Parse Trees:

Input: (id + id) * id $

```
                    E
                    |
                    T
                 /  |  \
                T   *   F
               /         \
              F           id
            / | \
           ( E )
           / | \
          E + T
         /    / \
        T    F
       / \    \
      F      id
     /
    id
```

Input: id * id $

```
        E
        |
        T
      / | \
     T  *  F
    /       \
   F        id
  /
 id
```

CFG:
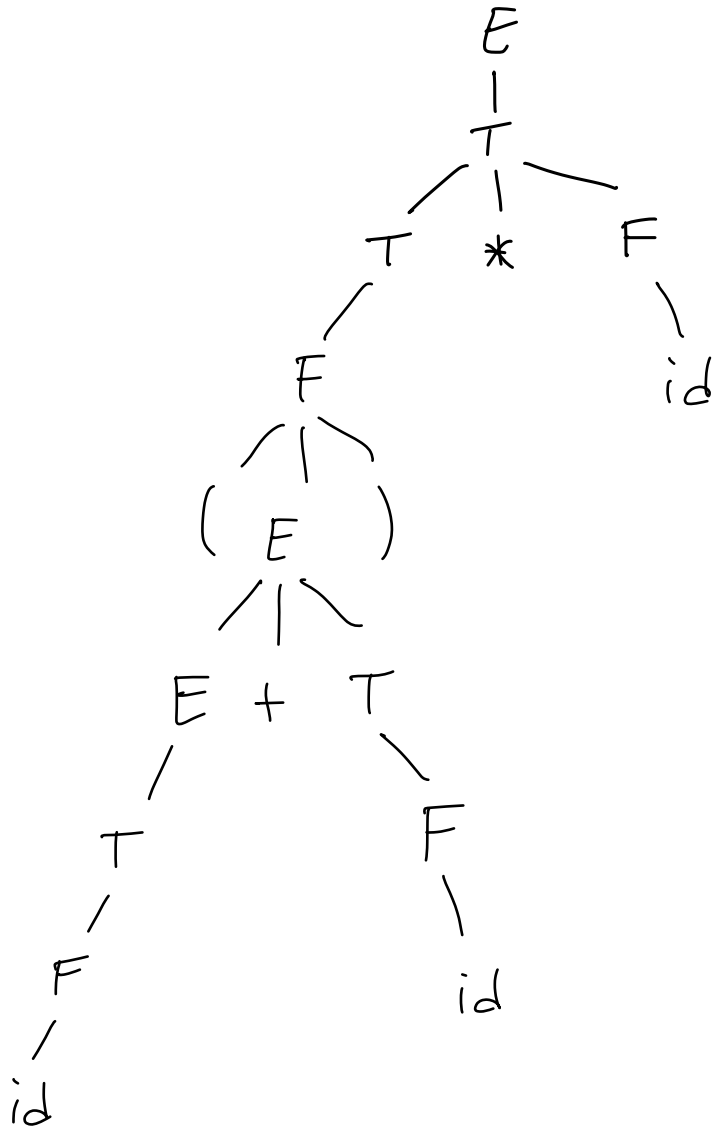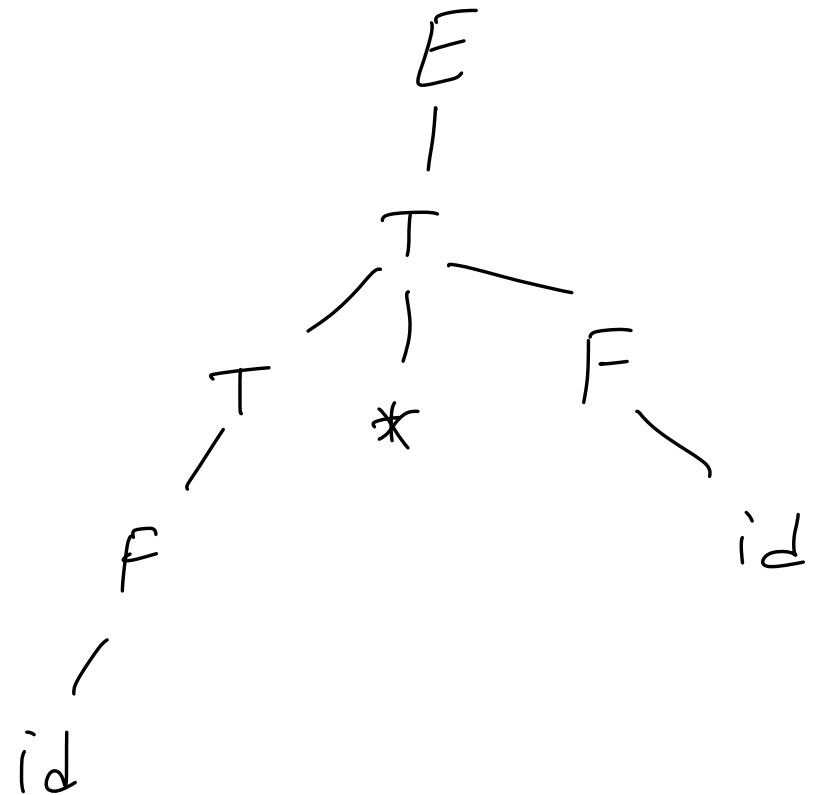
E → E + T ①
E → T ②
T → T * F ③
T → F ④
F → (E) ⑤
F → id ⑥

Input: (id + id) * id $

Parse Tree:
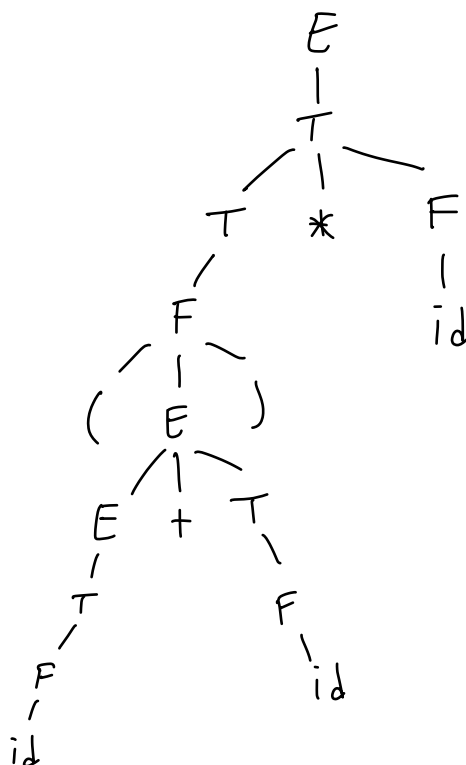
Stack	Input	action
$0	(id + id) * id $	
$0 (4	id + id) * id $	Shift 4
$0 (4 id 5	+ id) * id $	Shift 5
$0 (4 F 3	+ id) * id $	Reduce 6
$0 (4 T 2	+ id) * id $	Reduce 4
$0 (4 E 8	+ id) * id $	Reduce 2
$0 (4 E 8 + 6	id) * id $	Shift 6
$0 (4 E 8 + 6 id 5	) * id $	Shift 5
$0 (4 E 8 + 6 F 3	) * id $	Reduce 6
$0 (4 E 8 + 6 T 9	) * id $	Reduce 4
$0 (4 E 8	) * id $	Reduce 1
$0 (4 E 8) 11	* id $	Shift 11
$0 F 3	* id $	Reduce 5
$0 T 2	* id $	Reduce 4
$0 T 2 * 7	id $	Shift 7
$0 T 2 * 7 id 5	$	shift 5
$0 T 2 * 7 F 10	$	Reduce 6
$0 T 2	$	Reduce 3
$0 E 1	$	Reduce 2
$0 E 1	$	Accepted

Input: id * id $

CFG:
E → E + T ①
E → T ②
T → T * F ③
T → F ④
F → (E) ⑤
F → id ⑥

Parse Tree:

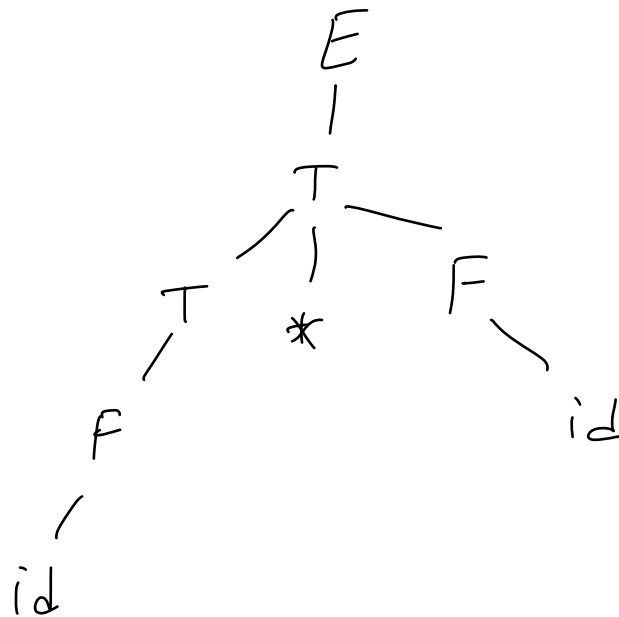| Stack	Input	action
$θ	id * id $	
$θid5	*id $	shift 5
$θF3	* id $	Reduce 6
$θT2	* id $	Reduce 4
$0T2*7	id $	Shift 7
$0T2*7id5	$	Shift 5
$0T2*7F10	$	Reduce 6
$0T2	$	Reduce 3
$0E1	$	Reduce 2
$0E1	$	Accepted

CFG:

$E \rightarrow E + T$ ①

$E \rightarrow T$ ②

$T \rightarrow T * F$ ③

$T \rightarrow F$ ④

$F \rightarrow (E)$ ⑤

$F \rightarrow id$ ⑥

Input: $(id *) $$

Stack	Input	action
$0	(id *) $	
$0(4	id * $	shift 4
$0(4id5	* $	shift 5
$0(4F3	* $	Reduce 6
$0(4T2	* $	Reduce 4
$0(4T2*7	$	Not accepted

Not Accepted!