

Aufgabe 1 Lösung

a)

$$S \rightarrow 0S1 \mid 01$$

eindeutig, es gibt für jedes Wort nur genau eine mögliche Ableitung.

$$L = \{0^n 1^n \mid n \geq 1\}$$

b)

$$S \rightarrow +SS \mid -SS \mid a$$

eindeutig; die rechte Seite jeder Produktion fängt mit einem eindeutigen Terminalsymbol an, dadurch gibt es für jedes Wort genau eine left-most derivation.

Die Sprache beschreibt Prefix-notierte arithmetische Ausdrücke über a

c)

$$S \rightarrow S(S)S \mid \epsilon$$

mehrdeutig. Beispiel: $()()$

$$1. S \xrightarrow{lm} S(S)S \xrightarrow{lm} \epsilon(S)S \xrightarrow{lm} \epsilon(\epsilon)S \xrightarrow{lm} \epsilon(\epsilon)S(S)S \xrightarrow{lm}^* ()()$$

$$2. S \xrightarrow{lm} S(S)S \xrightarrow{lm} S(S)S(S)S \xrightarrow{lm}^* ()()$$

Die Sprache beschreibt wohl-geklammerte Ausdrücke

d)

$$S \rightarrow aSaS \mid bSaS \mid \epsilon$$

mehrdeutig

e)

$$S \rightarrow a \mid S+S \mid SS \mid S^* \mid (S)$$

mehrdeutig

Beispiel: $a + a + a$

$$\begin{array}{c}
 S \\
 / \quad | \quad \backslash \\
 S \quad | \quad | \\
 / \quad | \quad \backslash \quad | \quad | \\
 S \quad | \quad S \quad | \quad S \\
 | \quad | \quad | \quad | \quad | \\
 a + a + a
 \end{array}$$

$$\begin{array}{c}
 S \\
 / \quad | \quad \backslash \\
 | \quad | \quad S \\
 | \quad | \quad / \quad | \quad \backslash \\
 S \quad | \quad S \quad | \quad S \\
 | \quad | \quad | \quad | \quad | \\
 a + a + a
 \end{array}$$

Aufgabe 2 Lösung

ModifierSet -> PFS | PSF | SPF | SFP | FPS | FSP

P -> public | eps

F -> final | eps

S -> static | eps

Ein realer Compiler würde z.B. folgende Grammatik nehmen:

ModifierSet -> ModifierSet Modifier | eps

Modifier -> public | static | final

Und in einer späteren Analysephase die Menge auf Korrektheit überprüfen.