Q&A Session for Programming Languages Lecture 13

Session Number: 1209223705
Date: 2020-10-20
Starting time: 14:27

ANON - 14:24
Q: From Lecture 12, what is meant by the "visible state"? Why was it underlined on a slide? This was about side effects.
Priority: N/A

Steven Haussmann - 14:33
A: State is "visible" if it can affect our computations. Non-visible state would be something that exists, but has no effect on us.

ANON - 14:26
Q: For today's lecture 13 content, when Professor Milanova talks about slide 14, what did she call "fun"? Did she say null array function?
Priority: N/A

Ana L. Milanova - 14:40
A: Did you mean Lec 12?

ANON - 14:26
Q: When Professor Milanova talks about the pair predicate, what's the spelling of the term she refers to that has two pointers? "Cons sells"?
Priority: N/A

Steven Haussmann - 14:34
A: Cons cells

ANON - 14:32
Q: Are exam grades expected to be released today?
Priority: N/A

ANON - 14:33
Q: Iirc, because the test got pushed back hw4 was going to be pushed back. is that still happening?
Priority: N/A

Ana L. Milanova - 14:41
A: Yep, it did get pushed back too.

ANON - 14:36
Q: just making sure, append is not allowed for the hw right
Priority: N/A

Ana L. Milanova - 14:42
A: It is not on the white list, if I remember correctly. Cons should be sufficient for what we need in the homework.

ANON - 14:37
Q: what's the difference between calling a function with ? or without.
Priority: N/A
Steven Haussmann - 14:41
A: The ? is just part of the name. It does not have any specific meaning; it's a convention to make it obvious that the function checks for something and returns true or false.
Ana L. Milanova - 14:43
A: Yes, what Steven says. fun? denotes a predicate, a function that returns true or false.

ANON - 14:37
Q: Is pair a builtin function?
Priority: N/A
Ana L. Milanova - 14:41
A: Yes, it is.

ANON - 14:38
Q: Can we use list function in assignment
Priority: N/A
Ana L. Milanova - 14:41
A: No, it is not on the white list, if I remember correctly. If you use other functions we may take points off.

ANON - 14:38
Q: what is tail optimization and why does it allow the stack to not run out of space?
Priority: N/A
Steven Haussmann - 14:39
A: If a function's last step is to return another function, then the function is effectively done -- so we can discard its stack frame. This means that the stack doesn't grow during recursive calls.

ANON - 14:39
Q: How do you write contract when the function has multiple parameters? Additionally, how do you show that a list uses data (quote) in the contract?
Priority: N/A
Steven Haussmann - 14:40
A: Quoting is only necessary when you want Racket to treat something as piece of data, rather than trying to evaluate and execute

it. You should not have to mention quoting at all in your contracts.
Steven Haussmann - 14:40
A: Also, I believe the syntax is arg1 * arg2 * ... * argn
Ana L. Milanova - 14:44
A: Any syntax that you adopt for multiple arguments will be fine. arg1 * arg2 * arg3 is one way, but you can use , if you want. As long as it is clear, it should be ok.

ANON - 14:40
Q: when will lectures 10-12 be uploaded to mediasite?
Priority: N/A
Steven Haussmann - 14:41
A: I believe they're appearing at the start of the list, because it's sorting their names lexicographically. So you'll see 1, 10, 11, 12, 2, 3, ...

ANON - 14:40
Q: I only see up to lecture 9
Priority: N/A
Konstantin Kuzmin - 14:43
A: Make sure you check all pages or select some convenient sorting.

ANON - 14:41
Q: Oh great, my audio for the streaming is finally working. Before I only got audio when the hosts are speaking.
Priority: N/A
Steven Haussmann - 14:42
A: For anyone else having trouble with this, check what your default communication device is! Webex will send audio from streamed videos to that, no matter what you have configured in its settings.

ANON - 14:43
Q: In our homework, the interpreter takes in a list of lists. In the contract, would we say the parameter is (list list) or something else?
Priority: N/A
Steven Haussmann - 14:43
A: I would write a list of lists of numbers as (list (list number))

ANON - 14:43
Q: Ana L. Milanova: Did you mean Lec 12?
A: No. I watched Lec 13 a little bit earlier. I actually mean lecture 13, today's lecture.
Priority: N/A

Ana L. Milanova - 14:56
A: Is it slide 17 on Higher-order functions?

ANON - 14:43
Q: I don't really understand what we have to write for "Contract" part of the comment even after this slide. We write the types it takes in and then the output type?
Priority: N/A
Steven Haussmann - 14:44
A: Yes, you're basically writing the type signature of the function. If it takes an integer and a list of integers and produces a list of integers, then its contract is integer * (list integer) -> (list integer)

ANON - 14:45
Q: Follow up on the (list (list a)) question, what if the lists are arbitrarily nested, as in the homework?
Priority: N/A
Steven Haussmann - 14:46
A: I believe we're just saying there's a type called something like "boolprogram" that represents a single boolean program, but I'll defer to Milanova on the specific name
Ana L Milanova - ???
A: Yes, what Steven says. We will assume that there is a type called "Boolprogram" or "Expr", or "Boolexpr", which happens to be a recursive type, that is, defined in terms of components that are of this type. Assume we can have this type and use it in our signatures.

ANON - 14:46
Q: Hey Steven. I didn't change any settings. Just joined the event and realized it's working. Not sure what was the issue.
Priority: N/A

ANON - 14:46
Q: Does equal? work for both atoms and lists
Priority: N/A
Ana L. Milanova - 14:48
A: Yes.

ANON - 14:46
Q: What is an unchecked type check? I didn't really understand that part
Priority: N/A
Steven Haussmann - 14:47

A: We're writing a type signature in the comments above a function. However, this is purely for our sake; Racket does not pay attention to it. Hence being "unchecked"

Ana L. Milanova -

A: Exactly what Steven says. This is in contrast to Java for example, where Java checks the type signature --- if you have declared a parameter to be of class Integer but they try to use it as a String, Java will complain.

ANON - 14:47
Q: unchecked type signature ^ sorry
Priority: N/A

ANON - 14:48
Q: Is equal? implemented using atom?
Priority: N/A

Ana L. Milanova - 14:49
A: I suppose it would be using pair?.

ANON - 14:49
Q: The reason we do not have explicit cases for (atom? x) and (not (atom? y)) and the other similar case because those would just return false?
Priority: N/A

Ana L. Milanova - 14:51
A: Yes, that is exactly it.

ANON - 14:49
Q: what's the difference between our eql? and the built-in equal?
Priority: N/A

Ana L. Milanova - 14:51
A: Eql? is designed to behave as equal?, yes.

ANON - 14:49
Q: Why is the output of (atomcount2 '(1 '() 2)) 4?
Priority: N/A

Steven Haussmann - 14:53
A: You have an extra quote in there. I believe this results in the second element being a list of 'quote and '() -- it's a bit weird!

Steven Haussmann - 14:54
A: (caadr '(1 '() 2)) -- car of car of cdr -- is 'quote

Steven Haussmann - 14:54
A: Unless you actually want a literal quote symbol in your list, you don't need to quote anywhere but at the outermost level.

```
ANON - 14:52
Q: How is apply different from foldl and foldr?
Priority: N/A
        Ana L. Milanova - 14:53
        A: apply is essentially the built-in foldl.
```

```
ANON - 14:53
Q: what common imperative langauges use the value model and what
commons imperative langauges use the reference model?
Priority: N/A
        Ana L. Milanova - 14:55
        A: We give some examples on the following slides. Normally we
have a discussion in class.
```

```
ANON - 14:54
Q: Can someone give a quick definition of deep and shallow recursion
Priority: N/A
        Steven Haussmann - 14:55
        A: Shallow recursion won't delve into sublists.
```

```
ANON - 14:54
Q: Will the tutorial of Dr. Racket from the beginning be posted to
mediasite?  Will it be in the Q&A or something else?
Priority: N/A
        Konstantin Kuzmin - 14:54
        A: It will be uploaded to Mediasite as Lec13 Q&A.
```

```
ANON - 14:59
Q: does a cons cell contain/refer to the head and tail of the list?
Priority: N/A
        Ana L. Milanova - 15:01
        A: Yes. The cons cell is made up of two pointers, the left
pointer points (i.e., refers) to the car (head) of the list, and the
right pointer points to the cdr (tail) of the list.
```

```
ANON - 15:01
Q: Ana L. Milanova: Is it Slide 17 on Higher-order functions? Whoops.
You're right, I mean't slide 17.
Priority: N/A
        Ana L. Milanova - 15:02
        A: Ok, now it makes sense! "fun" is the nullary function that
returns the +1 function. Fun is a higher-order function because it
returns a function value as a result.
```

ANON - 15:02
Q: The value model seems easier / less confusing to understand, but I get how the reference model would save space. How do people who create programming langauges decide this?
Priority: N/A
Ana L. Milanova - 15:14
A: There are many factors that go into the decision. More "static" languages would use the value model and more dynamic languages would use the reference model. With languages that are more "static", meaning that things like type and size of a variable is known at compile time, we can plan and create the appropriate size location on the stack or on the heap. We can plan the layout of memory and optimize. With languages that are more dynamic the type of object a variable refers to is often not known until runtime, and it is more convenient to just create and have the object on the heap with a reference pointing to that object from the stack.

ANON - 15:07
Q: This is from part 2. Is atom? a built-in function? I've been trying to use atom? but keep getting an error. only when i define it, it works of course.
Priority: N/A
Ana L. Milanova - 15:09
A: No, it is a function we defined: (define (atom? x) (not (pair? x)))

ANON - 15:08
Q: If Python follows the reference model, why does ['a'] == ['a'] yield true?
Priority: N/A
Steven Haussmann - 15:09
A: Each python class has an __eq__ method on it, which is invoked when you use the equality operator. It handles the comparison.
Steven Haussmann - 15:10
A: If you compare strictly by the default hashcodes of objects (which are implementation defined; I think they're based on addresses in memory), then two things will only be equal if they're literally the same object
Ana L. Milanova - 15:17
A: To add to Steven's answer, in Python "is" is the reference equality, so if you do "['a'] is ['a']" that will return false. == runs the value equality, which is what we need most of the time in our programs.

ANON - 15:10
Q: does a higher-order function just mean it treats fucntions as a first-class value?
Priority: N/A
        Steven Haussmann - 15:11
        A: It means that the function accepts other functions. That doesn't actually require functions to be first-class values -- you just have to be able to pass them around. C lets you pass function pointers, but you can't create functions at runtime, for example

---

ANON - 15:12
Q: Following up on the higher-order value question, I thought C did not treat function as first-class, but it is still higher order?
Priority: N/A
        Steven Haussmann - 15:14
        A: C allows you to pass function pointers as arguments. However, you cannot define functions at runtime, or define them inside of other functions.

ANON - 15:13
Q: When I try to run the rev procedure defined on slide 33, I get an error saying "expected a procedure that can be applied to arguments" highlighting the list I gave it. (rev (1 2 3))
Priority: N/A
        Steven Haussmann - 15:14
        A: You need to quote the list of numbers. (rev `(1 2 3)) should work. It's trying to execute a function named "1" with arguments "2" and "3".

ANON - 15:11
Q: Does atom work the same way as symbol? ?
Priority: N/A
        Steven Haussmann - 15:13
        A: We defined it in the last lecture; it just checks if something is not a pair. (symbol? 3) is false; (atom? 3) is true

---

ANON - 15:15
Q: is map similar to looping through every element in a list, and then applying a function?
Priority: N/A
        Steven Haussmann - 15:15
        A: Yes, the mutation-based imperative equivalent would be something like (for i=0; i<list.len; i++) { list[i] = fun(list[i]) }

ANON - 15:24

Q: what exactly does apply do?
Priority: N/A

Ana L. Milanova - 15:24
A: In this case, "apply +", it just sums up the elements of the list.

Ana L. Milanova - 15:26
A: More generally, it is a higher order function that "folds" or "reduces" a list around an initial value using a binary operation.

ANON - 15:25
Q: how come the second argument in the last example on slide 22 doesn't have a quote?
Priority: N/A

Ana L. Milanova - 15:27
A: In the last example, there is just a single argument, that list with many nested lists.

ANON - 15:27
Q: All lists of length > 2 are considered pairs, correct?
Priority: N/A

Ana L. Milanova - 15:29
A: Yes, that is correct. A stronger statement will be that all lists of length > 0 are pairs.

ANON - 15:30
Q: What does apply do again?
Priority: N/A

Ana L. Milanova - 15:35
A: It "folds" the list around an initial value using a binary operation. E.g., (apply - 0 '(1 2 3)) is ((0-1)-2)-3) = -6. apply is the built-in "foldl".

ANON - 15:30
Q: what does apply append do
Priority: N/A

Ana L. Milanova - 15:38
A: (apply '() append '((a b) (c d))) is (append (append '() '(a b)) '(c d)). I have pictures later in lecture that will make this clearer.

ANON - 15:33
Q: what do you mean by fold?
Priority: N/A

Ana L. Milanova - 15:38
A: "folding" or "reducing" the list into a result.

ANON - 15:38

Q: does fold just mean taking nested lists into one lists?

Priority: N/A

Ana L. Milanova - 15:40

A: The case of folding using append, yes.

Ana L. Milanova - 15:41

A: But fold is a higher order function, we can give different binary operations to the fold.

ANON - 15:42

Q: How does foldr know that x is the List Element and y is the Partial Result? Or is that the Common Accepted Convention?

Priority: N/A

Ana L. Milanova - 15:45

A: This comes from the way foldr is defined: the operation op is applied to (car lis) as the first operand. So the first operand of "op" is the current list element. In our case we had a (lambda (x y) ...) and the first argument x corresponds to the element.

ANON - 15:45

Q: So foldr and foldl are kind of just syntactic shorthand for a shallow recursive function?

Priority: N/A

Ana L. Milanova - 15:47

A: Yes, this is a good way to think about it. The folds traverse the list along the length of the list.

ANON - 15:47

Q: I'm not sure why Apply is foldl and not foldr as it does not seem to pass an accumulated value argument rather append to return solution

Priority: N/A

Ana L. Milanova - 15:50

A: apply is foldl: e.g., (apply - '(1 2 3)) computes (1-2)-3 = -4. Unlike our foldl apply allows for the "id" value to be optional, and it just "folds" using the first element of the list. But we can give an explicit id: (apply - 0 '(1 2 3)) which yields -6.

ANON - 15:49

Q: So foldr and foldl are kind of just syntactic shorthand for a shallow recursive function? This was answered as true but I thought they work in both Shallow/Deep Applications.

Priority: N/A

Ana L. Milanova - 15:51

A: Yes, you can use it in deep recursion applications, by giving an operation that recurses into individual elements.

ANON - 15:52
Q: Why does (apply 0 '(1 2 3)) yield -6? Should this still not yield -4 if foldl?
Priority: N/A
        Ana L. Milanova - 15:53
        A: (apply - '(1 2 3)) folds around the first element: (1-2)-3. (apply - 0 '(1 2 3)) folds around the 0: ((0-1)-2)-3)

ANON - 15:59
Q: what is the purpose of the id for foldl on slide 36?
Priority: N/A
        Ana L. Milanova - 16:00
        A: This is the same "id" this initial value around which we are folding the list. We are just writing the contract (i.e., the type signature) of foldl.

ANON - 16:00
Q: does a binary operation just refer to an operator like + or *
Priority: N/A
        Ana L. Milanova - 16:00
        A: Yes, that's the most common example, and the one we most often we use in our programs.
        Ana L. Milanova - 16:02
        A: We call (foldl + 0 '(1 2 3)) and this means, we sum the elements around 0: ((0 + 1) + 2) + 3). This is what we mean by "folding" the list.

ANON - 16:02
Q: could you explain the difference between foldr and foldl conceputally? im still not quite getting it
Priority: N/A
        Ana L. Milanova - 16:03
        A: Ok! foldl folds from left-to-right: e.g., (foldl - 0 '(1 2 3)) that is ((0-1)-2)-3. foldr folds from right-to-left: e.g, (foldl - 0 '(1 2 3)) is (2-(3-0))
        Ana L. Milanova - 16:04
        A: Sorry, pressed enter too soon. (foldr - 0 '(1 2 3)) is 1-(2-(3-0)).

ANON - 16:03
Q: since foldl uses tail recursion and foldr doesn't, does that mean one is better than the other?
Priority: N/A

Ana L. Milanova - 16:04
A: Yes, tail recursion is usually better.

ANON - 16:06
Q: just to be clear, foldr and foldl should provide the same result even though the sentntial forms are different?
Priority: N/A
Ana L. Milanova - 16:07
A: If the binary operation is associative, e.g., append, or +, then they will compute the same results. (Assuming an id of 0 or '() respectively.) But if the binary operation is not associative, then they won't in general produce the same result.

ANON - 16:07
Q: would lenh be a more efficient implementation than len?
Priority: N/A
Ana L. Milanova - 16:09
A: Yes, that would be

ANON - 16:08
Q: after this ends can you go back briefly to the 2 len functions example. thanks
Priority: N/A
Ana L. Milanova - 16:31
A: Yes, we did! I hope we answered your questions.

ANON - 16:10
Q: so is the benefit of tail recursion just efficiency?
Priority: N/A
Ana L. Milanova - 16:33
A: Yes, this is the primary benefit.

ANON - 16:10
Q: Is tail recursion required on the homework?
Priority: N/A
Ana L. Milanova - 16:33
A: Not for HW4, it is not.

ANON - 16:10
Q: is there a built-in version of foldr?
Priority: N/A
Ana L. Milanova - 16:37
A: Not that I know of. But I'll check the r5rs standard.

ANON - 16:11
Q: When will exam 1 grades be released? I seem to remember you had said it would be today
Priority: N/A
        Ana L. Milanova - 16:38
        A: We have been working around the clock. We still estimate they will be released by the end of the day today.

ANON - 16:14
Q: Is tail recursion typically used as a tool to form more efficient functions?
Priority: N/A
        Ana L. Milanova - 16:41
        A: Yes, this is the primary benefit. Calls to tail recursive functions forgoe stack frames, which allows for long list inputs.

ANON - 16:15
Q: what about the out of the time
Priority: N/A

ANON - 16:15
Q: problem
Priority: N/A

ANON - 16:15
Q: for the exam
Priority: N/A

ANON - 16:17
Q: are we expected to write tail recursive scheme functions for exam 2?
Priority: N/A
        Ana L. Milanova - 16:42
        A: We will define the topics are when we get closer to the test.

ANON - 16:17
Q: who should I send email to?
Priority: N/A

ANON - 16:17
Q: dr. racket seems to be complaining about foldr being undefined, is

there anything special we have to do to use it?
Priority: N/A

Ana L. Milanova - 16:44

A: foldl and foldr are not part of r5rs. We use these functions to illustrate this important class of higher-order functions. You can add the definitions to your program and call them.

ANON - 16:18
Q: Are we expected to apply Lecutre 13 matieral (excluding commentys) to HW4?
Priority: N/A

Ana L. Milanova - 16:47

A: No, I wouldn't expect you would need these, except for mymap, our implementation of map. That one may prove useful.

ANON - 16:18
Q: i asked earlier, but is there a built-in version of foldr?
Priority: N/A

ANON - 16:19
Q: I'm still confused about the application of lambda in function definitions. Do you mind reexplaining that?
Priority: N/A

Ana L. Milanova - 16:49

A: Yes, (lambda (x y) (+ 1 y)) is just an anonymous function of 2 parameters, x and y. Very often when we use those higher-order functions, the are these "small" functions, and it is convenient to create an anonymous function value.

Ana L. Milanova - 16:50

A: We can expand on this in class next time.

ANON - 16:20
Q: I sent an email directly to you Professor and Professor Kuzmin. I got a reply from Professor Kuzmin and also spoke to Professor Milanova. Do I still need to send an email to the proglang or i'm good?
Priority: N/A

Ana L. Milanova - 16:53

A: That should be fine, if you received a response or spoke to us. There is no need to send more email.

ANON - 16:20
Q: Is there a built-in version of foldr?
Priority: N/A

ANON - 16:21
Q: Will the hidden tests for the homework always be hidden?
Priority: N/A

Ana L. Milanova - 16:51
A: For now we expect to leave those hidden but we might change our mind later on.