Assignement-3 on Bottom-Up Parsing
CSC 340
First term 2016-2017

Consider the following CFG describing a simple programming language.

A language with integers and integer operations

$$P \rightarrow D; P \mid D$$

$$D \rightarrow \text{def id(ARGS)} = E;$$

$$\text{ARGS} \rightarrow \text{id, ARGS} \mid \text{id}$$

$$E \rightarrow \text{int} \mid \text{id} \mid \text{if } E_1 = E_2 \text{ then } E_3 \text{ else } E_4 \mid E_1 + E_2 \mid E_1 - E_2 \mid \text{id}(E_1,\ldots,E_n)$$

In this assignment you are required to write a bottom-up SLR(1) parser for the language described above. You can use the compiler construction tools (hackingoff.com) to generate the necessary finite automata and tables. The input to your code must be provided by the lexical analyzer you wrote for assignment 2).

Deliverables

1) **The NFA for recognizing the viable prefixes**
2) **The corresponding DFA**
3) **Your code**
4) **Several samples of input output.**

Due date Tuesday 3 Jan 2017