Week 2

子论坛
所有
Assignment: Homework 5
Assignment: Homework 5 (Auto-Grader)

← Week 2

Need help with "mupl-map"

Nikita Luparev Week 2 · 4 years ago

Hello.

I'm working on *mupl-map* and have hard time to implement it. As always I try to build intuition on how function works and this helps me to implement it correctly, but here I have hard times to build those intuition.

First, I assume that variable *mupl-map* will be bound to closure. So in order to evaluate MUPL fun to closure I need to either use *eval-exp* or *eval-under-env.* At this point everything is fine, problem arises when I try to use previously evaluated closure in order to make recursive call and here I'm really stuck.

2nd function that takes list of MUPL values which I need to map using previously given function also need to be evaluated to closure, but I don't understand how make environment for it where environment will contain binding to closure that points to *mupl-map* ?

I hope this gives you enough information about my thought process on this problem and you can point me where I wrong.

Thank you in advance, Nikita.

CS分钟 **Charilaos Skiadas** Mentor · 4 years ago

I think the problem is that you are thinking of evaluating. You shouldn't. Your job in both problems is to write a "MUPL program" that would do the right thing if and when it is evaluated. Imagine writing the map function in SML for example. Then translate that to MUPL terms (more or less).

1 赞 隐藏 4 回复

Nikita Luparev · 4 years ago

OK. But in order to map one particular element of a given list onto its new value I need to use given function. So, in order to do this I need to construct MUPL *call* expression

```
1  (call (var "f") (fst (var "xs")))
```

where (var "f") should point to closure because it's required by MUPL *call-exp* definition. This is first point that isn't yet clear to me.

Second.

I need to be able to call this function recursively, so I need to have binding in an environment pointing to closure. So my current way of thinking assumes that I need to manage environment in some way or another, maybe I completely wrong ?

0 个赞

CS分钟 **Charilaos Skiadas** Mentor · 4 years ago

Well for the first point, mupl-map needs to be a MUPL function which is curried, meaning it takes one argument (probably called "f") and its body is itself another function that takes as argument the list. So when mupl-map is to be called, the first argument provided to it is meant to be the function closure. So when later in the body you evaluate the expression you wrote above, the (var "f") expression will make the system look for a binding in the current environment, and hopefully find a function closure. It is up to the future callers of mupl-map to provide a function closure, and it is up to

your interpreter to ensure that when the body is evaluated there is a binding for "f", since that was the formal to your mupl-map function. Hope that makes sense.

As for the recursive calls, you have two function in the outer layer of this complex thing, one is mupl-map itself, the other is its body which is itself a function in order to do the currying. You can use their nameopt spots to give them names for the purposes of the recursive call. But you'll have to figure out which of the two you need to name.

6 个赞

Nikita Luparev · 4 years ago

thank you, it's now makes more sense

0 个赞

OG **Odin Moron Garcia** · 3 years ago

This is difficult, thanks for your advices!

回复 个赞

回复

回复

回复