Matricule **Nom** **Prénom**

EMD(Attention 2 pages Réponses QCM sur cette feuille)

QCM (3Pts): Cochez la ou les réponse(s) juste(s). Bonne réponse +0.5, Fausse réponse -0.5

1) $(\forall x)[P(x) \vee Q(x)] \equiv (\forall x)P(x) \vee (\forall y) Q(y)$ Vraie ☐ Faux ☐
2) $(\exists x)[P(x) \vee Q(x)] \equiv (\exists x)P(x) \vee (\exists y)Q(y)$ vraie ☐ Faux ☐
3) Soient 2 heuristiques *admissibles* h_1 et h_2 telles que $h_1(n) < h_2(n)\ \forall n$, alors avec h_2 A*:
   - explore moins ou autant de nœuds avant d'arriver au but qu'avec h_1 ☐
   - explore autant de nœuds avant d'arriver au but qu'avec h_1 ☐
   - explore plus de nœuds avant d'arriver au but qu'avec h_1 ☐
4) Soit une fonction heuristique non admissible, l'algorithme A* donne toujours une solution lorsqu'elle existe :
   - On est certain que le chemin est optimal ☐ On n'est pas certain que le chemin soit optimal ☐
5) Si les coûts des arcs dans un graphe sont tous égaux à 1 et la fonction heuristique $h(n)=0 \forall n$ alors:
   - A* retourne toujours une solution optimale lorsqu'elle existe. ☐
   - A* retourne toujours une solution qui n'est pas optimale lorsqu'elle existe ☐
6) Si $h(n) = h^*(n)\ \forall n$ *où* $h^*(n)$ est le cout du chemin optimal de n au but, alors :
   - A* donne un chemin optimal ☐ A* donne un chemin non optimal ☐

Exercice 1 (4Pts): Donner la définition de la substitution et de l'unification. La composition de la substitution est-elle commutative, justifier ? sinon dans quel cas elle est commutative ?

Exercice 2 (4.5Pts): Transformez les expressions suivantes en logique du 1er ordre.

1) Pour tout x et y, x est le maximum de x et y si et seulement si x est ≥ y sinon y est le maximum. Utiliser uniquement les prédicats ≥, < et = ainsi que la fonction *max(x,y)*.
2) Tout nombre impair ou égal à 2, peut etre premier. Utiliser les prédicats *impair*, = et *premier*.
3) Une variable x possède la propriété p si et seulement si, x possède les 2 propriétés a et b ou la propriété c. Utiliser les prédicats *a(x), b(x), c(x)* et *p(x)*.

Exercice 3 (3.5Pts) Rappeler les étapes de transformations de fbf en clauses, puis l'appliquer à:

$$((\forall x)(\exists y) P(x) \Rightarrow (q(x) \wedge m(y, x))) \vee \neg[(\exists x) R(x) \wedge S(x)]$$

Exercice 4 (3Pts) Une famille composée du père et de la mère pesant 80kg chacun et de 2 enfants (une fille et un garçon) pesant 50kg chacun. Ils sont au rez de chaussée et ils veulent prendre l'ascenseur pour rejoindre leur étage. L'ascenseur qui se trouve actuellement au rez de chaussée, ne peut prendre que 120 kg à la fois et il faut quelqu'un dans l'ascenseur pour actionner le bouton de marche. Comment faire ?

1-Exprimer ce problème par un système de production (donner la description des états, état initial, état but, les règles).

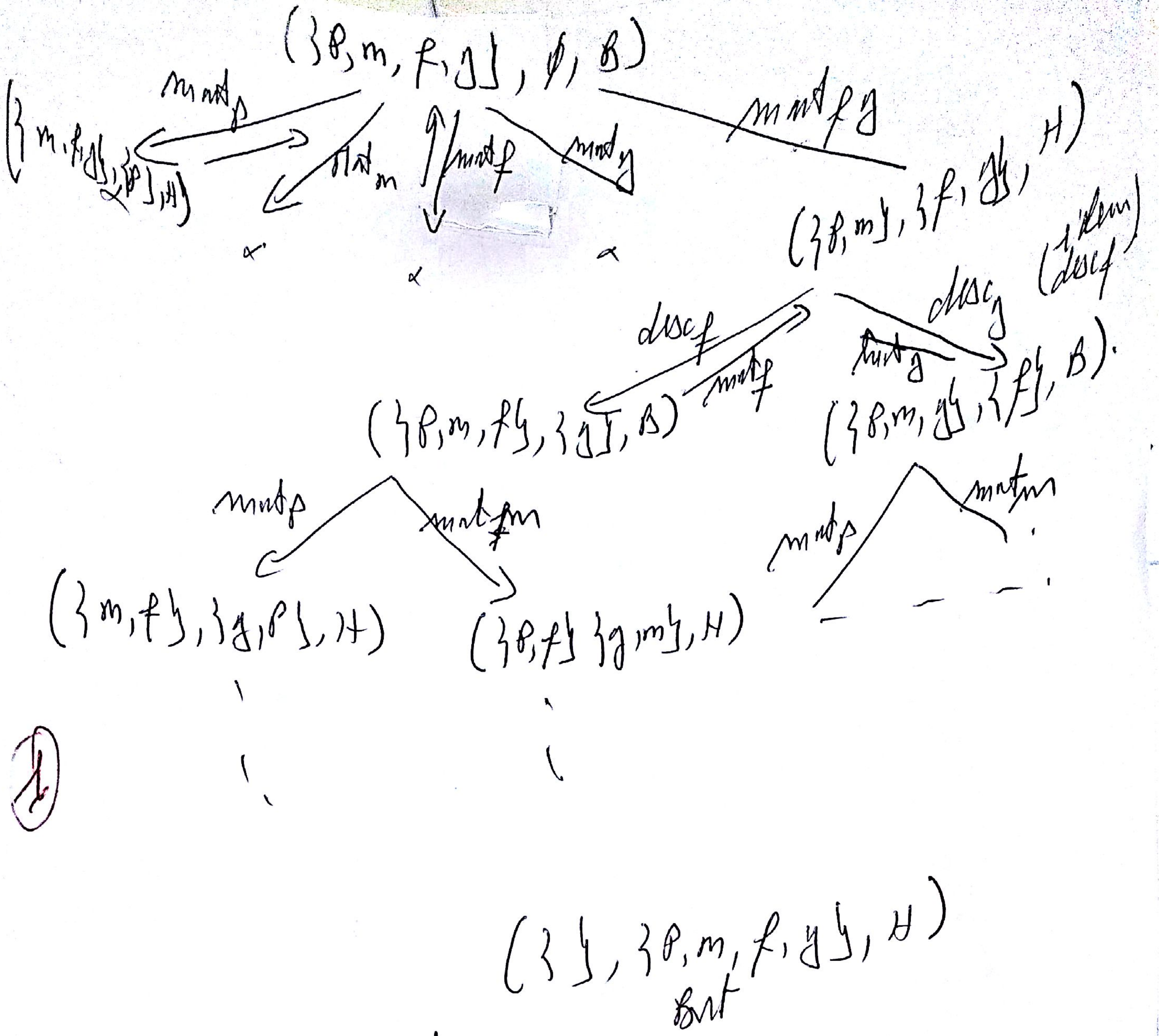
2-Donner l'espace de recherche avec la stratégie en largeur d'abord, déduire la séquence de règle pour atteindre le but.

Exercice 5 :(3Pts) Ali est un ingénieur en robotique et il a conçu un robot qui se déplace pour délivrer le courrier. Il a implémenté l'algorithme A* pour la recherche de chemin (en utilisant la distance en ligne droite comme fonction heuristique), toutefois l'algorithme ne semble pas fonctionner correctement, car le robot choisit souvent un chemin qui n'est pas optimal. Voici l'algorithme que le robot utilise (où D est le nœud de départ, F une file et l'heuristique).

F = {*D*}
Tant que *F* n'est pas vide
    Faire Prendre *F1*, le premier élément dans *F*
        Noeuds_enfants = développer(*F1*) /* tous les fils de F1 */
        Éliminer les noeuds dans *Noeuds_enfants* qui ont déjà été visités
        Pour tous les noeuds restant dans *Noeuds_enfants*,
          faire
              Si l'enfant est un état but ***Alors*** Retourner Succès et Sortir
        Fin pour tous
        Ajouter les *nœuds_enfants* à *F*
        Trier *F* selon la fonction suivante : *f = coûtChemin(D à noeud) + h(noeud)*
Fin tant que

a. Cet algorithme ne donne pas toujours la solution optimale, car il contient une erreur. Identifiez cette erreur et expliquez pourquoi la solution retournée n'est pas toujours optimale.

b. Corrigez alors l'erreur et écrivez le bon algorithme pour A*.



la sequence est :

mont$_{fg}$ / desc$_{enf}$ / mont$_{parent}$ / ~~desc~~ $_{enf}$ / mont$_{fg}$ / desc$_{enf}$ / mont$_{parent}$ / desc$_{enf}$ / mont$_{fg}$

enf → f ou g parent → p ou m.

- enum : - si l'état initial est But ⇒ détecté

(1.5) - sortie de l'alg avant le tri.

(1.5) - Alg du cours. (trier après l'ayant)

QCM ① F ② V ③ a ④ b ⑤ a ⑥ a   +0,5 / −0,5

exo 1/ a) Sub Cons (0,5)   b) mif Cons (0,5)   c) Non cible complet (1,5)   d) si l'ens des Vm = ∅ (1,5)

exo 2 a) $(\forall x)(\forall y)$ $(x \geq y \Leftrightarrow \max(x,y) = x) \vee$ $x < y \Leftrightarrow y = \max(x,y))$ (1,5)

b) $(\forall x)$ $(\text{impair}(x) \vee x = 2) \Leftrightarrow \text{premier}(x))$ (1,5)

c) $(\forall x)$ $((a(x) \wedge b(x)) \vee c(x)) \Leftrightarrow p(x))$ (1,5)

exo 3/ a) Cons (1) (0,5)

b) 2 clauses $\neg P(x) \vee q(x) \vee \neg R(z) \vee \neg S(z)$ (3)

$\neg P(x) \vee M(f(x), x) \vee \neg R(z) \vee \neg S(z)$

exo 4 état (l_B, l_H, P_m) avec l_B liste en Bas p, m, f, g

l_H — Haut

P_m H, B

(1) état initial $(\{p, m, f, g\}, \emptyset, B)$

" final $(\emptyset, \{p, m, f, g\}, H)$

actions

$desc_f$ si $f \in l_H$ et $P_m = H$ alors $l_H = l_H - f$ et $l_B = l_B + f$ et $P_m = B$

$desc_g$ si g — g — g

$mont_f$ si $f \in l_B$ et $P_m = B$ alors $l_B = l_B - f$ et $l_H = l_H + f$ et $P_m = H$

$mont_g$ — g — g — g

$desc_{fg}$ si f et $g \in l_H$ et $P_m = H$ alors $l_H = l_H - f - g$ et $l_B = l_B + f + g$ et $P_m = B$

$mont_{fg}$ si — l_B — B alors $l_B = l_B - f - g$ et $l_H = l_H + f + g$ et $P_m = H$

(1) $desc_p$, $mont_p$, $desc_m$, $mont_m$ idem que $desc_f$ et $mont_f$

règles $desc_X$ si $X \in l_H$ et $P_m = H$ alors $l_H = l_H - X$, $l_B = l_B + X$ et $P_m = B$

$mont_X$ — l_B — B — $l_B = l_B - X$ $l_H = l_H + X$ et $P_m = H$

avec X = f | g | f,g | p, m