Section 5 - Earley parser & Disambiguation

We're going to spend this section exploring ambiguity. We'll start with a small language that allows us to multiply and add the numbers 1, 2, and 3. It's not an especially useful language, but today we're super interested in the program 1 + 2 * 3, so it’s perfect for us! We want both operators to be left-associative, and we want multiplication to have higher precedence than addition.

Grammar 1

```
%left +
%left *
E -> E O E | V
O -> + | *
V -> 1 | 2 | 3
```

Is this grammar ambiguous?  
Please complete the earley parsing visualization for Grammar 1 on this string:  
1 + 2 * 3

---

Grammar 2

```
%left +
%left *
E -> E + E | E * E | V
V -> 1 | 2 | 3
```

Is this grammar ambiguous?  
Please complete the parsing visualization for Grammar 2 on this string:  
1 + 2 * 3

---

Grammar 3

```
E -> E + E2 | E2
E2 -> E2 * V | V
V -> 1 | 2 | 3
```

Is this grammar ambiguous?  
Please complete the parsing visualization for Grammar 3 on this string:  
1 + 2 * 3
Let’s play with if statements.

Grammar 4

S -> if E then S
  | if E then S else S
  | OTHERSTUFF

Is this grammar ambiguous?____________

Please draw a parse tree for Grammar 4 on this string:

if likeHorses then if likeFlying then print “Pegasus” else print “Mr. Ed”

Grammar 5

Using your answers above, complete grammar 5 to remove the problem.

Hint: We want to prevent: if(E1){if(E2){S3}}else{S4}.

Hint: We want all unmatched thens (thens without elses) to happen inside else cases.

S -> M # all thens matched!
  | U # some thens unmatched
M -> if ___ then ___ else ___
  | OTHERSTUFF
U -> if ___ then ___
  | if ___ then ___ else ___

Does this grammar accept the same language as Grammar 1?____________________

Please draw a parse tree for Grammar 5 on this string:

if likeHorses then if likeFlying then print “Pegasus” else print “Mr. Ed”

What is the issue with this language / parser?

Suggest a language specification that would resolve this issue?