ASSUMPTIONS:

This handout applies to LL(1) grammars G

- Upper case letters such as "A", "A'", "S" represent non-terminals of G.
- Lower-case letters such as "a" represent terminals of G.
- Greek letters such as "α", "β", "γ" represent strings of terminals and non-terminals of G.
- ε is the null string

LL(1) FIRST SETS

Definitions

- The **First set** of a non-terminal A is the set of all terminals that can begin a string derived from A. This set will include ε if the empty string can be derived from A.
- The **First set** of a string γ of terminals and non-terminals is the set of all terminals that can begin a string derived from γ. This set will include ε if the empty string can be derived from γ.

Algorithm to calculate First(γ) for a string γ of terminals and non-terminals

- If $\gamma = \varepsilon$, then First(γ) = $\{\varepsilon\}$
- If the first symbol in γ is a terminal a, then First(γ) = {a}
- If γ=Aγ' for some non-terminal A and (possibly empty) string γ'
  - If $\varepsilon \notin$First(A), then First(γ) = First(A)
  - If $\varepsilon \in$ First(A), then First(γ) = (First(A)- $\{\varepsilon\}$) $\cup$ First(γ')

Algorithm to calculate First Sets for all non-terminals of G

1. For each non-terminal A in G, Set First(A) = {}
2. Order all non-terminals of G such that for each production A→γ, γ cannot start with a non-terminal earlier in the list than A.

   (This can be done using algorithm to remove left-recursion in 3-6)
3. Reversing that order,
   For each rule A→γ, set First(A) = First(A) $\cup$ First(γ)

LL(1) FOLLOW SETS

Definition

- The **Follow set** of a non-terminal A is the set of all terminals that could appear right after A in a derivation. This set will include $ if end of file can follow A.

Algorithm to calculate Follow Sets for all non-terminals of G

1. For each non-terminal A in G,
   - Calculate First(A)
   - Set Follow(A) = {}
2. Separate productions with | into sets of productions.
3. As much as possible, order productions of the form $A \rightarrow \alpha A'\beta$ so that productions for A are before productions for A'
4. Following that order,

   For each rule $A \rightarrow \gamma$

   For each non-terminal A' in γ such that $\gamma = \alpha A'\beta$ i.e. $A \rightarrow \alpha A'\beta$
   - If $\beta=\varepsilon$ then Follow(A') = Follow(A') $\cup$ Follow(A)
   - Otherwise
     - Follow(A') = Follow(A') $\cup$ First(β)
     - If $\varepsilon \in$First(β), then Follow(A') = (Follow(A') – {ε}) $\cup$ Follow(A)
5. When one operation changes a set that another set depends on, recalculate. (Alternately, don't expand Follow sets until all Follow dependencies have been identified).

LL(1) PARSE TABLES

An LL(1) parse table for G is created as follows:

- The rows of the table are labeled with G's non-terminals
- The columns of the table are labeled with G's terminals
- Each entry of the table is either empty, or contains the rhs of a production:
  Look at each production $A \rightarrow \gamma$
  - $\forall a \in$First(γ) – {ε} Table(A,a) = γ
  - If $\varepsilon \in$First(γ), $\forall a \in$ Follow(A) Table(A,a) = γ