CS 2210: Compiler Design

Homework #2

Due Time: midnight February 22nd, 2020

1. (20 points) Given the following grammar, construct the *first* and *follow* sets for each non- terminal symbol. (Upper case indicates non-terminals whereas lower case indicates terminals)

(a) A → BAc | FE
B → bEF | g
E→e | ε
F → f | EH
H→h

(b) S→ ACB | CbB | Ba
A→da | BC
B→g | ε
C→h | ε

2 (60 points). For each of the below grammars, answer the following questions:

(1) Is the grammar LR(0)? If so, draw a DFA and the corresponding parse table. If not, point out the conflict.
(2) Is the grammar SLR(1)? If so, draw a DFA and the corresponding parse table. If not, point out the conflict.
(3) Is the grammar LALR(1)? If so, show how the lookahead component resolves the conflict. If not, show the conflict state and explain why it is not resolved.
(4) Is the grammar CLR(1)? If so, show how state splitting resolves the conflict. If not, show the conflict state and explain why it is not resolved.

⋆ Note LR(0) ⊂ SLR(1) ⊂ LALR(1) ⊂ LR(1). So, (2) needs answering only if (1) is false. (3) needs answering only if both (1) and (2) are false. (4) needs answering only if (1), (2), and (3) are false. (Upper case indicates non-terminals whereas lower case indicates terminals)

(a) E → E+T | T
T→ TF | F
F→ F* | a | b

(b) S → Aa | bAc | dc | bda

A→d

(c) S → Aa | bAc | Bc | bBa
A→d
B→d

3. (20 points) Given the grammar below:

S→ AA
A→ aA | b

Build LALR(1) parsing table and draw the content change of both **input buffer** and **parsing stack** on each step when you perform a shift or reduce (like what we have done on white board during class and use your own numbering/label for states and reduction rules) for the following input string.

“abaab”

Specifically, when a shift is performed, use the strikethrough line on the input token. When a popup is performed, use strikethrough line in the stack. (Do not just leave an empty stack and empty input string!)