Problem 1 – LR(1) Parsing

1. Compute the canonical collection of sets of LR(1) items
2. Construct the LR(1) parse table (ACTION and GOTO)
3. Is the grammar LR(1) or not? Justify your answer.
4. If the grammar is LR(1), show the behavior of the LR(1) parser on input
   \[ *id = id \]
   i.e., show stack content, current input, and selected action for each move of the machine.

Problem 2 – LR(0)

Show that the above grammar (Problem 1) is not LR(0). Note that it is sufficient to show one state where there is a conflict (Hint: you don’t need to enumerate all states).
Problem 3 – Type Systems

Assume a type system with the following inference rules

Rule $A_1$ : \[ E ⊢ e_1 : integer \quad E ⊢ e_2 : integer \quad \frac{\quad}{E ⊢ (e_1 + e_2) : integer} \]

Rule $A_2$ : \[ E ⊢ e : α \quad \frac{\quad}{E ⊢ &e : pointer(α)} \]

Rule $A_3$ : \[ E ⊢ e : pointer(α) \quad \frac{\quad}{E ⊢ *e : α} \]

Assuming that variable $a$ and constant $3$ are of type integer, and variable $b$ is of type boolean. Use the inference rules to determine the types of the following expressions. Note: if a proof does not exist, the type system reports a type error.

1. $&a$
2. $&b$
3. $(&a + 5)$
4. $*a$
5. $&3$
6. $*(a + b)$
7. $&&a$