

## Examen du Lundi, 21 mars 2016

Durée : 2 heures.

Documents autorisés : transparents du cours, énoncés et corrigés de TP, notes personnelles. Les ordinateurs ne sont pas autorisés. Les téléphones portables, comme tout autre moyen de communication vers l'extérieur, doivent être éteints.

### Exercice 1

Une entreprise de transport maritime doit transporter un certain nombre de conteneurs vers une destination lointaine, et pour cela elle doit affréter des bateaux de cargo.

Les bateaux disponibles pour la location sont classés en types, chaque type étant identifié par un nom. Les informations données pour chaque type sont sa capacité (en nombre de conteneurs par bateau), le nombre de bateaux de ce type disponibles, le prix de location d'un seul bateau de ce type, et le nombre de jours que prend un bateau de ce type pour le trajet qui nous intéresse ici. La description de la flotte est donnée en forme d'un tuple Oz, par exemple

```
fleet(a: ship(capacity: 100 number: 5 price: 1000 days: 10)
      b: ship(capacity: 500 number: 3 price: 3000 days: 14))
```

décrit une flotte de 5 bateaux de type *a* et 3 bateaux de type *b*. Un bateaux de type *a*, par exemple, peut transporter 100 conteneurs au maximum, coûte 1000 euros à la location, et prend 10 jours pour le trajet.

1. Écrire une fonction Oz, en utilisant des contraintes à domaines finis, qui prend en argument une description de la flotte comme au-dessus, et un nombre de conteneurs à transporter. La fonction doit retourner (dans une représentation de votre choix) les types et nombres de bateaux que la transporteur doit louer pour transporter tous les conteneurs au moindre coût.
2. Maintenant, le client exige que les conteneurs seront délivrés en 12 jours au plus tard, et oblige le transporteur de payer une pénalité de 3 euro par jour pour chaque conteneur qui arrive à sa destination en retard. Modifiez votre solution de la première question pour calculer la solution la moins chère (pour le transporteur) sous ces nouvelles conditions.

Faire attention au fait qu'un bateau peut être rempli au dessous de sa capacité, et que la pénalité est due seulement pour les conteneurs qui arrivent effectivement en retard.

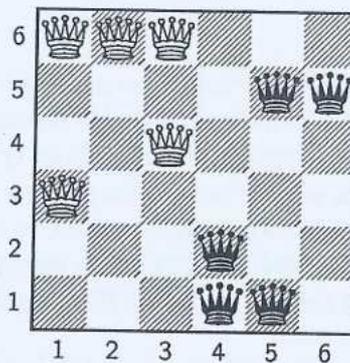
Les deux programmes demandés doivent trouver une solution optimale pour une flotte quelconque et un nombre de conteneurs quelconque.

### Exercice 2

On dispose d'un nombre illimité de dames blanches et de dames noires qu'on peut placer sur un échiquier de largeur  $N$  (qui donc a  $N \times N$  cases). Il n'est pas possible de placer deux dames sur la même case, mais une case peut rester inoccupée.

Les dames blanches et les dames noires placées sur l'échiquier forment deux armées, l'armée blanche et l'armée noire. Les deux armées sont en *paix* si aucune dame blanche ne menace une dame noire, et inversement.

On appelle un placement des dames sur l'échiquier *équitable* si les deux armées sont en paix et de taille égale. Un placement équitable avec un nombre maximal de dames et appelé *optimal*. Voici par exemple un placement optimal pour  $N = 6$  :



Écrire une fonction  $OZ$ , en utilisant des contraintes à domaines finis, qui prend en argument une largeur  $N$  de l'échiquier, et qui retourne un placement optimal.

*Indications* : Modéliser chaque case de l'échiquier par une variable à domaine fini, avec trois valeurs différentes qui représentent *dame blanche*, *dame noire*, et *inoccupée* dans un ordre de votre choix. Précisez la représentation que vous avez choisie.

Quelles symétries du problème peut-on exploiter ?