**Travaux Dirigés Intelligence Artificielle (1h30)**

**Environnement common Lisp**

Dans une fenêtre Unix, taper clisp.

EXERCICE 1.

Evaluer les expressions suivantes. Noter les messages d’erreur et ce qui se passe lorsqu’il manque une parenthèse ou lorsqu’il y en a trop. Pour bien comprendre un résultat, évaluer l’expression la plus interne jusqu’à l’expression globale.

Ex : Pour (car (cdr ' (a b c) ) ), évaluer (cdr ' (a b c) ) = (b c) puis (car ' (b c) ) = b

Après une erreur, l’interpréteur affiche un message d’erreur et se met en mode debug qui permet éventuellement de corriger l’erreur. Taper **:q** pour sortir de ce mode.

(car ' (a b c) )

(cdr '(a b c))

(cdr '(a))

(car (a b c))

(car '((+ 2 3) (+ 6 7) ) )

(car ((+ 2 3) (+ 6 7) ) )

(car (list (+ 2 3) (+ 6 7) ) )

(cadar (cdr '((a b) (a (b c)) d) ) )

(caadr '((a b) (a (b c)) d) )

(cons '((a b) c) ' (a (b c)) )

(append '(h j) () ' ( o p))

(append ' ((a b)(c d)) ' (e f))

(atom (cadr '(a(b) c (d))))

(listp (car '((a b c) d e) ))

(null 'a)

(equal '(a (b c)) '(a (b) c))

(member 'a '((a) b) )

(member 'a 'a)

(member 'a '(1 a 2 b))

(+ 2 3)) ;Que provoque la présence d’une parenthèse fermante supplémentaire ?

(car '(a b) ;Que provoque le manque de parenthèse fermante ?

(nth 3 ' (1 2 3 4 5)) ; que fait la fonction nth?

(nthcdr 3 ' (1 2 3 4 5)) ; que fait la fonction nthcdr ?

EXERCICE 2.

Écrire les formes LISP qui renvoient l'atome **A** à partir de (c e (d a) f), puis de ((x y) ((a) b))

EXERCICE 3.

Remplir le tableau suivant

arg1 arg2 append list cons

(a b) (c d)

a (c d)

(a b) c

((a b)(c d)) (e f)

(a b) ()

() (a b)

a ()

**Fonction *load***

Ecrire les fonctions utilisateur dans un fichier texte, puis charger le fichier dans l’environnement lisp à l’aide de la fonction ***load***.

La fonction ***load*** permet de charger un fichier texte avec

(load "cheminAPartirDeVotreHome/nomDuFichier")

Ex : (load "IA/td1.txt")

**Fonction *trace***

La fonction ***trace*** permet de tracer une fonction utilisateur en affichant tous les appels récursifs de cette fonction. On écrit :

(trace nomDeLaFonction)

Ex : avec la fonction *rem* définie par

(defun rem(x l)

(cond ((null l) nil)

((eq (car l) x) (rem x (cdr l)) )

(T (cons (car l) (rem x (cdr l)))

) )

(trace rem)

(rem 'b ' (a b c b d) )

Pour enlever la trace de la fonction ***rem***, écrire (untrace rem) ou (untrace).

(untrace) supprime toutes les traces actives.

EXERCICE 4

Ecrire la fonction ***alt*** (L) qui retourne une liste dont les éléments sont les éléments de rang impair de la liste L.

? (ALT '( A B C D E))

= (A C E)

? (ALT NIL)

= NIL

? (ALT '(A))

= (A)

EXERCICE 5.

a) Ecrire la fonction ***aLaFin*** (x l) qui renvoie la liste obtenue en ajoutant ***x*** à la fin de la liste ***l***.

? (aLaFin 'D '( A B C)) = (A B C D)

Ecrire à l’aide de ***aLaFin,*** la fonction la fonction ***inverse*** (l) qui inverse les éléménts d'une liste au premier niveau.

? (INVERSE '( A B C D E))

= (E D C B A)

EXERCICE 6.

Écrire une version récursive de la fonction ***somme***(l) qui calcule la somme de tous les nombres d’une liste plate.

 ? (*somme* ' (1 a b 2 4 9) )

= 16

EXERCICE 7.

Écrire une fonction ***remplace*** (x y l) qui remplace toutes les occurrences de ***x*** par ***y*** dans la liste l.

? (*remplace* 'g ' f ' (z g l g m)) = (z f l f m)