

- 1) Betrachten Sie die KFG $G_1 = (\{S, A, B\}, \{a, b\}, R, S)$, wobei die Regeln R wie folgt definiert sind:

$$\begin{aligned} S &\rightarrow aSA \mid bSB \mid c \\ A &\rightarrow a \mid bB \\ B &\rightarrow b \mid aA \end{aligned}$$

- Geben Sie den Syntaxbaum in Bezug auf G_1 für das Wort $abcaaa$ an.
 - Ist die Grammatik G_1 eindeutig? Begründen Sie Ihre Antwort.
 - Geben Sie eine kontextfreie Grammatik G_2 für die Sprache $\{a^{2n}cb^n \mid n \geq 0\}$ an.
- 2) Gegeben sei die folgende Grammatik $G = (\{S, B\}, \{a, b\}, R, S)$, wobei die Regeln wie folgt definiert sind:

$$\begin{aligned} S &\rightarrow a \mid SB \mid aSa \mid \epsilon \\ B &\rightarrow b \mid BS \end{aligned}$$

- a) Zeigen Sie mittels rekursiver Inferenz, dass das Wort $abab$ Teil der Sprache $L(G)$ ist.
 - b) Zeigen Sie dass die Grammatik mehrdeutig ist.
- 3) Schreiben Sie einen lex/yacc parser für die Aussagenlogik (siehe Folien Woche 2). Sie können sich an den unten angeführten Quellcode für die KFG G_2 aus der Vorlesung orientieren. Unter Linux können die Dateien wie folgt kompiliert werden kann:

- `lex ex.l`
- `yacc -d ex.y`
- `cc lex.yy.c y.tab.c -o out`

Stellen Sie sicher, dass Sie auch Lex and Yacc installiert haben¹.

`ex.l`

```
%{
    #include <stdio.h>
    #include "y.tab.h"
}%

%%
" c"      return C;
```

¹`apt install bison flex`

```

"v"      return V;
"("      return LB;
")"      return RB;
\"+\"    return PLUS;
\"-\"    return MINUS;
\"*\"    return MUL;
\"/\"    return DIV;
[ \\t] ; /* skip whitespace */
\n      return ENTER;
%%

```

ex.y

```

%{
#include <stdio.h>
#include <string.h>
void yyerror(const char *);
int yylex();
int yyparse();
}%

%token C V PLUS MINUS MUL DIV LB RB ENTER

%%

prog :  exp ENTER { printf("Accepted\n"); return 0; }
      ;

exp:   term
      | PLUS term
      | MINUS term
      | exp PLUS term
      | exp MINUS term
      ;

term:  factor
      | term DIV factor
      | term MUL factor
      ;

factor: C
       | V
       | LB exp RB
       ;

%%

void yyerror(const char *str) {
    printf("Syntax error\n");
}

```

```
int yywrap() {  
    return 1;  
}  
  
int main() {  
    printf("Enter expression:");  
    yyparse();  
}
```