



Présentation 2

PSAR

David TOTY
Maxime TRAN





Objectif:
Modéliser et résoudre le problème du Sudoku



Qu'est ce que le Sudoku ?

- Jeu de reflexion
- Règles

| | | | | | | | | |
|---|--|---|---|---|---|---|--|---|
| | | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |

Qu'est ce qu'un solveur ?

- Programme informatique
- Représenté sous forme booléenne
- Syntaxe: \wedge (ET), \vee (OU), \neg (NON), \Rightarrow (IMPLICATION), \Leftrightarrow (EQUIVALENCE)

Ex: $((a \vee b) \wedge \neg b) \vee c$

Solution: 1) $(a, b, c) = (\text{true}, \text{false}, \text{false})$

2) $(a, b, c) = (\text{false}, \text{false}, \text{true})$

Problème SAT

- Boolean SATisfiability problem
- Déterminer s'il existe une assignation de valeur pour rendre la formule vraie.
- Formule booléenne : tester assignation
- Solveurs prennent les expressions en Forme Normale Conjonctive
- Pas d'imbrications, pas de NON devant une formule

Ex:

$A \wedge B, A \vee B, (A \vee B) \wedge C$ (FNC)

$\neg(A \wedge B)$ et $A \wedge (B \vee (C \wedge D))$ (pas FNC)

Et pour les nombres ?

- Remplacer booléens par des prédicats selon une théorie
- C'est ce qu'on appelle la Satisfiability Modulo Theories (SMT)

Ex:

"Si $x < 5$, alors $y = x + 3$ " : $x < 5 \wedge y = x + 3$

Et pour les nombres ?

Dans un programme C:

```
if ( x < 5)
```

```
    y = x + 3
```

```
else
```

```
    y = x + 2
```

$\Rightarrow (x < 5 \wedge y = x + 3) \vee (x \geq 5 \wedge y = x + 2)$

Pour trouver une solution

- Hypothèse valide
- Propager les conséquences
- Si absurdité, hypothèse contestée

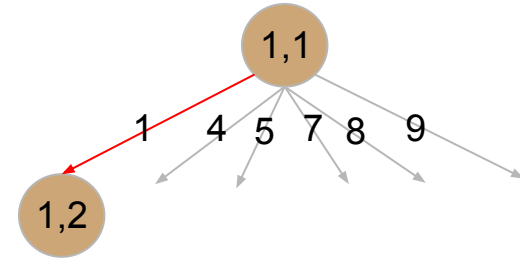
Exemple 1 - Sudoku

| | | | | | | | | |
|---|--|---|---|---|---|---|--|---|
| | | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |

Hypothèse: plus petit chiffre disponible, pas présent dans la ligne. Puis on teste sur la colonne.

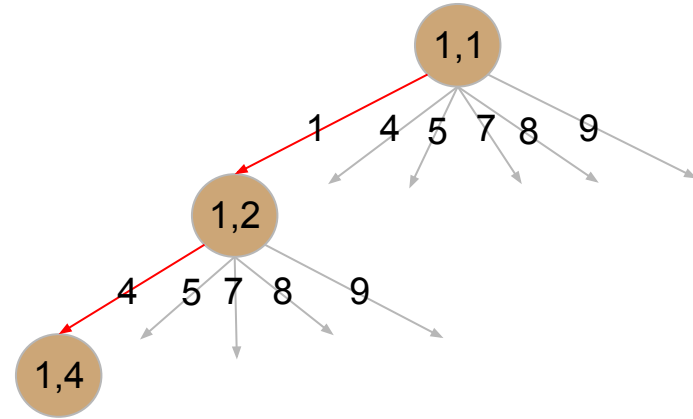
Exemple 1 - Sudoku

| | | | | | | | | |
|---|--|---|---|---|---|---|--|---|
| 1 | | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



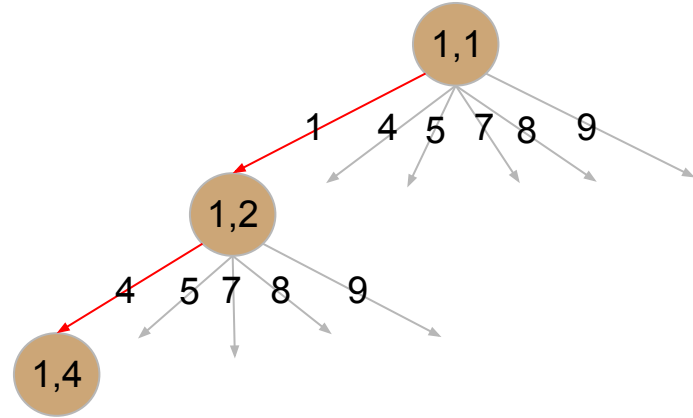
Exemple 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|--|---|
| 1 | 2 | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



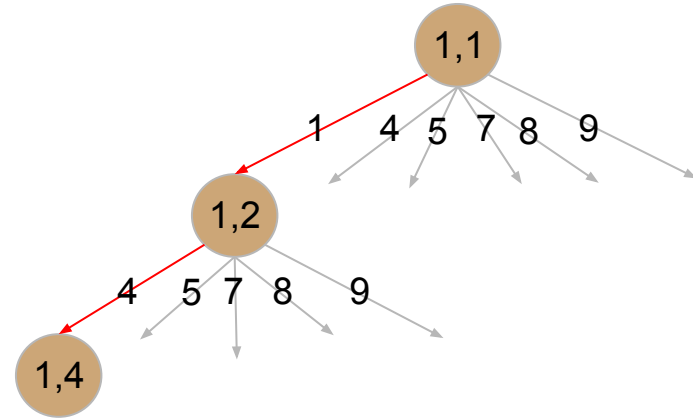
Example 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|--|---|
| 1 | 3 | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



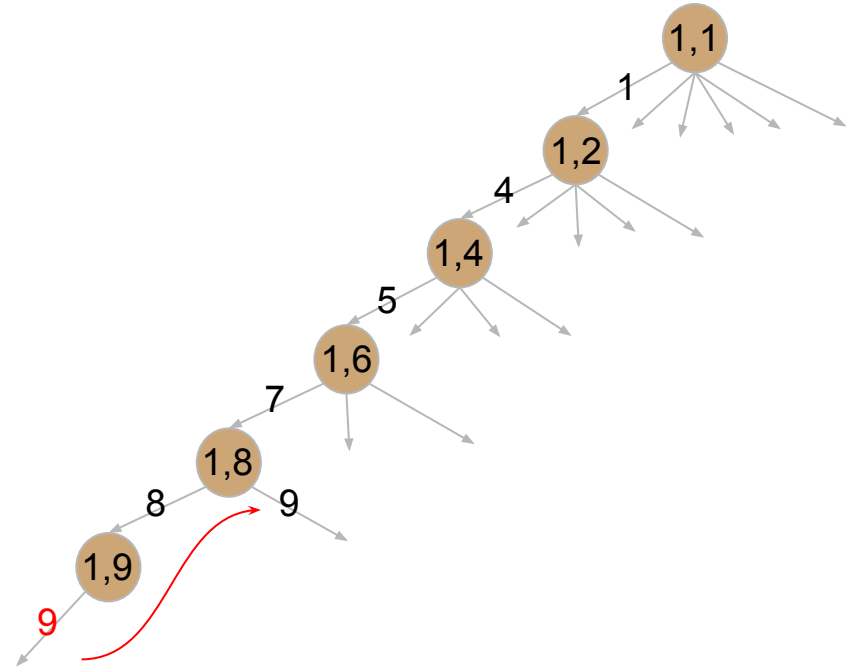
Example 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|--|---|
| 1 | 4 | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



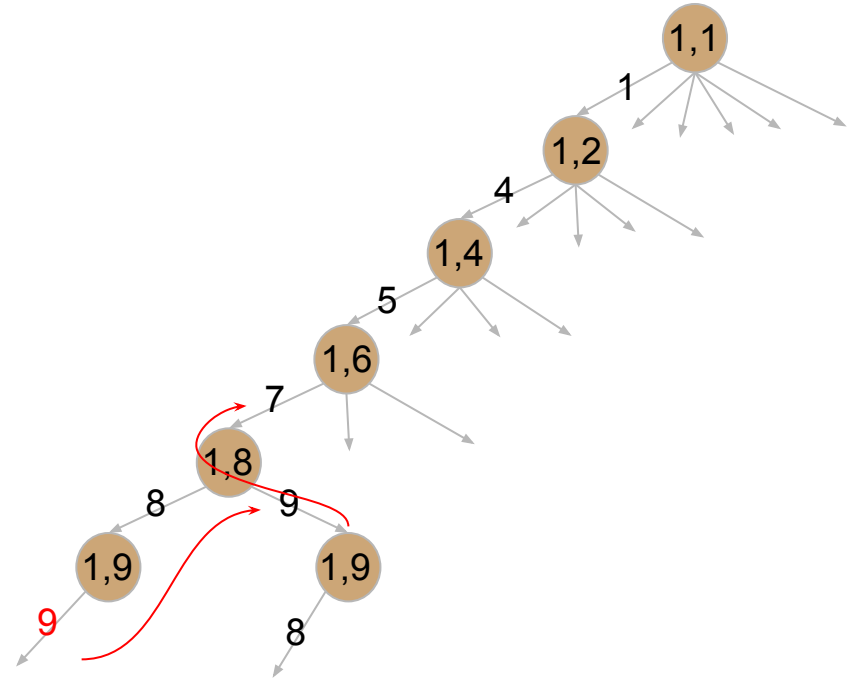
Exemple 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 3 | 5 | 2 | 7 | 6 | 8 | 9 |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



Exemple 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 3 | 5 | 2 | 7 | 6 | 9 | 8 |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |



Exemple 1 - Sudoku

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | 8 | 3 | 9 | 2 | 1 | 6 | 5 | 7 |
| 9 | 6 | 7 | 3 | 4 | 5 | 8 | 2 | 1 |
| 2 | 5 | 1 | 8 | 7 | 6 | 4 | 9 | 3 |
| 5 | 4 | 8 | 1 | 3 | 2 | 9 | 7 | 6 |
| 7 | 2 | 9 | 5 | 6 | 4 | 1 | 3 | 8 |
| 1 | 3 | 6 | 7 | 9 | 8 | 2 | 4 | 5 |
| 3 | 7 | 2 | 6 | 8 | 9 | 5 | 1 | 4 |
| 8 | 1 | 4 | 2 | 5 | 3 | 7 | 6 | 9 |
| 6 | 9 | 5 | 4 | 1 | 7 | 3 | 8 | 2 |

Brute force

Comment résoudre le problème ?

- Or-Tools
- Programmation par contraintes (variables, contraintes, domaines)
- Filtrage
- Propagation
- Backtracking

Exemple 1 - Sudoku

| | | | | | | | | |
|---|--|---|---|---|---|---|--|---|
| | | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |

Programmation Par Contrainte

Variables: Chacun des 81 cases de la grille

Contraintes: 1 occurrence d'un chiffre par colonne, ligne et cellule.

Domaines: {1, 2, 3, 4, 5, 6, 7, 8, 9}

Exemple 1 - Sudoku

| | | | | | | | | |
|---|--|---|---|---|---|---|--|---|
| | | 3 | | 2 | | 6 | | |
| 9 | | | 3 | | 5 | | | 1 |
| | | 1 | 8 | | 6 | 4 | | |
| | | 8 | 1 | | 2 | 9 | | |
| 7 | | | | | X | | | 8 |
| | | 6 | 7 | | 8 | 2 | | |
| | | 2 | 6 | | 9 | 5 | | |
| 8 | | | 2 | | 3 | | | 9 |
| | | 5 | | 1 | | 3 | | |

Programmation Par Contrainte

Variables: Chacun des 81 cases de la grille

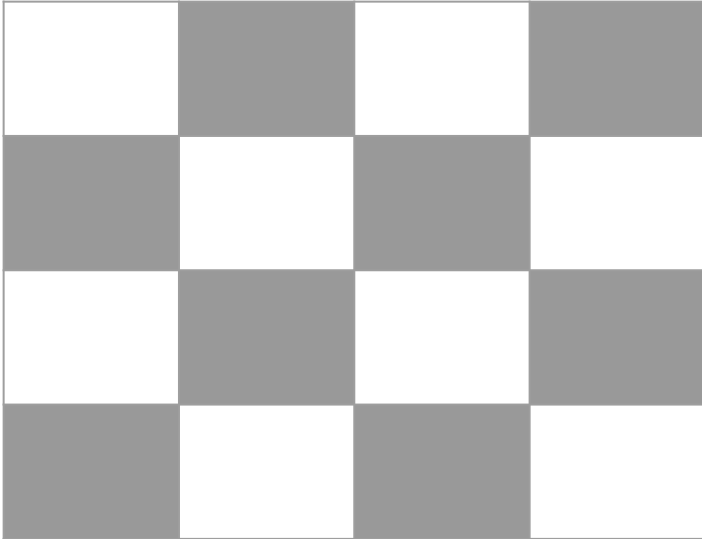
Contraintes: 1 occurrence d'un chiffre par colonne, ligne et cellule.

Domaines: {1, 2, 3, 4, 5, 6, 7, 8, 9}

Exemple: La case avec la croix rouge X

$D(X) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \Rightarrow D(X) = \{4\}$

Exemple 2 - N-Queens



Problème des 4-Queens

Variable: Q_i où i est la colonne

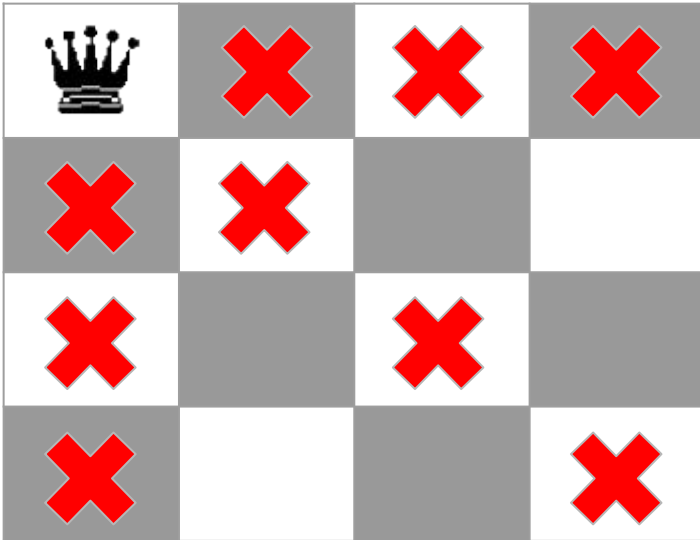
Domaine: $D_i = \{1, 2, 3, 4\}$ (ligne)

Contraintes:

$Q_i \neq Q_j$ (pas sur la même ligne)

$|Q_i - Q_j| \neq |i - j|$ (pas sur la même diagonale)

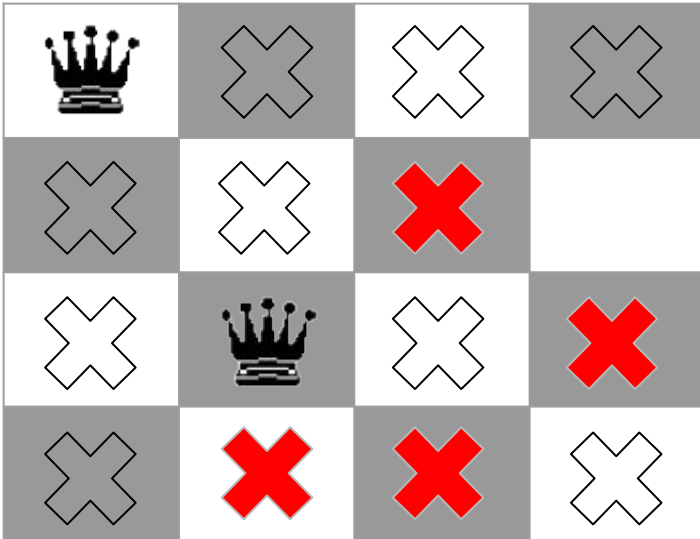
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

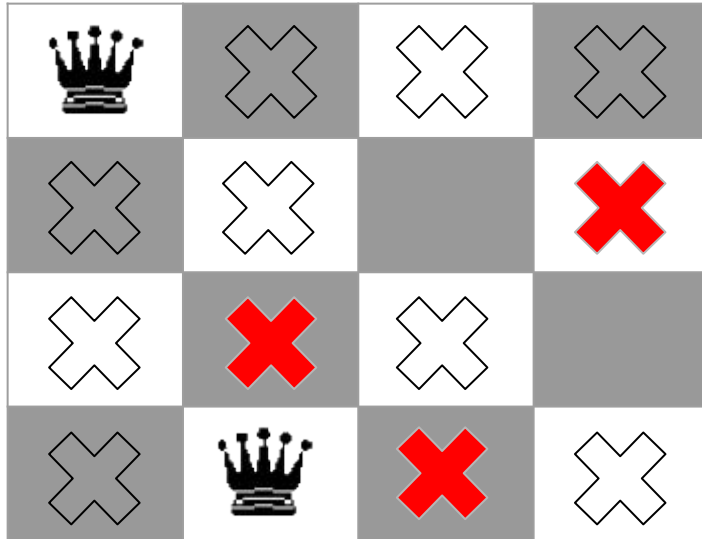
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

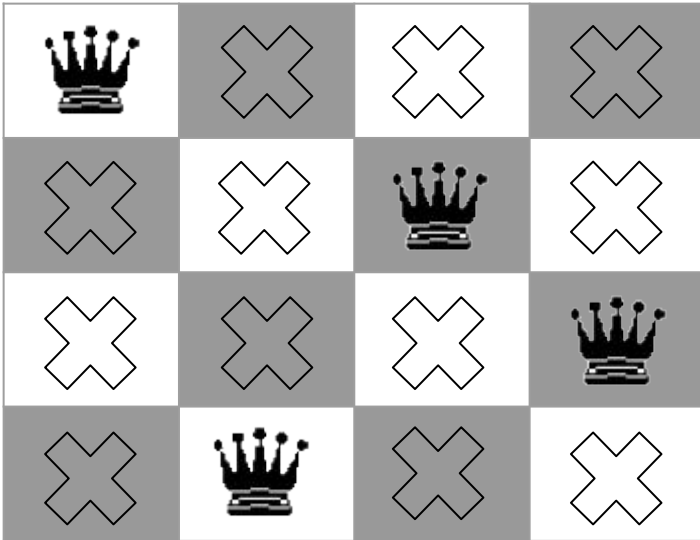
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

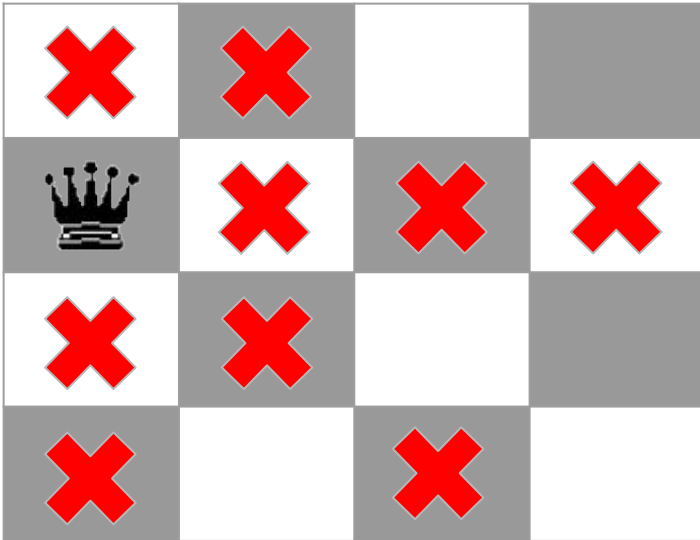
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

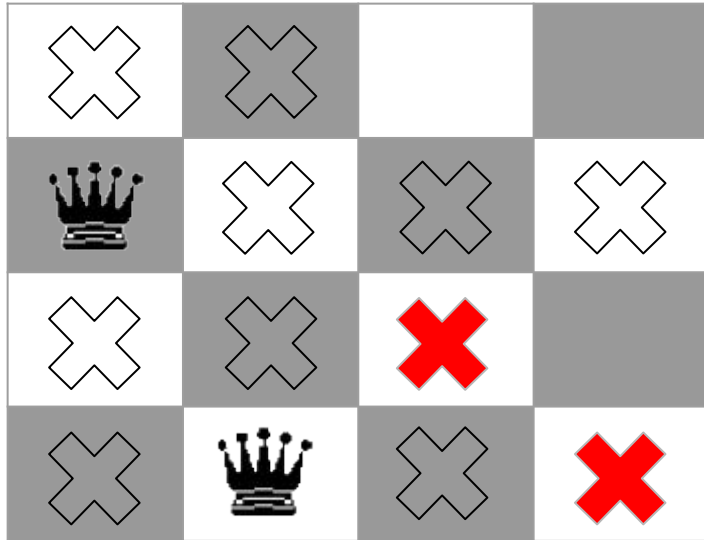
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

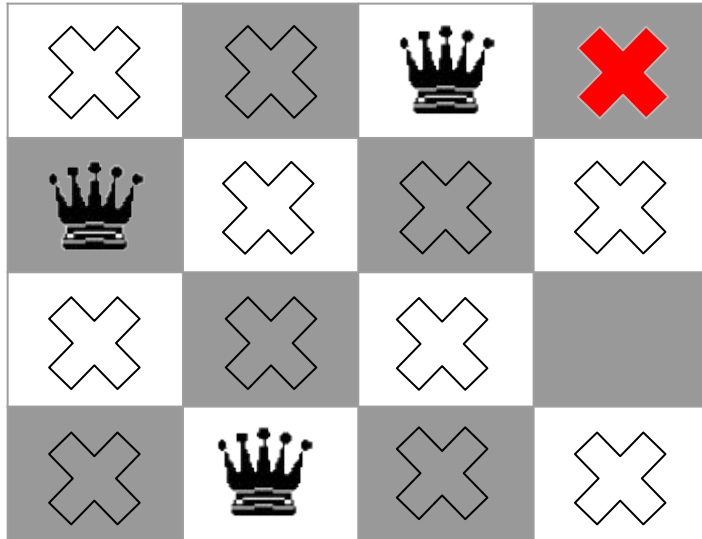
Exemple 2 - N-Queens



Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

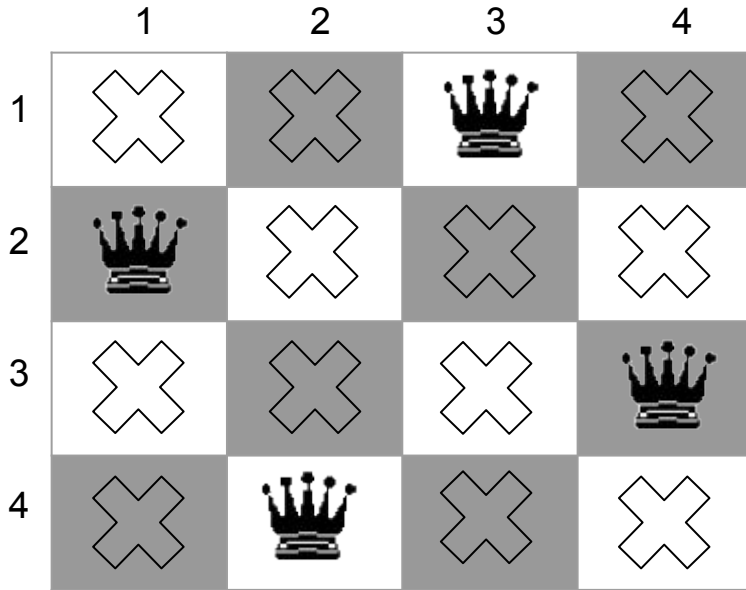
Exemple 2 - N-Queens



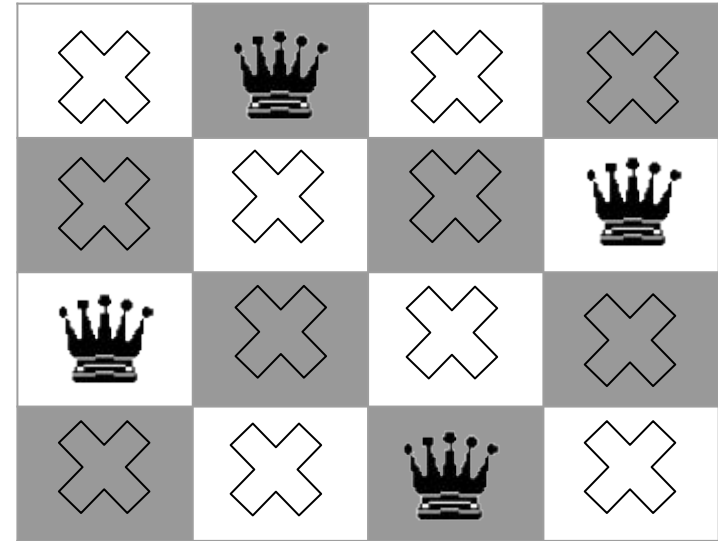
Problème des 4-Queens

- Pas sur la même ligne, colonne et diagonales

Exemple 2 - N-Queens



Solution: S1 = (Q₃, Q₁, Q₄, Q₂)



Solution: S2 = (Q₂, Q₄, Q₁, Q₃)

Modélisation

- Les contraintes du Sudoku

Soit x une valeur d'une case du Sudoku. x s'identifie par le triplet $(x_i, x_j, \text{Région}_x)$.

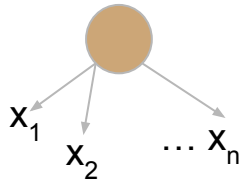
On a :

$$\forall (x, y), x = y \wedge (x_i \neq y_i) \wedge (x_j \neq y_j) \wedge (\text{Région}_x \neq \text{Région}_y) \quad (F)$$

\Leftrightarrow

$$\forall x, \nexists y, x = y \wedge (x_i = y_i) \vee (x_j = y_j) \vee (\text{Région}_x = \text{Région}_y)$$

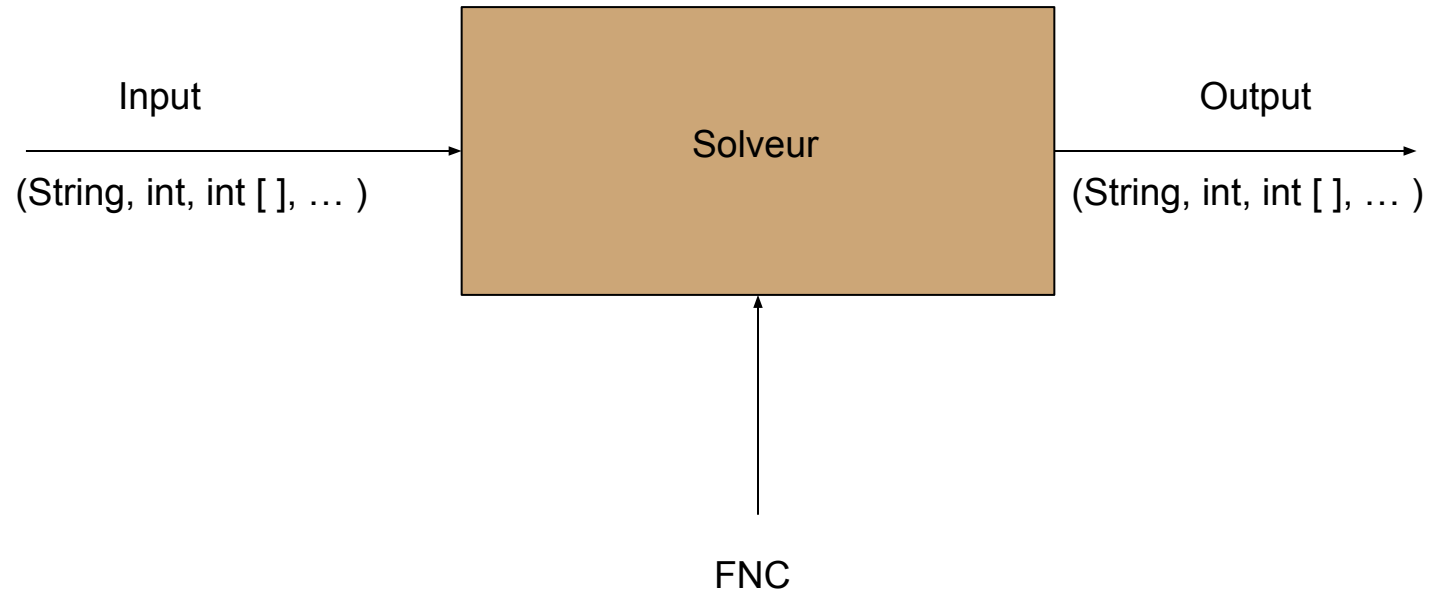
Arbre de recherche:



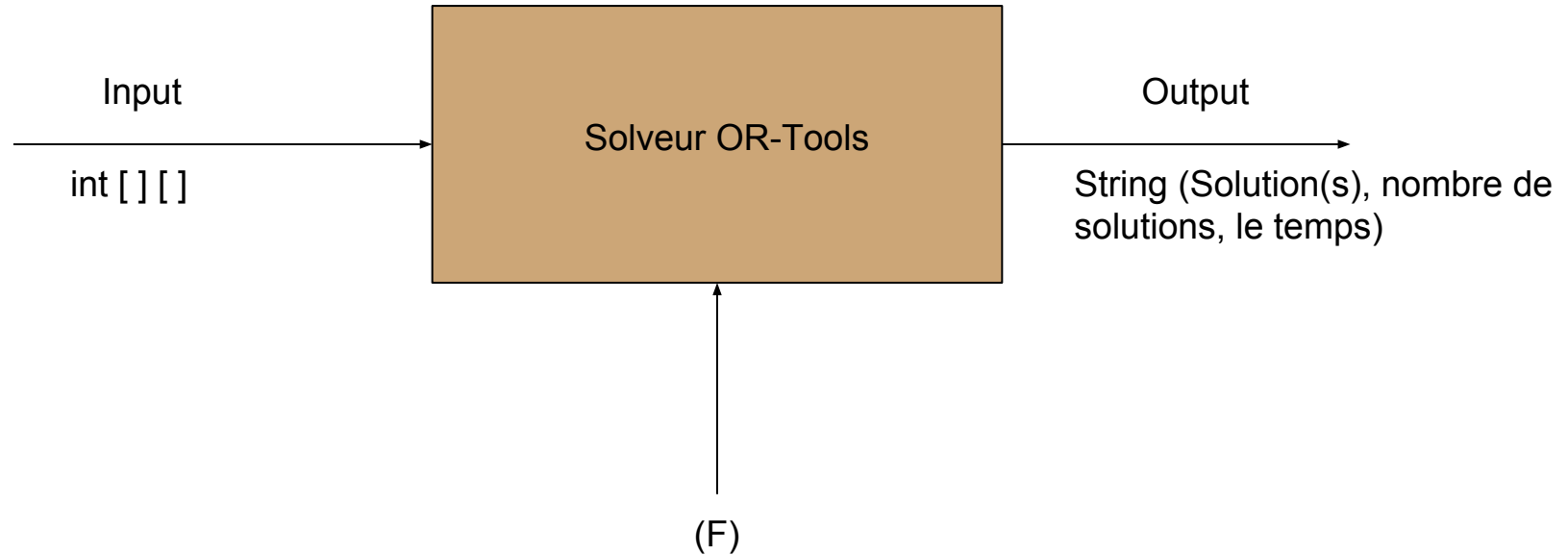
$$x \in (F) \{1, 2, \dots, 9\}$$

Parcours en profondeur

Principe de OR-Tools

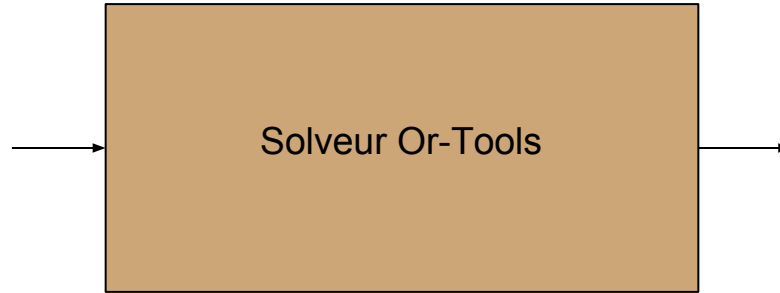


Principe de OR-Tools



Exemple

{ {0, 0, 3, 0, 2, 0, 6, 0, 0},
 {9, 0, 0, 3, 0, 5, 0, 0, 1},
 {0, 0, 1, 8, 0, 6, 4, 0, 0},
 {0, 0, 8, 1, 0, 2, 9, 0, 0},
 {7, 0, 0, 0, 0, 0, 0, 0, 8},
 {0, 0, 6, 7, 0, 8, 2, 0, 0},
 {0, 0, 2, 6, 0, 9, 5, 0, 0},
 {8, 0, 0, 2, 0, 3, 0, 0, 9},
 {0, 0, 5, 0, 1, 0, 3, 0, 0} }



4 8 3 9 2 1 6 5 7
9 6 7 3 4 5 8 2 1
2 5 1 8 7 6 4 9 3
5 4 8 1 3 2 9 7 6
7 2 9 5 6 4 1 3 8
1 3 6 7 9 8 2 4 5
3 7 2 6 8 9 5 1 4
8 1 4 2 5 3 7 6 9
6 9 5 4 1 7 3 8 2

Solutions: 1

Failures: 0

Branches: 0

Wall time: 16ms

Plan du Cahier des Charges

- I. Objectifs
 - A) A propos du Sudoku
 - B) A propos du solveur SMT
- II. Présentation du problème posé
 - A) Modélisation
 - B) Résolution
- III. Evaluation
 - A) Comparaison avec d'autres solveurs

Questions ?

Merci!

Nous contacter:

david.toty@etu.upmc.fr

maxime.tran@etu.upmc.fr

