

Zevi Rubin,
HW0 Written work

1. Rewrite the following s-expression so that it uses only binary arithmetic operations. You do not need to draw a tree, just write the s-expression. Show each step of your work:

`(* 1 2 3 (+ 4 5 6))`

my logic was that because `list (1 2 3) == list (1 (2 3))`, then `(+ 4 5 6)` must be the same as `(+ 4 (+ 5 6))`.

Based on that logic I got: `(* 1(* 2 (* 3 (+ 4(+ 5 6)))))`

2. Explain in your own words (2-3 sentences max) why the expression `(first (list 1 2))` evaluates to a number, but `(rest (list 1 2))` evaluates to a list.

The First evaluator fetches the first value of a list regardless of length, thus it being a single number. The Rest evaluator returns all the values of the list BUT the first, regardless of length, so therefore it must be prepared to return multiple numbers, thus it being a list.

3. Plai-typed has a procedure called `second` which returns the second element in a list.

a. Write an equivalent procedure using `first` and `rest`. Explain your work.

b. Consider the following s-expression:

`((1 (2 3)) (4 5) 6)`

What is the second element of this s-expression list? Explain.

(Note: Actually running this procedure requires a lot of type-casting. Just reason it out.)

a: given some list `l`;
`(define (findSecond (first (rest l)))) : (number)`

please ask for clarification if the above code doesn't seem to make sense to you. I might have written it unclearly.

b: in the state that is written, it looks like 3 is the second value a computer would find, but I think that 2 is in fact the second digit. This is because that whole expression is equal to `((123)(45)6)`, which I think is equal to `((12345)6)`? Therefore (45)?

4: it is Crucial that you use order of operations, specifically the parenthesis order when doing s-expression calculations. This is because the s-expression:

$(+ (*3 4) (* 2 3))$ (<- correct) does not equal $(* 3 (+ 4 (* 2 3)))$ (<-incorrect).

Basically, as long as there are the correct number of parenthesis (all are closed), then the short answer is yes.