

Q&A Session for Programming Languages Lecture 10

Session Number: 1204210457

Date: 2020-10-6

Starting time: 14:25

ANON - 14:25

Q: when will the quiz open?

Priority: N/A

Konstantin Kuzmin - 14:25

A: 2:30 pm EDT

ANON - 14:31

Q: I can't access submittity right now for some reason.

Priority: N/A

Konstantin Kuzmin - 14:39

A: Just retry in a few seconds.

ANON - 14:32

Q: Submittity is much slower today compared to past quizzes even

Priority: N/A

Konstantin Kuzmin - 14:39

A: It is. Just retry in a few seconds.

ANON - 14:33

Q: Is submittity down?

Priority: N/A

Ana L. Milanova - 14:36

A: Not that we know of. It is a bit slower loading some pages than usual though.

ANON - 15:01

Q: Would we be expected to submit .pl files for the exam?

Priority: N/A

Konstantin Kuzmin - 15:02

A: No, if there are any questions where you need to write Prolog, you will be expected to write code directly in the text box.

ANON - 15:01

Q: If we do not finish reviewing all Problem Sets, will the Solutions be posted?

Priority: N/A

Konstantin Kuzmin - 15:04

A: We will not cover all problem sets today. The intent is for students to practice themselves, not just watch us solving the problems. The remaining solutions will be posted the day before the test.

ANON – 15:03

Q: When we are creating attribute grammars, how would one represent multiple attribute expressions for a production? How are those attribute grammars chosen between?

Priority: N/A

Ana L. Milanova – 16:27

A: You have a lot of freedom to create attributes and rules. Essentially anything you'd like can be an attribute, e.g., count, strings, etc. The semantics of your interpretation determines the rules.

Ana L. Milanova – 16:29

A: E.g., if we have $A \rightarrow Aa$ and are counting the number of "a"'s at the fringe of the tree, we'll have an attribute "count", and the count of the parent A will be computed as plus one of the count of the child A: $A \rightarrow A1a$ gives rise to rule $A.c = A1.c + 1$.

ANON – 15:04

Q: ^ for last question, not attribute grammars, but attribute expressions for a production

Priority: N/A

ANON – 15:04

Q: will the exam questions be similar in rigor/style to the homework?

Priority: N/A

Ana L. Milanova – 16:31

A: Exam questions will cover similar material as HWs, however, questions will have shorter answers compared to HW questions.

ANON – 15:05

Q: What exactly is meant by LL Conflicts?

Priority: N/A

Ana L. Milanova – 16:34

A: There is a conflict when an LL(1) table entry contains 2 or more productions. E.g., the Dangling Else grammar exhibits an LL(1) conflict as we have two productions in entry `[else_part, else]`.

ANON – 15:07

Q: what does it mean for function to be third-class value again?

Priority: N/A

Steven Haussmann – 15:09

A: It means that something can't be treated like ordinary data. So, if functions are third-class, then you can't stick them in variables, pass them around, and so forth.

Ana L. Milanova - 16:35

A: Yes, what Steven says. A function is third-class value when it cannot be passed as argument to other functions or returned from another function.

ANON - 15:08

Q: So we will not have to do any L-Attributed Grammars?

Priority: N/A

Ana L. Milanova - 16:36

A: No. Only bottom-up evaluation of attribute rules. As we did in examples in class and in lecture today.

ANON - 15:09

Q: If we were using an online tool for making Parse Trees, can we use that same tool on the test, or should we be making it in a local app like MS Paint?

Priority: N/A

Ana L. Milanova - 16:37

A: Oh online tool. I might have answered this incorrectly in class. Rules specify CLOSED internet, so you WON'T be allowed to use that online tool.

ANON - 15:09

Q: In problem set 2, #5a says " $d(A, B, Q, R)$ does _____

Q contains _____

R contains _____". In what way do you want us to answer this? Do you want us to answer this specifically pertaining to $d(5, 3, Q, R)$?

Priority: N/A

Ana L. Milanova - 16:38

A: No, in this part, I'm asking what do Q and R stand for. What algorithm does "d" implement and what Q and R stand for in this algorithm. (You are right question is unclear in that regard, and we will do our best to avoid ambiguity on test.)

ANON - 15:10

Q: I missed that, it is equal precedence because both operators are in the same production?

Priority: N/A

Steven Haussmann - 15:11

A: Yes, that's one way to describe it. You could also argue that there's nothing enforcing the parsing of one operator before the other.

Steven Haussmann – 15:12

A: So, even if you had two separate nonterminals, you could have ambiguity if you can move freely between them. $E1 \rightarrow E1 + E1 \mid E2$ and $E2 \rightarrow E2 * E2 \mid E1$ would not enforce precedence, for example.

ANON – 15:10

Q: Is there an easy way to identify the # of parse trees an ambiguous grammar can produce?

Priority: N/A

Steven Haussmann – 15:13

A: If you're asked to count, it should be easy to enumerate them all. More generally, you can get a very large number of parse trees in some cases -- possibly quadratic or even exponential

ANON – 15:15

Q: An operator is both left and right associative only if it is possible to have expressions with that operator recurse either to the left or to the right?

Priority: N/A

Ana L. Milanova – 16:40

A: Yes, that is correct. If it is possible to parse so that tree recurses to the left, and it is possible to parse so that tree recurses to the right, then we say it is "both left and right associative".

ANON – 15:15

Q: What lectures of material will be covered on the exam?

Priority: N/A

Ana L. Milanova – 16:43

A: Lec 1-9, however not all of lec 9 material will be on the test. You don't have to worry about L-attributed grammars.

ANON – 15:18

Q: Will the solutions for the Practice problems be released?

Priority: N/A

Konstantin Kuzmin – 15:19

A: We will not cover all problem sets today. The intent is for students to practice themselves, not just watch us solving the problems. The remaining solutions will be posted the day before the test.

ANON – 15:18

Q: what about the extra credit?

Priority: N/A

Ana L. Milanova – 16:44

A: No time for extra credit unfortunately :). But answer to Quiz 1 extra credit was 42 (the answer to the ultimate question of life, universe, and everything :)).

ANON - 15:18

Q: Can we go over the extra credit problem from quiz 1 if we have time?

Priority: N/A

Ana L. Milanova - 16:46

A: Sorry, we don't have time for any of the extra credits, in any of the quizzes or homeworks unfortunately...

ANON - 15:18

Q: what exactly does "dangling else" mean?

Priority: N/A

Steven Haussmann - 15:22

A: An example would be something like "if a then if b then c else d". It's unclear whether "else" goes with the first "if" or the second "if"

ANON - 15:19

Q: is there any other way to verify the ambiguity of grammar instead of drawing more than one parse trees?

Priority: N/A

Ana L. Milanova - 16:47

A: This is the easiest way, and what we will be asking for in the context of this class.

ANON - 15:20

Q: what was the answer to the quiz 1 extra credit problem?

Priority: N/A

Ana L. Milanova - 16:47

A: 42.

ANON - 15:20

Q: $S \rightarrow ABC$

Priority: N/A

ANON - 15:20

Q: Is an operator both left and right associative only if it is possible to have expressions with that operator recurse either to the left or to the right?

Priority: N/A

Steven Haussmann - 15:27

A: Yes, that's the core problem that you have if your grammar is both left and right associative.

ANON - 15:22

Q: $S \rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAbbBccC \Rightarrow aabbcc$. Isn't this in the form $a^n b^n c^n$ for $n = 2$?

Priority: N/A

Steven Haussmann - 15:24

A: The problem with this is that you can only replace one non-terminal at a time. You can't enforce the language's rules here.

Steven Haussmann - 15:25

A: So you do, indeed, produce all strings of $a^n b^n c^n$ -- but you also produce strings with different numbers of a's/b's/c's

Ana L. Milanova - 16:48

A: Yes, what Steven says. The problem is that the grammar generates a much larger language than what we need. We need it to generate exactly that language, no more, no less.

ANON - 15:26

Q: do proper lists have to have an empty list `[]` as a leaf node?

Priority: N/A

Steven Haussmann - 15:36

A: If you're referring to how you can draw a list as a tree, then yes, you'll eventually terminate with a `[]` (assuming the list has a finite length, of course!)

ANON - 15:26

Q: does prolog unify the structures from Q4? I know it shouldn't in theory, but I recall in lecture hearing that prolog allows it for some reason (maybe when unifying X to $[X]$?)

Priority: N/A

Steven Haussmann - 15:43

A: Interesting case! It seems like Prolog winds up producing infinitely nested lists when I trace something like $X = [X]$, $X = [X]$.

Steven Haussmann - 15:44

A: In general, though, it's going to refuse to unify conflicting values. So $[3] = 3$ will always fail

ANON - 15:27

Q: How does prolog handle $A > B$ where either A or B are unbound?

Priority: N/A

Steven Haussmann - 15:31

A: It doesn't! You will get an error, since it doesn't have enough information to pick values.

ANON - 15:27

Q: Can you explain what invertible is again?

Priority: N/A

Steven Haussmann - 15:32

A: "invertible" means that the predicate can be run backwards -- giving it a predicate in a position that would intuitively be considered an input, not an output

Steven Haussmann - 15:33

A: Of course, Prolog doesn't have a notion of 'inputs' and 'outputs'! It just tries to satisfy predicates.

ANON - 15:28

Q: Can you go over why a list might be improper or proper again?

Priority: N/A

Steven Haussmann - 15:29

A: A proper list has, as its tail, another proper list. The empty list is a proper list.

ANON - 15:28

Q: a quick question the most recent frame of A is frame of A or the most recent frame of frame of A?

Priority: N/A

Steven Haussmann - 15:34

A: The most recent frame of a function is the first one you come across when working your way through the stack. It can be the frame most recently put on the stack.

ANON - 15:29

Q: Oh so the rule specifies that we need to only produce strings of a^n , b^n , c^n and no other?

Priority: N/A

Steven Haussmann - 15:29

A: Yes, you need to produce exactly that language -- otherwise, a grammar that accepts any string would be a valid answer.

ANON - 15:29

Q: I know we cannot use the internet for the test but what about opening PDFs for notes or using mediasite for videos?

Priority: N/A

Ana L. Milanova - 16:53

A: Yes, lecture ppt and pdfs from the course website as well as Mediasite are allowed. Also, crib sheets, derived from lecture notes and textbook are allowed as well.

ANON - 15:31

Q: Can you give an example of an operator being left and right associative?

Priority: N/A

Steven Haussmann - 15:33

A: Something like $E \rightarrow E + E \mid id$ would do.

ANON - 15:31

Q: would quiz 1 question 4 also be false because $n = 0$, $a^0b^0c^0 = 111$ cannot be produced?

Priority: N/A

Ana L. Milanova - 16:59

A: No, $a^0b^0c^0$ can be generated by the grammar. The problem is that the grammar generates a much larger language.

ANON - 15:32

Q: how quickly will the answers to todays quiz be released?

Priority: N/A

Ana L. Milanova - 17:01

A: In an hour or so. I will update the pdf of the slides with the answers to all quizzes 1-4, as well as this Q&A session. You should also have your quiz grade in rainbow grades pretty soon today.

ANON - 15:34

Q: Where can we find these practice problems?

Priority: N/A

Steven Haussmann - 15:35

A: The questions are posted on Submittity. Check <https://submittity.cs.rpi.edu/courses/f20/csci4430/forum/threads/193>

Steven Haussmann - 15:35

A: <https://submittity.cs.rpi.edu/courses/f20/csci4430/forum/threads/193>

Konstantin Kuzmin - 15:37

A: <https://submittity.cs.rpi.edu/courses/f20/csci4430/forum/threads/193>

ANON - 15:34

Q: How is $[1,2|3]$ a proper list? How would you know that it ends in $[]$? Same with $[1,2|3]$ how do you know that it doesn't end with $[]$?

Priority: N/A

Steven Haussmann - 15:35

A: A proper list has a tail that is a proper list. $[3]$ is a proper list, so $[1,2|3]$ must also be a proper list.

Steven Haussmann - 15:35

A: $[1,2|3]$ has a tail of 3 ; 3 is not a proper list, so $[1,2|3]$ is not a proper list

ANON – 15:40

Q: Why do we have T.pre again?

Priority: N/A

Ana L. Milanova – 17:02

A: T.pre is the attribute that synthesizes the prefix form of the expression represented by T.

ANON – 15:42

Q: could you draw the parse tree?

Priority: N/A

Ana L. Milanova – 17:02

A: We did!

ANON – 15:43

Q: option 1

Priority: N/A

Ana L. Milanova – 17:02

A: We did!

ANON – 15:43

Q: Generate parse tree

Priority: N/A

Ana L. Milanova – 17:02

A: We did!

ANON – 15:44

Q: What lectures of material will be covered on the exam? Will it be all of 1–9?

Priority: N/A

Ana L. Milanova – 17:03

A: 1–9, but you don't have to worry about L-attributed grammars.

ANON – 15:44

Q: Why didn't you do it for Spre?

Priority: N/A

Ana L. Milanova – 17:03

A: S.pre = E.pre.

ANON – 15:47

Q: Why is star first?

Priority: N/A

Konstantin Kuzmin – 15:49

A: We are asked in the problem to "construct an attribute grammar that translates the expressions into prefix notation." Hence, the prefix notation.

Konstantin Kuzmin – 15:51

A: <https://demonstrations.wolfram.com/PrefixInfixAndPostfixNotation/>

ANON – 15:48

Q: Why do you put + and * first if they are in the middle in the parse tree?

Priority: N/A

Konstantin Kuzmin – 15:49

A: We are asked in the problem to "construct an attribute grammar that translates the expressions into prefix notation." Hence, the prefix notation.

Konstantin Kuzmin – 15:51

A: <https://demonstrations.wolfram.com/PrefixInfixAndPostfixNotation/>

ANON – 15:49

Q: Yes please

Priority: N/A

ANON – 15:49

Q: yes going over part 2 would be great

Priority: N/A

Ana L. Milanova – 17:03

A: We did!

ANON – 15:50

Q: will the answers for hw 2 be available so that we can see how attribute works for the parser problem?

Priority: N/A

Ana L. Milanova – 17:04

A: No, we won't make any code available. However, you can find the attribute grammar that solves the problem in HW2 in Lecture 9.

ANON – 15:57

Q: How come she is not calculating the len of the 2nd line $S \rightarrow L$?

Priority: N/A

Ana L. Milanova – 17:05

A: We don't really need the length in S. We only need the length in order to properly account for the contribution of the binary number to the right of the decimal point. (I.e., len is a helper attribute.)

ANON - 15:58

Q: Should S.len be updated at the top-level?

Priority: N/A

Ana L. Milanova - 17:05

A: See above question and answer.

ANON - 15:59

Q: If len is only used for L₂, why not make L₂ a separate production, M?

Priority: N/A

Ana L. Milanova - 17:06

A: We can do that, sure! We will have the same productions in the CFG for L and M though, so we'll make the the underlying CFG more complex.

ANON - 15:59

Q: Could you go over the attribute for L --> L1B again? Thanks.

Priority: N/A

Ana L. Milanova - 17:10

A: First, length of L is length of L1 plus 1: $L.len = L1.len + 1$. Second, val of L is val of L1 shifted left plus value of B: $L.val = L1.val * 2 + 1$. (left shift is just $x2$)

ANON - 15:59

Q: Will we have to draw DFAs?

Priority: N/A

Ana L. Milanova - 17:10

A: You might.

ANON - 16:01

Q: Oh I see it is because S is never on the RHS of any Production so we do not need it.

Priority: N/A

Ana L. Milanova - 17:10

A: Correct. We don't need len of S. S is the top level binary number. We only need the val of S.

ANON - 16:01

Q: For example, $S \rightarrow L$ wouldn't need any len values, and $S \rightarrow L.M$ would only need len for the M part

Priority: N/A

Ana L. Milanova - 17:11

A: Yes, you are correct!

ANON - 16:01

Q: Ok, Thank you!

Priority: N/A

ANON - 16:01

Q: Will these notes be available to us in PDF form?

Priority: N/A

Ana L. Milanova - 17:11

A: Yes, we will make the Quiz answer notes available today.
And we will make the Problem Set answer notes available on Thursday.

ANON - 16:03

Q: If we finished the Problem Sets, are there any more Practice Problems for us?

Priority: N/A

Ana L. Milanova - 17:12

A: Great! We have no plans for more problem sets for now.

ANON - 16:03

Q: what does the + at the end mean for this question?

Priority: N/A

Ana L. Milanova - 17:14

A: This is the Kleene +, which means one or more occurrences.
(w* | z*)+ means one or more occurrences of any string generated by (w* | z*). The key observation here is that w* | z* generates the empty string.

Ana L. Milanova - 17:14

A: And therefore, (w* | z*)+ generates all strings of w's and z's, including the empty string.

ANON - 16:05

Q: "Construct an attribute grammar that associates an attribute count to the root of the tree S such that count contains the number of digits in the expression." Do this mean we just count the number of digits in the expression?

Priority: N/A

Ana L. Milanova - 17:15

A: Yes, that is correct.

ANON - 16:05

Q: Will the answers to these practice problems be posted somewhere?

Priority: N/A

Konstantin Kuzmin – 16:06

A: We will not cover all problem sets today. The intent is for students to practice themselves, not just watch us solving the problems. The remaining solutions will be posted the day before the test.

ANON – 16:10

Q: Sorry I didn't catch what the professor said. What does Kleene plus do?

Priority: N/A

Konstantin Kuzmin – 16:10

A: One or more occurrences. https://en.wikipedia.org/wiki/Kleene_star#Kleene_plus

ANON – 16:11

Q: i thought the kleene plus meant it couldnt be an empty string

Priority: N/A

Konstantin Kuzmin – 16:13

A: You are right. Kleene plus is $V_1 \cup V_2 \cup V_3 \dots$. So, it excludes $\{\epsilon\}$

Konstantin Kuzmin – 16:14

A: You are right. Kleene plus is $V_1 \cup V_2 \cup V_3 \dots$. So, it excludes $\{\epsilon\}$

Konstantin Kuzmin – 16:15

A: You are right. Kleene plus is $V_1 \cup V_2 \cup V_3 \dots$. So, it excludes $\{\epsilon\}$

Ana L. Milanova – 17:16

A: It depends what the expression we apply the Kleene plus does. Since our expression E generates the empty string then E^+ would generate the empty string too. But if E is something like $(w|z)^+$, then we don't have the empty string in the language.

ANON – 16:13

Q: 1 can generate any number of x's or y's but not both right?

Priority: N/A

Ana L. Milanova – 17:17

A: 1 being $(x|y)^*$ generates all strings of x's and y's, including just x's and just y's, and empty string.

ANON – 16:13

Q: so then the answer to b is wrong?

Priority: N/A

Ana L. Milanova – 17:19

A: Nope. Because $(w^* | z^*)$ does generate the empty string. So "one or more occurrence of the empty string" generates the empty string.

ANON – 16:14

Q: because it uses an empty string generated by 3

Priority: N/A

Konstantin Kuzmin – 16:14

A: You are right. Kleene plus is $V_1 \cup V_2 \cup V_3 \dots$. So, it excludes $\{\epsilon\}$

ANON – 16:16

Q: so if the Kleene + excludes ϵ , why is an empty string accepted as valid?

Priority: N/A

Ana L. Milanova – 17:19

A: See answers to questions earlier.

ANON – 16:16

Q: I undersand what Professor Milanova just did to show they have the same precedence and I agree as I did it this way, but what she wrote included paranthesis and the Grammar did not include this. Is this alright?

Priority: N/A

Ana L. Milanova – 17:21

A: I used the parentheses to illustrate the precedences shown in the parse trees. You are correct that the grammar does not include parentheses and does not generate parenthesized expressions.

ANON – 16:17

Q: How many questions are expected on the exam?

Priority: N/A

Ana L. Milanova – 17:22

A: An appropriate number.

ANON – 16:18

Q: When asked to give evidence like at part c, do we just show different parse trees for the same expression?

Priority: N/A

Ana L. Milanova – 17:22

A: Yes.

ANON – 16:18

Q: $(w^* | z^*)$ generates 0 or more x's or z's. so an empty string can be produced there. Then the + reproduces the empty string..? or is the + ignored in that case?

Priority: N/A

Ana L. Milanova - 17:22

A: Correct!

ANON - 16:20

Q: Should/could it also be $T \rightarrow -(E)$ because it seems that this Grammar limits you from applying $-$ to a larger expression which was intended in original grammar?

Priority: N/A

Ana L. Milanova - 17:25

A: You are correct, the grammar limits you in this way. If we add parenthesized expressions we can handle this case: $T \rightarrow -T \mid id \mid (E)$.

ANON - 16:21

Q: Sorry if this is a duplicate question, but the solutions (the ones not reviewed currently) to the problem sets will be posted later correct?

Priority: N/A

Ana L. Milanova - 17:25

A: Yes, this was addressed earlier.

ANON - 16:22

Q: Hi, could you give an recursive example of frames on stack? Thank you!

Priority: N/A

Ana L. Milanova - 17:26

A: One of the questions on Problem sets has recursion. We will post the answers to the problem sets later this week.

ANON - 16:25

Q: So was the given answer for 1b on practice test 1 incorrect?

Priority: N/A

Ana L. Milanova - 17:27

A: The answer to 1b given is correct. See earlier questions in this Q&A session.

ANON - 16:25

Q: won't making it $T \rightarrow (T)$ for example on the lowest line of a grammar restrict what can be put inside of the parantheses to only things that T can derive (things on its level and anything with higher precedence)?

Priority: N/A

Ana L. Milanova - 17:28

A: Yes, making $T \rightarrow -T$ is restrictive. We can only have $----id$ basically. If we want to apply the minus on larger expressions we need

T -> -T | id | (E)

ANON - 16:26

Q: So for the prefix notation questions, are they just basically the same as what we did in parseandSolve but just trying to match the desired pattern rather than to compute arithmetic?

Priority: N/A

Ana L. Milanova - 17:29

A: Yes, you are correct. We are interpreting the expression differently. We are generating the prefix form here, while in parse and solve, we were evaluating. Both are "interpretations" of the underlying expression string. The CFG does not give this.

ANON - 16:28

Q: Could you give an recursive example of frames on stack? Thank you!

Priority: N/A

Ana L. Milanova - 17:30

A: See the earlier answer.

ANON - 16:28

Q: the one about regular expressions

Priority: N/A

ANON - 16:28

Q: and dividing it to parts 1 2 and 3 with 3 having a kleene plus.

Priority: N/A

ANON - 16:29

Q: for b part 3 was used as an empty string, which I thought wasnt allowed due to the kleene plus

Priority: N/A

Ana L. Milanova - 17:31

A: See earliser answer to questions about 1b from ProblemSet1.

ANON - 16:31

Q: Regarding stack questions, for example HW3, we shouldn't remove procedures off the stack unless they return right? For example, recursion shouldn't be removed off the stack unless the base case is met and returns?

Priority: N/A

Ana L. Milanova - 17:31

A: Correct. When the procedure completes and returns, its frame is popped off the stack.

ANON - 17:03

Q: Could you give an recursive example of frames on stack? Thank you!

Priority: N/A

ANON - 17:27

Q: This is in response to my $T \rightarrow -(E)$ Question: Then technically since the $T \rightarrow -T \mid id$ would not capture what we intended with the original grammar, is it then the case that there cannot exist an equivalent unambiguous grammar since this is not equivalent?

Priority: N/A

Ana L. Milanova - 17:33

A: The equivalence is in the sense of "generating the same language". So in that sense, they should be equivalent. But the original grammar is ambiguous and allows for different structures/interpretation of the evaluation. The new one chooses specific struct.