

## TP4IA\_Hategekimana\_Fabrice

### 5.1 Formulation du problème sous forme d'un csp

règles à vérifier: adjacence, côté

1. Déterminer:

1. Variables du problème:

- la position de la cave C ( $D=\{1,2,3,4\}$ )
- la position de la fenêtre F ( $D=\{1,2,3,4\}$ )
- la position de la porte P ( $D=\{1,2,3,4\}$ )

2. Possibilité:

- Je ne vois qu'une seule possibilité mais ce choix n'est pas unique
- On a donc les variables  $A=\{C, F, P\}$  ou  $A=\{C, W, D\}$

3. Contraintes du problèmes (forme  $\langle X1, \dots, Xn \rangle: \langle a1, \dots, an \rangle, \dots, \langle b1, \dots, bn \rangle$ )

- $\langle F, C \rangle: \langle \text{piece}(F) = X, \text{piece}(C) = Y \rangle, X \neq Y$
- $\langle F, C \rangle: \langle \text{piece}(F) = X, \text{adjacent}(C) = Y \rangle, X \neq Y$
- $: \langle \text{nb}(\text{adjacent}(F) = X) \rangle, X \geq 2$
- $: \langle \text{nb}(\text{adjacent}(P) = X) \rangle, X \geq 2$
- La fonction `piece()` retourne le numéro de la pièce où se trouve l'objet
- La fonction `adjacent()` retourne toutes les pièces adjacentes

### 5.2 Backtracking

**On utilise l'algorithme du backtracking:**

PSC\_BACKTRACKING(A: affectation, D:domaines) 1. Si A= S\_G alors retourner A 2. Sélectionner une variable x\_p non affectée 3. Pour chaque valeur v\_pi de D\_p faire: - Ajouter x\_p <- v\_pi dans A - D <- FORWARD\_CHECKING(A,D) - si aucun domaine de D n'est vide: - Retourner PSC\_BACKTRACKING(A,D) - sinon: - Retourner échec

Résultats:

Etape 1. CU C=0; D=0; W=0  
Etape 2. AV C=1; D=0; W=0  
Etape 3. FC C=1; D=0; W=0  
Etape 4. AV C=1; D=3; W=0

Etape 5. FC C=1; D=3; W=0  
Etape 6. AV C=1; D=3; W=3  
Etape 7. FC C=1; D=3; W=3  
final: { 'C': 1, 'D': 3, 'W': 3}}