

Name: _____

Sample Solutions

1. (1 pt.) [Instructions removed for sample solutions.]

Print your name clearly in the space provided above.

2. (4 pts.) Answer Questions 2–4 in the context of the PLY system as discussed in class.

Provide a code snippet that uses PLY's lexer to define two *literal tokens*: S and F.

(A)

```
from ply import lex
literals = ["S", "F"]
# maybe definitions for other tokens
lex.lex()
```

3. (5 pts.) Provide a code snippet that uses PLY's lexer to define a *non-literal token* DOW. The token should match one of the seven strings denoting days of the week: Sunday, Monday, ..., Saturday (and nothing else).

(A)

```
from ply import lex
tokens = ("DOW", )
t_DOW = r"Sunday|Monday|Tuesday|Wednesday|Thursday|Friday|Saturday"
# maybe definitions for other tokens
lex.lex()
```

4. (5 pts.) Redo Question 3 here to allow the DOW token to match the strings in that question even if the first letter of each string is not capitalized. (Provide the full answer here, including both the modified and unmodified parts.)

(A)

```
from ply import lex
tokens = ("DOW",)
t_DOW = r"""[Ss]unday|[Mm]onday|[Tt]uesday|[Ww]ednesday|
            [Tt]hursday|[Ff]riday|[Ss]aturday""
# maybe definitions for other tokens
lex.lex()
```

5. (5 pts.) Consider the following context-free grammar.

$$S \rightarrow h E L L o \quad (1)$$

$$E \rightarrow e \quad (2)$$

$$| E e \quad (3)$$

$$L \rightarrow y L z \quad (4)$$

$$| o \quad (5)$$

Ignore all white space. **Clearly state** whether the following *sentence* is *valid* (belongs to the language of the grammar).

h e o y o z o

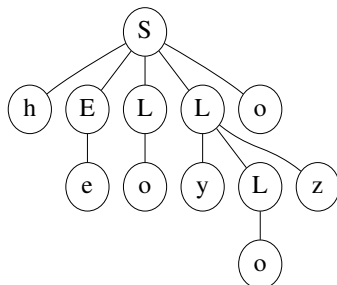
If it is valid then provide a *leftmost derivation* for it **using classroom conventions** (in particular, underlining replaced symbols and annotating arrows with rule numbers); else explain (as precisely as possible) why it is not valid. [Hint: It may be easier to answer Question 6 first.]

Ⓐ *It is valid. Leftmost derivation:*

$$\begin{aligned} \underline{S} &\xRightarrow{1} h \underline{E} L L o \\ &\xRightarrow{2} h e \underline{L} L o \\ &\xRightarrow{5} h e o \underline{L} o \\ &\xRightarrow{4} h e o y \underline{L} z o \\ &\xRightarrow{5} h e o y o z o \end{aligned}$$

6. (5 pts.) If the sentence of of Question 5 is not valid then make as small a change as possible to the sentence to yield a valid sentence (else use the unchanged sentence here). In either case, provide a *parse tree* for the (original or modified) sentence.

Ⓐ *No changes are needed since the original sentence is valid.*



7. (5 pts.) Repeat Question 5 using the following sentence instead of the one there:

h e y o o o z

- Ⓐ The sentence is not valid. Rule 4 of the grammar is the only one that generates a y (and z). Since the sentence has exactly one y (and z), Rule 4 must be used exactly once in any derivation of it. Doing so results in a subsequence $y L z$ in the sentential form. The desired sentence has three o terminals between the y and z . That means, this L must expand to exactly the substring ooo . However, the only way to get an o from L is Rule 5, and using that rule once leaves no additional nonterminals between the y and z , so no further terminals can be generated in that region. Therefore, the grammar cannot generate any sentence with more than one o between a y and the next z , such as the one in the question.
8. (5 pts.) Repeat Question 6 using the sentence of Question 7 instead of the one there.
- Ⓐ If exchanging two adjacent terminals is considered a single change, then we may swap the pair of adjacent characters $y o$ and similarly also the pair $o z$ to yield the sentence of Question 5 and the parse tree of Question 6. [There are several other reasonable options.]