Q&A Session for Programming Languages Lecture 3

Session Number: 1208941292
Date: 2020-9-11
Starting time: 14:24

ANON - 14:23
Q: Will you be discussing how the quiz will be working?
Priority: N/A
        Konstantin Kuzmin - 14:25
        A: It should be pretty self-explanatory, if you have used Submitty before. If you need techincal assistance, just ask us here.

ANON - 14:23
Q: How much time do we have for the quiz again?
Priority: N/A
        Konstantin Kuzmin - 14:24
        A: We expect you to be done by 3 pm but you can submit until 4:19:59 pm with no penalty.

ANON - 14:32
Q: submitty seems to be down for me
Priority: N/A
        Konstantin Kuzmin - 14:33
        A: It is just slow with hundreds of students trying to load the same page at once... Just give it a minute or two or try refreshing the page.

ANON - 14:33
Q: I clicked refresh on submitty at 2:30 and it is still buffering. My wifi is fine. Is anyone else having the same problems?
Priority: N/A
        Konstantin Kuzmin - 14:34
        A: Try refreshing the page...

ANON - 14:33
Q: If we're working in a quiz group, does everyone have to submit a quiz gradeable or just one person?
Priority: N/A
        Konstantin Kuzmin - 14:33
        A: Everyone. Answers do not have to be the same. They will be graded independently.

ANON - 14:46
Q: in question 4 is it saying n needs to be the same for each a,b,c or can they be different n?
Priority: N/A

Steven Haussmann - 14:51
A: All variables with the same name are equivalent.

ANON - 14:55
Q: Would a superset be considered as "generating" a language?
Priority: N/A

Ana L. Milanova - 16:42
A: Oh this is from the quiz. No.

ANON - 14:55
Q: Do we only get one submission?
Priority: N/A

Konstantin Kuzmin - 14:57
A: You can submit multiple times.

ANON - 14:58
Q: for Q4, when you ask if something generates the language C, do you mean strictly just C? so if it generates B, but C is in B, it doesnt count right?
Priority: N/A

Steven Haussmann - 15:01
A: It must generate every string from the language, and nothing more than that.

ANON - 14:58
Q: I'm having some trouble understanding how to approach problem 3 from the homework. Would it be possible (if time permits after the lecture recording), to go over a small example or operator precedence and right-associativity?
Priority: N/A

Ana L. Milanova - 15:06
A: Yes.

Ana L. Milanova - 16:43
A: Unfortunately, we ran out of time in class. Stop by our office hours.

ANON - 15:02
Q: So for the name do we put our RCS id, what is the name that submitty knows us by?
Priority: N/A

Richard Massimilla - 15:04

A: Put in your first and last name. You can see what name Submitty has for you by going to 'My Profile' on the Submitty sidebar.

Konstantin Kuzmin - 15:06

A: It is the name/preferred name that you chose in your Submitty profile. You can quickly check what it is directly from the quiz if you expand the "Show Details" for Test 14 and look at the "name".

ANON - 15:02

Q: Is it an issue if I get this when submitting: Something went wrong when grading this submission. Please contact your instructor about this.

Priority: N/A

Konstantin Kuzmin - 15:12

A: Just refresh the page. It should go away.

ANON - 15:03

Q: Nevermind, just had to refresh

Priority: N/A

Konstantin Kuzmin - 15:13

A: You are correct!

ANON - 15:11

Q: Quick question about the quiz. For question 4, are we referring to context-free grammar?

Priority: N/A

Steven Haussmann - 15:13

A: Yes, any grammar defined by a series of replacement rules for single non-terminal characters will be a context-free one.

ANON - 15:13

Q: does anyone know where this "check 14 honor pledge" is?

Priority: N/A

Konstantin Kuzmin - 15:34

A: Scroll all the way down the Quiz page. You will see "Test 14 Check Honor Pledge". Click "Show Details" next to it.

ANON - 15:13

Q: Can you repeat what NFA stands for?

Priority: N/A

Ana L. Milanova - 15:14

A: NFA = Nondeterministic Finite Automaton

Steven Haussmann - 15:15

A: the important thing about an NFA is that there can be more than one move for a given symbol, making it non-deterministic

ANON - 15:14
Q: So the Scanner emits tokens to the Parser but if the token is a letter it will try to see if those letters are a keyword? And after emitting tokens it resets the DFA back to Start State?
Priority: N/A
Ana L. Milanova - 15:16
A: Yes, it has this "hack" in the DFA that checks whether identifiers are keywords. And yes.

ANON - 15:16
Q: When will we be able to see our grade on the quiz?
Priority: N/A
Ana L. Milanova - 15:17
A: By next class on Tuesday. Or maybe earlier.

ANON - 15:17
Q: I'm not understanding why we have to push back the current character to the Input Stream. Can you kindly explain this?
Priority: N/A
Ana L. Milanova - 15:30
A: Because we have consumed a character that is part of the next token in the stream.

ANON - 15:18
Q: Would we be expected to draw a table scanner for the exam?
Priority: N/A

ANON - 15:19
Q: Can prof Milanova go over a short but detailed example of table driven scanning?
Priority: N/A
Ana L. Milanova - 15:31
A: Yes, I will go over again at the end of class. But generally, we focus a lot more on parsing than on scanning in the class.

ANON - 15:20
Q: Question 4 agian. Not sure how I should be reading a^n b^n c^n. Is it correct to assume that for instance a^n for n = 0 is 1? Not sure how that works.
Priority: N/A
Richard Massimilla - 15:26
A: Yes. An expression like a^n is not an exponential operation

like you would see in arithmetic. It is shorthand for how many a's are repeated in succession. So, for n=0, there would be no a's, hence an empty string.

ANON - 15:20
Q: why I write my name, but still get -12 on honor pledge
Priority: N/A
Konstantin Kuzmin - 15:35
A: I do not see any penalties on your quiz.

ANON - 15:20
Q: I do not understand how a Scanner is generated automatically and no manual code is made. Can this be elaborated?
Priority: N/A
Ana Milanova - ?:?
A: Because all steps involved are well known in automata theory, they can be coded into the scanner generator. We do need to manually write the regular expressions that specify our tokens.

ANON - 15:21
Q: could we assume that 1 (for n = 0) is an empty string?
Priority: N/A
Richard Massimilla - 15:26
A: Yes. An expression like a^n is not an exponential operation like you would see in arithmetic. It is shorthand for how many a's are repeated in succession. So, for n=0, there would be no a's, hence an empty string.

ANON - 15:22
Q: What is the difference between the scanner tbale and the token table?
Priority: N/A
Ana Milanova - ?:?
A: The scanner table encodes the actions: move, recognize, or error. The token table encodes the token in case of recognize action.

ANON - 15:22
Q: Can you explain again how a scanner table work
Priority: N/A
Ana Milanova - ?:?
A: This question comes up a lot. We can go over at some point if we have time. We won't have scanning on HW.

ANON - 15:24

```
Q: yep thanks
Priority: N/A
        Konstantin Kuzmin - 15:36
        A: Anytime!
```

```
ANON - 15:25
Q: What does "action of" mean in the table driven scanning definition
slide?
Priority: N/A
        Ana Milanova - ?:?
        A: Action is one of Move, Recognize, or Error. We'll go over
the Scanner later in more detail if we have time, but generally there
is more emphasis on parsing than scanning in the class.
```

```
ANON - 15:25
Q: I've noticed that I can only see the slides when I'm using windows.
I just want to make sure, is it OK if I watch lectures with my friend
on his laptop on a TV in our shared living room (And not attend webex
on my account?)
Priority: N/A
        Ana L. Milanova - 15:32
        A: Yes sure whatever works.
        Konstantin Kuzmin - 15:39
        A: ^ Just make sure that starting next week you complete one
of the activities that count as "participation" at least once a week.
```

```
ANON - 15:26
Q: What does the "..." represent under "move:" in the table-driven
scanning algorithm on slide 9?
Priority: N/A
        Steven Haussmann - 15:32
        A: I believe it just indicates that there are more tokens
after id and +; they just aren't being shown here
```

```
ANON - 15:31
Q: As the paser is being fed the tokens, how does it know where to put
them in the parse tree?
Priority: N/A
        Ana L. Milanova - 15:33
        A: That
        Ana L. Milanova - 15:33
        A: That's what we'll see in lectures today and on Tuesday. The
parser is driven by a context-free-grammar and the grammar determines
how the parser puts tokens on the tree.
```

ANON - 15:33
Q: So in the case of Scanning what is the Recognize Step? I'm still not sure why the current character needs to be placed back on Input Stream. That is, how do we now it is needed for the next token based on this Step?
Priority: N/A

Ana L. Milanova - 15:36
A: Will go back to that slide at the end of class.

Ana L. Milanova - 15:39
A: This is how the recognize step works: suppose the scanner is in some final state, say state 2, and suppose it sees any of the valid characters (space, *, + digit, or letter). The scanner emits the token "times" (taken from the token_tab on the right). As it has read (i.e., consumed) the next character, it has to push it back. When the scanner resets to state 0 it will read that character again and recognize the corresponding token. We can go in more detail on scanning if we have time later in the class. But our focus is more on parsing than scanning.

Ana L. Milanova - 15:42
A: Sorry, meant the scanner, not the parser.

ANON - 15:34
Q: What does LL and LR stand for again?
Priority: N/A

Alex Mankowski - 15:38
A: The first L's refer to 'left-to-right' scanning of input. The second letters refer to either 'left-most' or 'right-most' derivation.

Steven Haussmann - 15:39
A: LL is Left-to-right, Leftmost derivation; LR is Left-to-right, Rightmost derivation

---

ANON - 15:38
Q: What are the advantages and disadvtanges of top-down parsing vs bottom-up parsing?
Priority: N/A

Ana L. Milanova - 15:41
A: More on this to come. Top-down parsing is simpler. We can easily code top-down parsers. Bottom-up parsing is more complex and more powerful, can parse more languages.

ANON - 15:42
Q: Why is bottom up called "rightmost in reverse" ? (Last slide)
Priority: N/A

Steven Haussmann - 15:45
A: I'll want Milanova to weigh in, but I believe it's because you're constructing the deepest, left-most side of the parse tree

first; that's the last step of a right-most derivation.
        Ana Milanova - ?:?
        A: The way the parser constructs the parse tree corresponds to a rightmost derivation in reverse. What Steven says: the first (reduction) step that the parser takes corresponds to the last step in the rightmost derivation because we are going from left to right in the input. And the last reduction step it takes corresponds to the first step in the rightmost derivation. (We won't be covering bottom up parsing unfortunately.)

ANON - 15:43
Q: there were two terms - leftmost.. terminal(?) and look-ahead..terminal.
Priority: N/A
        Ana L. Milanova - 15:51
        A: Part 3 will actually answer those questions!

ANON - 15:47
Q: bottom up
Priority: N/A

ANON - 15:47
Q: slide 18
Priority: N/A

ANON - 15:47
Q: The lookahead terminal question is referring to slide 17 i think though
Priority: N/A

ANON - 15:49
Q: Would a true right-most parse tree be top to bottom then?
Priority: N/A
        Steven Haussmann - 15:52
        A: You're parsing something that was derived in a right-most manner; the order in which you construct the parse tree isn't really important. I suppose you could have a right-to-left parser that scans backwards...

ANON - 15:50
Q: can you turn your microphone off?
Priority: N/A
        Ana L. Milanova - 15:51
        A: Sorry!

```
ANON - 15:50
Q: Please mute
Priority: N/A
        Konstantin Kuzmin - 15:58
        A: Done.
```

```
ANON - 15:50
Q: Anyone else getting super bad echo?
Priority: N/A
        Konstantin Kuzmin - 15:58
        A: Should be fixed now.
```

```
ANON - 15:52
Q: Those hidden cases should be ignored correct?
Priority: N/A
        Konstantin Kuzmin - 15:52
        A: Yes, they are for internal scoring purposes.
```

```
ANON - 15:52
Q: What is meant by production on slide 20?
Priority: N/A
        Ana L. Milanova - 15:53
        A: E.g., list -> id list_tail is a production. We pick a
production and expand the parse tree accordingly.
```

```
ANON - 15:55
Q: i missed what LL stands for in LL(k).  Lookahead Length?
Priority: N/A
        Ana L. Milanova - 15:56
        A: LL = first L: left-to-right scan of the input, second L:
leftmost derivation. k is lookahead length.
```

```
ANON - 15:55
Q: For the extra credit question, does single number means a single
digit number?
Priority: N/A
        Konstantin Kuzmin - 15:57
        A: No, just one number with no whitespace. Can be one digit or
multiple digits. Like "6", "105" or "17" (with no quotes, of course).
```

```
ANON - 15:58
Q: Would RR(1) not allow for predictive parsing?
```

Priority: N/A
Ana L. Milanova - 16:01
A: I don't think there is RR, at least not in compilers. I think you might mean LR(1). LR is a larger set than LL, and most LR grammars do not admit predictive parsing.

ANON - 16:00
Q: Is prediction just "which element in the union to pick next"?
Priority: N/A
Ana L. Milanova - 16:04
A: Yes essentially. Prediction is: which production, in case there is more than one for a given non-terminal, to pick next.

ANON - 16:01
Q: I want to rewatch this part of lecture later, but I don't see it on mediasite, will this be uploaded?
Priority: N/A
Ana L. Milanova - 16:05
A: Mediasite tells me it's visible and available to everyone... But Mediasite does not always behave, we'll check.

ANON - 16:05
Q: can you post the parse tree for this slide?
Priority: N/A
Ana L. Milanova - 16:18
A: Will do.

ANON - 16:06
Q: Do leftmost derivations always create parse trees that recurse to the right?
Priority: N/A
Ana L. Milanova - 16:18
A: Not always. Depends on the grammar.

ANON - 16:07
Q: Just to follow up, I don't see part 4 on mediasite either
Priority: N/A
Konstantin Kuzmin - 16:13
A: Please check Page 2 on your Mediasite.

ANON - 16:07
Q: To the person who cannot see the video on mediasite, maybe they need to go to the next page of videos
Priority: N/A

Konstantin Kuzmin - 16:13
A: Thank you, it does appear to be the solution exactly.

ANON - 16:07
Q: Yes parts 3 and 4 are not available on Mediasite
Priority: N/A
Konstantin Kuzmin - 16:09
A: Check Page 2 please.

ANON - 16:10
Q: Sorry! I can see them on page 2
Priority: N/A
Konstantin Kuzmin - 16:10
A: Great!

ANON - 16:10
Q: mute your mic please
Priority: N/A
Konstantin Kuzmin - 16:12
A: I muted Prof. Milanova, so it should be good now.

ANON - 16:18
Q: Why are we verifying that the next token is + or $$ instead of just accepting empty string and allowing the next token to throw the ParseError
Priority: N/A
Ana L. Milanova - 16:21
A: If the next token is + or $$ there won't be a parse error, at least not yet. If next token is +, then we are expanding by + term term_tail. If the next token is $$, then we are at the end of input, so accept empty string and we're done. If next token is anything else, then here's where we have to throw the parse error. (I might be misinterpreting the question, I wish we could have a follow up.)

ANON - 16:20
Q: Why does (term_tail, *) in Table not have Epislon like (factor_tail, +)?
Priority: N/A
Ana L. Milanova - 16:23
A: More on this next time. A term_tail is something like this: + term + term + ... term. So a term_tail always begins with +, or we are at the end of the string and the term_tail is followed by $$.
Ana L. Milanova - 16:25
A: A term_tail cannot be followed by *, just by $$. We put \epsilon only when we want to "get rid" of the nonterminal in the

tree. This should be more clear on Tuesday!

ANON - 16:25
Q: So the reason factor_tail, + predicts Epsilon is because that means we have reached the end of a factor and now are dealing with a new term. Whereas, in the case of term, if you see *, this should not happen so we shoud thrown an ERROR.
Priority: N/A

Richard Massimilla - 16:38
A: That is correct. Due to * having higher precedence than + in the grammar we've defined, it should be impossible for a factor_tail to result in anything other than an empty string when it is followed by a +.

Richard Massimilla - 16:39
A: Similarly, it should be impossible for term to ever be followed by a * at all given the rules of our grammar

---

ANON - 16:25
Q: Apart from the text book, what other resource do you recommend for practicing these concepts?
Priority: N/A

ANON - 16:26
Q: I'm not understanding why we have to push back the current character to the Input Stream. Can you kindly explain this? Just re-posting because answer was not covered at end of Lecture.
Priority: N/A

Steven Haussmann - 16:27
A: If that's what I think it is, it's because we were just looking ahead at the next character, and not actually consuming it yet

Ana L. Milanova - 16:29
A: This is the Scanner question. The way our ad-hoc scanner works, it reads (i.e., consumes the next character from the input stream).

Ana L. Milanova - 16:33
A: Then if the scan table entails a "recognize" action, that means we've scanned the current token fully and we can emit it to the parser and the last read (i.e., consumed) character is part of the next token. But we have consumed the character that is part of the next token, so we'll need to push it back, so when the scanner DFA resets

Ana L. Milanova - 16:34
A: to the start state, it can start scanning the next token from that character.

ANON - 16:27

Q: why isn't list its own column? (slide 32)
Priority: N/A

Richard Massimilla - 16:32

A: The columns are for terminals, the rows are for nonterminals. List has its own row in that table because it is a nonterminal.

ANON - 16:28
Q: So in the case of Scanning what is the Recognize Step? I'm still not sure why the current character needs to be placed back on Input Stream. That is, how do we now it is needed for the next token based on this Step? Just re-posting because not answered.
Priority: N/A

Ana L. Milanova - 16:38

A: The Recognize step happens when scanner is in a final state (of the DFA) and is ready to emit a token. E.g., states 2, 3, 4 in our simple automaton. Referring to slide 9. Suppose you are at state 3. State 3 is a final state. Upon seeing any other valid character, the scanner Recognizes the plus token. The Recognize step is encoded in the table as follows: look at current state 3 and the incoming character,

Ana L. Milanova - 16:39

A: if you have a - in the table and a token on the right, in our case plus, then the step is Recognize.

Ana L. Milanova - 16:40

A: But if you have a - and no token on the right, e.g., in state 1 on "other", then the action is an Error.

Ana L. Milanova - 16:41

A: And if you have a state in the table, e.g., on 1 and * you have state 2, that means Move action. Scanner/DFA is moving into state 2.

Ana Milanova - ?:?

A: We are not covering this in detail now, but I'll make a note to go back to scanning in the "Catch-up" class. Generally, there is more emphasis on parsing than scanning in the class.

ANON - 16:28
Q: I'm confused by the parsing table in general, it seems weird which boxes are filled and which arent
Priority: N/A

Steven Haussmann - 16:34

A: The table describes what can happen given the current non-terminal symbol and the terminal symbol being looked-ahead at. Empty squares represent situations with no valid production rule.

ANON - 16:30

Q: Apart from the text book, what other resource do you recommend for practicing these concepts? Seeing the concepts with examples would make them easier to understand, eg. generating strings (CFG), moving from ambiguous to unambiguous etc.
Priority: N/A

ANON - 16:31
Q: Oh, I think I see now. Is the parse tree saying "when I'm sitting at start, and encounter an id, I produce list $$" ?
Priority: N/A

> Ana L. Milanova - 16:35
> A: Yes.