Parsing id+id*id (Slide 11; See the alg. in Slide 8; grammar in Slide 9 and parse table in Slide 10).
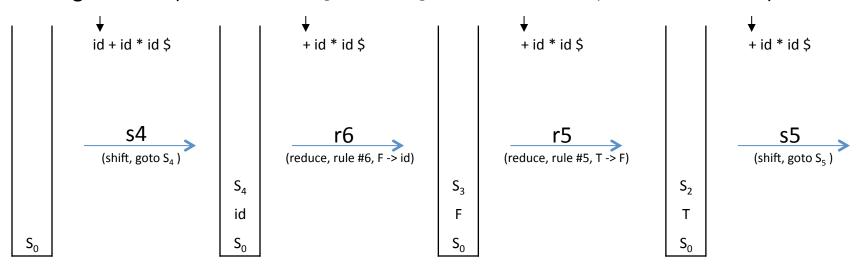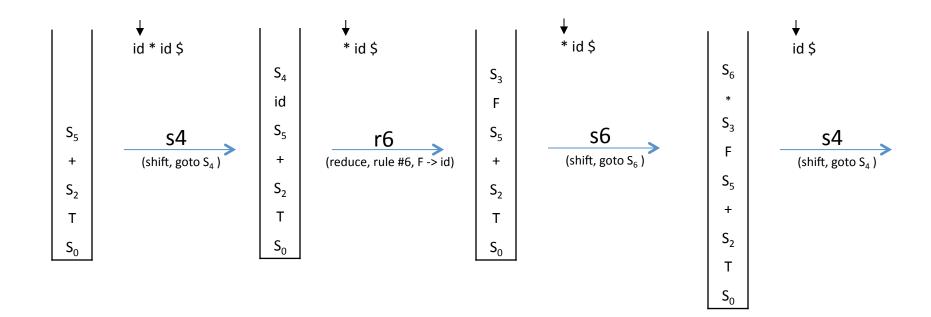
Row 1:

id + id * id $

$\xrightarrow{\text{s4}}$ (shift, goto S_4)

Stack: S_0

+ id * id $

$\xrightarrow{\text{r6}}$ (reduce, rule #6, F -> id)

Stack: S_4 / id / S_0

+ id * id $

$\xrightarrow{\text{r5}}$ (reduce, rule #5, T -> F)

Stack: S_3 / F / S_0

+ id * id $

$\xrightarrow{\text{s5}}$ (shift, goto S_5)

Stack: S_2 / T / S_0

Row 2:

id * id $

$\xrightarrow{\text{s4}}$ (shift, goto S_4)

Stack: S_5 / + / S_2 / T / S_0

---

* id $

$\xrightarrow{\text{r6}}$ (reduce, rule #6, F -> id)

Stack: S_4 / id / S_5 / + / S_2 / T / S_0

---

* id $

$\xrightarrow{\text{s6}}$ (shift, goto S_6)

Stack: S_3 / F / S_5 / + / S_2 / T / S_0

---

id $

$\xrightarrow{\text{s4}}$ (shift, goto S_4)

Stack: S_6 / * / S_3 / F / S_5 / + / S_2 / T / S_0

Parsing id+id*id (cont.)

Stack 1: S_4, id, S_6, *, S_3, F, S_5, +, S_2, T, S_0 — input: $

→ **r6** (reduce, rule #6, F -> id)

Stack 2: S_3, F, S_6, *, S_3, F, S_5, +, S_2, T, S_0 — input: $

→ **r5** (reduce, rule #5, T -> F)

Stack 3: S_8, T, S_6, *, S_3, F, S_5, +, S_2, T, S_0 — input: $

→ **r4** (reduce, rule #4, T -> F*T)

Stack 4: S_2, T, S_5, +, S_2, T, S_0 — input: $

→ **r3** (reduce, rule #3, E -> T)

Stack 5: S_7, E, S_5, +, S_2, T, S_0 — input: $

→ **r2** (reduce, rule #2, E -> T + E)

Stack 6: S_1, E, S_0 — input: $

Accept