

Programming Languages and Compilers, WS 2010/2011
Test Context-free Grammars, Nov 11 2010

U. Kastens

Institut für Informatik, Fakultät für Elektrotechnik, Informatik und Mathematik,
 Universität Paderborn

This test will be graded anonymously and returned.
 Please enter your nickname:

uk

20

1. A context-free grammar consists of (precisely):

- 2
- T set of terminals
 - N set of non terminals
 - $S \in N$ start symbol
 - P set of productions

2. Use the productions

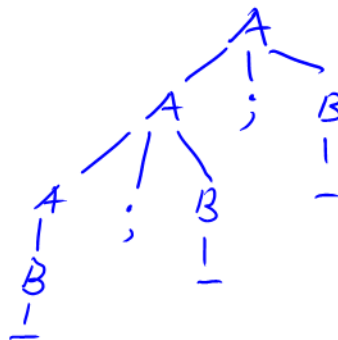
- A ::= A ';' B
- A ::= B
- B ::= '-'

$A \Rightarrow A ; B \Rightarrow A ; B ; B \Rightarrow B ; B ; B$
 $\Rightarrow - ; B ; B \Rightarrow - ; - ; B \Rightarrow - ; - ; -$

2 to derive - ; - ; - from A.

3. Draw a derivation tree for (2.).

2



4. A CFG is ambiguous if ...

2 there is a sentence in $L(G)$ that has two derivation trees

5. Describe the language of (2.).

2 a non-empty sequence of - separated by ;

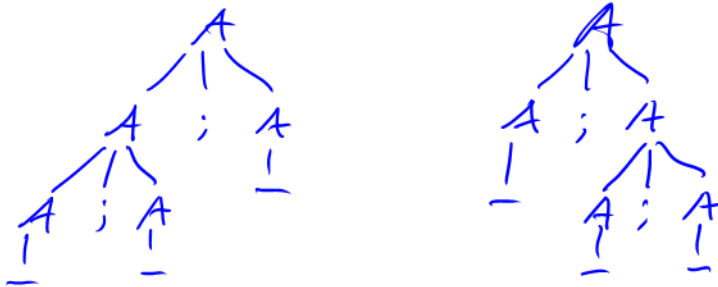
6. Write an ambiguous grammar for (5.)

2

$A ::= A ; A$
 $A ::= '-'$

7. Demonstrate that (6.) is ambiguous. *The sentence has 2 derivation trees:*

2



8. Write a CFG for expressions with binary operators **ODER** and **UND** and a unary prefix operator **NICHT**. Terminal operands are **id**. **NICHT** has highest precedence, **ODER** lowest. Parentheses can be used to group subexpressions.

2

$E ::= E \text{ 'ODER' } F \mid F$
 $F ::= F \text{ 'UND' } T \mid T$
 $T ::= \text{'NICHT' } T$
 $T ::= \text{'(' } E \text{ ')'}$
 $T ::= \text{id}$

9. Write an abstract syntax for (8.).

2

$Ex ::= Ex \text{ BOp } Ex$
 $Ex ::= \text{uOp } Ex$
 $Ex ::= \text{id}$
 $\text{BOp} ::= \text{'ODER' } \mid \text{'UND'}$
 $\text{uOp} ::= \text{'NICHT'}$

10. Write a strict BNF grammar for "A is a non-empty sequence of B or C where the sequence elements are separated by ',' or '!'.

2

$A ::= A \text{ Sep } \text{BoC}$
 $A ::= \text{BoC}$
 $\text{BoC} ::= B$
 $\text{BoC} ::= C$
 $\text{Sep} ::= \text{' ,'}$
 $\text{Sep} ::= \text{' !'}$