Alec Guilin

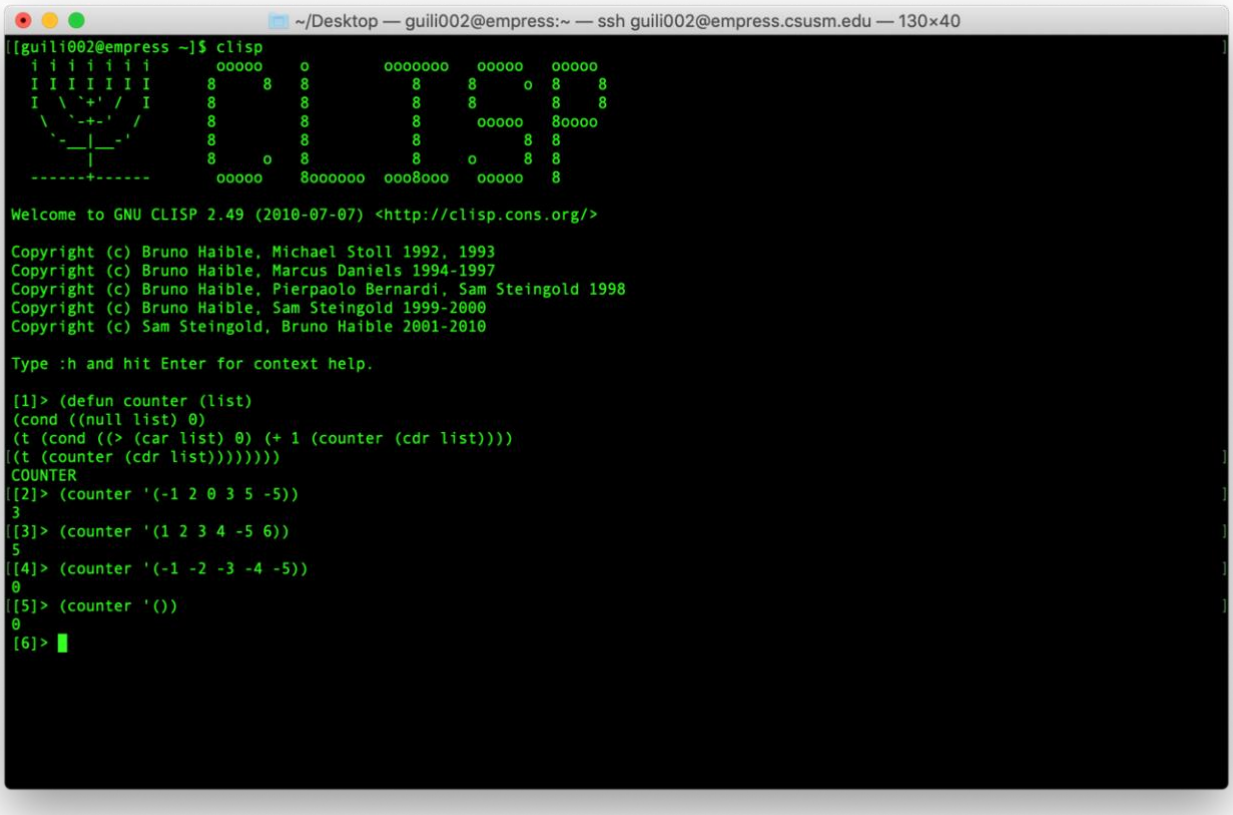
Professor Guillén

CS 351

19 April 2019

HW #5 and #6

1) (defun counter (list)
   (cond ((null list) 0)
   (t (cond ((> (car list) 0) (+ 1 (counter (cdr list))))
   (t (counter (cdr list)))))))

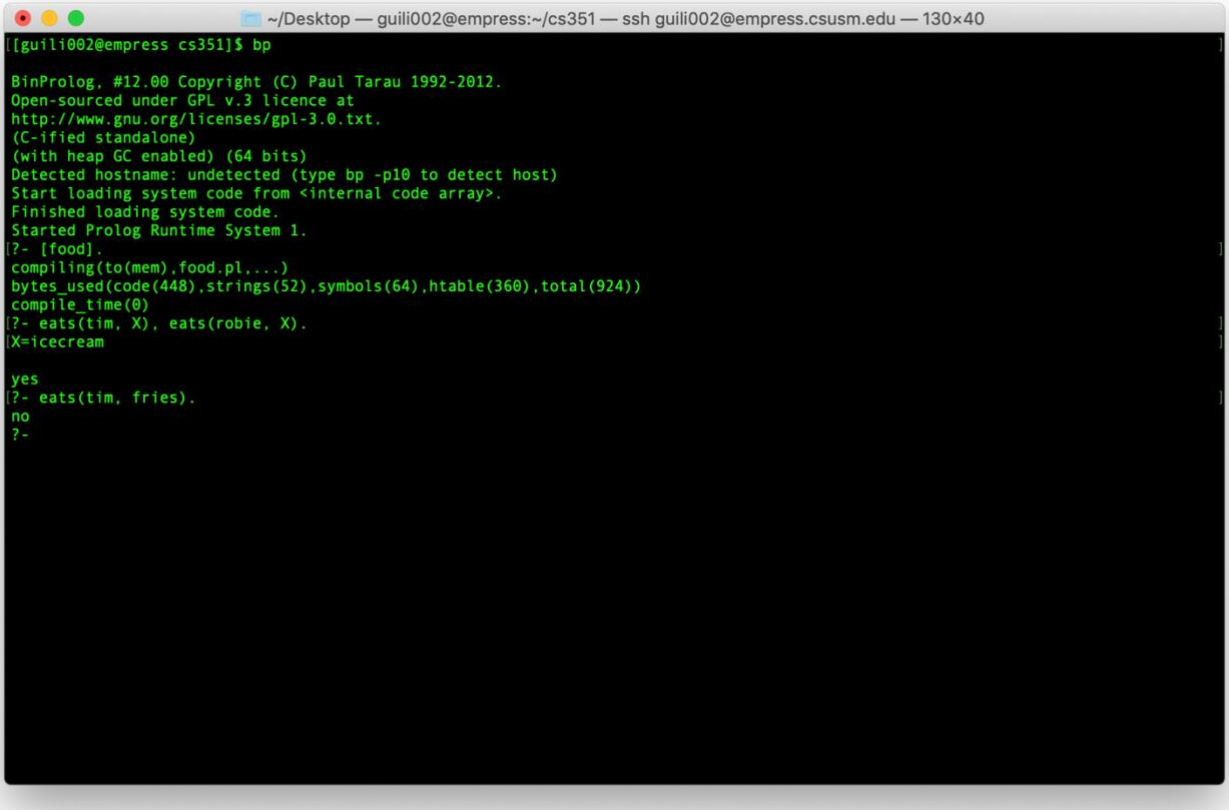


2) (a b c) (c d)
(defun union(seta setb)
(cond ((null seta) setb
((member (car seta) setb) (union (cdr seta) setb))
(t (cons (car seta) (union (cdr seta) setb))))

Manual trace: call 1: (Union (a b c) (c d))
call 2: (Union (b c) (c d))
call 3: (Union (c) (c d))
call 4: (Union () (c d))

call 4 returns (c d)
call 3 returns (c d)
call 2 returns (b c d)
call 1 returns (a b c d)

3) eats(robie, icecream).
eats(robie, burgers).
eats(robie, fries).
eats(tim, veggies).
eats(tim, juice).
eats(tim, icecream).



4) Who are cousins?

parent(min,gre).
parent(per,gre).
parent(tag,bound).
parent(upper,bound).
siblings(min,upper).
cousin(X,Y) :- parent(U,X), parent(V,Y), siblings(U,V).

(1) 1 Call: cousin(_0, _1)?
(1) 1 Call: parent(_2, _0)?
(1) 1 Exit: parent(min, gre)
(2) 1 Call: parent(_2, _1)?
(2) 1 Exit: parent(min, gre)
(3) 1 Call: siblings(min, min)?
(3) 1 Fail: siblings(min, min)
(2) 1 Redo: parent(_2, _1)?
(2) 1 Exit: parent(per, gre)
(4) 1 Call: siblings(min, per)?
(4) 1 Fail: siblings(min, per)
(3) 1 Redo: parent(_2, _1)?
(3) 1 Exit: parent(tag, bound)
(5) 1 Call: siblings(min, tag)?
(5) 1 Fail: siblings(min, tag)
(4) 1 Redo: parent(_2, _1)?
(4) 1 Exit: parent(upper, bound)
(6) 1 Call: siblings(min, upper)?
(6) 1 Exit: siblings(min, upper)
(1) 1 Exit: cousin(gre, bound)