1. 1. first(H) = {h}

first(E) = {e, ε}

first(B) = {b, g}

first(F) = {f} + first(E) + first(H) = {f, e, h, ε}

first(A) = first(B) + first(F) = {b, g, f, e, h, ε}

follow(A) = {$, c}

follow(B) = first(A) = {b, g, f, e, h}

follow(E) = follow(A) + first(F) + first(H) = {$, c, f, e, h}

follow(F) = first(E) + follow(B) + follow(A) = {$, e, b, g, f, h, c}

follow(H) = follow(F) = {$, e, b, g, f, h, c}

* 1. first(C) = {h, ε}

first(B) = {g, ε}

first(A) = {d} + first(B) + first(C) = {d, g, h, ε}

first(S) = first(A) + first(C) + {b} + first(B) + {a} = {d, g, h, ε, b, a}

follow(S) = {$}

follow(A) = first(C) + first(B) + follow(S) = {h, g, $}

follow(B) = follow(S) + {a} + first(C) + follow{A} =w {$, a, h, g}

follow(C) = first(B) + follow(S) + {b} + follow(A) = {g, $, b, h, g}



**LR(0): A picture containing clock

Description automatically generated**

this grammar is not LR(0). As shown by the partial DFA, there occurs a shift-reduce conflict in state 5

**SLR(1):**

this grammar is SLR(1)A close up of a clock

Description automatically generated

1. E → E+T
2. E → T
3. T→ TF
4. T→ F
5. F→ F\*
6. F → a
7. F → b

Follow(E) = {+, $}

Follow(T) = {$, +, a, b}

Follow(F) = {\*, $, +, a, b}

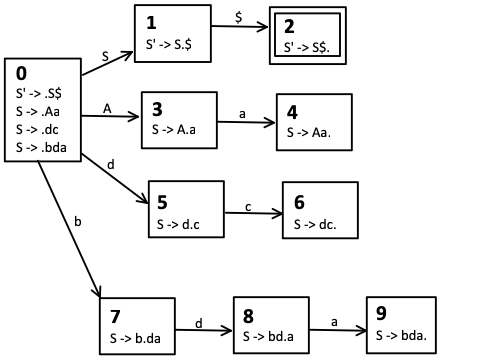
Action Goto

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | \* | + | a | b | $ | **E** | **T** | **F** |
| 1 |  |  | S9 | S7 |  | I2 | I10 | I9 |
| 2 |  | S3 |  |  | accept |  |  |  |
| 3 |  |  | S6 | S7 |  |  | IT |  |
| 4 |  | R1 | S6 | S7 | R1 |  |  | I5 |
| 5 | S8 | R3 | R3 | R3 | R3 |  |  |  |
| 6 | R6 | R6 | R6 | R6 | R6 |  |  |  |
| 7 | R7 | R7 | R7 | R7 | R7 |  |  |  |
| 8 | R5 | R5 | R5 | R5 | R5 |  |  |  |
| 9 | S8 | R4 | R4 | R4 | R4 |  |  |  |
| 10 |  | R2 | S6 | S7 | R2 |  |  | I9 |



**LR(0):**

the grammar is LR(0)



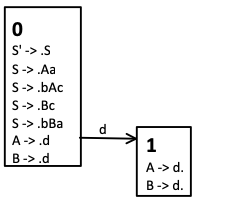
1. S→Aa
2. S-> bAc
3. S-> dc
4. S -> bda
5. A→d

Action Goto

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | $ | **S** | **A** |
| 0 |  | S7 |  | S5 |  | I1 | I3 |
| 1 |  |  |  |  | accept |  |  |
| 2 |  |  |  |  |  |  |  |
| 3 | S4 |  |  |  |  |  |  |
| 4 | R1 | R1 | R1 | R1 | R1 |  |  |
| 5 |  |  | S6 |  |  |  |  |
| 6 | R3 | R3 | R3 | R3 | R3 |  |  |
| 7 |  |  |  | S8 |  |  |  |
| 8 | S9 |  |  |  |  |  |  |
| 9 | R4 | R4 | R4 | R4 | R4 |  |  |



**LR(0):**

****

This grammar is not LR(0), in state 1 there is a reduce-reduce conflict. Thus LR(0) fails.

**SLR(1):**

This grammar is not LR(1) either, in the partial DFA of LR(0), state 1 has two reduce rules deriving from non-terminals A and B. If we check the follow set of non-terminals we will find out:

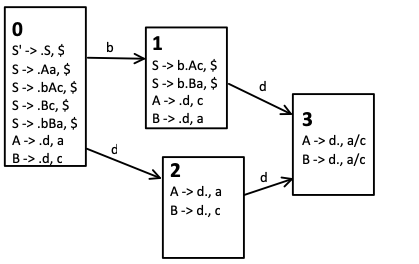
Follow(S) = {$}

Follow(A) = {a, c}

Follow(B) = {a, c}

among which the follow sets of A and B are identical. Thus, SLR(1) cannot resolve the reduce-reduce conflict in state 1.

**LALR(1):**



This grammar is not LALR(1) either, the rules derived from non-terminal A and B have different lookaheads in state 1 and 2. Yet in state 3, after being dynamically merged, these two rules face reduce-reduce conflict again because their sets of lookahead symbols are identical

**CLR(1)**

This grammar is CLR(1).

**A close up of a clock

Description automatically generated**

CLR(1) solves the reduce-reduce conflict in LALR(1) by splitting the old state 3 into two states with identical grammatical rules but different sets of lookaheads. Unlike in LALR(1), state 1 and 2 herein branch to two separates states, leading to more items than LR(0). Other branches in this CLR(1) DFA are all led to by one-symbol derivation and thus they shouldn’t have any conflicts. Therefore, this grammar is CLR(1).