

Exam - Décisions sous incertitude : Troll

École des Mines de Douai – FI 2A

Retro-engineering

Vous êtes en possession du script suivant décrivant le comportement d'un Troll dans un jeu vidéo et vous souhaitez comprendre ce comportement.

Listing 1 – Script

```
String behavior( int [] s )
{
    if( s[0] == 1 )
    {
        if( s[1] == 1 )
            return "run_away";
        else
            return "attack";
    }
    else
    {
        if( s[1] < 3 )
            return "sleep";
        else
        {
            if( s[2] == 0 )
                return "explore";
            else
                return "socialize";
        }
    }
}
```

Question 1 (3 points) : Construire l'arbre de décision correspondant au comportement du Troll.

Question 2 (1.5 points) : Énumérer les variables et les domaines de variation possibles pour ce comportement.

Question 3 (0.5 point) : Quel est la taille de l'espace d'état que couvre cet arbre de décision ?

Question 4 (1 point) : Qu'elle est l'action retournée par l'état [1, 2, 1] ?

Question 5 (1 point) : Combien d'états sont représentés par la feuille pointée par l'état [1, 2, 1] ? (calculs et résultats)

Question 6 (1 point) : Proposer des noms cohérents pour chacune des variables d'état avec une courte justification.

1 Probabilities on attack outcome

On souhaite modéliser un combat entre un Troll et un Chevalier (Knight) de façon à améliorer le comportement du Troll. Le Troll et le Chevalier ont tous les deux un niveau de vie (health) débutant à 3 et descendant jusqu'à 0 (3 : healthy; 2 : hurt; 1 : dying; 0 : dead). Pendant une attaque d'un Troll, le Troll peut potentiellement toucher le chevalier et réduire sa vie avec de 1 à 3 dommages. Dans des conditions normales, le Troll a 0.1 chance de faire 1 dommage, 0.4 chance de faire 2 dommages et 0.2 chance de faire 3 dommages, autrement, le chevalier esquivé l'attaque. Si la vie du Troll est diminuée (hurt, dying), ses chances de faire des dommages sont réduites de la moitié de leur valeur pour chaque point de vie perdu par le Troll. De plus, le chevalier possède potentiellement un bouclier (shield), dans ce cas, les dommages qu'il reçoit sont diminués de 1.

Question 7 (2 points) : Proposer un réseau Bayésien permettant de calculer les probabilités sur les dommages causés par le Troll lors d'une attaque.

Question 8 (2 points) : Donner le tableau des probabilités conditionnelles attachées au nœud représentant les dommages infligés par le Troll.

Question 9 (1 point) : En considérant que le Troll est en pleine forme (healthy) et que la moitié des Chevaliers rencontrés utilise un bouclier, quelle est la probabilité que le Troll tue le chevalier à la première attaque ? (calculs sur la base du réseau Bayésien proposé et résultat)

Question 10 (2 points) : Un combat entre un Troll et un Chevalier est composé par plusieurs attaques du Troll où le chevalier pourra contre-attaquer s'il est en vie et s'il n'a subi aucun dommage (après une esquive ou si le bouclier a absorbé tous les dommages). Une contre-attaque fait 1 dommage avec une probabilité de 0.5 et 2 dommages avec une probabilité de 0.2 quelque soit l'état de santé du Troll ou du Chevalier. Enfin, les dommages causés par le Chevalier sont augmentés de 1 s'il n'a pas de bouclier (lui permettant d'être mieux armé). Proposer un nouveau réseau Bayésien modélisant l'évolution des points de vie du Troll et du Chevalier.

Question 11 (1 point) : Quel est le nom de ce type de réseau bayésien ?

Question 12 (3 points) : Implémenté en Java la fonction qui retourne les probabilités sur l'évolution des points de vie du Troll en s'appuyant sur le réseau Bayésien proposé. La fonction prend autant de paramètres qu'il y a de variables utiles au calcul de ces probabilités (sous forme d'entiers avec des noms appropriés) et retourne un tableau de 4 probabilité (une pour chacune des 4 niveaux de vie possibles). Exemple de signature de fonction attendue (avec *nva* pour nom de variable approprié) :

`double [] trollHealthEvolution(int nva1, int nva2, int nva3, [...])`

2 Markov Decision Process

Basé sur le réseau Bayésien modélisant un combat, on propose de développer un processus de décision Markovien pour modéliser un combat entre un Troll et un Chevalier utilisant un bouclier. Les états sont simplement construits sur les points de vie du Troll et du Chevalier (vivant) et avec trois états supplémentaires modélisant le fait que le Troll ou le Chevalier est mort et la fin du combat. La décision pour le Troll consiste à attaquer (attack) ou fuir (run away). L'action "run away" termine systématiquement dans l'état *End* fin du combat, avec une probabilité à 1.0. Les transitions de l'action "attack" sont modélisées par la figure ci après. Le modèle est composé de 12 état et 2 actions. La fonction de récompense représente le plaisir qu'a le Troll d'attaquer et tuer du Chevalier.

$$r(s, a) = \begin{cases} -10 & \text{if Troll is dead} \\ 8 & \text{if Knight is dead} \\ 0 & \text{if combat is ended} \\ 2 & \text{if Troll is attacking (combat not ended and Troll or Knight not dead)} \\ 0 & \text{if Troll is running away} \end{cases}$$

Question 13 (1 point) : En considérant l'équation de Bellman, quelle est la meilleure action et sa valeur associée au pas de temps $t = 1$ pour chacun de ces 5 états : (*End* ; *Troll(0)* ; *Knight(0)* ; *Troll(2)-Knight(1)* ; *Troll(1)-Knight(1)*).

équation de Bellman :
$$V_t(s) = \max_a \left(r(s, a) + \sum_{s' \in S} t(s, a, s') V_{t-1}(s') \right) \quad \text{avec} \quad V_0(s) = 0$$

Question 14 (2 points) : Même question, pour le pas de temps $t = 2$.

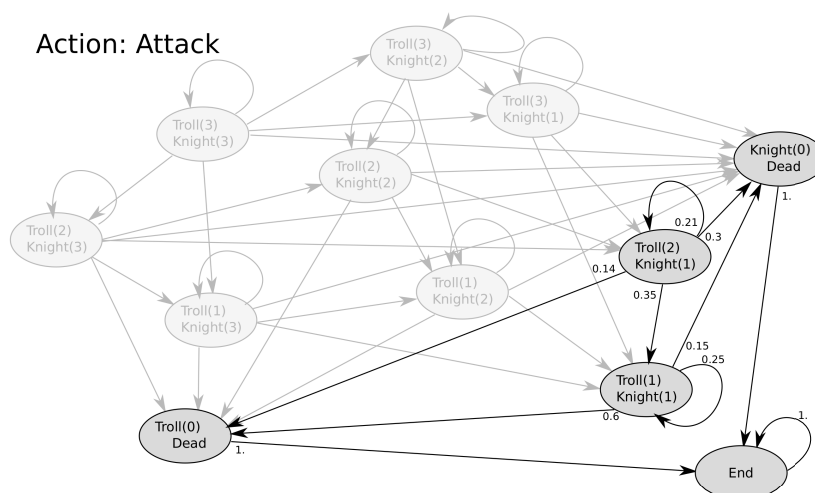


FIGURE 1 – Fonction de transition pour l'action "attack" (focalisé sur les états ciblés par les questions 13 et 14)