

**ÉCOLE POLYTECHNIQUE DE MONTRÉAL**

**INF8215**

Intelligence artif.: méthodes et algorithmes

**A2018 - Travail pratique #2**

**Présenté à :**

Mathieu Kabore 1708751

Florence Gaborit ...

Reph dauphin Mombrun 1662298

**Date : 11 Novembre 2018**

Intelligence artif.: méthodes et algorithmes	0
Choix d'implantations, méthode et motivations	2
Exercise 1	2
Exercise 2	3
Exercise 3	3
Exercise 4	4

# Choix d'implantations, méthode et motivations

## Exercice 1

Tout d'abord, nous avons considéré le problème de cet exercice comme étant un problème de satisfaction de contraintes tel que suggéré dans la description du travail. Il s'agit précisément d'une spécialisation des problèmes d'identification que nous avons traité selon les étapes ci-dessous.

Dans un premier temps, l'analyse du problème nous a permis d'identifier cinq (5) types d'entité qui sont composés de 5 variables chacun, d'où un total de **25 variables** listées ci-dessous:

### Variables :

- Les couleurs des **maisons**: *rouge, vert, blanc, jaune, bleu*
- Les différentes **nationalités**: *anglais, espagnol, ukrainien, norvégien, japonais*
- Les **animaux**: *chien, escargot, renard, cheval, zèbre*
- Les **professions** des individus: *sculpteur, diplomate, violoniste, médecin, acrobate*
- Les **boissons**: *café, thé, lait, jus, eau*

Dans un second temps, nous avons procédé à l'identification du domaine du problème, c'est à dire les valeurs possibles que peuvent prendre les 25 variables trouvées. Nous avons représenté le domaine par des identifiants de position qui permettent de mettre en relation les variables lors de la représentation des contraintes.

**Domaine** : { 1, 2, 3, 4, 5 }

Enfin, nous avons procédé à l'identification des divers contraintes selon l'énoncé du problème en suivant le formalisme suivant:

**anglais = rouge** signifie que l'anglais habite à la maisons rouge

**norvégien = 1** signifie que le norvégien habite dans la première maison à gauche, ainsi de suite jusqu'à épuisement des toutes les contraintes.

La liste exhaustive des contraintes peut être trouvée dans le code, accompagné de commentaires bien explicites.

Cette formulation du problème nous a permis de créer rapidement un modèle avec l'outil **Minizinc** et de procéder à sa résolution en utilisant les fonctions internes du langage.

## Exercice 2

L'objectif de cet exercice est de résoudre un problème de planification et d'ordonnancement d'un tournoi de football connu sous le nom de *Round-Robin Tournament*. Tout comme l'exercice précédent, il s'agit également d'un problème de satisfaction de contraintes, même si elle comporte plusieurs particularités au niveau des contraintes. La méthode de résolution est donc la même, et est basée sur l'identification des **variables**, la définition du **domaine**, la modélisation des **contraintes**, puis la résolution du modèle défini.

### Variables:

- **Calendrier[ i ][ j ]**: tableau à deux (2) dimensions, **i** allant du **1** à **14**, le nombre d'équipe
- **Pv[ k, l ]**: permet d'identifier la location d'un match entre l'équipe **k** et **l**
- **nTeam**: Le nombre d'équipes dans le tournoi

### Domaines:

- Pv: { 0, 1 }, Calendrier: { À compléter }
- Il faut noter que nous avons opté d'utiliser le fichier de données de **MiniZinc** N14.dzn qui contient déjà 14 équipes et le tableau des locations des matchs pv.

### Contraintes:

Nous avons identifié trois (3) principales contraintes pour la représentation du modèle:

- Appliquer une contrainte d'unicité sur les matchs par date dans le calendrier à remplir
- Appliquer une contrainte d'unicité sur les dates par match dans le calendrier à remplir
- Appliquer une contrainte sur le nombre de matchs successifs qui est limité à 4 dans le calendrier à remplir et sur les locations (à domicile ou à l'extérieur)

### Expliquez une symétrie dans le problème et ajouter une contrainte redondante afin de la briser.

Une symétrie évidente que nous avons trouvée dans le problème se trouve dans la représentation d'un match entre l'équipe qui joue à domicile et celle qui joue à l'extérieur. Le tableau à deux (2) dimensions nous permet de profiter de cette symétrie pour réduire le coût de la résolution en temps.

### Qu'observez-vous quant au temps de résolution ? Pourquoi une si grande différence ?

Le temps de résolution est pratiquement considérable entre les deux (2) implémentations. La différence est grande parce qu'il faut un temps aussi grand que la version améliorée pour résoudre l'autre moitié.

Bonus

## Exercice 3

La façon dont nous avons procédé est la suivante:

Un prédicat a été défini pour chacun des cours, chacune des relations directes de prérequis et de corequis mentionnées dans l'énoncé. Une règle de commutativité a été ajoutée pour assurer la symétrie de corequis/2. Une règle de transitivité a aussi été ajoutée pour corequis.

La convention établie pour l'ordre de prérequis/2 est que  $\text{prerequis}(a, b)$  se lit "a est prérequis de b". Finalement, une règle nommée "coursAPrendreComplet/2" dont le but est de renvoyer les cours corequis ou prérequis à un autre a été défini.

## Exercise 4

Nous avons défini plusieurs prédicats pour modéliser plusieurs catégories. En premier lieu, nous avons catégorisé la liste de personnes en hommes et femmes. Ces deux catégories constituent des partitions de l'ensemble de personnes. Ensuite, nous avons listé les catégories film, jeu vidéo, réel, en vie, politicien, religieux, vieux (né avant 1960 par convention), tafta, acteur, musicien, artiste. Nous avons aussi listé des prédicats pour définir les objets.