Nom :…………………………….. **Prénom :………………………** **CIN :………………………** **Salle :………**

Année universitaire 2019-2020

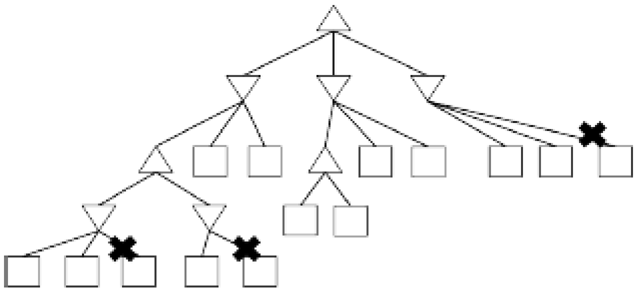
Correction Examen Intelligence artificielle - Final

Enseignants	:M. Fourati, H. Ghazouani	**Date**	: 30/06/2020	**Nbre. de pages**	**: 6**
Filière / Classe	: 2ème Ing. Inf.	**Calculatrices et documents**	: non autorisés	**Durée**	: 1h30

Exercice 1 : Questions de cours (2,5 points : 0,25 pour une bonne réponse, -0,25 pour une mauvaise réponse et 0 si pas de réponse) (V : Vrai, F : Faux)

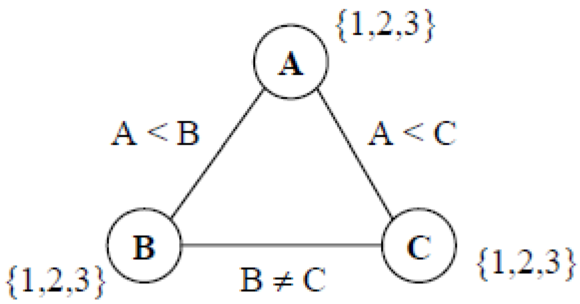
+0.25/bonne réponse

-0.25/mauvaise réponse

	VRAI	FAUX
Les heuristiques utilisées dans les problèmes à satisfaction de contraintes (CSP) ne peuvent pas être généralisées pour tous les problèmes de ce type.	☐	X
L'algorithme AC-3 de cohérence d'arc n'est pas toujours suffisant pour éliminer toutes les incohérences dans un CSP.	X	☐
La contrainte globale AllDiff ne peut pas être exprimée par des contraintes binaires.	☐	X
L'heuristique du degré permet de privilégier la variable ayant le moins de valeurs dans son domaine.	☐	X
L'algorithme α-β pruning est généralement plus rapide que miniMax mais il perd la garantie d'optimalité.	☐	X
L'algorithme α-β pruning retourne toujours les mêmes valeurs minimax pour tous les nœuds de l'arbre que l'algorithme miniMax.	☐	X
L'algorithme α-β pruning retourne toujours la même valeur minimax du nœud racine que miniMax.	X	☐
La valeur minimax d'un état est toujours inférieure ou égale à la valeur expectiminimax du même état.	X	☐
L'algorithme MiniMax retourne toujours le meilleur coup à jouer pour un joueur indépendamment du fait que les joueurs jouent ou non d'une manière optimale.	☐	X
Un algorithme α-β pruning opérant de gauche à droite, l'élagage suivant peut être atteint : 	☐	X

<u>Exercice 2 :</u> (Consistance d'arcs : 3 points) 3 points

Dans le graphe de contraintes suivant, les contraintes sont identifiées sur les liens et le domaine de chacune des variables est indiqué entre accolades. Montrez toutes les étapes de l'algorithme de cohérence des arcs AC-3 sur ce problème pour le rendre arc-cohérent. Vous devez identifier tous les arcs qui sont vérifiés et montrer les changements aux domaines de valeur des variables à chaque étape. Identifiez les arcs initiaux et ceux ajoutés tout au long de l'application de l'algorithmes ; <u>pour ces derniers identifiez les par une autre couleur.</u>



Arc à traiter	Domaine de A	Domaine de B	Domaine de C	File des Arcs
INIT	1, 2, 3	1, 2, 3	1, 2, 3	**0.25**
A→B	1, 2, ~~3~~ **0.25**	1, 2, 3	1, 2, 3	A→B ,A→C , B→A, B→C, C→A, C→B **1.5**
A→C	1, 2	1, 2, 3	1, 2, 3	A→C , B→A, B→C, C→A, C→B
B→A	1, 2	~~1~~, 2, 3 **0.25**	1, 2, 3	B→A, B→C, C→A, C→B, **A→B 0.25**
B→C	1, 2	2, 3	1, 2, 3	B→C, C→A, C→B, **A→B**
C→A	1, 2	2, 3	~~1~~, 2, 3 **0.25**	C→A, C→B, **A→B, A→C 0.25**
C→B	1, 2	2, 3	2, 3	C→B, **A→B, A→C**
A→B	1, 2	2, 3	2, 3	**A→B, A→C**
A→C	1, 2	2, 3	2, 3	**A→C**
Domaines finaux	**1, 2**	**2, 3**	**2, 3**	

<u>Exercice 3 :</u> (CSP - Définition: 2,5 points) 2.5

On vous demande de disposer trois statuettes dans un hall : une sculpture d'un cygne en glace (g), un lion en or (o) et une pièce abstraite en marbre (m). Il y a trois tables : 1, 2 et 3, disposées en ligne, dont la première (1) à côté de la porte d'entrée. Comme c'est une journée un peu chaude, la sculpture en glace ne peut se placer à côté de la porte. Le responsable de la salle vous informe également qu'il ne serait pas judicieux de placer deux statuettes d'animaux l'une à côté de l'autre. Il est évident que chaque table doit contenir une statuette différente.

1. Formulez ce problème comme un problème de satisfaction de contrainte CSP en précisant les variables, les domaines et les contraintes unaires et binaires.

   P=(X,D,C)

   Deux solutions existent :

Solution 1

X	g, o, m **0.25**
D	1, 2, 3 **0.25**
C	g≠1 (unaire) **0.25** g≠o, g≠m, o≠m (binaires) **0.75** |g-o| ≠1 (binaire pour l'adjacence) **0.5**

Solution 2

X	X_1, X_2, X_3 **0.25**
D	g, o, m **0.25**
C	X_1≠g (unaire) **0.25** Binaires : (X_1, X_2)∈{(g,m),(m,g), (o,m), (m,o)} (X_2, X_3)∈{(g,m),(m,g), (o,m), (m,o)} (X_1, X_3)∈{(g,m),(m,g), (o,m), (m,o), (g,o), (o,g)} **1.25** Il est à noter que l'absence d'adjacence entre g et o est implicite puisque les domaines sont explicités. De même pour AllDiff.

2. Imaginez qu'on rend le CSP nœud-cohérent, quelle variable sera assignée en premier si on décide d'appliquer l'heuristique MRV ? Expliquez.

   Évidemment c'est la variable g ou X_1 puisque leur domaine a été réduit par l'application de la cohérence de nœud. 0.5 (0.25 pour la variable et 0.25 pour l'explication)

<u>Exercice 4 :</u> (CSP - Résolution : 4.5 points) 5,5 points

Votre ami a oublié le mot de passe (mdp) d'accès à son PC. Vous proposez de l'aider. Il vous donne les indications suivantes, qu'il a notées en définissant son mot de passe :

- Le mdp se compose de 4 chiffres **<u>non nuls</u>** C_1,C_2,C_3,C_4.
- Il n'a jamais utilisé les chiffres 1 et 9
- Les chiffres sont ordonnés en ordre décroissant ($C_1 > C_2 > C_3 > C_4$).
- $C_1 \leq 7$,
- $C_4 > 2$
- $C_2 < 6$

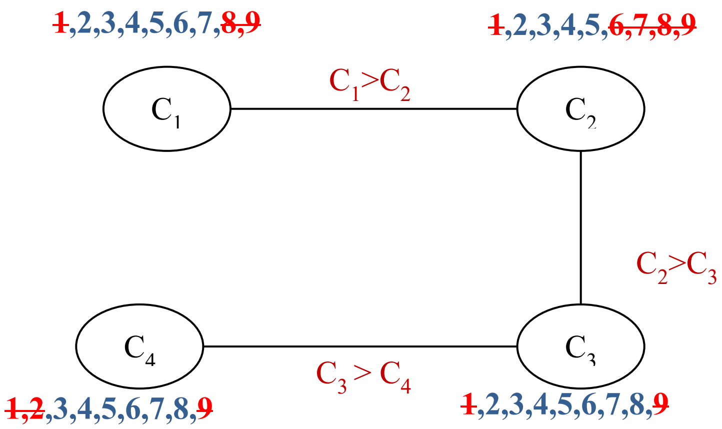
Nous nous proposons de considérer ce problème comme un CSP, dont les digits {C_1, C_2, C_3, C_4} sont les variables le domaine de chaque variable est [1,2,3,4,5,6,7,8,9] et les contraintes sont celles décrites ci-dessus, afin de récupérer le mot de passe.

1. Dessinez le graphe de contraintes de ce CSP (préciser les contraintes binaires sur les arcs et les domaines sur les nœuds). Ensuite <u>le rendre nœud-cohérent</u>.

Les modifications en rouges montrent le changement des domaines après avoir rendu le CSP nœud cohérent.

Le graphe non nœud-cohérence : **1**

Nœud-cohérence : **1**



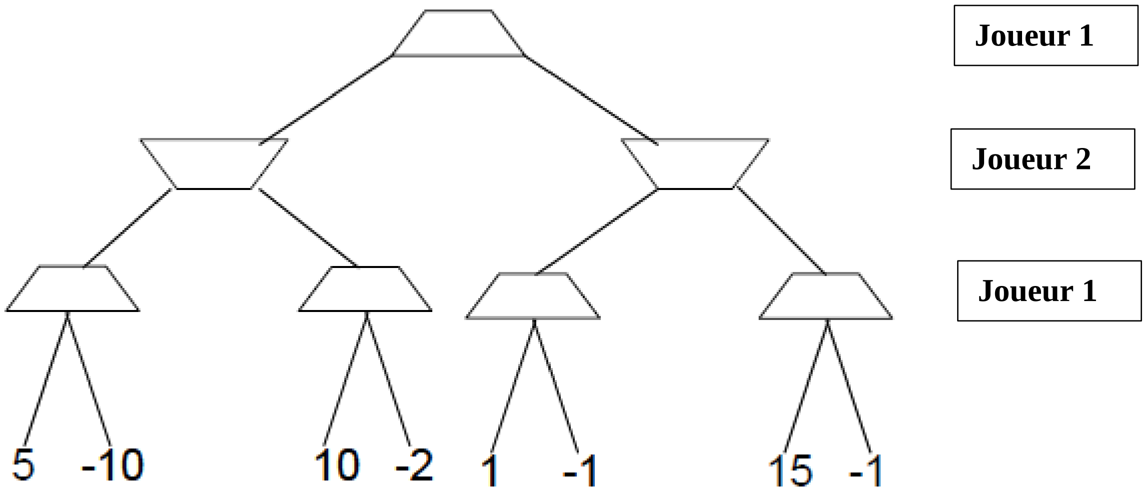
2. Utilisez l'algorithme MAC avec l'heuristique MRV pour résoudre ce problème. Spécifier les domaines des variables à chaque étape. En cas d'égalité, il faut supposer l'ordre d'assignation suivant : C_1, C_2, C_3, C_4. Pour les choix de valeurs, favorisez toujours les plus petites. Spécifiez les domaines des variables à chaque étape.

Assignation	**Dom C_1**	**Dom C_2**	**Dom C_3**	**Dom C_4**	**Arcs**
Init	2,3,4,5,6,7	2,3,4,5	2,3,4,5,6,7,8,9	3,4,5,6,7,8	
$C_2=2$ **0.25**	~~2~~,3,4,5,6,7 3,4,5,6,7	**2** **2**	2,3,4,5,6,7,8,9 ~~2,3,4,5,6,7,8,9~~ **Back 0.25**	3,4,5,6,7,8 3,4,5,6,7,8	$C_1 \rightarrow C_2$ $C_3 \rightarrow C_2$
$C_2=3$ **0.25**	~~2,3~~,4,5,6,7 4,5,6,7	**3** **3** **0.25**	2,3,4,5,6,7,8,9 2,~~3,4,5,6,7,8,9~~	3,4,5,6,7,8 3,4,5,6,7,8	$C_1 \rightarrow C_2$ $C_3 \rightarrow C_2$
AC-3	4,5,6,7	**3**	2	~~3,4,5,6,7,8~~ **0.25** **BACK**	$C_4 \rightarrow C_3$
$C_2=4$ **0.25**	~~2,3,4,~~5,6,7 5,6,7	**4** **0.25** **4**	2, 3,4,5,6,7,8,9 2,3,~~4,5,6,7,8,9~~	3,4,5,6,7,8 3,4,5,6,7,8	$C_1 \rightarrow C_2$ $C_3 \rightarrow C_2$
AC-3	5,6,7	**4**	2,3	~~3,4,5,6,7,8~~ **0.25** **BACK**	$C_4 \rightarrow C_3$
$C_2=5$ **0.25**	~~2,3,4,5,~~6,7 6,7	**5** **5** **0.25**	2,3,4,5,6,7,8,9 2,3,4,~~5,6,7,8,9~~	3,4,5,6,7,8 3,4,5,6,7,8	$C_1 \rightarrow C_2$ $C_3 \rightarrow C_2$

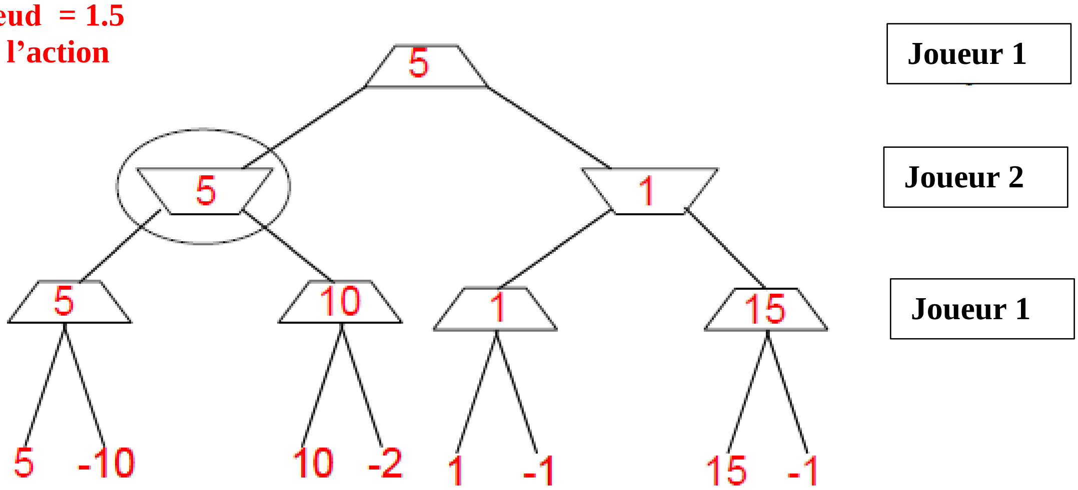
Assignation	Dom C_1	Dom C_2	Dom C_3	Dom C_4	Arcs
AC-3	6,7	**5**	**2,3,4**	**3,~~4,5,6,7,8~~** **0.25**	$C_4 \rightarrow C_3$
C4=3	6,7	**5**	**0.25** ~~3,~~4	**3**	$C_3 \rightarrow C_4$
C_3=4 **0.25**	6,7	**5**	**4**	**3**	$C_4 \rightarrow C_3$
C_1=6 **0.25**	**6**	**5**	**4**	**3**	$C_3 \rightarrow C_4$

<u>Exercice 5 :</u> (Algorithmes de jeux : 7,5 points) 6.5 points

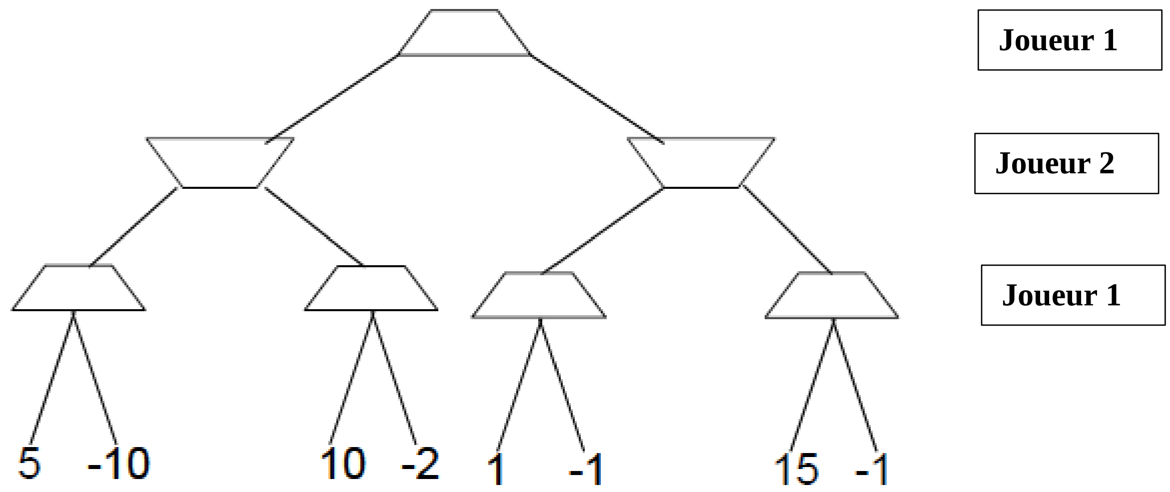
1. Considérez le jeu à somme nulle et à deux joueurs suivant. Les valeurs d'utilité des feuilles sont indiquées dans la figure. C'est le joueur 1 (MAX) qui commence. Supposons que les deux joueurs jouent d'une façon optimale. Remplissez les losanges par les valeurs minimax en appliquant l'algorithme MiniMax, ensuite encerclez le meilleur coup pour le joueur 1 (racine).

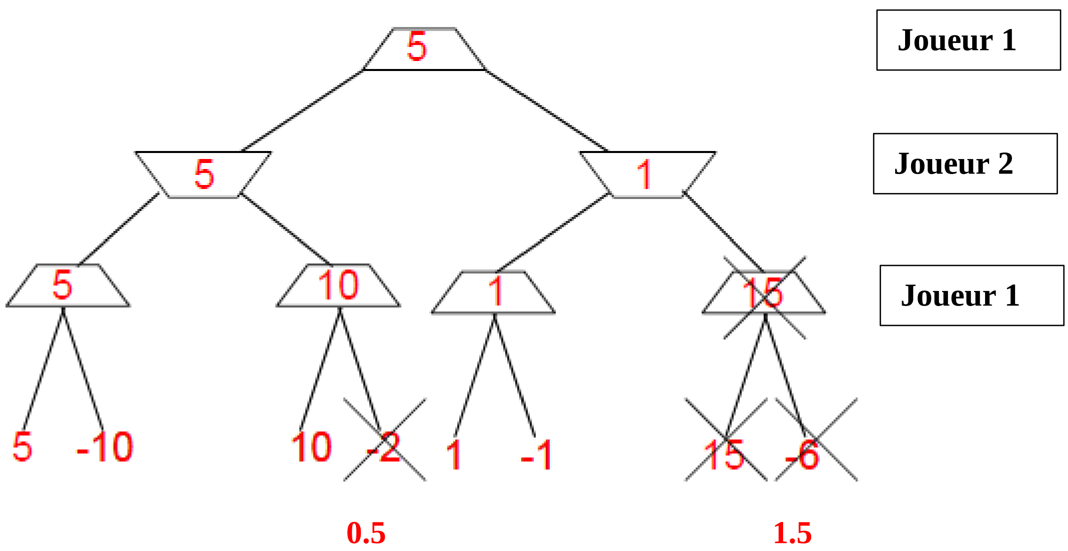


0.25 / nœud = 1.5
0.5 pour l'action

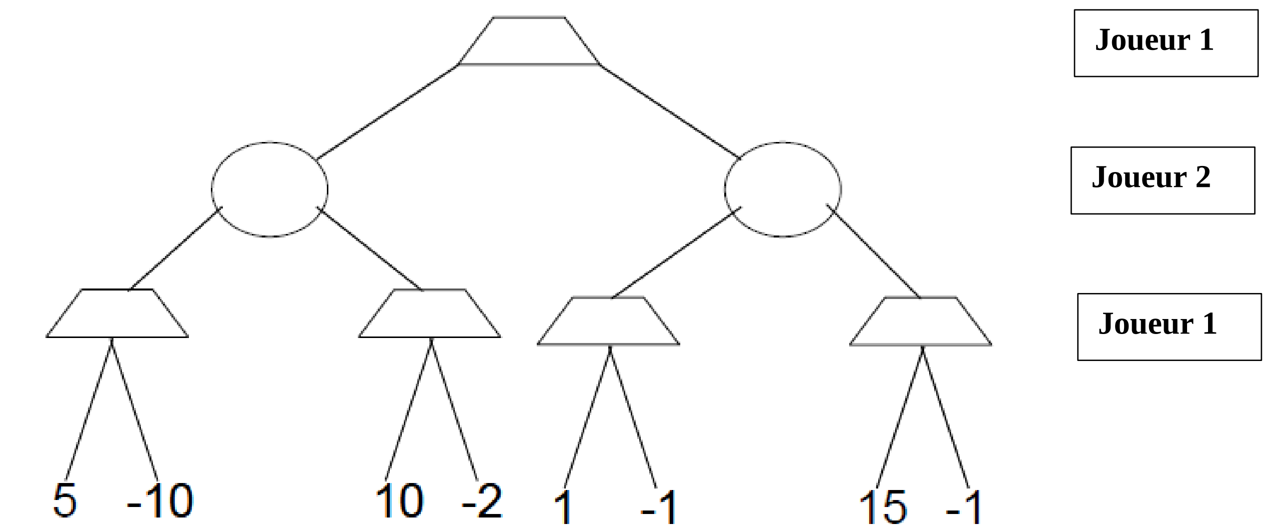


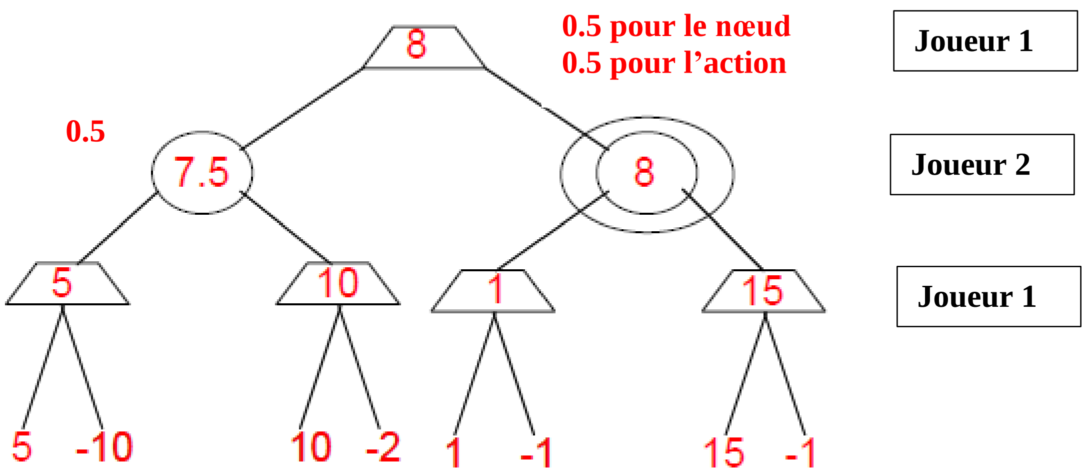
2. Nous restons dans le même contexte et nous vous demandons d'appliquer l'algorithme α-β d'élagage. Montrez les branches élaguées.



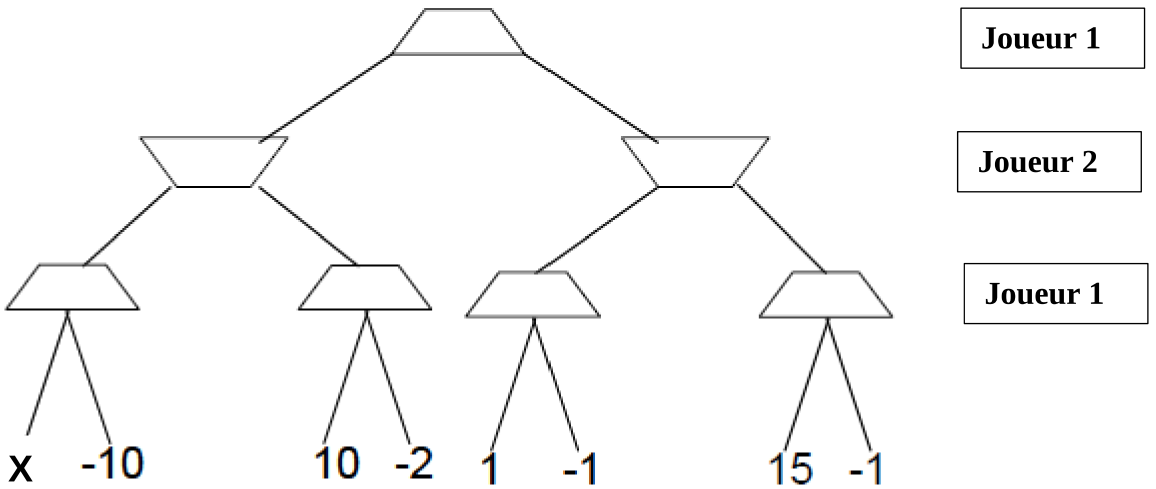


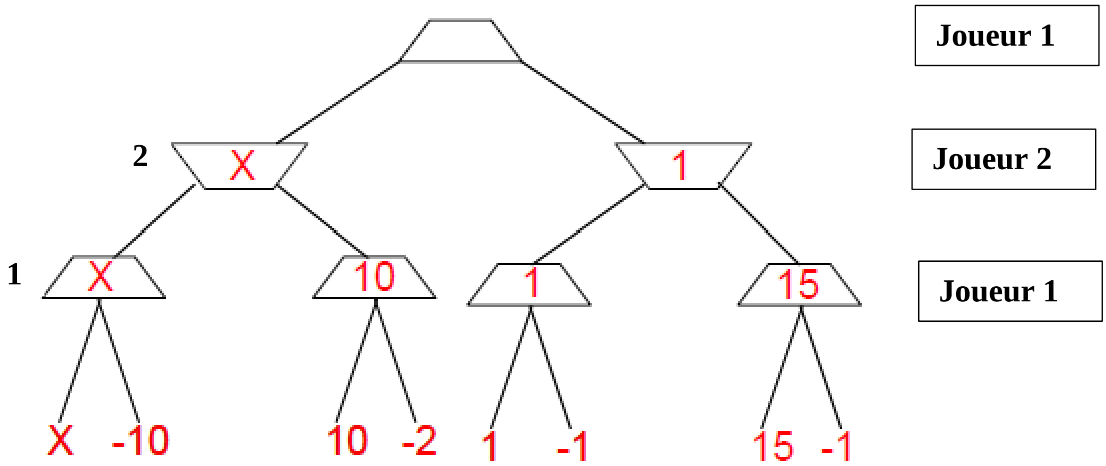
3. Nous restons dans le même contexte et nous supposons que le joueur 2 choisit à chaque tour une action d'une façon aléatoire (chaque action peut être choisie avec 50% de chance). Supposons également que le joueur 1 choisit encore ses actions de façon à maximiser ses chances de gagner. Remplissez les losanges par les valeurs minimax et expectiminimax ensuite encerclez l'action optimale pour le joueur 1 (racine).





4. Considérez la modification suivante sur l'arbre de jeu ; nous restons dans le premier contexte où les deux joueurs cherchent à maximiser leur gain. Imaginons que la valeur X est inconnue. Pour quelles valeurs de X, le joueur 1 choisira la branche gauche (votre gauche en regardant de face) ? Expliquez.





Il est à noter que si on cherche à ce que la branche gauche sera choisie par le joueur 1, il faut que le nœud **2** soit ≥ 1. Or, ce dernier prendra le minimum entre le nœud **1** et 10.

Pour le nœud **1**, il prendra la valeur maximale entre -10 et X. Si X<-10, le nœud **1** aura cette même valeur -10 qui sera remontée au nœud **2** et dans ce cas le joueur 1 choisira la branche droite. Donc, pour que le joueur 1 choisisse la branche de gauche, il faut que X soit >-10 (c'est pour cette raison que les nœuds **1** et **2** auront X comme valeur). On sait donc que -10<X<10, donc pour choisir la branche de gauche, le X doit être ≥1 qui est la valeur du fils droit de la racine. **(1 point)**