

Prof. Dr. Knut Reinert
Enrico Siragusa
Sascha Meiers
Christoph Hartmann

Institut für Informatik
AG Algorithmische Bioinformatik

Algorithmen und Datenstrukturen in der Bioinformatik

Elftes Übungsblatt WS 11/12

Abgabe Montag, 16.01.2012, 15:00 Uhr

Name:

Übungsgruppe:

A B C

Matrikelnummer:

Niveau I

Aufgabe 1: Grammatiken

Gegeben sei eine Grammatik mit den folgenden Produktionsregeln:

- I $S \rightarrow Ta$
- II $T \rightarrow Sb$
- III $T \rightarrow Ub$
- IV $U \rightarrow Ub$
- V $U \rightarrow \epsilon$

- a) Geben Sie eine umgangssprachliche, aber exakte Beschreibung dieser Sprache
- b) Konstruieren Sie einen DFA, der diese Sprache akzeptiert.

Aufgabe 2: Automaten

- a) Konstruieren sie einen *deterministischen finiten Automaten*, der Ganzzahlen über dem Alphabet $\Sigma = -, 0\dots 9$ akzeptiert. Beispiele für akzeptierte Zahlen sind 0, -170 und 13, nicht akzeptiert werden sollen z. B. 007, 13 - -0 und -0.
- b) Mit welchem regulären Ausdruck lassen sich *rationale Zahlen* ausdrücken, die alle in a) genannten Bedingungen erfüllen, aber zudem noch einen Punkt mit beliebig vielen Nachkommastellen haben können?

Niveau II

Aufgabe 3: Beweise Beweisen oder widerlegen Sie folgendes:

- a) Ist eine Sprache regulär, so ist auch die Komplementsprache \bar{L} regulär. Die Komplementsprache enthält genau die Worte, die nicht in L sind.
 - b) Ist eine Sprache L_1 Teilmenge einer regulären Sprache L_2 , so ist L_1 regulär. Teilmenge heißt, dass alle Worte aus L_1 auch Worte von L_2 sind.
 - c) Seien L_1 und L_2 reguläre Sprachen, so ist auch L_1L_2 regulär. (Alle Worte der Sprache L_1L_2 beginnen mit einem Wort aus L_1 , auf das ein Wort aus L_2 folgt).
-

Aufgabe 4: Bonusaufgabe: RU.

Sie haben ein formales System über dem Alphabet $\sigma = R, I, U$. Sie können aus Worten, die im System sind, neue Worte aus dem System erzeugen, indem Sie die folgenden Regeln anwenden:

- I** Wenn Sie eine Kette besitzen, deren letzter Buchstabe I ist, können Sie am Schluss ein U zufügen ($RUUI \rightarrow RUUIU$).
- II** Angenommen Sie haben Rx Dann können es durch Rxx ersetzen. x steht hier für Zeichenfolgen beliebiger Länge ($RUIUU \rightarrow RUIUUUIUU$).
- III** Wenn in einer der Ketten Ihrer Sammlung III vorkommt, können Sie es durch U ersetzen ($RIII \rightarrow RU$).
- IV** Wenn UU in einer Ihrer Ketten vorkommt, kann man es streichen ($UUU \rightarrow U$).

Sie haben zu Beginn nur ein Wort: RI. Wie können sie die folgenden Wörter erzeugen:

- a) RUI
 - b) RR
 - c) RU
-

Programmieraufgabe (Abgabe Montag, 23.01.2012, 15:00)

P-Aufgabe 6: Implement an *object oriented* DFA in C++.

Input arguments from the command line are:

- a) a file containing the text T
- b) a text containing a full definition of a DFA A (see below)
- c) an output filename where occurrences \mathcal{O} must be written

Your program must:

- a) read T
- b) read and build A
- c) scan T using A
- d) remember in \mathcal{O} positions where A accepts T
- e) write occurrences \mathcal{O} into an output file (see below)
- f) write time scanning T in milliseconds and the number of occurrences $|\mathcal{O}|$ to standard output (see below)

Remember that a DFA A is a 5-tuple $(\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$. The file format describing the DFA is as follows:

- a) line 1: number of states $|\mathcal{Q}|$
- b) line 2: initial state q_0
- c) line 3: final states \mathcal{F} (possibly more than one, separated by spaced)
- d) line 4: alphabet symbols Σ (other ASCII symbols encountered in T reset A to q_0)
- e) lines 5...: transition function $\delta : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$ (row i defines all explicit transitions for state q_i)

Example:

```
user@linux:~$ ./aufgabe6 english.50MB whatever.dfa results.txt
245, 687
```

```
user@linux:~$ head results.txt
46916
60678
71674
74695
96572
248819
251678
252348
260717
260791
```

```
user@linux:~$ cat whatever.dfa
9
0
8
aehrtvw
0 0 0 0 0 0 1
0 0 2 0 0 0 0
```

```
3 0 0 0 0 0 0
0 0 0 0 4 0 0
0 5 0 0 0 0 0
0 0 0 0 0 6 0
0 7 0 0 0 0 0
0 0 0 8 0 0 0
0 0 0 0 0 0 1
```

Hints:

Check out the material at <https://svn.imp.fu-berlin.de/agbio/aldabi/ws11/documents/aufgabe6>. You will find a code template *aufgabe6.cpp*, its class diagram *aufgabe6.pdf* and an example DFA *whatever.dfa*.

To test your program, you can download and unpack the input file <http://pizzachili.dcc.uchile.cl/texts/nlang/english.50MB.gz>.