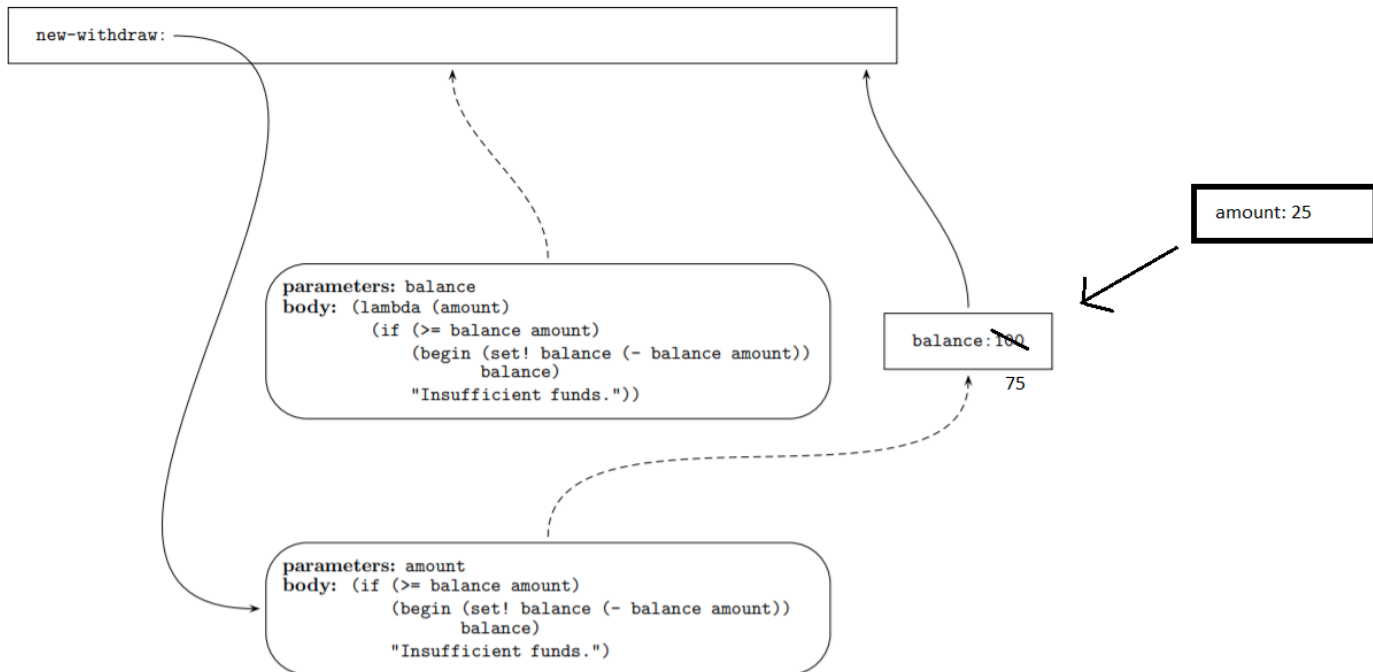
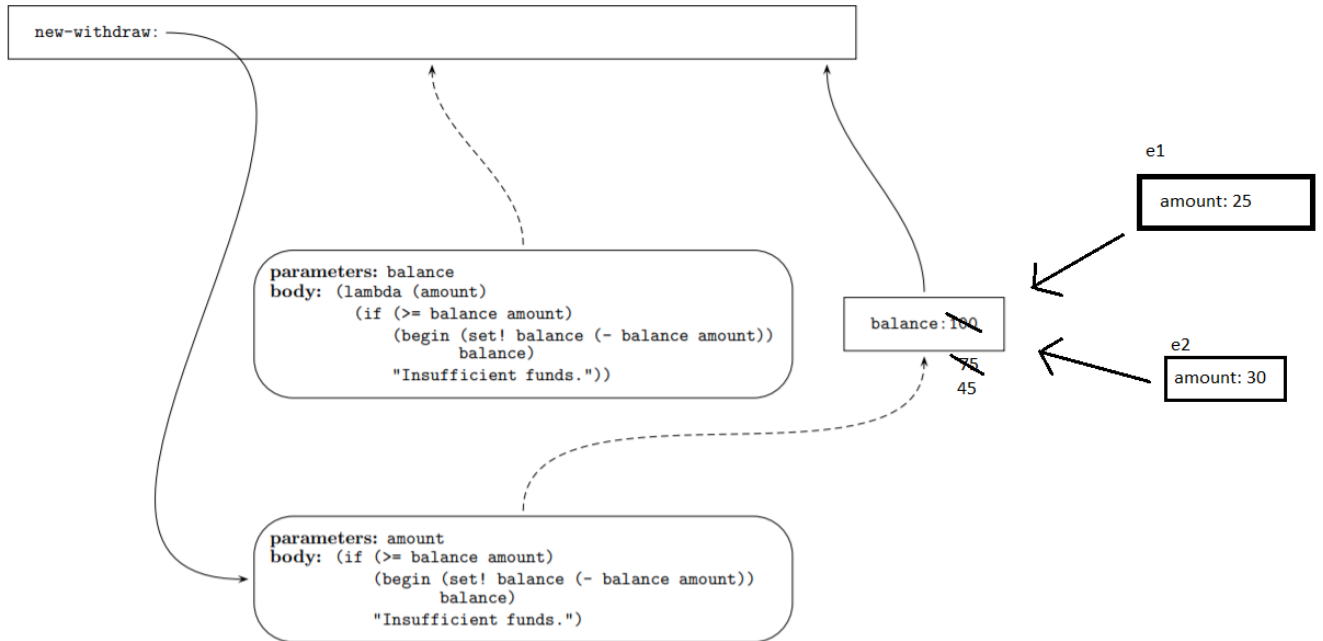


Jesse Huynh
CS 450
Homework 4

Question 2.
New-withdraw 25

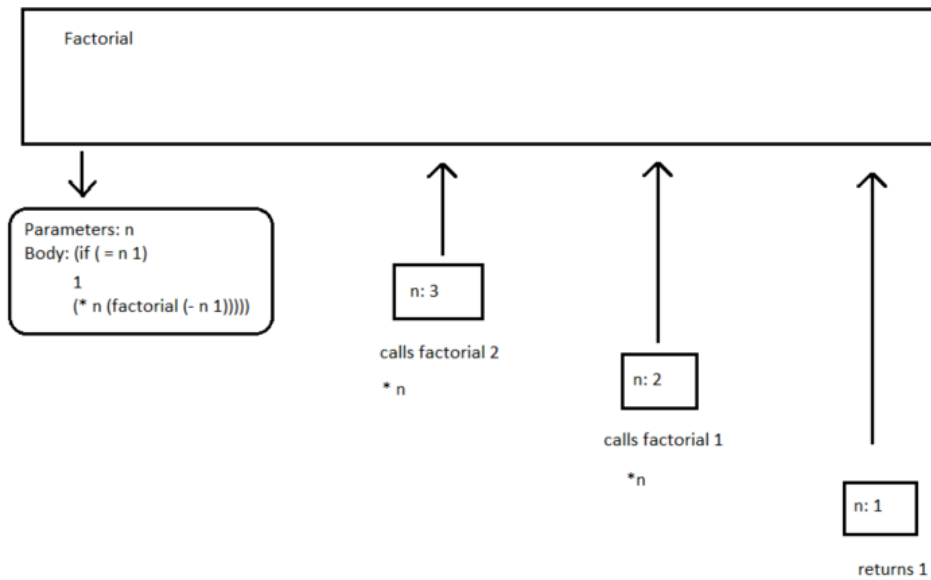


New-withdraw 30



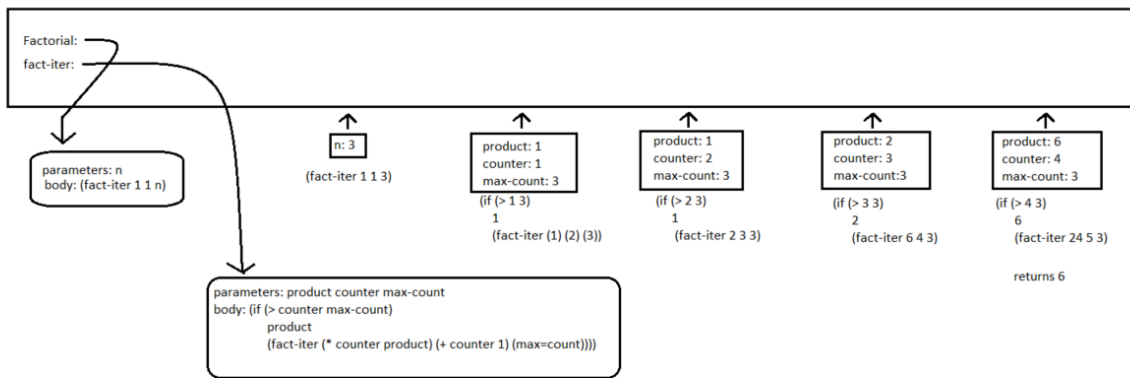
Question 5

- a. I was not sure how to do this part, as in the example, sqrt has functions defined within but in the recursive factorial, there were not. In the sqrt example, there is an instance with x2 and all the calls point back to it, however I was not sure how to do that with just 3 and no function calls in factorial so I did it differently.



b. Iterative

I once again tried to do it like sqrt, but realized that it did not have fact-iter under the definition of factorial, but as a helper function. I then proceeded to do this problem similarly to I did for a, with all the arrows pointing back towards the main environment.



Part 2

Task1

This part took me a lot longer than I think it should have. I spent a lot of time just understanding the math and going over the java example. Once I understood the math, I tried turning the java example into scheme however I got stuck trying to implement the for loops.

I then went to a different solution, which involves making a copy of the original list and using that to iterate through each possible value of k from which to split my lists left and right. I was having trouble with infinite looping as well as not looping at all, which I fixed by adjusting my loop function and also by making sure my code was correct (I kept calling min-cost-naïve on the original list instead of my iterator list which caused me a lot of headache until I saw the issue).

By the time I finished this problem, the due date was already passed so I was unable to implement DP and the function to print the tree.

Question 2.

I did not complete the DP function but below is the output from my naïve function.

```
> (timed min-cost-naive data)
(time: 0)
142
>

> (timed min-cost-naive keys-medium)
(time: 46)
2.84
> (timed min-cost-naive keys-large)
(time: 8931)
4.5600000000000005
>
```

Question 3.

Did not complete

Question 4.

