To-Do

Lecture 7 covered derivations and briefly introduced $LL(k)$ grammars/parsers. Before the next lecture, do the following.

1. ***Wrap up W-1.*** Please be sure to keep up to date with piazza and post questions that you have (although it is pretty late now for questions).
2. ***See additional information below.*** Another commonly used notation for PL grammars is EBNF. See below for differences and links to example grammars.
3. ***Do the practice questions below.*** You must understand how to perform derivations and how to create parse trees. Do the practice questions below before the next lecture. Solutions are provided on the last page.

Additional Information

Extended Backus-Naur Form (EBNF). EBNF is another way to write context-free grammars, which is similar to BNF but adds additional operators. Non-terminals are written as names (without angle-brackets), terminals are written using double quotes (e.g., `"0"`), `=` is used instead of `::=`, and a bar is used for alternation (`|`). Instead of star, repitition of zero or more symbols is denoted using curly braces `{ ... }`. Optional items are enclosed in square brackets `[ ... ]` (e.g., $S \to \texttt{a} \mid \varepsilon$ is represented as `s = [ a ]`). There are many variants of both BNF and EBNF (e.g., instead of using `=`, `:` is sometimes used, sometimes commas are used for concatenation, and so on).

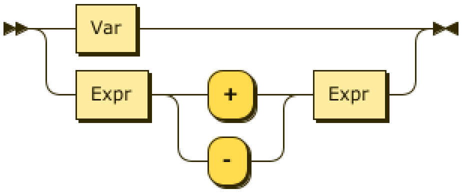
PL Syntax Examples. The following are examples of grammars that use variants of BNF/EBNF notation.

- Java: https://docs.oracle.com/javase/specs/jls/se21/html/jls-19.html
- Python: https://docs.python.org/3/reference/grammar.html
- C++ (summary): https://alx71hub.github.io/hcb/

Railroad Diagrams. "Railroad" (i.e., syntax) diagrams are a visual notation that are sometimes used for representing programming language syntax rules. More information can be found here: https://en.wikipedia.org/wiki/Syntax_diagram. For example, the BNF rule

```
Expr ::= Var | Expr ( '+' | '-' ) Expr
```

corresponds to this railroad diagram



You can play around with using BNF syntax and generating corresponding railroad diagrams on this website: https://bottlecaps.de/rr/ui.

Practice Questions

Provide the specified derivations for the following. Note that your derivations must be single-step, using the ⇒ notation from class. An answer key is provided on the last page.

1. Consider the grammar rule $S \rightarrow$ `ab` | `aa`S`bb`. Give a derivation of the string `"aaaaabbbbbb"`.
2. Consider the grammar rule $S \rightarrow$ `a` | `a`S | `a`S`b`. Give a derivation of the string `"aaaab"`.
3. Consider the grammar rule $S \rightarrow \varepsilon$ | `[`S`]` | SS. Give a left-most derivation of the string `"[[][[]]][]"`.
4. Consider the grammar $S \rightarrow$ `0` | `1` | S+S. Give two *different* left-most derivations of the string `"0 + 1 + 0"`.
5. Repeat Question 4 but give two different right-most derivations of the string.
6. Use the following grammar to give a left-most derivation of `"¬[true ∨ [false ∧ true]]"`.

   $S \rightarrow E$ | `[` S `]` | ¬ S

   $E \rightarrow$ `true` R | `false` R

   $R \rightarrow \vee\, S$ | $\wedge\, S$ | ε
7. Create a parse tree for the derivation of `"if true then 1 else 0"` using the following grammar.

   $S \rightarrow$ `if` B `then` V `else` V

   $B \rightarrow$ `true` | `false`

   $V \rightarrow$ `0` | `1`

Answer Key

1. $S \Rightarrow \texttt{aa}S\texttt{bb} \Rightarrow \texttt{aaaa}S\texttt{bbbb} \Rightarrow \texttt{aaaaabbbbb}$

2. $S \Rightarrow \texttt{a}S\texttt{b} \Rightarrow \texttt{aa}S\texttt{b} \Rightarrow \texttt{aaa}S\texttt{b} \Rightarrow \texttt{aaaab}$

3. $S \Rightarrow SS \Rightarrow \texttt{[}S\texttt{]}S \Rightarrow \texttt{[}SS\texttt{]}S \Rightarrow \texttt{[[}S\texttt{]}S\texttt{]}S \Rightarrow \texttt{[[]}S\texttt{]}S \Rightarrow \texttt{[[][}S\texttt{]]}S \Rightarrow \texttt{[[][[}S\texttt{]]]}S \Rightarrow \texttt{[[][[]]]}S \Rightarrow \texttt{[[][[]]][}S\texttt{]} \Rightarrow \texttt{[[][[]]][]}$

4. (a) $S \Rightarrow S\texttt{+}S \Rightarrow \texttt{0+}S \Rightarrow \texttt{0+}S\texttt{+}S \Rightarrow \texttt{0+1+}S \Rightarrow \texttt{0+1+0}$

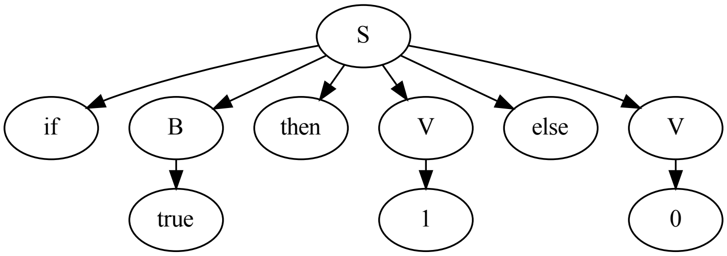
   (b) $S \Rightarrow S\texttt{+}S \Rightarrow S\texttt{+}S\texttt{+}S \Rightarrow \texttt{0+}S\texttt{+}S \Rightarrow \texttt{0+1+}S \Rightarrow \texttt{0+1+0}$

5. (a) $S \Rightarrow S\texttt{+}S \Rightarrow S\texttt{+0} \Rightarrow S\texttt{+}S\texttt{+0} \Rightarrow S\texttt{+1+0} \Rightarrow \texttt{0+1+0}$

   (b) $S \Rightarrow S\texttt{+}S \Rightarrow S\texttt{+}S\texttt{+}S \Rightarrow S\texttt{+}S\texttt{+0} \Rightarrow S\texttt{+1+0} \Rightarrow \texttt{0+1+0}$

6. $S \Rightarrow \neg S \Rightarrow \neg\texttt{[}S\texttt{]} \Rightarrow \neg\texttt{[}E\texttt{]} \Rightarrow \neg\texttt{[true}\,R\texttt{]} \Rightarrow \neg\texttt{[true} \lor S\texttt{]} \Rightarrow \neg\texttt{[true} \lor \texttt{[}S\texttt{]]} \Rightarrow \neg\texttt{[true} \lor \texttt{[}E\texttt{]]} \Rightarrow \neg\texttt{[true} \lor \texttt{[false}\,R\texttt{]]} \Rightarrow \neg\texttt{[true} \lor \texttt{[false} \land S\texttt{]]} \Rightarrow \neg\texttt{[true}\lor\texttt{[false} \land E\texttt{]]} \Rightarrow \neg\texttt{[true}\lor\texttt{[false} \land \texttt{true}\,R\texttt{]]} \Rightarrow \neg\texttt{[true}\lor\texttt{[false} \land \texttt{true]]}$

7.


S
if
B
then
V
else
V
true
1
0