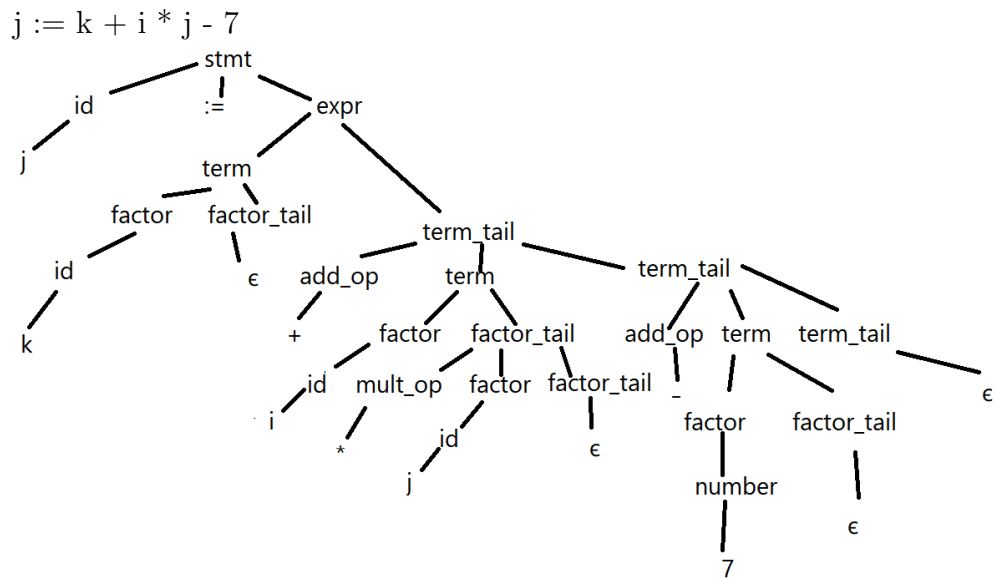
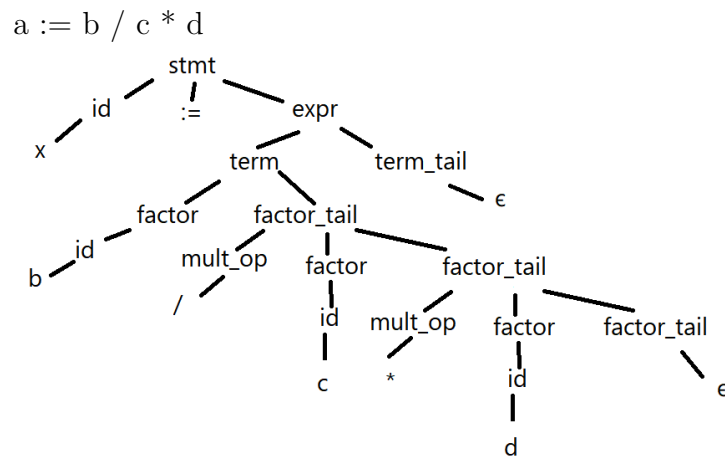
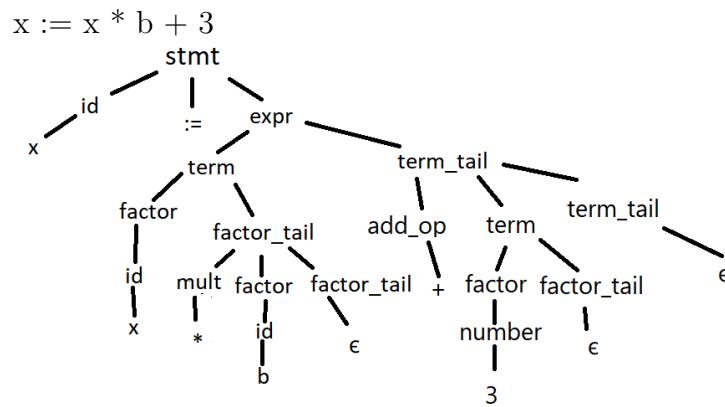


1



2 (a) **hoheyheyho**
$$\begin{aligned} S &\Rightarrow Y \\ &\Rightarrow \text{ho } S \text{ hey } Z \\ &\Rightarrow \text{ho } S \text{ heyheyho} \\ &\Rightarrow \text{ho } Z \text{ heyheyho} \\ &\Rightarrow \text{ho } \epsilon \text{ heyheyho} \\ &\Rightarrow \text{hoheyheyho} \end{aligned}$$

- (b) The grammar is ambiguous as there are multiple ways to derive the same sequence.  
For example **hohey** can be derived by

$$\begin{aligned} S &\Rightarrow Y \\ &\Rightarrow \text{ho } S \text{ hey } S \\ &\Rightarrow \text{ho } S \text{ hey } Z \\ &\Rightarrow \text{ho } S \text{ hey } \epsilon \\ &\Rightarrow \text{ho } Z \text{ hey} \\ &\Rightarrow \text{ho } \epsilon \text{ hey} \\ &\Rightarrow \text{hohey} \end{aligned}$$

Or it can be derived by

$$\begin{aligned} S &\Rightarrow Z \\ &\Rightarrow \text{hohey} \end{aligned}$$

3 (a)

$$S \rightarrow xA$$

$$\rightarrow yA$$

$$\rightarrow \_A$$

$$A \rightarrow xB$$

$$\rightarrow yB$$

$$\rightarrow \_B$$

$$\rightarrow 0B$$

$$\rightarrow 1B$$

$$\rightarrow \epsilon$$

$$B \rightarrow xA$$

$$\rightarrow yA$$

$$\rightarrow \_A$$

$$\rightarrow 0A$$

$$\rightarrow 1A$$

$$\rightarrow \epsilon$$

(b)

$$S \Rightarrow xA$$

$$\Rightarrow xyB$$

$$\Rightarrow xyxA$$

$$\Rightarrow xyx\_B$$

$$\Rightarrow xyx\_0A$$

$$\Rightarrow xyx\_01B$$

$$\Rightarrow xyx\_011A$$

$$\Rightarrow xyx\_011\epsilon$$

$$\Rightarrow xyx\_011$$

4

$$\begin{aligned} S &\rightarrow aaAa \\ A &\rightarrow aaAa \\ &\rightarrow B \\ B &\rightarrow bB \\ &\rightarrow \epsilon \end{aligned}$$

We know this will generate only the strings we are looking for as we ensure that once we reach the B state we can no longer add to the a characters. This allows us to make sure we maintain the proper amount of a's and allowing for as many b's as needed.

5

$$\begin{aligned} S &\rightarrow aCb \\ &\rightarrow aA \\ A &\rightarrow aA \\ &\rightarrow \epsilon \\ B &\rightarrow bB \\ &\rightarrow \epsilon \\ C &\rightarrow aCb \\ &\rightarrow aA \\ &\rightarrow abbB \end{aligned}$$

We know this will generate only the strings we are looking for as we ensure that, by adding our state C, we can never hit the condition of the a's and b's begin equal.