

## Q&A Session for Programming Languages Lecture 9

Session Number: 1208492367

Date: 2020-10-2

Starting time: 14:24

---

ANON - 14:25

Q: No quiz today, right?

Priority: N/A

Konstantin Kuzmin - 14:26

A: No, we were not planning a quiz today. Next one is on 10/6.

---

ANON - 14:34

Q: Syntax analysis = the parser stuff?

Priority: N/A

Ana L. Milanova - 14:35

A: The scanner and the parser, yes. Parser being way more important.

---

ANON - 14:30

Q: Thank you Professor Kuzmin and Professor Milanova for extending the homework deadline!

Priority: N/A

Konstantin Kuzmin - 14:39

A: You are welcome! It's our pleasure to ensure that students are learning the material without any unnecessary stress as much as possible.

---

ANON - 14:39

Q: How would I create a predicate that takes in two lists as arguments, but initializes the first two indexes of the second list to some values before iterating the function with the first list?

Priority: N/A

Ana L. Milanova - 14:41

A: Here we'll be answering questions on the current lecture, not on Prolog... At the end of lecture we might take some Prolog questions, and we'll devote extra office hours to Prolog and HW.

Ana L. Milanova - 16:24

A: I might be misinterpreting this question. Maybe like this:

Ana L. Milanova - 16:25

A: I'll need more detail to answer actually, you can ask on the forum or in OH?

---

ANON - 14:40

Q: Quiz and Exam are both next week right? Will we go over exam topics?

Priority: N/A

Ana L. Milanova - 14:42

A: Yes, I'll post the topics and practice tests later tonight (after we get out of Office Hours). We'll have a live review on Tuesday.

---

ANON - 14:42

Q: Semantic Analysis = interpreter stuff?

Priority: N/A

Ana L. Milanova - 14:44

A: No, not necessarily. Static semantic analysis usually involves having a more complete view of the program than an interpreter has. Most often static semantic analysis is done in compilers.

---

ANON - 14:44

Q: Probably a bad question, but if I had the following: `pop([X | L], X, L)`

If I had `Stack = [1, 2, 3]` then ran `pop(Stack, 1, Stack)`, would `Stack` be unified with `[2, 3]` afterwards?

Priority: N/A

Ana L. Milanova - 14:45

A: We will get back to those at the end of class. And in office hours.

Ana L. Milanova - 16:28

A: No, if you call with `pop(Stack, 1, Stack)` prolog will return false. This is because `L` will unify with `[2,3]` due to the first unification (`[1,2,3]=Stack = [X|L]`) and it will unify with `[1,2,3]` due to the second unification (`[1,2,3] = Stack = L`).

Ana L. Milanova - 16:29

A: `pop(Stack,1,Rest)` should work as you'll get in `Rest` the tail of `Stack`.

Ana L. Milanova - 16:30

A: Or you can just do this: `[Head | Rest] = Stack`. Then if `Stack` is `[1,2,3]`, then `Head` gets bound to `1` and `Rest` gets bound to `[2,3]`.

---

ANON - 14:57

Q: So `E1` is `E` but we put a subscript so that we don't get confused right?

Priority: N/A

Ana L. Milanova - 14:58

A: Yes, the kind of non-terminal is the same, `E`. But there are different nodes that are `E`-nodes in the parse tree, and the subscripts distinguish between the different nodes in the parse tree.

---

ANON – 14:57

Q: This AG1 is not LL(1), right?

Priority: N/A

Ana L. Milanova – 14:59

A: The underlying CFG can be LL(1), or not. And yes, the underlying CFG of AG1 is not LL(1).

---

ANON – 15:06

Q: Would you mind discussing the attribute() slides in more detail in relation to LL1() as it pertains to the last function of the homework

Priority: N/A

Ana L. Milanova – 15:07

A: We do discuss this in detail in the last two parts of lecture :).

---

ANON – 15:09

Q: For when HW2 questions are answered at the end: should our parseAndSolve solution logic look something like this tree shown on slide 13 during the presentation? If so, how do we address when the production rules do not have both subterms for the operators?

Priority: N/A

Ana L. Milanova – 15:11

A: Last parts of lecture discuss several attribute grammars that solve the "attribute" question in HW2. We will discuss more if questions arise (if my internet connection, which is breaking, permits!)

---

ANON – 15:23

Q: Does it matter if we put  $A \rightarrow aA$ ? same goes for B and C

Priority: N/A

Ana L. Milanova – 15:23

A: No, it will be exactly the same.

---

ANON – 15:24

Q: when we write  $a^k b^k c^k$  as a regular expression, shouldn't it represent all expressions of equal number of a's, b's, and c's?

Priority: N/A

---

ANON – 15:25

Q: Is .c the "pointer"?

Priority: N/A

Ana L. Milanova – 15:27

A: .c is the count in the example on slides 19 and 20.

Ana L. Milanova - 15:29

A: The attributes can be basically anything we want, strings, numerical values, pointers. And the attribute rules can perform essentially arbitrary operations on the attribute values.

---

ANON - 15:26

Q: Never-mind

Priority: N/A

---

ANON - 15:30

Q: on slide 22, was that num, man, max, or mam?

Priority: N/A

Ana L. Milanova - 15:33

A: I believe it was max.

Ana L. Milanova - 16:32

A: Yes, I believe it was max as the other 3 don't make sense :). My connection was bad on that slide blurring the slides...

---

ANON - 15:30

Q: Why is it  $F.c = E.c + 1$ ?

Priority: N/A

Ana L. Milanova - 15:34

A: In  $F \rightarrow (E)$  we will be increasing the nesting depth by 1. The nesting depth of the factor F will be 1 plus the nesting depth of the expression in parentheses.

---

ANON - 15:45

Q: So all Rules (1) are the corresponding sibling values that we are going pass down the Recursion as a Helper Value? Then when the Recursion comes back up, those values can then be used to compute the .val values via (2) Rules?

Priority: N/A

Ana L. Milanova - 15:46

A: Yes, precisely.

---

ANON - 15:54

Q: If we have attended this lecture, will our office hours check in for the week be satisfied?

Priority: N/A

Konstantin Kuzmin - 15:55

A: Yes.

---

ANON - 15:56

Q: Will the + from AG3 act the same way as the \* num factor\_tail from

the homework? My thinking is no since we have a new non\_terminal in factor\_tail

Priority: N/A

Ana L. Milanova - 16:05

A: Yes, + and \* are analogous and handled similarly.

---

ANON - 15:56

Q: So is it basically that we create predicates for each of those sub and val values?

Priority: N/A

Ana L. Milanova - 16:00

A: If you want to implement AG3, then yes, you'll create structures that represent and handle val and sub. But it will be much easier to implement the "hack" that allows us to evaluate the expression bottom up.

Ana L. Milanova - 16:01

A: I can go over this more after we are done with streaming.

---

ANON - 16:05

Q: For L-attributed grammars, we would visit nodes top-down for the inherited attributes but in order to compute the synthesized attributes we would still need to go bottom-up, right?

Priority: N/A

Ana L. Milanova - 16:06

A: Yes, when the parse of the child node returns (i.e., the corresponding recursive descent procedure returns), the current node (i.e., procedure) passes the return value up the recursion chain.

Ana L. Milanova - 16:08

A: The return value of the recursive descent procedure is typically a synthesized attribute.

---

ANON - 16:11

Q: Is today's lecture on the test next week

Priority: N/A

Ana L. Milanova - 16:33

A: Only Part1 and Part2, mostly Part2.

---

ANON - 16:11

Q: I can see the screen

Priority: N/A

---

ANON - 16:16

Q: in the 2nd parsehelper, is that "RHS + Stack?"

Priority: N/A

Ana L. Milanova - 16:33

A: Beware, there are bugs/omissions in what I wrote as outline.

---

ANON – 16:18

Q: Does the rule on office hour still apply (not answering attributes and extra credit)?

Priority: N/A

Ana L. Milanova – 16:33

A: Yes, that's the plan now. But if we get a sparse queue, we'll take questions on attributes as well.

---

ANON – 16:18

Q: Is Tuesday's lecture a review lecture? Does that mean there are no pre-recorded stream for it?

Priority: N/A

Ana L. Milanova – 16:34

A: Yes, we'll do a live review lecture. That's the plan at least.

---

ANON – 16:20

Q: yeah i just wanted to make sure i was seeing a +. Thank you. I will visit office hours to get through with parseLL. still having some issues with that. Once I get parseLL working, I will tackle the attributes.

Priority: N/A

---

ANON – 16:21

Q: thanks for the lecture

Priority: N/A

Ana L. Milanova – 16:37

A: You are welcome. We are all doing our best, given the situation...

---

ANON – 16:22

Q: when will the office hour start?

Priority: N/A

Ana L. Milanova – 16:23

A: 4:45pm. I might be a little late taking care of the Q&A but the TAs should be there

---

ANON – 16:23

Q: I was kind of like stuck into some bugs and several implementation details

Priority: N/A

Ana L. Milanova - 16:37

A: We'll be helping with foxes and hens and parserLL, so bring your code.

---

ANON - 16:24

Q: Can I get help for the detail coding problems during the office hour ?

Priority: N/A

---

ANON - 16:26

Q: I am using the parsehelper as you suggested. My issue is not handling the epsilon characters. I'm trying to check whether prodnum is 3 or 6 and then pass in the bottom half of the stack with the entire input. Not sure if I am approaching it the correct way

Priority: N/A

Ana L. Milanova - 16:38

A: You should have a clause handling that case, that will just pop off the epsilon symbol off the stack.

---

ANON - 16:31

Q: when will office hours end today?

Priority: N/A

Ana L. Milanova - 16:35

A: If you join the queue before the time's up, 6:35 we'll do our best to help. But we cannot guarantee about afterwards.