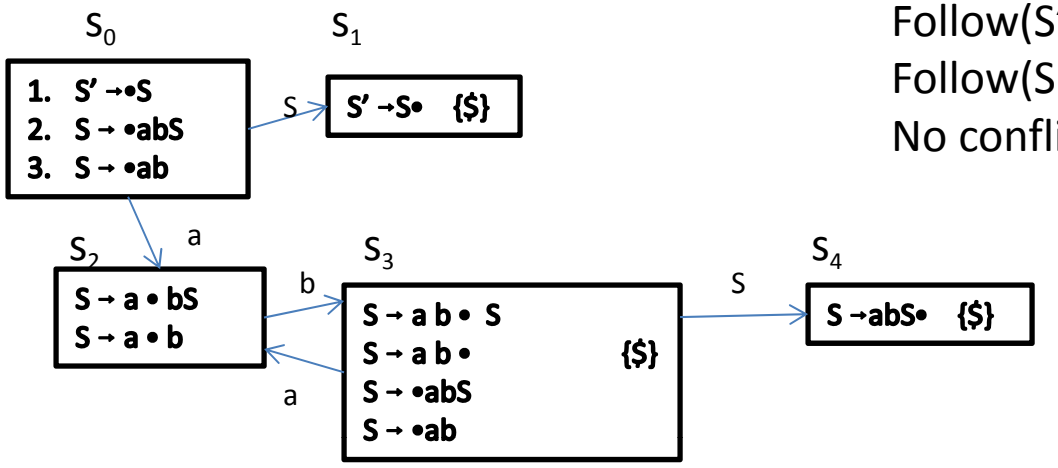
1. See the table below.

| Stack<br>(Bottom Top) | Input | Action |
|---|---|---|
| 0 | id*id+id*id$ | Initial state |
| 0 id 4 | *id+id*id$ | s4 |
| 0 F 3 | *id+id*id$ | r6 |
| 0 F 3 * 6 | Id+id*id$ | s6 |
| 0 F 3 * 6 id 4 | +id*id$ | s4 |
| 0 F 3 * 6 F 3 | +id*id$ | r6 |
| 0 F 3 * 6 T 8 | +id*id$ | r5 |
| 0 T 2 | +id*id$ | r4 |
| 0 T 2 + 5 | id*id$ | s5 |
| 0 T 2 + 5 id 4 | *id$ | s4 |
| 0 T 2 + 5 F 3 | *id$ | r6 |
| 0 T 2 + 5 F 3 * 6 | id$ | s6 |
| 0 T 2 + 5 F 3 * 6 id 4 | $ | s4 |
| 0 T 2 + 5 F 3 * 6 F 3 | $ | r6 |
| 0 T 2 + 5 F 3 * 6 T 8 | $ | r5 |
| 0 T 2 + 5 T 2 | $ | r4 |
| 0 T 2 + 5 E 7 | $ | r3 |
| 0 E 1 | $ | r2 |
| 0 E 1 | $ | accept |

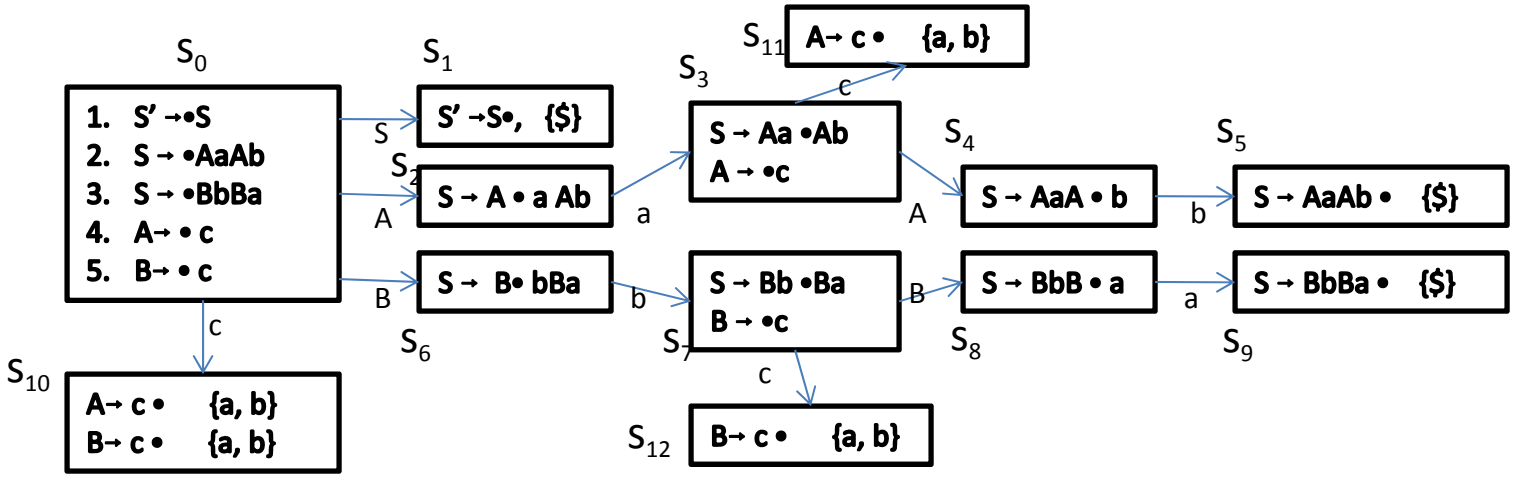
2. (a)




Follow(S') = {$}
Follow(S) = {$}
No conflict. It is SLR.

| | Action | | | GOTO |
|---|---|---|---|---|
| | a | b | $ | S |
| $S_0$ | s2 | | | 1 |
| $S_1$ | | | ! | |
| $S_2$ | | s3 | | |
| $S_3$ | s2 | | r3 | 4 |
| $S_4$ | | | r2 | |

2. (b)

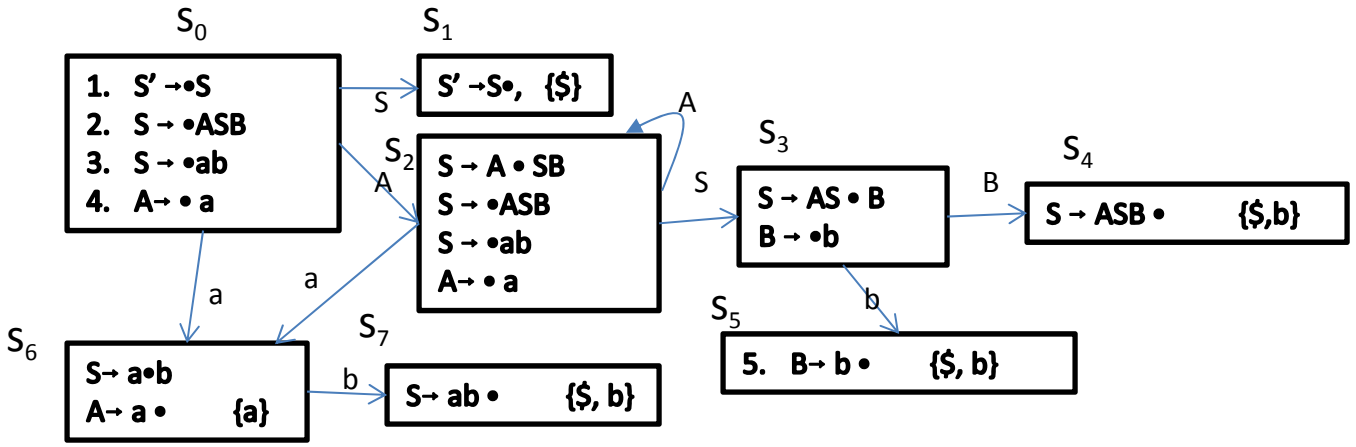



| | Action | | | | GOTO | | |
|---|---|---|---|---|---|---|---|
| | a | b | c | $ | S | A | B |
| $S_0$ | | | s10 | | 1 | 2 | 6 |
| $S_1$ | | | | ! | | | |
| $S_2$ | s3 | | | | | | |
| $S_3$ | | | s11 | | | 4 | |
| $S_4$ | | s5 | | | | | |
| $S_5$ | | | | r2 | | | |
| $S_6$ | | s7 | | | | | |
| $S_7$ | | | s12 | | | | 8 |
| $S_8$ | s9 | | | | | | |
| $S_9$ | | | | r3 | | | |
| $S_{10}$ | r4 or r5? conflict | r4 or r5? conflict | | | | | |
| $S_{11}$ | r4 | r4 | | | | | |
| $S_{12}$ | r5 | r5 | | | | | |

Follow(S') = {$}
Follow(S) = {$}
Follow(A) = {a, b}
Follow(B) = {a, b}

It is NOT SLR.
Reduce-reduce conflict at state $S_{10}$.

2. (c)




Follow(S') = {$}
Follow(S) = {$, b}
Follow(A) = {a}
Follow(B) = {$, b}

It is SLR. No conflicts.

| | Action | | | GOTO | | |
|---|---|---|---|---|---|---|
| | a | b | $ | S | A | B |
| $S_0$ | s6 | | | 1 | 2 | |
| $S_1$ | | | ! | | | |
| $S_2$ | s6 | | | 3 | 2 | |
| $S_3$ | | s5 | | | | 4 |
| $S_4$ | | r2 | r2 | | | |
| $S_5$ | | r5 | r5 | | | |
| $S_6$ | r4 | s7 | | | | |
| $S_7$ | | r3 | r3 | | | |

3. Problem 2.18(a)

First(Es) = { atom, ’ , (  // based on rule Es -> E Es, add first(E)
ε  // Es -> ε, textbook does not include ε in first sets
}

Follow(E) = { $$,  // P → E $$
// E → ’ E  add follow(E), not change
atom, ’, (, )  // E → ( E Es) add first(Es))
//  Es → E Es add first(Es) and follow(Es), no change.
}

Predict(Es → ε) = (first(ε)- ε) ∪ follow(Es) = { ) }
where follow(Es) = { )  // E → ( E Es)
}  // Es → E Es, not change

4. Follow symbol order B, T, C. The predict sets are calculated based on the first and follow sets. Note that there is no conflict for predict sets.

| | | | |
|---|---|---|---|
| B → B or T | | B → T B’ | predict={not, (, true, false} |
| \| T | Remove left recursion → | B’ → or T B’ | predict={or} |
| | | \| ε | predict={$, )} |

| | | | |
|---|---|---|---|
| T → T and C | | T → C T’ | predict={not, (, true, false} |
| \| C | Remove left recursion → | T’ → and C T’ | predict={and} |
| | | \| ε | predict={$, ), or} |

| | |
|---|---|
| C → not C | predict={not} |
| \| (B) | predict={(} |
| \| true | predict={true} |
| \| false | predict={false} |

| | First | Follow |
|---|---|---|
| B | {not, (, true, false} | {$, )} |
| B’ | {or, ε} | {$, )} |
| T | {not, (, true, false} | {$, ), or} |
| T’ | {and, ε} | {$, ), or} |
| C | {not, (, true, false} | {$, ), or, and} |

5.

```
S()
{
  if (match('+') || match('-'))
  {
    if (S())
      return T();
    else return Error;
  } else if (match('a'))
    return Ok;
 else
    return Error;
}
```

```
T()
{
  if (match('/') || match('*'))
     return S();
 else if (match('b'))
    return Ok;
 else
    return Error;
}
```