

TP n°2

L'objectif de ce TP est de prendre en main le solveur **Minizinc** permettant de modéliser et de résoudre des problèmes à base de contraintes. Les liens et les sources sont donnés sur la page du cours sur **madoc**.

Un modèle Minizinc du problème des reines apparaît ci-dessous.

```
include "globals.mzn";
int: n = 4;
array[1..n] of var 1..n: q; % q[1],...,q[n] de domaine 1..n
constraint alldifferent(q);
constraint forall(i in 1..n, j in i+1..n) (abs(q[j] - q[i]) != j - i);
solve satisfy;
output ["q = ", show(q), "\n"];
```

Ce modèle est structuré de la manière suivante.

- L'instruction **include** permet d'importer la spécification des contraintes globales se trouvant dans le fichier **globals.mzn**.
- Le nombre de reines **n** est défini comme une constante entière de valeur 4.
- Les variables entières sont définies comme un tableau **q** d'indices **1..n**, chaque variable ayant pour domaine l'intervalle d'entiers **1..n**.
- Les deux contraintes (connues) apparaissent à la suite.
- L'instruction **solve satisfy** indique au solveur de résoudre un problème de satisfaction de contraintes.
- L'instruction **output** permet de définir le format d'affichage des solutions.

La commande **minizinc queens.mzn** permet de lancer la résolution. Le processus s'arrête dès que la première solution a été trouvée ou si aucune solution n'a été trouvée.

```
q = [3, 1, 4, 2]
-----
```

L'option **--all-solutions** en ligne de commande permet de chercher toutes les solutions.

Dans ce TP, vous pourrez accomplir les tâches suivantes.

- Lancer le solveur sur le modèle des reines donné ci-dessus. Comparer avec votre code.
- Tester un autre modèle des reines se trouvant dans le fichier **queen_cp2.mzn** du répertoire **doc/examples** de la distribution.
- Commencer la lecture du tutoriel et consulter la spécification si besoin.
- Modéliser et résoudre des problèmes déjà vus en TD (certains se trouvent dans le répertoire **doc/examples**, d'autres sont récupérables facilement sur le web).