

## 32. „Dangling-else“ ambiguity

Gegeben: Grammatik  $G = (T, N, P, S)$  mit

- $T := \{ \text{if, else, s1, s2, c1, c2} \}$
- $N := \{ E, S \}$  mit  $S$  = Startsymbol
- Produktionsregeln  $P$ :
  - (1)  $S \rightarrow \text{if } E \ S$
  - (2)  $S \rightarrow \text{if } E \ S \ \text{else } S$
  - (3)  $S \rightarrow s1 \mid s2$
  - (4)  $E \rightarrow c1 \mid c2$

*Bedeutung der Terminals:*

$s1, s2$  (statement1, statement2) stehen jeweils für Anweisungen oder Anweisungsblöcke

$c1, c2$  (condition1, condition2) stehen jeweils für Boolesche Terme

Zeige:

Für das Wort `if c1 if c2 s1 else s2` gibt es verschiedene Syntaxbäume.

*Möglichkeiten, um der Mehrdeutigkeit zu begegnen:*

- Der else-Zweig bezieht sich immer auf das nächststehende if.
- Kennzeichnung von Anweisungsblöcken durch entsprechende Strukturierung des Quelltextes

Python: durch Einrücken;

Pascal: mit den Schlüsselwörtern **begin** und **end**;

C++, Java: mit **{** und **}**.

<pre>if c1:     if c2:         s1     else:         s2</pre>	<pre>if c1:     if c2:         s1     else:         s2</pre>
--	--

33. Gegeben ist die Grammatik  $G = (T, N, P, S)$ , bestehend aus der Menge  $T$  der Terminalzeichen, der Menge  $N$  der Nonterminalzeichen, der Menge  $P$  der Produktionen und dem Element  $S \in N$  als Startzeichen:

$T := \{ a, b, p, q, \text{if, then, else} \}$

$N := \{ S, S_1, S_2, B, T \}$  mit  $S$ =Startzeichen

Produktionen  $P$ :

- (1)  $S \rightarrow S_1 \mid S_2$
- (2)  $S_1 \rightarrow T \mid \text{if } B \text{ then } S_1 \text{ else } S_2$
- (3)  $S_2 \rightarrow T \mid \text{if } B \text{ then } S \mid \text{if } B \text{ then } S_1 \text{ else } S_2$
- (4)  $B \rightarrow p \mid q$
- (5)  $T \rightarrow a \mid b$

Zeige:

Das Wort `if p then if q then a else b`

besitzt in dieser Grammatik nur einen einzigen Syntaxbaum!

*Bedeutung der Terminals:*

$p, q$  bezeichnen Boolesche Terme;

$a, b$  stehen für Anweisungen oder Anweisungsblöcke.