Complexité des Algorithmes et des Problèmes

Application pratique du problème SAT

Configuration de la sécurité d'un réseau informatique

Dr Moustapha Diaby – ESATIC

Exercice : configuration sécurisée d'un réseau

Une entreprise souhaite mettre en place une configuration de sécurité pour son réseau informatique. L'administrateur dispose de trois mécanismes de sécurité :

- un **pare-feu** ;
- un **VPN** ;
- un **IDS** (Intrusion Detection System).

On définit les variables booléennes suivantes :

$$P = \begin{cases} 1 & \text{si le pare-feu est actif,} \\ 0 & \text{sinon,} \end{cases}$$

$$V = \begin{cases} 1 & \text{si le VPN est actif,} \\ 0 & \text{sinon,} \end{cases}$$

et

$$I = \begin{cases} 1 & \text{si l'IDS est actif,} \\ 0 & \text{sinon.} \end{cases}$$

Ainsi, une configuration

$$(P, V, I) = (1, 0, 1)$$

signifie que le pare-feu et l'IDS sont actifs, tandis que le VPN est désactivé.

Contraintes de sécurité

L'entreprise impose les règles suivantes :

Règle 1. Au moins le **pare-feu ou l'IDS** doit être actif.

Règle 2. Si le **VPN est actif**, alors le **pare-feu doit obligatoirement être actif**.

Règle 3. Le **VPN et l'IDS ne doivent pas être actifs simultanément**.

Règle 4. Si le **pare-feu est désactivé**, alors **l'IDS doit obligatoirement être actif**.

Rappel

L'objectif est de transformer ces règles de sécurité en une formule booléenne, puis de répondre à la question :

Existe-t-il une configuration qui respecte toutes les règles ?

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

Travail demandé

1. **Modélisation des contraintes**
   Traduire chacune des quatre règles de sécurité en une expression logique utilisant les opérateurs :
   $$\wedge, \qquad \vee, \qquad \neg.$$
2. **Transformation d'une implication**
   La deuxième règle peut d'abord s'écrire :
   $$V \Rightarrow P.$$
   Montrer que cette implication est équivalente à :
   $$\neg V \vee P.$$
3. **Construction de la formule SAT**
   Construire une formule booléenne F représentant simultanément toutes les contraintes de sécurité.
4. **Test d'une première configuration**
   L'administrateur propose :
   $$(P, V, I) = (1, 1, 1).$$
   Cette configuration signifie que le pare-feu, le VPN et l'IDS sont tous actifs.
   Vérifier chaque contrainte et déterminer si cette configuration satisfait la formule F.
5. **Test d'une deuxième configuration**
   L'administrateur propose maintenant :
   $$(P, V, I) = (1, 1, 0).$$
   Cette configuration respecte-t-elle toutes les règles ?
   Justifier la réponse.
6. **Recherche de toutes les configurations**
   Trois variables booléennes donnent :
   $$2^3 = 8$$
   configurations possibles.
   Compléter le tableau suivant.

P	V	I	$P \vee I$	$\neg V \vee P$	$\neg V \vee \neg I$	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				

Donner toutes les configurations satisfaisantes.
En déduire la réponse au problème :

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

7. Une exigence supplémentaire

Le responsable de la sécurité souhaite maintenant que le **VPN soit obligatoirement actif**.
La nouvelle contrainte est donc :

$$V = 1.$$

Construire une nouvelle formule :

$$F' = F \wedge V.$$

Existe-t-il une configuration satisfaisant F' ?
Si oui, donner une telle configuration.

8. Généralisation

Supposons maintenant que l'entreprise possède n mécanismes de sécurité pouvant chacun être activés ou désactivés.

a) Combien existe-t-il de configurations possibles ?
b) Combien existe-t-il de configurations pour $n = 10$?
c) Pour $n = 20$?
d) Pour $n = 50$?
e) Expliquer pourquoi une recherche exhaustive devient rapidement difficile.

9. Lien avec la classe NP

Supposons qu'une personne fournisse directement une configuration candidate.
Est-il difficile de vérifier que cette configuration respecte toutes les règles ?
Expliquer pourquoi une configuration satisfaisante peut être considérée comme un **certificat**.
En déduire le lien avec :

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

Correction

1. Modélisation

La première règle est :

> Au moins le pare-feu ou l'IDS doit être actif.

Elle devient :

$$\boxed{C_1 = P \vee I.}$$

La deuxième règle est :

> Si le VPN est actif, alors le pare-feu est actif.

Elle s'écrit :

$$V \Rightarrow P.$$

Donc :

$$\boxed{C_2 = \neg V \vee P.}$$

La troisième règle interdit que le VPN et l'IDS soient simultanément actifs :

$$\neg(V \wedge I).$$

En utilisant la loi de De Morgan :

$$\neg(V \wedge I) = \neg V \vee \neg I.$$

Ainsi :

$$\boxed{C_3 = \neg V \vee \neg I.}$$

Enfin, la quatrième règle est :

$$\neg P \Rightarrow I.$$

Or :

$$A \Rightarrow B \equiv \neg A \vee B.$$

En prenant $A = \neg P$, on obtient :

$$\neg(\neg P) \vee I = P \vee I.$$

Donc :

$$\boxed{C_4 = P \vee I.}$$

On remarque que C_4 est équivalente à C_1.

2. **Implication**
   L'implication :

$$V \Rightarrow P$$

   est fausse uniquement lorsque :

$$V = 1 \qquad \text{et} \qquad P = 0.$$

   La formule :

$$\neg V \vee P$$

   est également fausse uniquement dans cette situation.
   Ainsi :

$$\boxed{V \Rightarrow P \equiv \neg V \vee P.}$$

3. **Formule SAT**
   Toutes les contraintes doivent être vérifiées simultanément.
   Elles sont donc reliées par l'opérateur ET :

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

   On obtient :

$$\boxed{F = (P \vee I) \wedge (\neg V \vee P) \wedge (\neg V \vee \neg I)}$$

   La quatrième contrainte n'a pas besoin d'être répétée puisqu'elle est équivalente à la première.
4. **Configuration** $(1, 1, 1)$
   Considérons :

$$P = 1, \qquad V = 1, \qquad I = 1.$$

   Première clause :

$$P \vee I = 1 \vee 1 = 1.$$

   Deuxième clause :

$$\neg V \vee P = 0 \vee 1 = 1.$$

   Troisième clause :

$$\neg V \vee \neg I = 0 \vee 0 = 0.$$

   Donc :

$$F = 1 \wedge 1 \wedge 0 = 0.$$

   Ainsi :

$$\boxed{(1, 1, 1) \text{ n'est pas une configuration valide.}}$$

   La raison est simple : le VPN et l'IDS sont actifs simultanément, ce qui viole la troisième règle.

5. **Configuration** $(1, 1, 0)$
   Considérons :

$$P = 1, \qquad V = 1, \qquad I = 0.$$

   Nous obtenons :

$$P \vee I = 1 \vee 0 = 1,$$

$$\neg V \vee P = 0 \vee 1 = 1,$$

   et :

$$\neg V \vee \neg I = 0 \vee 1 = 1.$$

   Donc :

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

   Par conséquent :

$$\boxed{(P, V, I) = (1, 1, 0)}$$

   est une configuration satisfaisante.

6. **Toutes les configurations**
   Le tableau complet est :

P	V	I	$P \vee I$	$\neg V \vee P$	$\neg V \vee \neg I$	F
0	0	0	0	1	1	0
0	0	1	1	1	1	1
0	1	0	0	0	1	0
0	1	1	1	0	0	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	0	0

   Les configurations satisfaisantes sont donc :

$$\boxed{(0, 0, 1), \quad (1, 0, 0), \quad (1, 0, 1), \quad (1, 1, 0).}$$

   Puisqu'il existe au moins une configuration satisfaisante :

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

7. **VPN obligatoirement actif**
   On ajoute :

$$V = 1.$$

La nouvelle formule devient :

$$F' = F \wedge V.$$

Donc :

$$F' = (P \vee I) \wedge (\neg V \vee P) \wedge (\neg V \vee \neg I) \wedge V.$$

Une configuration satisfaisante est :

$$\boxed{(P, V, I) = (1, 1, 0).}$$

En effet :

$$F'(1, 1, 0) = 1.$$

Ainsi, il est possible d'activer le VPN tout en respectant toutes les contraintes.
Dans ce cas, les contraintes imposent :

$$\boxed{V = 1 \Longrightarrow P = 1 \quad \text{et} \quad I = 0.}$$

8. Explosion combinatoire

Chaque mécanisme possède deux états :

$$0 \quad \text{ou} \quad 1.$$

Avec n mécanismes, le nombre de configurations possibles est :

$$\boxed{2^n.}$$

Par exemple :

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

Une recherche exhaustive peut donc nécessiter l'examen de :

$$\boxed{2^n}$$

configurations.
La croissance est exponentielle.

9. Lien avec NP

Supposons que l'on fournisse directement la configuration :

$$(P, V, I) = (1, 1, 0).$$

Il suffit de vérifier les contraintes :

$$P \vee I, \qquad \neg V \vee P, \qquad \neg V \vee \neg I.$$

La vérification est donc facile.

Pour une formule comportant un grand nombre de variables et de clauses, le principe reste identique : on parcourt les clauses et on vérifie qu'elles sont satisfaites.
Une affectation satisfaisante constitue donc un **certificat**.
Sa validité peut être vérifiée en temps polynomial par rapport à la taille de la formule.
On obtient ainsi :

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

Interprétation pratique

Cet exercice illustre le principe général d'utilisation de SAT :

Problème réel

$\Downarrow$

Variables de décision booléennes

$P,\ V,\ I$

$\Downarrow$

Traduction des règles en contraintes logiques

$\Downarrow$

Construction d'une formule SAT

$\Downarrow$

Existe-t-il une affectation satisfaisante ?

$\Downarrow$

Configuration réalisable du système

Ainsi, SAT permet de répondre à une question concrète :

Existe-t-il une configuration des différents mécanismes de sécurité qui respecte simultanément toutes les contraintes imposées ?

Remarque. SAT est ici utilisé comme un problème de **faisabilité**. Il ne cherche pas nécessairement la meilleure configuration. Il cherche d'abord à déterminer s'il existe au moins une configuration respectant toutes les contraintes.