Q&A Session for Programming Languages Lecture 2

Session Number: 1205196106
Date: 2020-9-4
Starting time: 14:22

ANON - 14:28
Q: last lecture when the video plays it switches to a different audio device having me take a couple seconds to put on that device and then adjust the audio. Anyone know why this is the case or has a solution?
Priority: N/A

Steven Haussmann - 14:30
A: You might want to try having Webex use your default system setting for the output device (or picking a specific device if you were already doing that).

ANON - 14:31
Q: I have a device selected for webex specifically. But it ignores that device for the lecture video and then switches back to that device when the video ends
Priority: N/A

ANON - 14:32
Q: Yeah it did it again
Priority: N/A

ANON - 14:33
Q: How do we know which expression the Kleene Star applies to? An example from the homework: 0*10*10*10*, for the second star, does this mean that any number of 0s or does it mean any numbers of 10? If instead:0*(10)*10*10*, does the 2nd star mean any 10s?
Priority: N/A

Ana L. Milanova - 14:36
A: We cover this in lecture today! * has higher precedence than concatenation. So 10* is 1(0)*, i.e., 1 followed by any number of 0's, it is not (10)*.

ANON - 14:35
Q: Where is the lecture video? Webex audio isn't working?
Priority: N/A

Ana L. Milanova - 14:37
A: It is available on Mediasite. But try increasing the volume, the recording came out low.

Ana L. Milanova - 14:38

A: Mediasite: Mediasite: https://mediasite.mms.rpi.edu/Mediasite5/Channel/programming_languages

---

ANON - 14:43
Q: Could you explain "push-down automation" in more details?I don't quite get the concept. Thanks
Priority: N/A
Ana L. Milanova - 14:51
A: A push-down automaton, or PDA, is just a DFA with a stack. I.e., in addition to the DFA, we have a stack to store additional information.

ANON - 14:47
Q: the audio is sounding kind of faint and muffled. I have it turned as loud as my computer will go
Priority: N/A
Ana L. Milanova - 14:53
A: I think it will get better!

ANON - 14:48
Q: what denotes an empty string? the feed is a little blurry for me
Priority: N/A
Steven Haussmann - 14:49
A: Lowercase epsilon: ε

ANON - 14:49
Q: is it possible for you to talk closer to the microphone in the future? This is really difficult to hear/understand even with my computer at max volume.
Priority: N/A
Ana L. Milanova - 14:54
A: Yes, it will get better.

ANON - 14:52
Q: What exactly does the Kleene star mean in this context? I think I'm getting mixed up
Priority: N/A
Steven Haussmann - 14:56
A: It means that the thing to its left -- like f or (abcde) -- can appear 0 or more times. So 01* represents things like 0, 01, 011, 0111, and so forth. 0(11)* would be 0, or 011, or 01111

ANON - 14:54
Q: With Kleene * operations does it go left to right?

Priority: N/A

ANON - 14:54
Q: Does the kleem star represent 0 or more or 1 or more?
Priority: N/A
Steven Haussmann - 14:54
A: 0-or-more.

ANON - 14:55
Q: is 'digit digit*' different from 'digit*'?
Priority: N/A
Ana L. Milanova - 14:55
A: Yes, digit digit* is sometimes denoted digit+, it means one or more. digit* is 0 or more.

ANON - 14:55
Q: for previous slides example 3, could it also be a non-negative decimal constant?
Priority: N/A
Ana L. Milanova - 15:00
A: That particular regex can only generate a non-negative real. We have at least one digit after the decimal point.

ANON - 14:57
Q: would it be possible to adjust the volume of the lecture videos. my enviorment is not the quietest but usually noise canceling headphones do the trick. After trying multiple noise canceling headphones i still cannot hear much.
Priority: N/A
Ana L. Milanova - 15:01
A: I probably won't be able to do this for this video. But I think it gets better.

ANON - 14:58
Q: i know it wont happen during live lecture, but being able to watch the videos later would be better than nothing!
Priority: N/A

ANON - 15:01
Q: BNF stands for what? I couldn't make it out <> Normal Form
Priority: N/A
Ana L. Milanova - 15:01
A: BNF = Backus Naur Form

ANON - 15:01
Q: What does BNF stand for?
Priority: N/A

Ana L. Milanova - 15:02
A: Backus Naur Form. After John Backus and Peter Naur

Ana L. Milanova - 15:03
A: They are Turing Award Winners. Peter Naur won for the use of formal languages to describe the syntax of programming languages, if I remember correctly.

ANON - 15:02
Q: Are we able to use either notation in homework or is one preferred over the other?
Priority: N/A

Ana L. Milanova - 15:03
A: You can use any conventional notation. You can use the BNF we have adopted in class and in textbook, but if you have used a different notaiton in Focs that will be fine too.

ANON - 15:04
Q: Can you go over what regular grammer is?
Priority: N/A

Ana L. Milanova - 15:05
A: A regular grammar is a special kind of context-free grammars. It generates a regular language.

Ana L. Milanova - 15:06
A: It has specific structure that is more constraint than a general CFG. This structure was defined in slide 23.

ANON - 15:05
Q: What does the term "terminal" and "non-terminal" mean?
Priority: N/A

Steven Haussmann - 15:06
A: A terminal symbol can't be replaced by anything else. A non-terminal has at least one way to replace it.

ANON - 15:06
Q: So for something like A -> 1A2 would it generate 1^n2^n?
Priority: N/A

Ana L. Milanova - 15:07
A: Yes, assuming that you also have an A -> \epsilon production.

ANON - 15:06

```
Q: So, terminal is 1, 0 and nonterminal is S, A?
Priority: N/A
        Ana L. Milanova - 15:07
        A: Correct.
        Ana L. Milanova - 15:08
        A: The terminals come from the alphabet, 1, 0. The
nonterminals construct strings using the rules of grammar.
```

```
ANON - 15:08
Q: How does making that change make it context free?
Priority: N/A
        Ana L. Milanova - 15:09
        A: Will get to this later, longer answer.
        Steven Haussmann - 15:10
        A: In brief, a context-free language has things you couldn't
describe with just a regular grammar. For exmaple, you can't write a
regular expression for a^n b c^n because you have to make sure there
are the same number of a's and c's
```

```
ANON - 15:08
Q: how did you change a^n b^m c^n d^m in slide 27?
Priority: N/A
        Ana Milanova - ?:?
        A: a^n b^m c^m d^n (just switched the powers n and m).
```

```
ANON - 15:08
Q: So is non-context-free grammar just regular grammar with some
conditions?
Priority: N/A
        Ana L. Milanova - 15:10
        A: No. regular grammars are context-free grammars, just more
constrained with some conditions (i.e., the conditions make them less
expressive). Non-context-free grammars can be context-sensitive
grammars which are more general and more complex.
```

```
ANON - 15:09
Q: what was the example that made a^n b^m c^n d^m a context free
language?
Priority: N/A
        Ana Milanova - ?:?
        A: a^n b^m c^m d^n (just switched the powers n and m).
```

```
ANON - 15:09
Q: What was LL and LR grammar? I could not hear it well!
Priority: N/A
```

Ana L. Milanova - 15:11
A: Special kinds of grammars used in compilers and programming languages. Will talk about them next time.

ANON - 15:10
Q: Why was the original not context free?
Priority: N/A
Ana L. Milanova - 15:12
A: It is a context-sensitive language. There is a proof using standard techniques from automata theory, but that is out of scope for us.

ANON - 15:10
Q: ah my internet died what did I miss in the last minute?
Priority: N/A
Ana L. Milanova - 15:13
A: Everything in Q&A will be posted on course webpage!

ANON - 15:11
Q: Why is a^n b^m c^n d^m is non context free but a^m b^n c^m d^n is context free?
Priority: N/A
Ana L. Milanova - 15:12
A: This came up a lot but it's a longer answer, will get back to this later.
A: The proof of why a^n b^m c^n d^m is not context-free is out of scope. a^m b^n c^n d^m is context-free:
A -> b A c | \epsilon
B -> a B d | A (B is the starting non-terminal)

ANON - 15:15
Q: does anything differ in computation from doing leftmost compared to rightmost (or neither) derivation?
Priority: N/A
Ana L. Milanova - 15:19
A: This is broad and I might be misinterpeting the question. Generally, it depends on the grammar. We can expand later.

ANON - 15:16
Q: When parsing the parse trees, is an in-order traversal performed?
Priority: N/A
Ana L. Milanova - 15:20
A: The parse constructs the parse tree, it does not traverse the parse tree. (I might be misinterpreting the question.)

ANON - 15:17
Q: What is the difference betwwen context free languages and context free grammar?
Priority: N/A

Steven Haussmann - 15:19
A: A language is a set of strings that obey some set of rules; a grammar is the set of rules that describe a language.

Steven Haussmann - 15:20
A: So, for example, a language might be {ε, A, AA, AAA, AAAA, ...}, and the grammar for it would be this regular expression: A*

ANON - 15:18
Q: I'm having trouble understanding the audio with the static, if I read the chapters assigned on the course website, will I get the same information?
Priority: N/A

Ana L. Milanova - 15:22
A: Yes, the ppt slides can be useful as well. If you have questions, you can post on the Submitty forum.

ANON - 15:22
Q: with most languages it seems you can build multiple parse trees, is the point that they get parsed in a certain way (such as starting from the right side) or do we just need to avoid languages that can create ambiguity?
Priority: N/A

Ana L. Milanova - 15:26
A: If the language/grammar is unambiguous you should be able to build just one parse tree, i.e., have unambiguous evaluation. Yes, the point is that they are parsed in a certain unambiguous way that gives rise to unambiguous evaluation.

ANON - 15:23
Q: so to summarize, the way you derive to get a parse tree is decided by order of operations as well as computing from left to right?
Priority: N/A

Ana L. Milanova - 15:26
A: Yes.

ANON - 15:24
Q: Can you explain again all the reasons ambiguity is bad?
Priority: N/A

Steven Haussmann - 15:26

A: In this case, ambiguity would allow for multiple interpretations of the same statement. (1 + 2) ∗ 3 vs 1 + (2 ∗ 3), for example

---

ANON - 15:27
Q: real −→ integer exponent | decimal (exponent | ε)
Priority: N/A

ANON - 15:28
Q: Can you explain the meaning and purpose of this statement? real −→ integer exponent | decimal (exponent | ε)
Priority: N/A
Ana L. Milanova - 15:31
A: That's slide 18. This construct describes a Real number (token). A real number can be an integer followed by an exponent, or a decimal (i.e., a number with a floating point) followed by an exponent. The "exponent" uses the E notation.

ANON - 15:29
Q: in the example of expr+expr -> expr+term, would there be a case where expr+expr turns into term+expr?
Priority: N/A
Steven Haussmann - 15:35
A: There isn't a replacement rule for "expr + expr -> expr + term" or anything like that -- the slide meant that we replaced the production rule of "expr -> expo + expr" with the rule of "expr -> expr + term"

ANON - 15:32
Q: Did adding the new nonterminal 'factor' force ∗ to be left associated?
Priority: N/A
Ana L. Milanova - 15:34
A: Having the new nonterminal 'factor' as the right operand in 'term -> term ∗ factor' forces ∗ to be left associative.

ANON - 15:34
Q: How come an integer constant token is digit digit∗ instead of just digit∗? Someone posted a question earlier but am still a little confused.
Priority: N/A
Ana L. Milanova - 15:36
A: digit digit∗ generates at least one digit in the constant. digit∗ generates the empty string as well, and we want to exclude the empty string from the language of valid integer constants, i.e., we

want at least one digit.

ANON - 15:35
Q: When you say followed by the exponent do you mean the integer in raised by an exponent or it in multiplied by another value that is raised by an integer?
Priority: N/A
Ana L. Milanova - 15:40 (corrected answer I typed in class)
A: The meaning is, the integer x 10 raised to the exponent. Where the exponent is defined as specified later in the expression. This is the so-called exponent in E-notation, e.g., 10 E -5 means 10x10^-4.

ANON - 15:35
Q: Why does " real −→ integer exponent | decimal (exponent | ε)" contain ε, an empty string?
Priority: N/A
Ana L. Milanova - 15:41
A: Because we may not need an exponent, just generate the number, e.g., 10.10

ANON - 15:37
Q: For the final example: Why after we make the new nonterminal, term, why didn't the entire expression become: expr--->expr * expr | term?
Priority: N/A
Richard Massimilla - 15:58
A: That definition is ambiguous with regards to associativity. We use expr -> expr * term | term which puts expr to the left of term in 'expr * term' to make it left associative

---

ANON - 15:38
Q: Did anyone catch the very last line she wrote? She said "in order to make ^ right associative, we need to put term's recursive operand on the right"
Priority: N/A
Richard Massimilla - 15:41
A: For the left associative * operator, expr -> expr * term | term was used, with expr to the left of term in the expr * term part. For the right associative ^ operator, we write term -> id ^ term | id

---

ANON - 15:39
Q: did you say + * / - are all left associative or is associativity based on your choice
Priority: N/A
Richard Massimilla - 15:57

A: In mathematics, that's true of those operators. Like if you do additions on a calculator it'll parse left-to-right. When you define your own operators, associativite direction will be up to you

---

ANON - 15:39
Q: Why is a^n b^n a context-free grammer, but a^n b^n c^n is not a context-free grammar?
Priority: N/A
Steven Haussmann - 15:40
A: My rule of thumb for that is to see if generating the language requires anything more than adjacent insertions. S -> ASB -> AASBB -> ... always puts in A and B right next to S, whilst something like S -> ABSC -> AABBSCC -> ... can't be done at all

---

ANON - 15:39
Q: How do we do the homework with a partner? Do we both submit the same assignment?
Priority: N/A
A: HW is individual unless otherwise specified. HW1 is individual. So you do the final write up on your own but you can discuss the HW with friends, mentors, TAs or instructors.

ANON - 15:39
Q: On reg expression slide 19 is digit*.digit* equivalent to digit*(digit.|.digit)digit*?
Priority: N/A
Richard Massimilla - 16:03
A: No, the middle portion in parentheses ensures that there is no empty string on both sides of the decimal point. That we cannot generate just the decimal point .

ANON - 15:39
Q: At the end for the Final Exercise, how come we did not introduce a Factor Non-Terminal as was done in the Previous Example because that Factor also introduced Paranthesis and the Solution draw by Professor did not include any Paranthesis.
Priority: N/A

ANON - 15:40
Q: Going back to the 2nd part of the lecture, where the regular expression was (a | b)(a a | bb), why wasn't "aab" not listed as a regular expression that can be produced by these rules?
Priority: N/A
Steven Haussmann - 15:42

A: That grammar generates four strings: "abb", "aaa", "baa", and "bbb" (assuming it's aa, not a a). You pick either a or b, then pick either aa or bb

---

ANON - 15:44
Q: Does this "count" as office hours. I recall you saying that office hours would be held after class and if it does, would this count for the weekly office hours?
Priority: N/A

Konstantin Kuzmin - 15:47
A: Most likely, yes. We are still figuring out the best way to keep all students engaged but at the same time not to cause any major inconveniences. In any case, participation will be taken only starting from week 3 of classes.

ANON - 15:44
Q: yes
Priority: N/A

ANON - 15:48
Q: Can you explain what this sentence means in the homework? "Your description should be a high level characterization — one that would still make
sense if we were using a different regular expression for the same language."
Priority: N/A

ANON - 15:50
Q: Can you explain how one writes an equivalent unambiguous expression form to make sure one operand has precendence over another opearnd
Priority: N/A

ANON - 15:50
Q: Please use example from final slide if possible?
Priority: N/A

ANON - 15:50
Q: Is there any special meaning to expr, term, and id?
Priority: N/A

Richard Massimilla - 15:53
A: They're similar to variable names. All the meaning in those symbols comes from the definitions that we write, but we chose to name them in a sensible and descriptive way. Expr is the word 'expression' abbriviated and 'id' is the terminal

---

ANON - 15:50
Q: why wcw is not a context free language?
Priority: N/A
Steven Haussmann - 15:51
A: in that language, w was defined to be any sequence of 0 and 1. The problem is that you need to repeat the sequence on both sides of c -- you can easily make two mirrored sequences, like 11001c10011, but you can't get two identical sequences.

ANON - 15:51
Q: are mostly questions that we type up in either LaTex or a Word Doc and then there is a question in Python for the last one, would we submit both a python file and the pdf and just refer to the python file for that one and then would there be an autograder?
Priority: N/A
Ana L. Milanova - 16:12
A: The Python question should not be submitted, it is just for you to work on something in case first 5 are too easy.
Ana L. Milanova - 16:12
A: You will be typing and submitting 1-5 only.

ANON - 15:53
Q: Just skimming the homework - what does LL(1) mean?
Priority: N/A
Richard Massimilla - 15:55
A: It means you do left derivations and operators are left associative
Ana Milanova - ?:?
A: More on LL next week.

ANON - 15:53
Q: Can we get these question and answers saved?
Priority: N/A
Konstantin Kuzmin - 15:55
A: Yes, they are all-going to be posted: https://www.cs.rpi.edu/~milanova/csci4430/schedule.html

---

ANON - 15:55
Q: Can you go over how to construct a parse tree?
Priority: N/A

ANON - 15:55
Q: Actually, could you re-explain the final example of rewriting

expression grammars very very slowly while justifying each step? It was just a little fast in the video...The example was as followsd:
expr -> expr x expr|expr ^ expr | id
Priority: N/A
Ana Milanova - ?:?
A: If the live lecture explanation didn't help, you can go to our office hours.

ANON - 15:55
Q: At the end for the Final Exercise, how come we did not introduce a Factor Non-Terminal as was done in the Previous Example because that Factor also introduced Paranthesis and the Solution draw by Professor did not include any Paranthesis.
Priority: N/A
Ana L. Milanova - 16:25
A: Correct, because our grammar was simplified and we did not have a parenthesized expression. So a factor can only be an id, and introducing a Factor nonterminal was redundant.

ANON - 15:56
Q: Only the quizzes are in groups, correct?
Priority: N/A
Konstantin Kuzmin - 15:59
A: Correct. And possibly a homework assignment if it is explicitly tagged as a Team assignment.
Ana L. Milanova - 16:26
A: Correct. There might be some group homework but we will make this explicit in the homework assignment if this is the case.

ANON - 15:57
Q: when will the hw be posted?
Priority: N/A

ANON - 15:58
Q: never mind i see it now
Priority: N/A

ANON - 15:58
Q: The quizzes are at the start of lecture time, are on Submitty, and are collaborative, right? How much time do we have to complete them?
Priority: N/A
Konstantin Kuzmin - 16:00
A: Correct, correct, and correct. We will announce the time limit additionally. Some time before the first quiz, of course.

ANON - 15:59

Q: So the Context Free Grammar is constructed based on the Input String such as "id * id ^ id * id" and not in general terms?

Priority: N/A

Steven Haussmann - 16:03

A: A grammar is a set of rules. You use the rules to construct valid strings that make up a language -- you might have the terms switched around.

ANON - 16:02

Q: What is id referring to in "factor-→ id|number|-factor|(expr)"?

Priority: N/A

Richard Massimilla - 16:05

A: id is a terminal. In this context, it'd make sense to assume it represents an identifier. However, note that it doesn't require a concrete definition and is simply a terminal

---

ANON - 16:02

Q: yes

Priority: N/A

ANON - 16:02

Q: yes

Priority: N/A

ANON - 16:05

Q: in this example, parentheses have even higher precedence than *, right?

Priority: N/A

Richard Massimilla - 16:06

A: That is correct

ANON - 16:07

Q: Regarding this question: So the Context Free Grammar is constructed based on the Input String such as "id * id ^ id * id" and not in general terms?

Priority: N/A

Ana L. Milanova - 16:29

A: The CFG is a set of rules that can generate strings such as id*id^id*id. A derivation derives the string from the rules by repeatedly applying productions. The parse takes a string, such as id*id^id*id, and either builds a parse tree according to the rules of the CFG, or it rejects the string because it cannot be generated by the CFG rules.

ANON - 16:07
Q: So higher precedence = lower in the parse tree?
Priority: N/A
Richard Massimilla - 16:14
A: Yes, higher precedence operations will be evaluated further from the root of the tree

ANON - 16:07
Q: So the higher up the operation should be in the tree, the lower its precedence in the order of operations?
Priority: N/A
Richard Massimilla - 16:14
A: Yes, lower precedence operations will be evaluated closer to the root of the tree

ANON - 16:09
Q: Why does the second "expr" have to be term again?
Priority: N/A
Steven Haussmann - 16:09
A: It enforces ordering by giving you a left hand side and a right hand side.
Ana L. Milanova - 16:32
A: Because we want to generate expressions that are just products: e.g., id*id*id. That second production is the "base case" in the recursive rules for expo. So expr => term => term * id => term * id * id => id * id * id. The application of expr -> term production terminates the derivation.

ANON - 16:09
Q: So if we want right associativitiy, we would make term the left operand?
Priority: N/A
Richard Massimilla - 16:12
A: In 'expr -> term ^ expr | term' we have right associativity because expr is to the right of term in 'term^expr'

ANON - 16:10
Q: For the Final Expression, based on the Grammar we did it seems that you would not be able to have expressions with more than one *? I do not think this was what we intended, and is why I think we might need the Factor Non-Terminal.
Priority: N/A
Ana L. Milanova - 16:35
A: If we have term -> term * id | id, then id*id*id is broken

into term*id, then term is broken into term*id and finally term goes to id.

ANON - 16:10
Q: Also, I am sorry because I sent a question twice by mistake.
Priority: N/A

ANON - 16:11
Q: Can you explain what this sentence means in the homework? "Your description should be a high level characterization — one that would still make
sense if we were using a different regular expression for the same language."
Priority: N/A

> Steven Haussmann - 16:15
> A: You should avoid describing the exact contents of the regular expression. Describe (AB)* as "all strings made up of any number of A's and B's in any order", rather than "repetition of either A or B"
>
> Steven Haussmann - 16:16
> A: Even if I found another regex to describe that language, the explanation would still make sense
>
> Ana L. Milanova - 16:36
> A: There is a note in the HW right after that sentence that should explain it with an example.

ANON - 16:13
Q: Could you speak a little bit about the logistics of the quiz next week?
Priority: N/A

> Konstantin Kuzmin - 16:15
> A: There is no quiz next week. Our first quiz is 9/11. We will post more details closer to the first quiz date.
>
> Konstantin Kuzmin - 16:15
> A: https://www.cs.rpi.edu/~milanova/csci4430/schedule.html
>
> Konstantin Kuzmin - 16:16
> A: Oops, sorry, it is next week. :)
>
> Konstantin Kuzmin - 16:17
> A: In any case, we will post more details closer to the quiz date.

ANON - 16:17
Q: why do we make a preference of right or left associativity for some operands? is it because right associativity takes higher precedence
Priority: N/A

> Richard Massimilla - 16:20

A: No, it is because a choice must be made to avoid ambiguity. Especially for non-commutative operators like exponents where (x^y)^z yields a different result mathematically than x^(y^z)

ANON - 16:20
Q: How do we know something need sot be right or left associative?
Priority: N/A
Ana L. Milanova - 16:40
A: For arithmetic operators, precedence and associativity is established by longstanding convention. E.g., * has higher precedence than + or - and both are left-associative, I.e., they are evaluated from left to right. It just has been this way, and we follow it. If we didn't have conventional rules for precedence and associativity then we would have to resolve ambiguity with tons of parentheses.

ANON - 16:21
Q: What would the rest of the grammar look like?
Priority: N/A
Ana L. Milanova - 16:41
A: Not clear what grammar.

ANON - 16:23
Q: Are you currently holding office hours?
Priority: N/A
Ana L. Milanova - 16:41
A: If I figure out how to start the office hour queue.