

Problem 2.27. Let $G = (V, \Sigma, R, \langle \text{STMT} \rangle)$ be the following grammar.

$$\begin{aligned}\langle \text{STMT} \rangle &\rightarrow \langle \text{ASSIGN} \rangle \mid \langle \text{IF-THEN} \rangle \mid \langle \text{IF-THEN-ELSE} \rangle \\ \langle \text{IF-THEN} \rangle &\rightarrow \text{if condition then } \langle \text{STMT} \rangle \\ \langle \text{IF-THEN-ELSE} \rangle &\rightarrow \text{if condition then } \langle \text{STMT} \rangle \text{ else } \langle \text{STMT} \rangle \\ \langle \text{ASSIGN} \rangle &\rightarrow \text{a:=1}\end{aligned}$$

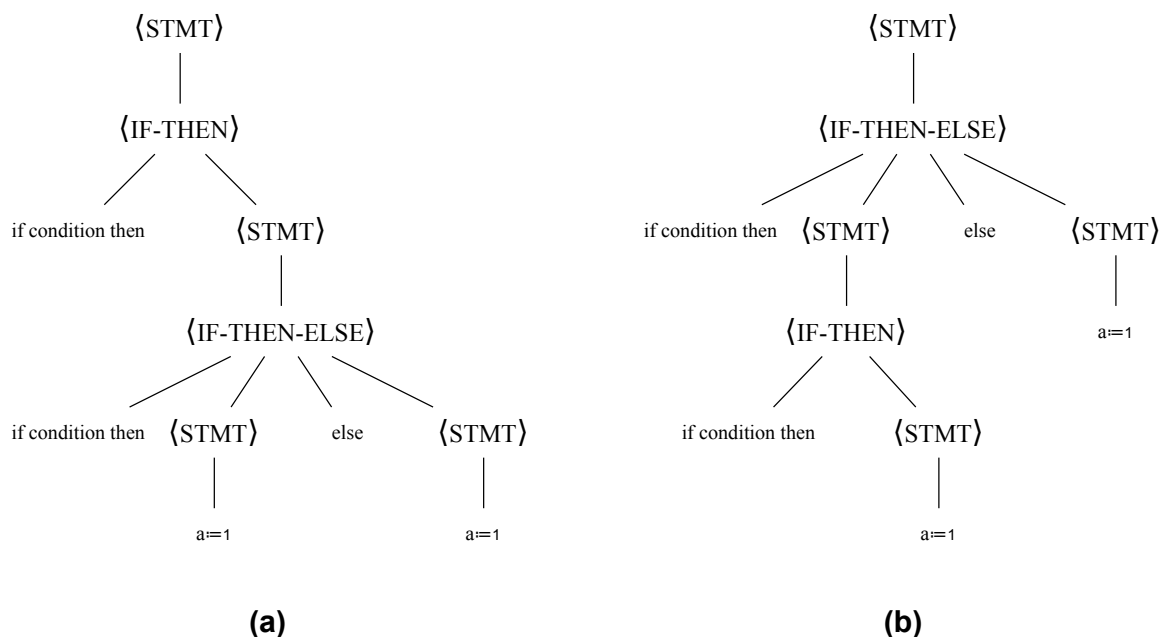
$$\Sigma = \{\text{if, condition, then, else, a:=1}\}$$

$$V = \{\langle \text{STMT} \rangle, \langle \text{IF-THEN} \rangle, \langle \text{IF-THEN-ELSE} \rangle, \langle \text{ASSIGN} \rangle\}$$

G is a natural-looking grammar for a fragment of a programming language, but G is ambiguous.

Part a. Show that G is ambiguous.

The string “if condition then if condition then a:=1 else a:=1” has two different parse trees in G .



Two different parse trees of the string
“if condition then if condition then a:=1 else a:=1”.

Part b. Give a new unambiguous grammar for the same language.