Département d'Informatique --- Faculté des Sciences --- UFAS

PROGRAMMATION LOGIQUE

CONTROLE ---- Janvier 2013

Durée 1h30 documents non autorisés

Exercice 01 (03 points)

Définir un prédicat **somme(L, S)** qui est vrai si **S** est la somme des éléments de la liste **L**.

Exemple **?-somme([2,4,7,5], S). S = 18.**

Exercice 02 (04 points)

On désire traduire des phrases du Français vers l'Anglais et vice versa On dispose de la base de connaissances suivante (qui peut être agrandie à loisir) :

change(chat, cat). change(toit, roof). change(le, the).

change(est, is). change(sur, on).

Définir une clause **traduire(L_1, L_2)** qui traduit la phrase **L_1** en la phrase **L_2**.

Exemple **?- traduire([le, chat, est, sur, le, toit], L). L = the cat is on the roof.**

N.B : **L_1** est donnée sous forme de liste, **L_2** sera donnée sous forme textuelle.

Exercice 03 (04 points)

- Définir une clause **inserer(X, L_1, L_2)** qui insère l'élément **X** dans la liste ordonnée d'entiers **L_1** pour avoir une liste ordonnée **L_2**.

Exemple **?- inserer(5, [2,4,17,24,88], L). L = [2, 4, 5, 17, 24, 88].**

- En déduire une algorithme de tri : **tri(L_1,L_2)** qui utilise cette clause.

Exemple **?- tri([14, 5, 8, 2, 6, 1], L). L = [1, 2, 5, 6, 8, 14].**

Exercice 04 (05 points)

Définir une clause **supprimer(N, L_1, L_2, X)** qui supprime l'élément **X** d'ordre **N** de la liste **L_1**. Le résultat est la liste **L_2** amputée de l'élément **X**.

Exemple **?-supprimer(3, [a, b, c, d, e, f], L_2, X). X = c, L_2 = [a, b, d, e, f]**

Exercice 05 (04 points)

La fonction d'Ackermann est définie récursivement comme suit :

$$A(m,n) = \begin{cases} n+1 & si\ m = 0 \\ A(m-1,1) & si\ m > 0\ et\ n = 0 \\ A(m-1, A(m,n-1)) & si\ m > 0\ et\ n > 0 \end{cases}$$

Définir une clause qui évalue cette fonction.