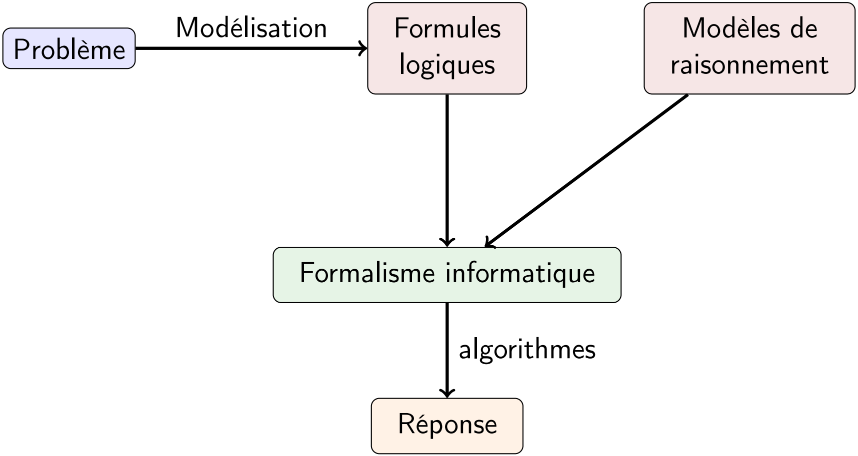
Logique et Déduction - Cours 2 - Utilisation de SAT-solveurs

Pierre-Cyrille Héam

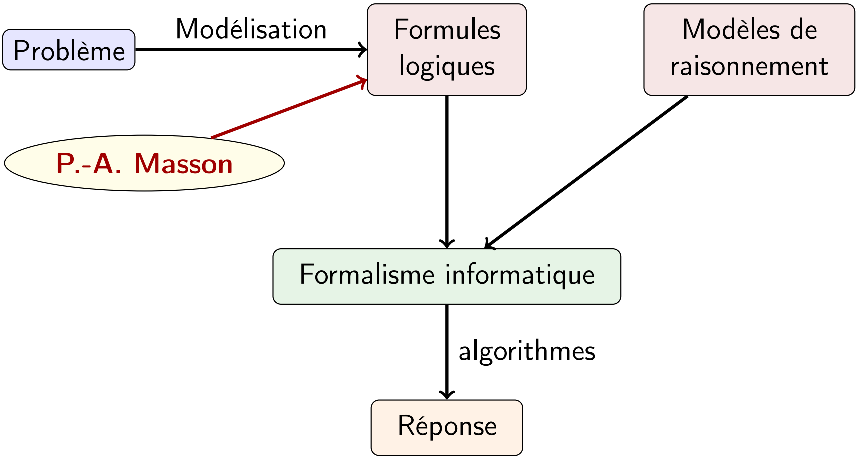
pheam [at] femto-st.fr

Licence 2 Informatique

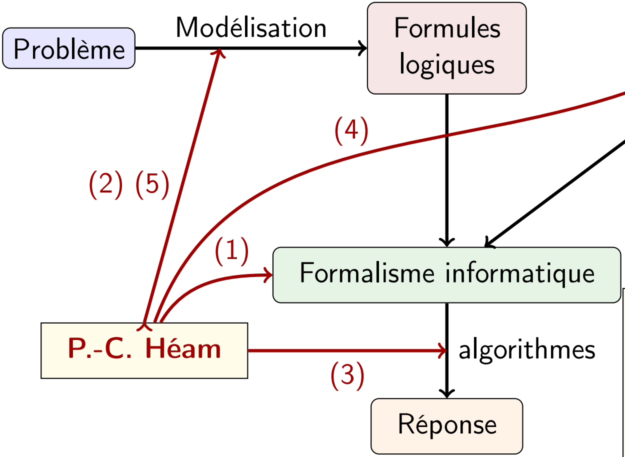
Logique et automatisation - organisation du module



Logique et automatisation - organisation du module



Logique et automatisation - organisation du module



Modélisation (5)
Modélisation SAT (2)
Modèle déductif (4)
Réécriture (1)
Résolution SAT (3)

Calculabité-Complexité

Problèmes demandant peu de ressource (temps)
Classe P (polynomiaux)

Problèmes demandant beaucoup de ressources
Classe EXP et au dessus (exponentiels et plus)

Problèmes indécidables
(non résolubles par un programme)

Calculabité-Complexité

Problèmes demandant peu de ressource (temps)
Classe P (polynomiaux)

Problèmes intermédiaires
Classe NP-complets

Problèmes demandant beaucoup de ressources
Classe EXP et au dessus (exponentiels et plus)

Problèmes indécidables
(non résolubles par un programme)

Exemple de problème NP-Complet : **SAT**

SAT

Donnée : Une formule de la logique propositionnelle avec k variables.
Sortie : Vrai, si la formule est satisfiable, faux sinon.

$$((A \wedge B) \vee C) \wedge \neg(B \vee A)) \wedge (B \vee \neg C)$$

La table de vérité contient-elle au moins une ligne avec un 1 ? (elle a 2^k lignes).

SAT est NP-complet (Théorème de Cook, 1971).

Reste NP-complet, même si la formule est sur forme normale conjonctive et toute les clauses ont au plus trois littéraux.

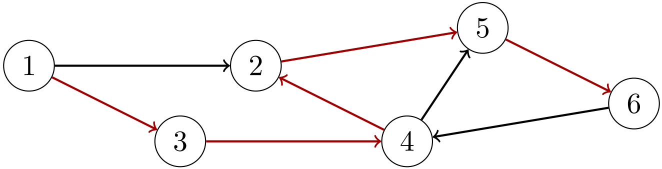
Exercice

- Combien de ligne possède une table de vérité d'une formule à n variables ?
- On suppose que l'on dispose d'un ordinateur qui tester 10^{10} lignes d'une table de vérité (beaucoup plus rapide que n'importe quel ordinateur actuel) par seconde.
- On suppose que l'on considère une formule avec 20 variables, combien de temps lui faut-il (à la louche) ? (on pourra prendre comme approximation que $2^{10} \simeq 10^3$)
- Même question avec 40 variables ?
- Même question avec 60 variables ?
- Même question avec 80 variables ?
- Même question avec 100 variables ?
- Même question avec 100 variables mais sur le plus gros calculateur du monde (environ 10^{17} opérations par seconde, en comptant un cycle de processeur par ligne de la table).

Exemple : **Chemin Hamiltonien**

Chemin Hamiltonien

Donnée : un graphe orienté $G = (V, E)$
Sortie : Vrai s'il existe un chemin qui passe une et une seule fois par chaque sommet, faux sinon.

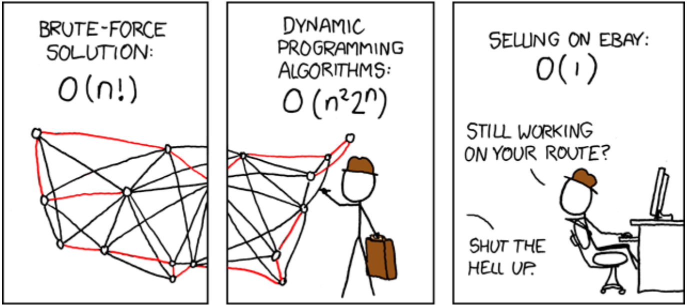


Exemple : **Voyageur de commerce**

Voyageur de Commerce

Donnée : un graphe orienté $G = (V, E)$
Sortie : un chemin de taille minimale qui passe par tous les sommets au moins une fois ?

De très nombreuses variantes utilisées en logistique.



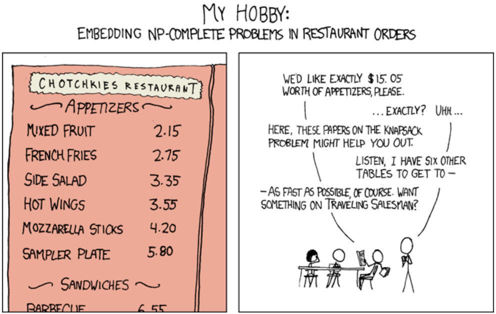
source https://xkcd.com/399/

Exemple : **Sac à dos**

Sac à dos

Données : un réel K et une suite de réels positifs $x_1, \ldots, x_k$ de valeurs d'objets et une suite $y_1, \ldots, y_n$ de poids d'objets.
Sortie : Un sous-ensemble des x_i de valeur maximale mais dont le poids total est inférieur à K.

De très nombreuses variantes d'optimisation des stocks.



source https://xkcd.com/287/

Exemple : **Mariages stables**

On se donne deux ensembles A et B, supposés disjoints, ayant chacun n éléments. On se donne aussi, pour chaque élément de A et B, une fonction de préférence, qui classe tous les éléments de l'autre ensemble. On cherche alors à associer de façon bijective les éléments de A avec ceux de B, pour qu'il n'existe pas $\in A$ et $b \in B$ tels que a préfère b à l'élément qui lui est associé, et b préfère a à l'élément qui lui est associé (définition `wikipedia`).

parcoursup

Entrez dans l'enseignement supérieur

Problèmes NP-complet

- Ordonnancement de tâches, allocation de ressources,
- Emplois du temps,
- Optimisation dans les réseaux,
- Finance,
- Sécurité informatiques,
- etc.

Il existe différentes approches : approximations, méta-heuristiques, utilisation de solvers.

Comment faire ?

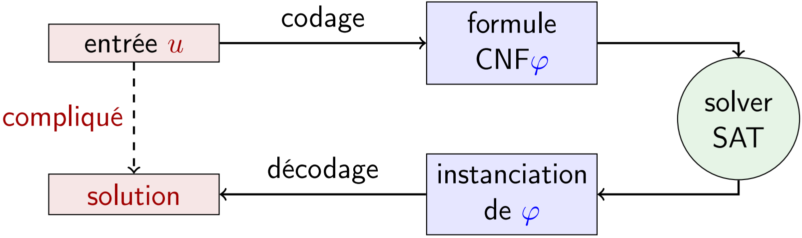
Résoudre un problème NP-complet par force brute (ou proche) demandera rapidement trop de ressources.

- Pour les problèmes de décision, utilisation de **SAT**-**solvers**.
- Pour les problèmes d'optimisation, utilisation de **solvers en nombres entiers**, d'approximation ou d'heuristiques, de méta-heuristiques.

Codage SAT

Codage SAT

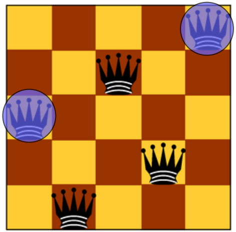
Résoudre un problème par codage SAT, c'est trouver une fonction calculable en temps polynomiale qui, à chaque entrée u du problème d'entrée, associe une formule SAT en CNF φ qui est satisfiable si et seulement si u est une instance positive du problème.
De plus, on chercher à ce qu'une instanciation de la formule (qui rende son interprétation vraie) donne une solution au problème.



Plan

Problème de complétion des dames

Comment placer $n - k$ dames sur un échiquier de taille $n \times n$ sans qu'aucune ne soit en prise d'une autre (au plus une par ligne, colonne et diagonales), sachant que k dames sont déjà placés.

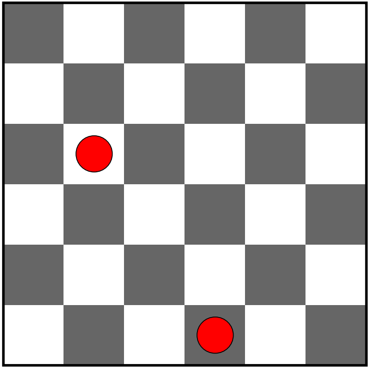


Le problème est pas NP-complet.

image `https://automaths.blog/2018/11/09/le-probleme-des-reines/`

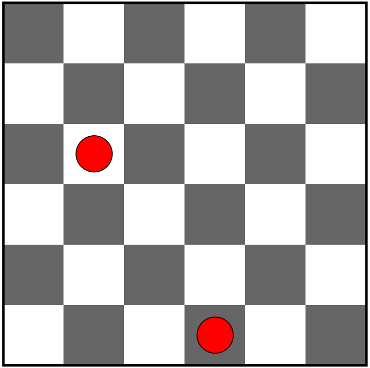
Exercice

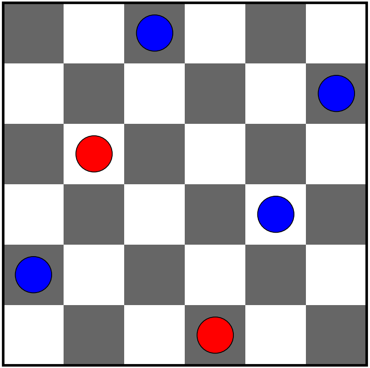
Placer 4 dames sur l'échiquier pour qu'aucune ne soit en prise avec une autre.



Exercice

Placer 4 dames sur l'échiquier pour qu'aucune ne soit en prise avec une autre.





Le dames : exemple

Placer 6 autres dames de telle sorte qu'elle ne puissent pas se prendre (une seule par ligne, colonne, diagonale).

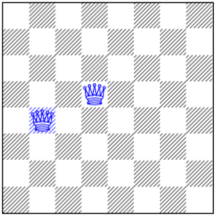
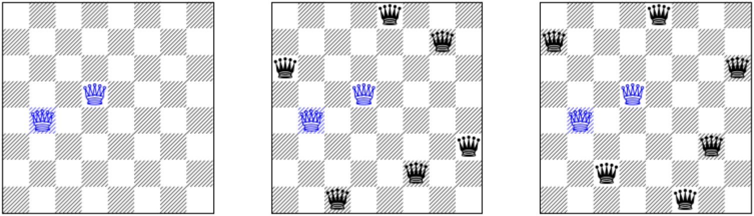


image tirée de *Complexity of n -Queens Completion*, de I.A. Gent, Ch. Jefferson et P. Nightingale, Journal of Artificial Intelligence Research, 59 (2017), pages 815 – 848.

Le dames : exemple

Placer 6 autres dames de telle sorte qu'elle ne puissent pas se prendre (une seule par ligne, colonne, diagonale).



Il n'y a que deux possibilités.

image tirée de *Complexity of n -Queens Completion*, de I.A. Gent, Ch. Jefferson et P. Nightingale, Journal of Artificial Intelligence Research, 59 (2017), pages 815 – 848.

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 1, choisir les variables : Positions des dames

- 9 variables $X_{i,j}$ pour coder la position de la première dame.
- 9 variables $Y_{i,j}$ pour coder la position de la deuxième dame.
- 9 variables $Z_{i,j}$ pour coder la position de la troisième dame.

$X_{1,3} = 1$ si et seulement si la première dame est en position $1, 3$.

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 1, choisir les variables : Positions des dames

- 9 variables $X_{i,j}$ pour coder la position de la première dame.
- 9 variables $Y_{i,j}$ pour coder la position de la deuxième dame.
- 9 variables $Z_{i,j}$ pour coder la position de la troisième dame.

$X_{1,3} = 1$ si et seulement si la première dame est en position $1, 3$.
27 variables, la table de vérité a 2^{27} (un peu plus de cent millions de lignes)

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 2, les contraintes :

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 2, les contraintes :

la première dame est sur une au moins case : $X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3}$

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 2, les contraintes :

la première dame est sur une au moins case : $X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3}$

la première dame est sur une case au plus :

$\neg(X_{i,j} \wedge X_{i',j'})$ pour $(i,j) \neq (i',j')$. (36 cas)

Codage sur un échiquier 3x3

1,1	1,2	1,3
2,1	2,2	2,3
3,1	3,2	3,3

Étape 2, les contraintes :
la première dame est sur une au moins case : $X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3}$
la première dame est sur une case au plus :
$\neg(X_{i,j} \wedge X_{i',j'})$ pour $(i,j) \neq (i',j')$. (36 cas)
Ce qui donne pour la première dame est sur exactement une case :

$$\varphi_X = X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(X_{i,j} \wedge X_{i',j'})$$

Codage sur un échiquier 3x3 (les contraintes)

la première dame est sur exactement une case :

$$\varphi_X = X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(X_{i,j} \wedge X_{i',j'})$$

la deuxième dame est sur exactement une case :

$$\varphi_Y = Y_{1,1} \vee Y_{1,2} \vee \ldots \vee Y_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(Y_{i,j} \wedge Y_{i',j'})$$

la deuxième dame est sur exactement une case :

$$\varphi_Z = Z_{1,1} \vee Z_{1,2} \vee \ldots \vee Z_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(Z_{i,j} \wedge Y_{i',j'})$$

Codage sur un échiquier 3x3 (les contraintes)

la première dame est sur exactement une case :

$$\varphi_X = X_{1,1} \vee X_{1,2} \vee \ldots \vee X_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(X_{i,j} \wedge X_{i',j'})$$

la deuxième dame est sur exactement une case :

$$\varphi_Y = Y_{1,1} \vee Y_{1,2} \vee \ldots \vee Y_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(Y_{i,j} \wedge Y_{i',j'})$$

la deuxième dame est sur exactement une case :

$$\varphi_Z = Z_{1,1} \vee Z_{1,2} \vee \ldots \vee Z_{3,3} \wedge \bigwedge_{(i,j)<(i',j')} \neg(Z_{i,j} \wedge Y_{i',j'})$$

Une instanciation qui satisfait $\varphi_X \wedge \varphi_Y \wedge \varphi_Z$ possède exactement un $X_{i,j}$ vrai, exactement un $Y_{i,j}$ vrai et exactement un $Z_{i,j}$ vrai.

Codage sur un échiquier 3x3 (les contraintes)

La première et la seconde dame ne peuvent pas être toute les deux sur la case $(1, 1)$: $\neg(X_{1,1} \wedge Y_{1,1})$

Codage sur un échiquier 3x3 (les contraintes)

La première et la seconde dame ne peuvent pas être toute les deux sur la case $(1,1)$: $\neg(X_{1,1} \wedge Y_{1,1})$
La première et la seconde dame ne peuvent pas être sur une même case :

$$\varphi_{X,Y} = \bigwedge_{(i,j)} \neg(X_{i,j} \wedge Y_{i,j})$$

Codage sur un échiquier 3x3 (les contraintes)

La première et la seconde dame ne peuvent pas être toute les deux sur la case $(1,1)$: $\neg(X_{1,1} \wedge Y_{1,1})$

La première et la seconde dame ne peuvent pas être sur une même case :

$$\varphi_{X,Y} = \bigwedge_{(i,j)} \neg(X_{i,j} \wedge Y_{i,j})$$

La première et la troisième dame ne peuvent pas être sur une même case :

$$\varphi_{X,Z} = \bigwedge_{(i,j)} \neg(X_{i,j} \wedge Z_{i,j})$$

La deuxième et la troisième dame ne peuvent pas être sur une même case :

$$\varphi_{Y,Z} = \bigwedge_{(i,j)} \neg(Y_{i,j} \wedge Z_{i,j})$$

Codage sur un échiquier 3x3 (les contraintes)

Un instanciation qui satisfait

$$\varphi_X \wedge \varphi_Y \wedge \varphi_Z \wedge \varphi_{X,Y} \wedge \varphi_{X,Z} \wedge \varphi_{Y,Z}$$

- possède exactement un $X_{i,j}$ vrai,
- exactement un $Y_{i,j}$ vrai et
- exactement un $Z_{i,j}$ vrai ;
- de plus couples de ces trois indices sont deux à deux distincts.

Le reines sont toutes sur l'échiquier sur des cases différentes.

Codage sur un échiquier 3x3 (les contraintes)

Un instanciation qui satisfait

$$\varphi_X \wedge \varphi_Y \wedge \varphi_Z \wedge \varphi_{X,Y} \wedge \varphi_{X,Z} \wedge \varphi_{Y,Z}$$

- possède exactement un $X_{i,j}$ vrai,
- exactement un $Y_{i,j}$ vrai et
- exactement un $Z_{i,j}$ vrai ;
- de plus couples de ces trois indices sont deux à deux distincts.

Le reines sont toutes sur l'échiquier sur des cases différentes.

Il reste à coder qu'elles ne sont pas en prises mutuelles (et les conditions initiales).

Codage sur un échiquier 3x3 (les contraintes)

Les autres contraintes (hors conditions initiales) sont

- Deux dames ne peuvent pas être sur une même ligne.
- Deux dames ne peuvent pas être sur une même colonne.
- Deux dames ne peuvent pas être sur une même diagonale.

Codage sur un échiquier 3x3 (les contraintes)

Les autres contraintes (hors conditions initiales) sont

- Deux dames ne peuvent pas être sur une même ligne.

$$\varphi_{\text{ligne}} = \bigwedge_{i,j \neq j'} \neg(X_{i,j} \wedge Y_{i,j'}) \wedge \neg(X_{i,j} \wedge Z_{i,j'}) \wedge \neg(Y_{i,j} \wedge Z_{i,j'})$$

- Deux dames ne peuvent pas être sur une même colonne.

- Deux dames ne peuvent pas être sur une même diagonale.

Codage sur un échiquier 3x3 (les contraintes)

Les autres contraintes (hors conditions initiales) sont

- Deux dames ne peuvent pas être sur une même ligne.

$$\varphi_{\text{ligne}} = \bigwedge_{i,j \neq j'} \neg(X_{i,j} \wedge Y_{i,j'}) \wedge \neg(X_{i,j} \wedge Z_{i,j'}) \wedge \neg(Y_{i,j} \wedge Z_{i,j'})$$

- Deux dames ne peuvent pas être sur une même colonne.

$$\varphi_{\text{colonne}} = \bigwedge_{j,i \neq i'} \neg(X_{i,j} \wedge Y_{i',j}) \wedge \neg(X_{i,j} \wedge Z_{i',j}) \wedge \neg(Y_{i,j} \wedge Z_{i',j})$$

- Deux dames ne peuvent pas être sur une même diagonale.

Codage sur un échiquier 3x3 (les contraintes)

Les autres contraintes (hors conditions initiales) sont

- Deux dames ne peuvent pas être sur une même ligne.

$$\varphi_{\text{ligne}} = \bigwedge_{i,j \neq j'} \neg(X_{i,j} \wedge Y_{i,j'}) \wedge \neg(X_{i,j} \wedge Z_{i,j'}) \wedge \neg(Y_{i,j} \wedge Z_{i,j'})$$

- Deux dames ne peuvent pas être sur une même colonne.

$$\varphi_{\text{colonne}} = \bigwedge_{j,i \neq i'} \neg(X_{i,j} \wedge Y_{i',j}) \wedge \neg(X_{i,j} \wedge Z_{i',j}) \wedge \neg(Y_{i,j} \wedge Z_{i',j})$$

- Deux dames ne peuvent pas être sur une même diagonale.

$$\varphi_{\text{diag}} = \bigwedge_{|i-j|=|i'-j'|} \neg(X_{i,j} \wedge Y_{i',j'}) \wedge \neg(X_{i,j} \wedge Z_{i',j'}) \wedge \neg(Y_{i,j} \wedge Z_{i',j'})$$

Codage sur un échiquier 3x3 (les contraintes)

Un instanciation qui satisfait

$$\varphi_X \wedge \varphi_Y \wedge \varphi_Z \wedge \varphi_{X,Y} \wedge \varphi_{X,Z} \wedge \varphi_{Y,Z} \wedge \varphi_{\text{ligne}} \wedge \varphi_{\text{colonne}} \wedge \varphi_{\text{diag}}$$

- possède exactement un $X_{i,j}$ vrai,
- exactement un $Y_{i,j}$ vrai et
- exactement un $Z_{i,j}$ vrai ,
- de plus couples de ces trois indices sont deux à deux distincts,
- deux variables à vrai ne sont ni sur la même ligne, ni sur la même colonne, ni sur la même diagonale.

Le reines sont toutes sur l'échiquier satisfont les contraintes.

Codage sur un échiquier 3x3 (les contraintes)

Un instanciation qui satisfait

$$\varphi_X \wedge \varphi_Y \wedge \varphi_Z \wedge \varphi_{X,Y} \wedge \varphi_{X,Z} \wedge \varphi_{Y,Z} \wedge \varphi_{\text{ligne}} \wedge \varphi_{\text{colonne}} \wedge \varphi_{\text{diag}}$$

- possède exactement un $X_{i,j}$ vrai,
- exactement un $Y_{i,j}$ vrai et
- exactement un $Z_{i,j}$ vrai ,
- de plus couples de ces trois indices sont deux à deux distincts,
- deux variables à vrai ne sont ni sur la même ligne, ni sur la même colonne, ni sur la même diagonale.

Le reines sont toutes sur l'échiquier satisfont les contraintes.
Il reste à coder les conditions initiales
$X_{3,1}$ impose qu'une dame (la première) soit dans la case $(3, 1)$.

Généralisation

$X_{i,j,k}$: la k-ième dame est sur la case (i, j).

Généralisation

$X_{i,j,k}$: la k-ième dame est sur la case (i,j).

- Chaque dame apparaît une et une seule fois :

$$\varphi_1 = \bigwedge_{k} \bigvee_{i,j} X_{i,j,k} \quad \text{et} \quad \varphi_2 = \bigwedge_{k} \bigwedge_{i,j} \bigwedge_{(i',j') \neq (i,j)} \neg(X_{i,j,k} \wedge X_{i',j',k})$$

Généralisation

$X_{i,j,k}$: la k-ième dame est sur la case (i,j).

- Chaque dame apparaît une et une seule fois :

$$\varphi_1 = \bigwedge_{k} \bigvee_{i,j} X_{i,j,k} \quad \text{et} \quad \varphi_2 = \bigwedge_{k} \bigwedge_{i,j} \bigwedge_{(i',j') \neq (i,j)} \neg(X_{i,j,k} \wedge X_{i',j',k})$$

- Pas deux dames sur la même case :

$$\varphi_3 = \bigwedge_{k \neq k'} \bigwedge_{i,j} \neg(X_{i,j,k} \wedge X_{i,j,k'})$$

Généralisation

$X_{i,j,k}$: la k-ième dame est sur la case (i,j).

- Chaque dame apparaît une et une seule fois :

$$\varphi_1 = \bigwedge_{k} \bigvee_{i,j} X_{i,j,k} \quad \text{et} \quad \varphi_2 = \bigwedge_{k} \bigwedge_{i,j} \bigwedge_{(i',j')\neq(i,j)} \neg(X_{i,j,k} \wedge X_{i',j',k})$$

- Pas deux dames sur la même case :

$$\varphi_3 = \bigwedge_{k\neq k'} \bigwedge_{i,j} \neg(X_{i,j,k} \wedge X_{i,j,k'})$$

- Pas deux dames sur la même ligne, ni même colonne :

$$\varphi_L = \bigwedge_{k\neq k'} \bigwedge_{i,j,j'} \neg(X_{i,j,k} \wedge X_{i,j',k'}) \quad \text{et} \quad \varphi_C = \bigwedge_{k\neq k'} \bigwedge_{i,j,i'} \neg(X_{i,j,k} \wedge X_{i',j,k'})$$

Généralisation

$X_{i,j,k}$: la k-ième dame est sur la case (i,j).

- Chaque dame apparaît une et une seule fois :

$$\varphi_1 = \bigwedge_{k} \bigvee_{i,j} X_{i,j,k} \quad \text{et} \quad \varphi_2 = \bigwedge_{k} \bigwedge_{i,j} \bigwedge_{(i',j')\neq(i,j)} \neg(X_{i,j,k} \wedge X_{i',j',k})$$

- Pas deux dames sur la même case :

$$\varphi_3 = \bigwedge_{k\neq k'} \bigwedge_{i,j} \neg(X_{i,j,k} \wedge X_{i,j,k'})$$

- Pas deux dames sur la même ligne, ni même colonne :

$$\varphi_L = \bigwedge_{k\neq k'} \bigwedge_{i,j,j'} \neg(X_{i,j,k} \wedge X_{i,j',k'}) \quad \text{et} \quad \varphi_C = \bigwedge_{k\neq k'} \bigwedge_{i,j,i'} \neg(X_{i,j,k} \wedge X_{i',j,k'})$$

- Pas deux dames en diagonale :

$$\varphi_D = \bigwedge_{k\neq k'} \bigwedge_{|i-i'|=|j-j'|} \neg(X_{i,j,k} \wedge X_{i',j',k'})$$

Sudoku

		2	5		3		4	1
							3	
	4			1	8			
		5	3					
9					5			
3	8		9				7	
5		4						
			2		7	3	9	
					9		8	4

En testant toutes les possibilités par colonne (5 cases vides), on obtient $(5!)^9 \simeq 10^{18}$ possibilités. Plusieurs dizaines d'années de calcul.

Sudoku

		2	5		3		4	1
							3	
	4			1	8			
		5	3					
9					5			
3	8		9				7	
5		4						
			2		7	3	9	
					9		8	4

En testant toutes les possibilités par colonne (5 cases vides), on obtient $(5!)^9 \simeq 10^{18}$ possibilités. Plusieurs dizaines d'années de calcul.

On va coder avec une formule SAT :

$X_{i,j,k}$ est vraie si la case (i,j) contient k.

Il y a n^3 variables (dans le cas classique $n = 9$).

Codage SAT du Sudoku.

La case (i, j) a au moins une valeur :

$\bigvee_{k=1}^{9} X_{i,j,k}$

La case (i, j) a au plus une valeur :

$\bigwedge_{k=1}^{9} \bigwedge_{k'=1,k'\neq k}^{9} \left(X_{i,j,k} \Rightarrow \neg X_{i,j,k'}\right)$ ce qui s'écrit aussi
$\bigwedge_{k=1}^{9} \bigwedge_{k'=1,k'\neq k}^{9} \left(\neg X_{i,j,k} \vee \neg X_{i,j,k'}\right)$

On le fait pour toutes les cases :

$\bigwedge_{i=1}^{9} \bigwedge_{j=1}^{9} \left(\bigwedge_{k=1}^{9} X_{i,j,k} \wedge \bigwedge_{k=1}^{9} \bigwedge_{k'=1,k'\neq k}^{9} \left(\neg X_{i,j,k} \vee \neg X_{i,j,k'}\right)\right)$

Codage SAT du Sudoku, Contrainte sur les colonnes

Le nombre 1 apparaît dans la première colonne

$\bigvee_{i=1}^{9} X_{i,1,1}$

Codage SAT du Sudoku, Contrainte sur les colonnes

Le nombre 1 apparaît dans la première colonne

$\bigvee_{i=1}^{9} X_{i,1,1}$

Chaque nombre apparaît dans la chaque colonne

$\bigwedge_{j=1}^{9} \bigwedge_{k=1}^{9} \bigvee_{i=1}^{9} X_{i,j,k}$

Par un argument de cardinalité, cela suffit à garantir la contrainte sur les colonnes.

Codage SAT du Sudoku, Contrainte sur les colonnes

Le nombre 1 apparaît dans la première colonne

$\bigvee_{i=1}^{9} X_{i,1,1}$

Chaque nombre apparaît dans la chaque colonne

$\bigwedge_{j=1}^{9} \bigwedge_{k=1}^{9} \bigvee_{i=1}^{9} X_{i,j,k}$

Par un argument de cardinalité, cela suffit à garantir la contrainte sur les colonnes.

On fait de même pour les lignes et les carrés.

Codage SAT du Sudoku, conditions initiales

$$X_{1,1,5} \wedge X_{1,2,3} \wedge X_{1,5,7} \dots$$

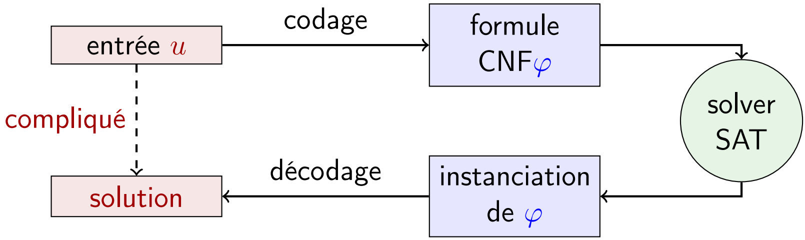
La formule obtenue a une solution si et seulement si la grille de Sudoku en a une.
Une instanciation des $X_{i,j,k}$ qui rend la formule vraie donne une solution de la grille de sudoku.

Plan

Codage SAT

Codage SAT

Résoudre un problème par codage SAT, c'est trouver une fonction calculable en temps polynomiale qui, à chaque entrée u du problème d'entrée, associe une formule SAT en CNF φ qui est satisfiable si et seulement si u est une instance positive du problème.
De plus, on chercher à ce qu'une instanciation de la formule (qui rende son interprétation vraie) donne une solution au problème.



CNF

Rappel : Une formule est sous forme normale conjonctive (CNF) si elle est une conjonction de clauses. Une *clause* est une disjonction de littéraux. Un *littéral* est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.

$$(x_1 \vee x_3 \vee \neg x_2) \wedge x_4 \wedge (\neg x_2 \vee \neg x_4)$$

CNF

Rappel : Une formule est sous forme normale conjonctive (CNF) si elle est une conjonction de clauses. Une *clause* est une disjonction de littéraux. Un *littéral* est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.

$$(x_1 \vee x_3 \vee \neg x_2) \wedge x_4 \wedge (\neg x_2 \vee \neg x_4)$$

Toute formule propositionnelle est équivalente à une formule en forme CNF (il existe des techniques et des algorithmes – voir cours P.A. Masson).

CNF

Rappel : Une formule est sous forme normale conjonctive (CNF) si elle est une conjonction de clauses. Une *clause* est une disjonction de littéraux. Un *littéral* est soit une variable propositionnelle, soit la négation d'une variable propositionnelle.

$$(x_1 \vee x_3 \vee \neg x_2) \wedge x_4 \wedge (\neg x_2 \vee \neg x_4)$$

Toute formule propositionnelle est équivalente à une formule en forme CNF (il existe des techniques et des algorithmes – voir cours P.A. Masson).

Mais en général on arrive à coder directement en CNF.

Exemples

$$A \Rightarrow B \sim \neg A \vee B$$
$$(B \vee A) \Rightarrow C \sim (\neg B \vee C) \wedge (\neg A \vee C)$$
$$A \Rightarrow (B \wedge C) \sim (\neg A \vee \neg B \vee C)$$

Avec un peu d'habitude, on écrit facilement les clauses directement.

Format de fichier .cnf

Les SAT-solvers lisent des fichiers .cnf qui codent la formule.

Exemple pour $(x_1 \vee x_3 \vee \neg x_2) \wedge x_4 \wedge (\neg x_2 \vee \neg x_4)$

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

- Le p et le cnf sont dans tous les fichiers.
- Le 4 est le nombre de variables, le 3 le nombre de clauses.
- Les variables sont des entiers strictement positifs.
- Chaque clause est sur une ligne, terminée par un 0.
- Le - devant un littéral signifie la négation.

Exercice

Donner le fichier format cnf pour la formule

$$x_1 \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (x_5 \vee \neg x_4 \vee x_3)$$

Exercice

Donner le fichier format cnf pour la formule

$$x_1 \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (x_5 \vee \neg x_4 \vee x_3)$$

```
p  cnf  5  4
1          0
2  -3   0
1  4    5  0
5  -4   3  0
```

Format de fichier .cnf

Les SAT-solvers lisent des fichiers `.cnf` qui codent la formule.

fichier `exemple.cnf`

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

```
cryptominisat5 --verb=0
exemple.cnf
```

```
s SATISFIABLE
v -1 -2 -3 4 0
```

```
cryptominisat5 --verb=0 --maxsol=3 exemple.cnf
```

```
s SATISFIABLE
v -1 -2 -3 4 0
s SATISFIABLE
v 1 -2 -3 4 0
s SATISFIABLE
v 1 -2 3 4 0
```

Certains solvers (comme `cryptominisat`) ont aussi des API pour certains langages comme Python ou C++.

format CNF : numéroter les variables

Pour les solvers, les variables doivent être numérotées.
$X_{i,j,k}$: la k-ième dame est sur la case (i, j).

```
def X(i,j,k,n):
    return i+(j-1)*n+(k-1)*n*n

def InvX(r,n):
    i=r%n
    if i == 0 :
        i+=n
    r=(r-i)//n
    j=1+r%n
    k=1+(r-j+1)//n
    return (i,j,k)
```

format CNF : numéroter les variables

Pour les solvers, les variables doivent être numérotées.
$X_{i,j,k}$: la k-ième dame est sur la case (i, j).

```
def X(i,j,k,n):
    return i+(j-1)*n+(k-1)*n*n

def InvX(r,n):
    i=r%n
    if i == 0 :
        i+=n
    r=(r-i)//n
    j=1+r%n
    k=1+(r-j+1)//n
    return (i,j,k)
```

$$X(4, 4, 2, 8) = 4 + (4 - 1) * 8 + (2 - 1) * 64 = 4 + 24 + 64 = 92$$

format CNF : numéroter les variables

Pour les solvers, les variables doivent être numérotées.
$X_{i,j,k}$: la k-ième dame est sur la case (i, j).

```
def X(i,j,k,n):
    return i+(j-1)*n+(k-1)*n*n

def InvX(r,n):
    i=r%n
    if i == 0 :
        i+=n
    r=(r-i)//n
    j=1+r%n
    k=1+(r-j+1)//n
    return (i,j,k)
```

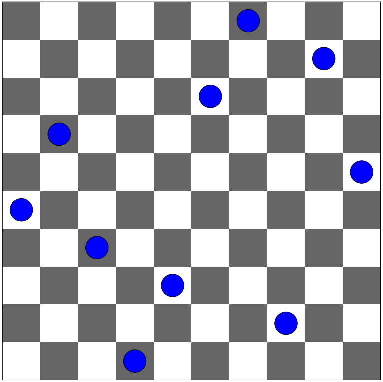
$X(4, 4, 2, 8) = 4 + (4 - 1) * 8 + (2 - 1) * 64 = 4 + 24 + 64 = 92$

$InvX(92, 8)$: $i = 92\%8 = 4$, $j = 1 + ((92 - 4)/8)\%8 = 1 + 11\%8 = 4$
$k = 1 + (11 - 4 + 1)/8 = 1 + 1 = 2$

Plan

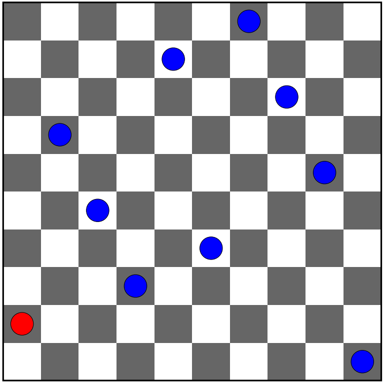
Les dames

Placer 10 dames sur l'échiquier, (0.3 secondes).



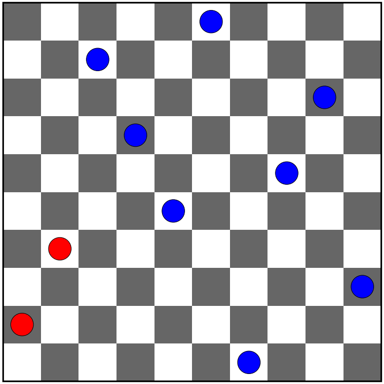
Les dames

Placer 10 dames sur l'échiquier, sachant qu'il y en a une en (1,2) (0.3 secondes).



Les dames

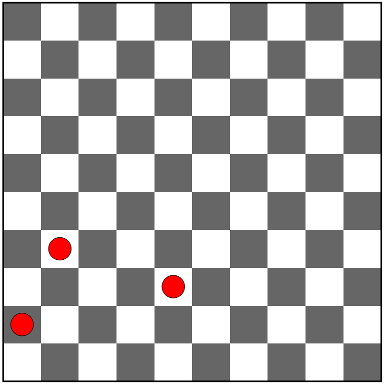
Placer 10 dames sur l'échiquier, sachant qu'il y en a une en (1,2) et en (2,4) (3.4 secondes).



En force brute, 10^{13} possibilités environ.

Les dames

Placer 10 dames sur l'échiquier, sachant qu'il y en a une en (1,2), en (2,4) et en (5,3) (7 minutes) : **pas de solution.**



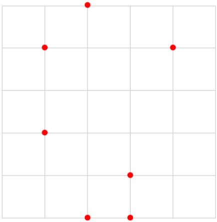
En force brute, 10^{12} possibilités environ.

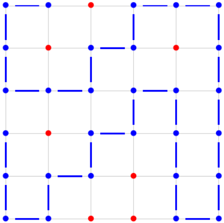
Cas pratique 2

Chemin autoevitant

Donnée : Un entier n et une liste ℓ de points du quadrillage $n \times n$ et un entier $k \leq n^2$

Question : est-il possible de faire un chemin sur le quadrillage qui évite les points de ℓ, de longueur k, sans passer deux fois par le même point.





Cas pratique

Chemin autoevitant

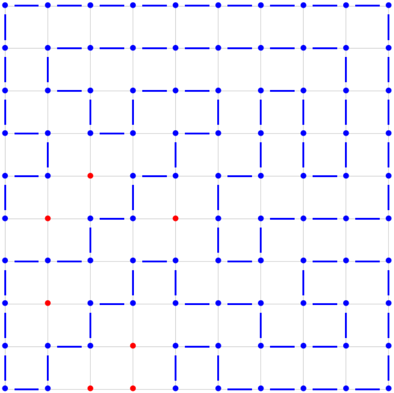
Donnée : Un entier n et une liste ℓ de points du quadrillage $n \times n$ et un entier $k \leq n^2$
Question : est-il possible de faire un chemin sur le quadrillage qui évite les points de ℓ, de longueur k, sans passer deux fois par le même point.

- Environ 120 lignes de Python, $n^3 + 4n^2$ variables (pour $n = 8$, 768 variables).
- Cryptominisat [1]
- Trouve une solution pour $n = 8$ en 20 secondes.
- Montre qu'il n'y a pas de solution pour $n = 7$ en 2 minutes.

1. `https://www.msoos.org/cryptominisat5/`

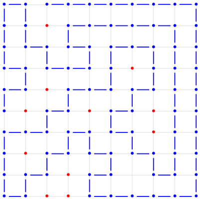
Cas pratique

- Solution trouvée pour $n = 9$ avec un sommet libre en 23s.
- Solution trouvée pour $n = 10$ avec un sommet libre en 2 minutes.
- Programme arrêté après 10h pour $n = 9$.
- Solution trouvée pour $n = 10$ avec en 20 minutes.

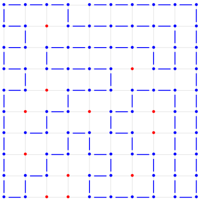


Cas pratique

- Solution trouvée pour $n = 10$ avec un sommet libre en 4 minutes.



- Solution trouvée pour $n = 10$ avec en 7 minutes.



Plan

Formules 2-CNF

Définition

Une formule est sous forme 2-CNF si

- Elle est sous forme CNF,
- Chaque clause est de longueur au plus 2.

$$(x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_1) \wedge (x_2 \vee \neg x_3) \wedge x_4$$

Formules 2-CNF

Définition

Une formule est sous forme 2-CNF si

- Elle est sous forme CNF,
- Chaque clause est de longueur au plus 2.

$$(x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_1) \wedge (x_2 \vee \neg x_3) \wedge x_4$$

2-SAT

Donnée : Une formule sous forme 2-CNF
Sortie : Vrai si la formule est satisfiable, Faux sinon.

Transformations en implications

$$x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_1 \wedge x_5) \wedge (x_5 \vee x_2)$$

L'idée de l'approche est la suivante :

- On supprime les clauses unaires (avec la bonne affectation des variables) et on propage pour n'avoir que des 2-clauses.

Transformations en implications

$$x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_1 \wedge x_5) \wedge (x_5 \vee x_2)$$

L'idée de l'approche est la suivante :

- On supprime les clauses unaires (avec la bonne affectation des variables) et on propage pour n'avoir que des 2-clauses.
  $x_1 = 1$ et $(x_2 \vee x_4) \wedge (x_3 \wedge \neg x_4) \wedge x_5 \wedge (x_5 \vee x_2)$

Transformations en implications

$$x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_1 \wedge x_5) \wedge (x_5 \vee x_2)$$

L'idée de l'approche est la suivante :

- On supprime les clauses unaires (avec la bonne affectation des variables) et on propage pour n'avoir que des 2-clauses.
  $x_1 = 1$ et $(x_2 \vee x_4) \wedge (x_3 \wedge \neg x_4) \wedge x_5 \wedge (x_5 \vee x_2)$
  $x_1 = 1$ et $x_5 = 1$ et $(x_2 \vee x_4) \wedge (x_3 \vee \neg x_4)$

Transformations en implications

$$x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_4) \wedge (x_3 \vee \neg x_4) \wedge (\neg x_1 \wedge x_5) \wedge (x_5 \vee x_2)$$

L'idée de l'approche est la suivante :

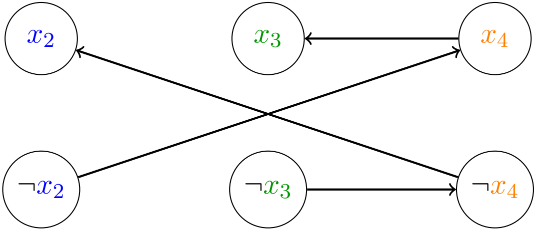
- On supprime les clauses unaires (avec la bonne affectation des variables) et on propage pour n'avoir que des 2-clauses.
  $x_1 = 1$ et $(x_2 \vee x_4) \wedge (x_3 \wedge \neg x_4) \wedge x_5 \wedge (x_5 \vee x_2)$
  $x_1 = 1$ et $x_5 = 1$ et $(x_2 \vee x_4) \wedge (x_3 \vee \neg x_4)$
- Les clauses 2-clauses sont transformées en implications avec $\alpha \vee \beta \equiv (\neg\alpha \Rightarrow \beta) \wedge (\neg\beta \Rightarrow \alpha)$.
  $x_1 = 1$ et $x_5 = 1$ et

$$(\neg x_2 \Rightarrow x_4) \wedge (\neg x_4 \Rightarrow x_2) \wedge (\neg x_3 \Rightarrow \neg x_4) \wedge (x_4 \Rightarrow x_3)$$

Graphes d'implications

Le graphe d'implication pour une formule (avec que des implications entre littéraux) est construit comme suit :

- Les sommets sont tous les littéraux possibles,
- On met une arête (flèche) d'un littéral ℓ_1 vers un littéral ℓ_2, chaque fois qu'il y a une implication $\ell_1 \Rightarrow \ell_2$.

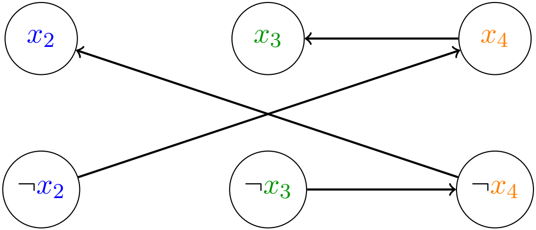


$$(\neg x_2 \Rightarrow x_4) \wedge (\neg x_4 \Rightarrow x_2) \wedge (\neg x_3 \Rightarrow \neg x_4) \wedge (x_4 \Rightarrow x_3)$$

Graphes d'implications

Le graphe d'implication pour une formule (avec que des implications entre littéraux) est construit comme suit :

- Les sommets sont tous les littéraux possibles,
- On met une arête (flèche) d'un littéral ℓ_1 vers un littéral ℓ_2, chaque fois qu'il y a une implication $\ell_1 \Rightarrow \ell_2$.



$$(\neg x_2 \Rightarrow x_4) \wedge (\neg x_4 \Rightarrow x_2) \wedge (\neg x_3 \Rightarrow \neg x_4) \wedge (x_4 \Rightarrow x_3)$$

- La formule est satisfiable si et seulement s'il n'y a pas de boucle contenant un littéral et sa négation.

Exemple 1

$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$

x_1	x_2	x_3	x_4	x_5
$\neg x_1$	$\neg x_2$	$\neg x_3$	$\neg x_4$	$\neg x_5$

Exemple 1

$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$

$$\begin{tikzcd}
x_1 & x_2 \arrow[l] & x_3 & x_4 & x_5 \\
\neg x_1 \arrow[r] & \neg x_2 & \neg x_3 & \neg x_4 & \neg x_5
\end{tikzcd}$$

$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$

Exemple 1

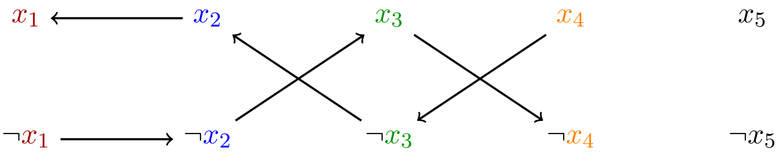
$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$

$$\begin{tikzcd}
x_1 & x_2 \arrow[l] & x_3 & x_4 & x_5 \\
\neg x_1 \arrow[r] & \neg x_2 \arrow[ur] & \neg x_3 \arrow[ul] & \neg x_4 & \neg x_5
\end{tikzcd}$$

$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$
$$(x_2 \vee x_3) \sim (\neg x_3 \Rightarrow x_2) \wedge (\neg x_2 \Rightarrow x_3)$$

Exemple 1

$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$



$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$
$$(x_2 \vee x_3) \sim (\neg x_3 \Rightarrow x_2) \wedge (\neg x_2 \Rightarrow x_3)$$

Exemple 1

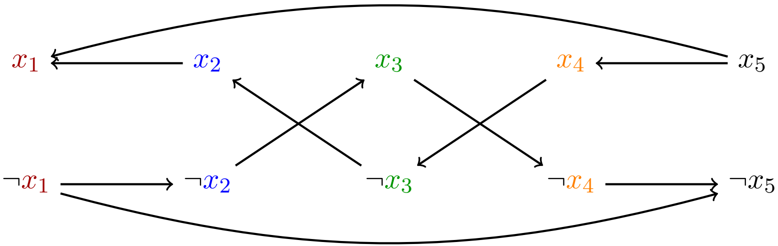
$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$

$$\begin{tikzcd}
x_1 & x_2 \arrow[l] & x_3 \arrow[dr] & x_4 \arrow[dl] & x_5 \arrow[l] \\
\neg x_1 \arrow[r] & \neg x_2 \arrow[ur] & \neg x_3 \arrow[ul] & \neg x_4 \arrow[r] & \neg x_5
\end{tikzcd}$$

$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$
$$(x_2 \vee x_3) \sim (\neg x_3 \Rightarrow x_2) \wedge (\neg x_2 \Rightarrow x_3)$$

Exemple 1

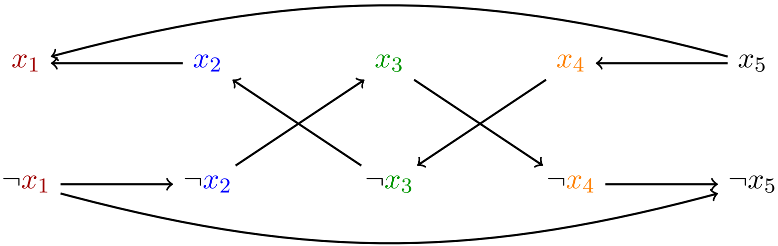
$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$



$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$
$$(x_2 \vee x_3) \sim (\neg x_3 \Rightarrow x_2) \wedge (\neg x_2 \Rightarrow x_3)$$

Exemple 1

$$(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_4 \vee \neg x_5) \wedge (\neg x_5 \vee x_1)$$



$$(x_1 \vee \neg x_2) \sim (\neg x_1 \Rightarrow \neg x_2) \wedge (x_2 \Rightarrow x_1)$$
$$(x_2 \vee x_3) \sim (\neg x_3 \Rightarrow x_2) \wedge (\neg x_2 \Rightarrow x_3)$$

Il n'y a pas de cycle contenant un littéral et sa négation, la formule est satisfiable

Exemple 2

$$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$$

Exemple 2

$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

Exemple 2

$$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$

Exemple 2

$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4)$

Exemple 2

$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4)$

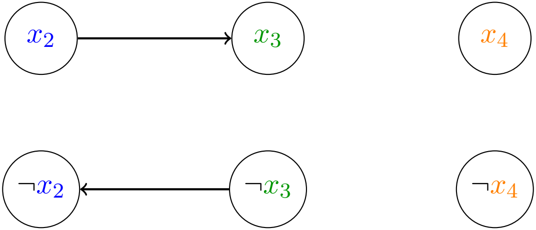
Exemple 2

$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4)$



Exemple 2

$$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$$

$$x_1 = 0 \text{ et } (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$$

$$x_1 = 0 \text{ et } x_5 = 1 \text{ et } (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$$

$$x_1 = 0 \text{ et } x_5 = 1 \text{ et } (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4)$$

$$x_2 \longrightarrow x_3 \longrightarrow x_4$$

$$\neg x_2 \longleftarrow \neg x_3 \longleftarrow \neg x_4$$

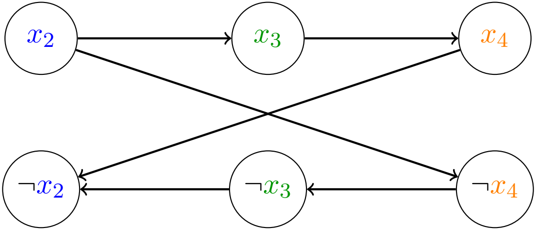
Exemple 2

$\neg x_1 \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_5) \wedge (x_5 \vee x_3)$

$x_1 = 0$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (0 \vee x_5) \wedge (x_5 \vee x_3)$

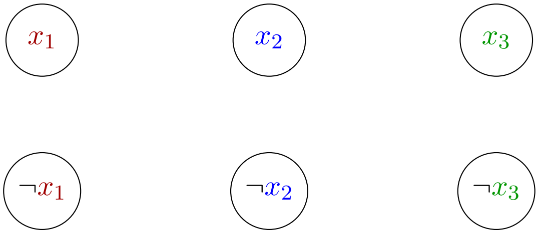
$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4) \wedge (\mathbf{1} \vee x_3)$

$x_1 = 0$ et $x_5 = 1$ et $(\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (x_2 \vee x_4)$



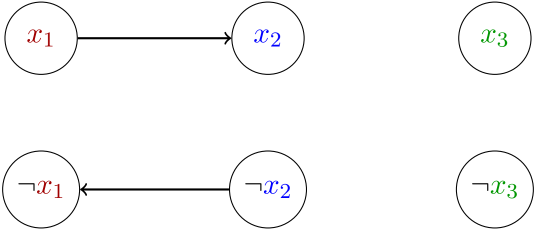
Exemple 3

$$(\neg x_1 \lor x_2) \land (\neg x_2 \lor x_3) \land (\neg x_3 \lor x_1) \land (\neg x_1 \lor \neg x_2) \lor (x_2 \lor x_3)$$



Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$



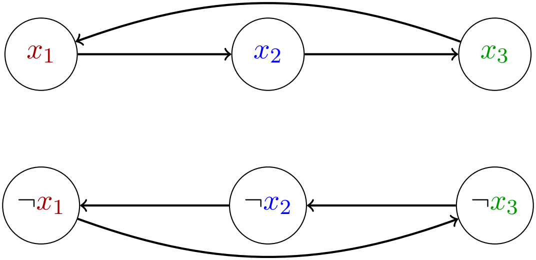
Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$

$$\begin{tikzcd}
x_1 \arrow[r] & x_2 \arrow[r] & x_3 \\
\neg x_1 & \neg x_2 \arrow[l] & \neg x_3 \arrow[l]
\end{tikzcd}$$

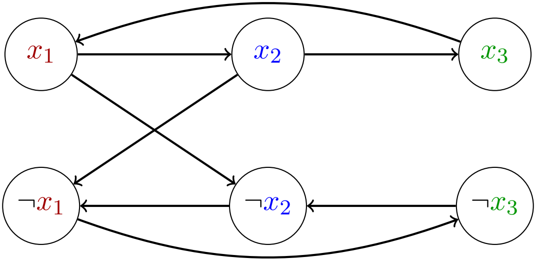
Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$



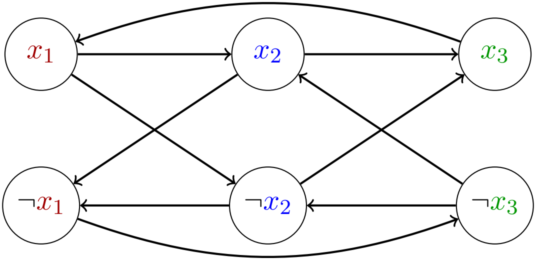
Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$



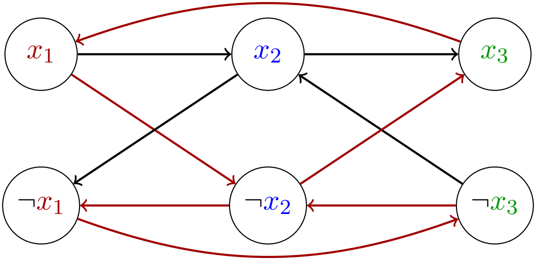
Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$



Exemple 3

$$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_1 \vee \neg x_2) \vee (x_2 \vee x_3)$$



Il existe une cycle $\neg x_1 \rightarrow \neg x_3 \rightarrow \neg x_2 \rightarrow x_3 \rightarrow x_1 \rightarrow \neg x_2$ passant par x_1 et $\neg x_1$, donc la formule n'est pas satisfiable.

Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

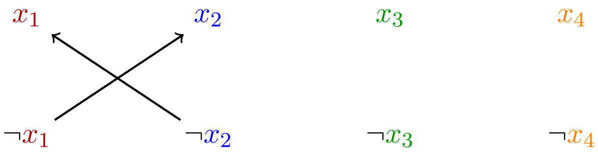
x_1 x_2 x_3 x_4

$\neg x_1$ $\neg x_2$ $\neg x_3$ $\neg x_4$

Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$



Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

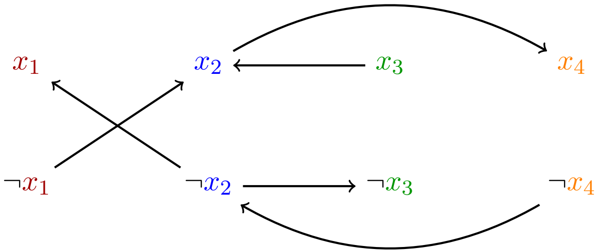
$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

$$\begin{tikzcd}
x_1 & x_2 & x_3 \arrow[l] & x_4 \\
\neg x_1 \arrow[ur] & \neg x_2 \arrow[ul] \arrow[r] & \neg x_3 & \neg x_4
\end{tikzcd}$$

Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

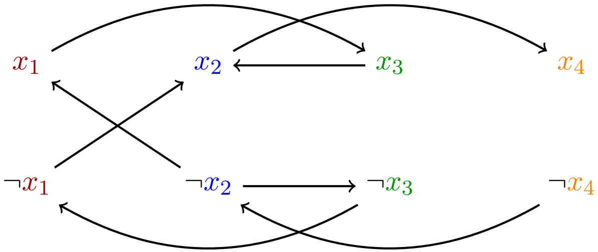
$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$



Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

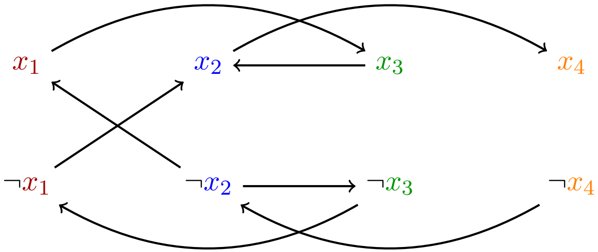
$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$



Exercice

En utilisant l'algorithme proposé, décider si la formule suivante est satisfiable :

$$(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$



Pas de circuit passant par un littéral et sa négation : la formule est satisfiable.

Plan

SAT solver

- Pour certains problèmes, développer un programme efficace est chronophage,
- L'utilisation de solvers, par exemple SAT, est une solution possible,
- Il existe d'autres types de solvers (programme linéaire, SMT solver), ou d'autres techniques (méta-heuristiques, approximations,...)