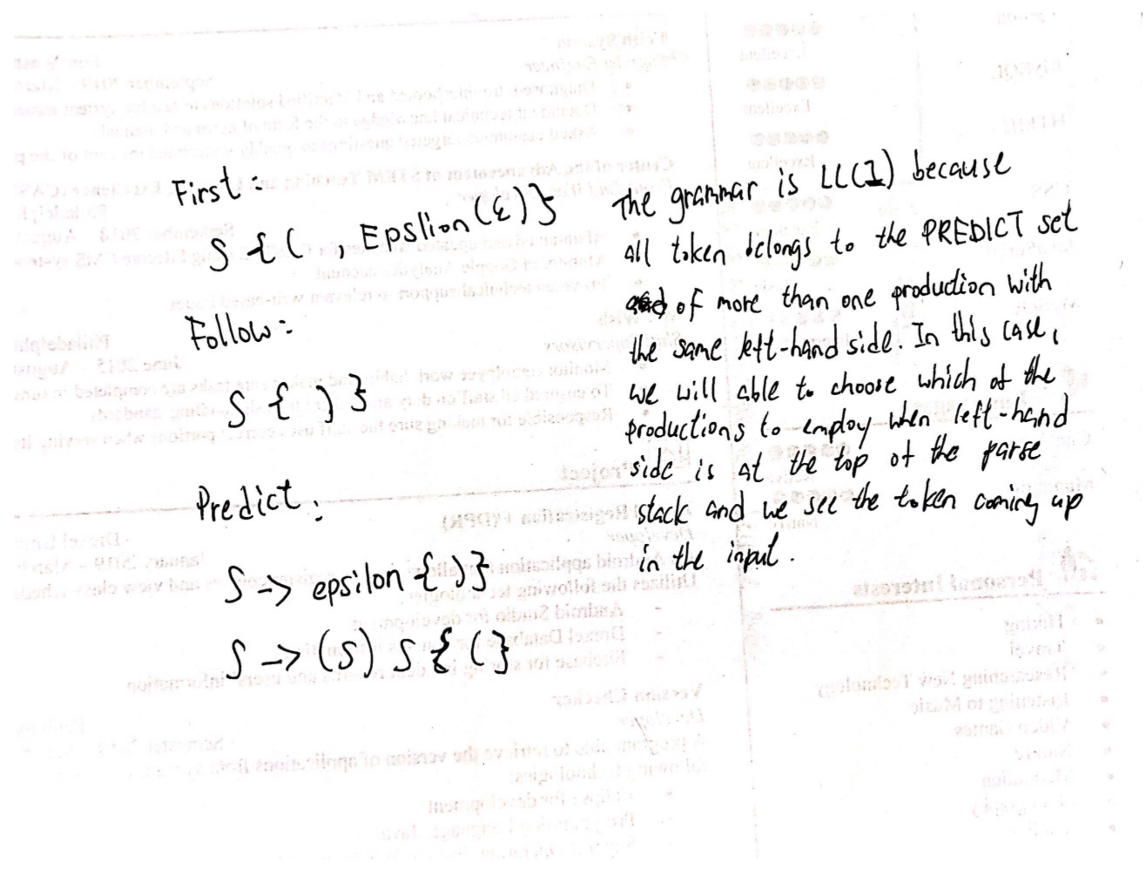
Man Tik Li
CS 360 - 002
Krzysztof Nowak
July 22, 2020

Quiz 2

1. Applicative order evaluates subexpressions when before the procedure is applied. When reducing using applicative order we reduce the function part and the argument part of the function to there smallest form before we substructure the simplest form lack into the function. The problem with applying applicative order here is that the receives of reductions here will never terminate like they would when sorting a normal order.

2. In figure 2.20, the first column top-of-stack being used in the predict set to every predict entry from left hand side. The remaining columns are used in the right-hand side for the predict set. Each non-terminal present in the left-hand side of the predict entries is unique. To construct this table, check all the attributes in "first's". The first set of entry given by the "first's" of left-hand side. If the epsilon (ϵ) pointing from left-hand side, add the current entry under the "follow's" of the left-hand side. For instance, the stmt_list in second row of predict set. It needs to include id, read, and write under the first's of the left-hand side and this provides why this entry is present under these three.

3.



First:

S { (, Epslion (ϵ) }

Follow:

S {) }

Predict:

S -> epsilon {) }

S -> (S) S { (}

The grammar is LL(1) because all token belongs to the PREDICT set ~~and~~ of more than one production with the same left-hand side. In this case, we will able to choose which of the productions to employ when left-hand side is at the top of the parse stack and we see the token coming up in the input.

Practice Problems:

2.9 – 3
2.11 – 2
2.13 - 2