = B \vee A.}$$

   La deuxième règle :

   $$B \Rightarrow E$$

   devient :

   $$\boxed{C_2 = \neg B \vee E.}$$

   La troisième règle :

   $$\neg E \Rightarrow A$$

   devient :

   $$\boxed{C_3 = E \vee A.}$$

   Enfin, la quatrième règle interdit :

   $$A \wedge E.$$

   On écrit donc :

   $$\neg(A \wedge E).$$

   D'après la loi de De Morgan :

   $$\neg(A \wedge E) = \neg A \vee \neg E.$$

   Ainsi :

   $$\boxed{C_4 = \neg A \vee \neg E.}$$

2. **Transformation des implications**
   On utilise l'équivalence :

   $$X \Rightarrow Y \equiv \neg X \vee Y.$$

   Pour :

   $$B \Rightarrow E,$$

   on obtient :

   $$\boxed{\neg B \vee E.}$$

   Pour :

$$\neg E \Rightarrow A,$$

on obtient :

$$\neg(\neg E) \vee A.$$

Donc :

$$\boxed{E \vee A.}$$

3. Formule SAT

Toutes les contraintes doivent être vraies simultanément.
On obtient donc :

$$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4.$$

Ainsi :

$$\boxed{F = (B \vee A) \wedge (\neg B \vee E) \wedge (E \vee A) \wedge (\neg A \vee \neg E)}$$

La question SAT est :

$$\boxed{\text{Existe-t-il } (B, E, A) \in \{0,1\}^3 \text{ tel que } F = 1 \text{ ?}}$$

4. Premier technicien

Pour :

$$(B, E, A) = (1, 1, 0),$$

on obtient :

$$C_1 = 1 \vee 0 = 1,$$

$$C_2 = \neg 1 \vee 1 = 0 \vee 1 = 1,$$

$$C_3 = 1 \vee 0 = 1,$$

et :

$$C_4 = \neg 0 \vee \neg 1 = 1 \vee 0 = 1.$$

Ainsi :

$$F = 1 \wedge 1 \wedge 1 \wedge 1 = 1.$$

Donc :

$$\boxed{\text{Accès autorisé.}}$$

5. Deuxième technicien

Pour :

$$(B, E, A) = (1, 0, 0),$$

la deuxième clause donne :

$$C_2 = \neg 1 \vee 0 = 0 \vee 0 = 0.$$

De plus :

$$C_3 = 0 \vee 0 = 0.$$

Donc :

$$F = 0.$$

Ainsi :

$$\boxed{\text{Accès refusé.}}$$

Le badge est valide, mais l'exigence biométrique n'est pas satisfaite.

6. Recherche de toutes les configurations

Le tableau complet est :

B	E	A	C_1	C_2	C_3	C_4	F
0	0	0	0	1	0	1	0
0	0	1	1	1	1	1	1
0	1	0	0	1	1	1	0
0	1	1	1	1	1	0	0
1	0	0	1	0	0	1	0
1	0	1	1	0	1	1	0
1	1	0	1	1	1	1	1
1	1	1	1	1	1	0	0

Les deux configurations satisfaisantes sont :

$$\boxed{(B, E, A) = (0, 0, 1)}$$

et :

$$\boxed{(B, E, A) = (1, 1, 0).}$$

Par conséquent :

$$\boxed{\text{SAT}(F) = \text{OUI.}}$$

7. Accès sans autorisation administrateur

La condition :

$$A = 0$$

s'écrit :

$$\neg A.$$

La nouvelle formule est donc :

$$\boxed{F' = F \wedge \neg A.}$$

D'après le tableau précédent, la configuration :

$$(B, E, A) = (1, 1, 0)$$

satisfait F et vérifie également :

$$\neg A = 1.$$

Ainsi :

$$\boxed{F' \text{ est satisfiable.}}$$

Une solution est :

$$\boxed{(B, E, A) = (1, 1, 0).}$$

L'interprétation est simple : sans autorisation administrateur, le technicien doit utiliser la procédure normale :

$$\boxed{\text{badge valide + empreinte biométrique validée.}}$$

8. Explosion combinatoire

Chaque mécanisme possède deux états possibles.

Pour n mécanismes, il existe donc :

$$\boxed{2^n}$$

configurations.

Par exemple :

Nombre de mécanismes	**Configurations possibles**
3	8
10	1 024
20	1 048 576
50	1 125 899 906 842 624

Une méthode naïve examinant toutes les configurations possède donc une croissance exponentielle :

$$\boxed{O(2^n)}$$

si l'on considère uniquement le nombre de configurations testées.

9. Lien avec NP

Trouver une configuration satisfaisante peut nécessiter l'exploration d'un grand espace de recherche.

En revanche, si une configuration candidate est fournie, par exemple :

$$(B, E, A) = (1, 1, 0),$$

il suffit de vérifier :

$$B \vee A,$$

$$\neg B \vee E,$$

$$E \vee A,$$

et :

$$\neg A \vee \neg E.$$

La vérification d'une affectation donnée peut être effectuée efficacement en parcourant les clauses de la formule.

L'affectation satisfaisante constitue donc un **certificat**.

On retrouve ainsi l'idée fondamentale de la classe NP :

une solution candidate peut être vérifiée en temps polynomial.

Par conséquent :

$$\boxed{\text{SAT} \in NP.}$$

Interprétation pratique

Problème réel

Contrôler l'accès à une salle serveur

$\Downarrow$

Variables booléennes

$B, \quad E, \quad A$

$\Downarrow$

Traduction des règles de sécurité

$\Downarrow$

Construction de la formule SAT

$\Downarrow$

Existe-t-il une configuration satisfaisante ?

$\Downarrow$

Configuration de sécurité réalisable

À retenir

Le problème SAT permet ici de transformer une question concrète de cybersécurité :

Existe-t-il une configuration du système permettant l'accès tout en respectant toutes les règles de sécurité ?

en un problème mathématique :

Existe-t-il une affectation des variables telle que $F = 1$?

SAT est donc ici utilisé comme un problème de **faisabilité**.

Il ne cherche pas nécessairement la meilleure configuration. Il cherche à déterminer s'il existe **au moins une** configuration respectant simultanément toutes les contraintes.