

## Contrôle Continu HMIN107 – Intelligence Artificielle

Octobre 2018 – Durée 1h – aucun document autorisé

### Exercice 1. Planification

4 points

On cherche à planifier des tâches pour mener à bien un mini-projet de développement logiciel. L'équipe dispose de 120 jours. Les différentes tâches identifiées sont :

- B : recueillir les besoins du client ; 1
- S : définir les spécifications du projet ; 2
- A : concevoir l'architecture du projet ; 3
- M : modéliser de façon détaillée le programme ; 4
- I : implémenter le projet ;
- U : écrire le manuel d'utilisation ;
- D : écrire la documentation de maintenance du programme ;
- P : présenter le projet fini au client durant une journée.

On souhaite associer à chacune des tâches un jour de début et un jour de fin. Le projet doit commencer dès le jour 1 par le recueil des besoins qui devrait durer au moins 15 jours et au plus 30 jours. Les spécifications ne peuvent commencer à être définies que 5 jours avant la fin du recueil des besoins. La modélisation détaillée ne peut commencer qu'après la fin de la phase de conception de l'architecture. La conception de l'architecture ne peut commencer qu'après la fin de la phase de spécification et ne doit pas durer au-delà du 50<sup>ème</sup> jour du projet. L'implémentation du projet ne peut commencer qu'après la fin de la phase de modélisation détaillée. La rédaction du manuel d'utilisation ne peut commencer qu'après la fin de la phase de spécification et celle de la documentation de maintenance après la fin de la modélisation détaillée. Tout doit être fini avant le jour de la présentation qui doit avoir lieu au plus tard le 120<sup>ème</sup> jour.

Modéliser ce problème sous la forme d'un réseau de contraintes en intension. Vous préciserez quelle est l'arité de vos contraintes.

### Exercice 2. Résolution d'un réseau de contraintes

16 points

Soit le réseau de contraintes  $\langle X, D, C \rangle$  suivant :

$$X = \{a, b, c, d, e, f, g, h\}$$

$$D(a) = D(e) = \{1, 2\}$$

$$D(f) = D(h) = \{1, 2, 3\}$$

$$D(b) = D(c) = D(d) = D(g) = \{1, 2, 3, 4, 5\}$$

$$C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\} \text{ avec :}$$

$$c_1 : a < b$$

$$c_2 : a \neq d$$

$$c_3 : b + 2 \leq c$$

$$c_4 : b > d + 1$$

$$c_5 : d + e + f = g$$

$$c_6 : a + 1 < g$$

$$c_7 : f > h$$

- 1- Exprimer les contraintes en extension.
- 2- Représenter graphiquement ce réseau de contraintes (on dessinera le graphe biparti variables / contraintes).

- 3- Représenter l'exécution de l'algorithme de Backtrack pour la **recherche d'une solution** en utilisant **d, b, a, g, e, f, c, h** comme ordre d'affectation des variables et l'**ordre croissant sur les entiers pour les valeurs**. On attend que vous dessiniez l'arbre de recherche dont les noeuds représentent une assignation (de plus en plus complète) ; la racine de l'arborescence représentant l'assignation vide. Chaque niveau de l'arbre est dédié à une variable. Pour faciliter la représentation **on se limitera à indiquer sur chaque noeud la valeur assignée** à la variable correspondant au niveau courant (l'assignation courante étant donc « lue » en remontant le chemin du noeud courant à la racine). Lorsqu'une assignation viole une contrainte, on indique par un x que le noeud n'est pas prolongé en **précisant l'une des contraintes violées**.
- 4- Donner la solution calculée.
- 5- Représenter l'exécution de l'algorithme de Forward Checking à la **recherche d'une solution** en utilisant l'heuristique dynamique *dom+deg+alpha* (plus petit domaine courant, puis en cas d'égalité plus grand degré dans le graphe de contraintes, et enfin ordre alphabétique si encore égalité) pour l'ordre d'application des variables et l'ordre **décroissant** sur les entiers pour les valeurs. On indiquera à chaque nœud de l'arbre la **variable assignée, la valeur choisie et les modifications apportées par propagation aux domaines** des autres variables.
- 6- Donner la solution calculée.
- 7- Ce réseau est-il arc-consistant (c'est à dire chaque valeur du domaine de chaque variable a-t-elle un tuple support dans chaque contrainte où la variable apparaît) ? Justifier votre réponse.
- 8- Calculer la fermeture arc-consistante de ce réseau.
- 9- Calculer l'ordre d'assignation des variables engendré par l'heuristique statique combinée suivante : *max-card* (variable connectée au plus grand nombre de variables déjà choisies), puis *max-degree* (plus grand degré dans le graphe de contraintes) en cas d'égalité, puis enfin l'ordre alphabétique si deux variables sont encore de même priorité. Vous préciserez quelles contraintes doivent « réellement » être vérifiées par l'algorithme de backtrack après chaque assignation de variable.
- 10- Réappliquer l'algorithme de backtrack à la recherche de **toutes les solutions** sur le réseau de contraintes arc-consistant obtenu à la question 6 en utilisant l'ordre calculé à la question précédente et l'**ordre croissant** sur les valeurs.
- 11- Donner le nombre de solutions à ce réseau.