

CIS681 – Artificial Intelligence

Some Predicates and Other Functions

Lisp comes with a number of predicates – functions for testing what kind of s-expression you have. These are used a lot in conjunction with flow-of-control functions. Notice, false=nil, true=anything else.

(null x) – returns t if the value of x is nil, nil otherwise.

(atom x) – returns t if the value of x is an atom (a number or identifier or nil).

(numberp x) – returns t if the value of x is a number.

(listp x) – returns t if the value of x is a list.

(equal x y) – returns t if the value and x and the value of y are the same atoms, or lists made up of the same elements in the same order (i.e., lists that look the same).

(= -numbers-), (< -numbers-), (> -numbers-), (<= -numbers-), (>= -numbers-) – each of these predicates take any number of numbers and returns t iff all the arguments comply with the specified predicate. E.g., \= returns t iff all its arguments are different.

(+ -numbers-) – returns the sum of the arguments. If no arguments are given, 0 is returned.

(1+ number) – returns number plus 1.

(- -numbers-) – returns the result of subtracting from the first argument all the subsequent arguments. If one argument is given, the result of subtracting that argument from 0 is returned.

(1- number) – returns the number minus 1.

(* -numbers-) – returns the product of all the arguments. 1 is returned if no arguments are given.

(/ -numbers-) – returns the result of dividing the first argument by succeeding ones. If there is only one argument, its reciprocal is returned.