Marlon Braun
Gruppe 05 6175858

Prof. Dr. Visvanathan Ramesh
Alperen Kantarcı
Gamze Akyol

Goethe-Universität Frankfurt am Main
Institut für Informatik

Programmierparadigmen und Compilerbau

Sommersemester 2023

Exercise Nr. 5

Deadline Thursday 06.07.2023, 10:00 in Moodle!

Question 0 (0 Points)

Read the following tips!

- Group work is **not allowed!** You can talk and discuss with your friends about the exercises, but you should solve the exercises **individually**. Plagiarism is not tolerated.
- You can use ChatGPT or other web sources but you cannot ask the whole question. You should be able to explain every step of your code.
- Upload all the files you want to submit into **a single .zip file on Moodle**. Please ensure that you **only submit .pdf and .hs files.**
- Please write your **matriculation number**, **name** and **group number** to each file that you are submitting.
- Your submission file name should be in the following order:

  Name_matriculationNumber_Group_GroupNumber_Exercise_ExerciseNumber.extension.

  For example **AlperenKantarci_1111111_Group_2_Exercise_5.zip**

Question 1 - Regular Expressions (1 + 1 + 1 + 1 = 4 Points)

Specify the regular expression for the following languages using the alphabet $\sum = \{a, b, c\}$

1. All words where every a is followed by **at least** one b
2. All words where the number of 'a' is divisible by 3 (including 0 'a')
3. All words which starts and ends with the same symbol.
4. All words that contain 'abc' as a substring

Question 2 - DFA & NFA (2 + 4 + 4 = 10 Points)

Let L be a formal language that has $\sum = \{a, b\}$ as an alphabet. All words that end with **'bab'** are part of the language L.

1. Write down the language as a regular expression.
2. Create a DFA for the language L that accepts the language.
3. Create an NFA with a maximum of 4 states that accepts the language.

Question 3 - Parsing **(3 + 3 = 6 Points)**

You are given a "**parser.hs**" file that contains parser combinator functions. You will understand and explain how this parser works.

1. First, explain **each** function of the parser and give a simple usage. For example 'symbol function defines a parser that checks if the first character of a string matches a specific character. If it matches, it returns a tuple with the remaining string and the matched character. One example usage: `symbol 'a' ''abc''`. This would return `[(''bc'','a')]` which is the remaining unmatched string and matched symbol.'

2. Run the following three parser commands and explain each step of the execution, with the result.

   - `runParser (chainl zahl (symbol '+' *> succeed (+) <|> symbol '-' *> succeed (-))) "346 + 847 - 346 + 223`
   - `runParser (chainl (chainl (chainl zahl (symbol '*' *> succeed (*))) (symbol '/' *> succeed div)) (symbol '+' *> succeed (+) <|> symbol '-' *> succeed (-))) "12*34/2+56-78*9"`
   - `runParser (many ((option (symbol '+') *> integer) <|> (symbol '-' *> integer <@ negate))) "+984-234+445"`

Specify the regular expression for the following languages using the alphabet $\sum = \{a, b, c\}$

1. All words where every a is followed by **at least** one b

$R = C^* b^* (ab^+)^*$

hinter dem (ab+)* sollten noch b und c folgen können

2. All words where the number of 'a' is divisible by 3 (including 0 'a')

$R = b^* c^* (aaa)^+ c^* b^*$

matcht z.B. nicht 'abaca'

3. All words which starts and ends with the same symbol.

$R = a^+(a^*b^*c^*)^* a^+ \mid b^+(a^*b^*c^*)^* b^+ \mid c^+(a^*b^*c^*)^* c^+$ ✓

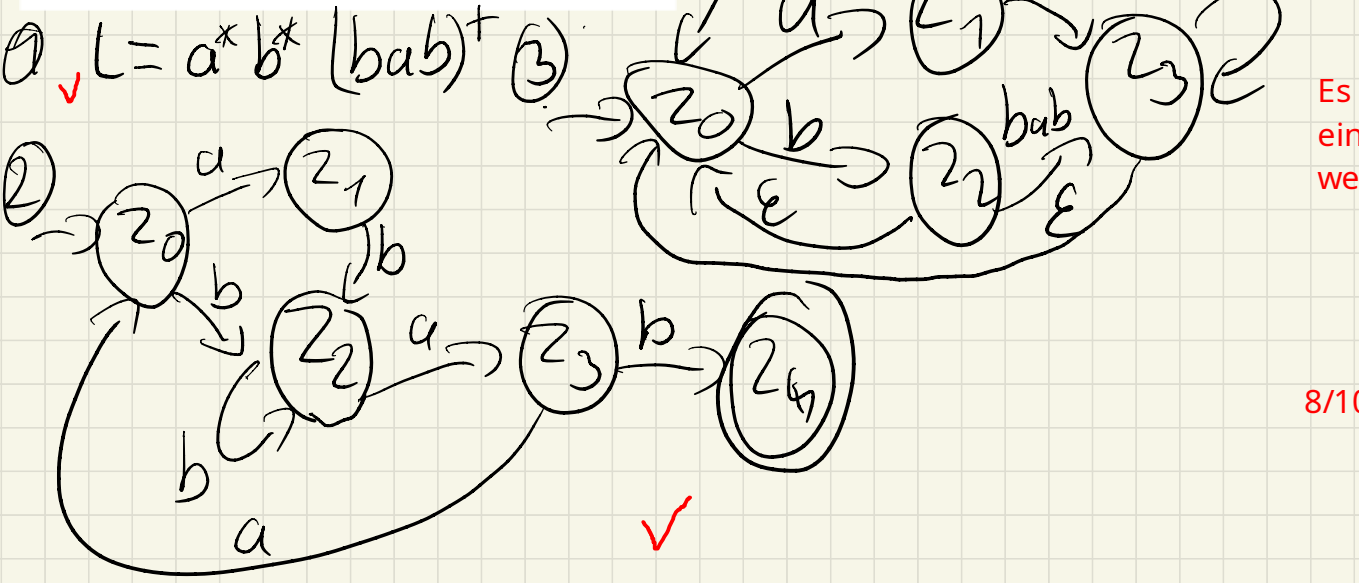
4. All words that contain 'abc' as a substring

$R = (a^*b^*c^*)^* (abc)^+ (a^*b^*c^*)^*$ ✓

Question 2 - DFA & NFA **(2 + 4 + 4 = 10 Points)**

Let L be a formal language that has $\sum = \{a, b\}$ as an alphabet. All words that end with **'bab'** are part of the language L.

1. Write down the language as a regular expression.
2. Create a DFA for the language L that accepts the language.
3. Create an NFA with a maximum of 4 states that accepts the language.



Es soll nur pro Schritt ein Buchstabe gelesen werden

8/10