

Aufgabentypen die in der Klausur vorkommen können

1. Nennen Sie fünf wichtige Anwendungsgebiete der Computerlinguistik.
2. Für welches der drei Anwendungsgebiete “Maschinelle Übersetzung”, “Rechtschreibkorrektur” und “Volltextsuche” müssen die meisten Komponenten eines Sprachmodells implementiert sein, für welches die wenigsten? Begründen Sie ihre Entscheidung in 1-3 Sätzen.
3. Nennen Sie jeweils zwei Lesarten für die folgenden Sätze. Welche Art von Ambiguität liegt jeweils vor?
 - (a) Der Keller ist voller Schimmel
 - (b) Jeder Mann liebt eine Frau.
 - (c) Im Schrank ist eine Wachstube.
4. Sei $\Sigma = \{a, b, c\}$ und $L = \{bba, aa, bb, ab\}$. Welche der folgenden Aussagen ist wahr?
 - (a) $\epsilon \in \Sigma$.
 - (b) $\epsilon \in \Sigma^*$.
 - (c) $\epsilon \in \Sigma^+$.
 - (d) $\epsilon \in L$.
 - (e) $L \in \Sigma^*$.
 - (f) $L \subseteq \Sigma$.
 - (g) $L \subseteq \Sigma^*$.
 - (h) L ist eine formale Sprache über dem Alphabet Σ .
 - (i) Wenn $x \in L$, dann $|x| < 4$.
 - (j) Wenn $x \subseteq L$, dann $|x| < 4$.
 - (k) $|\epsilon| = 1$.
 - (l) Für alle $x \in \Sigma^*$ gilt: $x \circ \epsilon = \epsilon \circ x$.
 - (m) $|\epsilon \circ \epsilon \circ \epsilon| = 0$
5. Setzen Sie bitte in die Lücken jeweils eines der Symbole “ \in , \ni , \subseteq , \supseteq , $=$, $>$ ” so ein, daß eine wahre Aussage entsteht:

$$\begin{array}{ll}
 \{a, b\} \cup \{a\} \underline{\quad} \{a, b, c\} & \{c, b\} \underline{\quad} \emptyset \\
 L((a + b)^*) \underline{\quad} L(a^*bb^*) & \{c, a\} \underline{\quad} a \\
 \epsilon \underline{\quad} L(a^*) &
 \end{array}$$

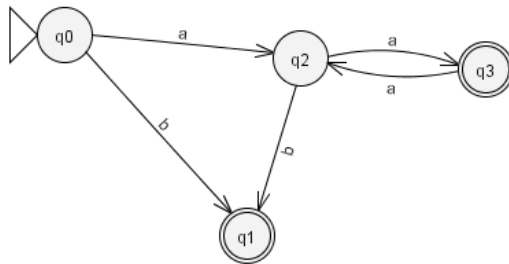
6. Sei $\Sigma = \{a, b, c\}$ ein Alphabet:
- Geben Sie ein Wort der Länge 4 über Σ an.
 - Welche der folgenden Ausdrücke sind Wörter über Σ und welche Länge haben sie:
'aa', 'caab', 'da'
 - Was ist der Unterschied zwischen Σ^* , Σ^+ und Σ ?
 - Wieviele Elemente haben Σ^* und Σ^+ ?
7. Seien $w = aabc$ und $v = bcc$ Wörter, berechnen Sie:
- $w \circ v$
 - $((w^R \circ v)^R)^2$
 - $w \circ (v^R \circ w^3)^0$
8. Sei $K = \{aa, ba\}$, $L = \{bb, aa\}$
- (a) Geben Sie die Sprachen $L \circ K$, $\{\epsilon\} \circ L$ und $K \circ \emptyset$ an.
 - (b) Geben Sie die Sprache L^3 an.
 - (c) Geben Sie die Sprache $K \setminus L$ an.
9. Woran erkennt man, dass ein endlicher Automat nicht deterministisch ist?
10. Wann akzeptiert ein endlicher Automat ein Wort?
11. Gegeben Sei die folgende Übergangstabelle eines endlichen Automaten, dessen Startzustand q_0 ist und der die Endzustände $F = \{q_0\}$ hat:
- | | | |
|-------|-------|-------|
| | a | b |
| q_0 | q_1 | |
| q_1 | q_0 | q_1 |
- (a) Zeichnen Sie den endlichen Automaten.
 - (b) Geben Sie eine rechtslineare Grammatik an, die die Sprache generiert, die von dem Automaten akzeptiert wird. (schwierig, da nicht geübt)
 - (c) Geben Sie den regulären Ausdruck der Sprache an, die von dem Automaten akzeptiert wird.

12. Gegeben sei folgende rechtslineare Grammatik:

$$\begin{aligned} S &\rightarrow aB \\ B &\rightarrow bA \\ B &\rightarrow \epsilon \\ A &\rightarrow aB \\ A &\rightarrow \epsilon \end{aligned}$$

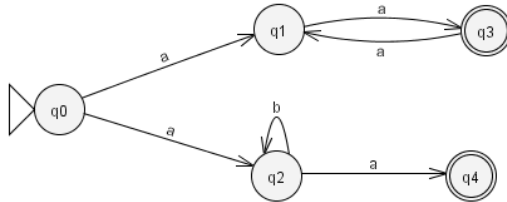
- Geben Sie einen endlichen Automaten an, der die Sprache akzeptiert, die von der Grammatik generiert wird.
- Geben Sie den regulären Ausdruck der Sprache an, die von der Grammatik generiert wird.
- Wählen Sie ein Wort der Länge 5, das von der Grammatik generiert wird und zeichnen Sie den Ableitungsbaum.

13. Gegeben sei folgender endlicher Automat:



- Geben Sie die Übergangstabelle des endlichen Automaten an.
 - Geben Sie den regulären Ausdruck der Sprache an, die von dem Automaten akzeptiert wird.
14. Zeichnen Sie einen deterministischen endlichen Automaten mit vollständiger Übergangsfunktion, der die Sprache $L(ab^*a)$ akzeptiert und geben sie eine rechtslineare Grammatik an, die die Sprache generiert. Zeichnen Sie einen endlichen Automaten, der das Komplement der Sprache $L(ab^*a)$ akzeptiert.

15. Gegeben sei folgender nichtdeterministischer endlicher Automat:

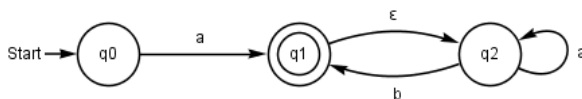


- (a) Zeichnen Sie einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert. (schwierig, da wenig geübt)
- (b) Geben Sie den regulären Ausdruck der Sprache an, die von dem Automaten akzeptiert wird.

16. Gegeben Sei die folgende Übergangstabelle eines endlichen Automaten, dessen Startzustand q_0 ist und der die Endzustände $F = \{q_1, q_2\}$ hat:

	a	b	c
q_0	q_1, q_2		
q_1		q_1	
q_2			q_2

- (a) Zeichnen Sie den endlichen Automaten.
 - (b) Zeichnen Sie einen deterministischen endlichen Automaten an, der ebenfalls die Sprache akzeptiert, die von dem Automaten akzeptiert wird.
17. Geben Sie einen regulären Ausdruck an, der die Sprache über dem Alphabet $\Sigma = \{a, b, c\}$ beschreibt, die aus allen Wörtern besteht, die genau ein c beinhalten, aber nicht auf c enden.
18. Geben Sie einen endlichen Automaten ohne ϵ -Übergänge an, der die Sprache akzeptiert, die von dem dargestellten Automaten akzeptiert wird.



19. Zeichnen Sie einen endlichen Automaten, der die Vereinigungsmenge der beiden Sprachen $L(b^*a^*)$ und $L(ba^+b^*)$ akzeptiert.

20. Geben Sie einen endlichen Automaten an, der die durch den Perl-Regex `/[aA]+/` beschriebene Sprache akzeptiert. Nennen Sie ein Wort, das akzeptiert wird und begründen Sie, warum es von dem Automaten akzeptiert wird. Geben Sie die Übergangstabelle für den Automaten an.
21. Geben Sie zu den folgenden Perl-Regexen formale reguläre Ausdrücke an, in denen nur die Operatoren `*`, `+` sowie die Konkatenation verwendet werden:
- (a) `/(ab){3}/`
 - (b) `/[aA]+/`
22. Vervollständigen Sie Zeile 6 des folgenden Programms so, dass nur Passwörter akzeptiert werden, die alle drei der folgenden Bedingungen erfüllen:
- (a) das Passwort besteht nur aus Buchstaben (ohne Umlaute und ohne ß); und
 - (b) das Passwort beginnt mit einem Kleinbuchstaben; und
 - (c) in dem Passwort folgt auf jeden Kleinbuchstaben ein Großbuchstabe und umgekehrt.

```

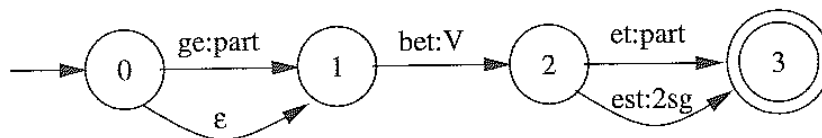
1  #!/perl -w
2  print "Please_enter_password:_";
3  $password = <>;
4  chomp($password);
5
6  while ($password !~ /^          $/) {
7
8      print "Access_denied.\n";
9      print "Please_enter_password:_";
10     $password = <>;
11     chomp($password);
12 }
13 print "Correct_password!_Welcome.";

```

23. Welche Aussagen lassen sich aus den folgenden Ergebnissen einer statistischen Textanalyse über den analysierten Text ziehen? Läßt sich Textart und Textinhalt ermitteln? Was ist an den Ergebnissen ungewöhnlich?

- die 20 häufigsten Wörter: die, in, mit, wurden, der, paderborn, am, das, und, polizei, eines, stadion, spiel, polizisten, fuballfans, düsseldorf, teile, freitagabend, anhänger, im.
- Zahl der Token: 183
Zahl der Typen: 120
Token pro Typ: 125
Anteil der 10 frequentesten Typen: 24.59%
Anteil der 20 frequentesten Typen: 35.51%
Anteil der 30 frequentesten Typen: 46.44%
Anteil der 40 frequentesten Typen: 56.28%
Anteil der 50 frequentesten Typen: 61.74%
Anteil der 60 frequentesten Typen: 67.21%
Anteil der 70 frequentesten Typen: 72.67%
Anteil der 80 frequentesten Typen: 78.14%
Anteil der 90 frequentesten Typen: 83.60%
Anteil der 100 frequentesten Typen: 89.07%
68.33% aller Typen kommen höchstens 1 mal vor.
91.66% aller Typen kommen höchstens 2 mal vor.
95.83% aller Typen kommen höchstens 3 mal vor.
98.33% aller Typen kommen höchstens 5 mal vor.
99.16% aller Typen kommen höchstens 6 mal vor.
100% aller Typen kommen höchstens 9 mal vor.

24. Was schreibt der folgende Transduktor bei der Eingabe von “gebetet” auf das zweite Band? Warum ist der Transduktor kein guter Transduktor zur Modellierung eines Ausschnitts der deutschen Morphologie?



25. Welche Antworten generiert die folgende DATR-Theorie auf die Anfragen:

- RON: <can fly>?
- RON: <has beak>?
- RON: <is quiet>?
- RIA: <is black>?
- RIA: <can sing>?
- RIA: <can fly>?

```

1 BIRD :
2   < > == no
3   <has claws> == yes
4   <has beak> == yes
5   <can fly> == yes.
6
7 RAVEN :
8   < > == BIRD
9   <is black> == yes.
10
11 RON :
12  < > == RAVEN
13  <can> == no
14  <is dead> == yes.
15
16 RIA :
17  < > == RAVEN.

```

26. Welche Antworten (yes/no) werden aus dem folgenden Prolog-Programm auf die Anfragen

- `liebt(frits,maria)?`
- `liebt(frits,paul)?`
- `liebt(frits,hasso)?`

inferiert?

```

1  liebt(frits ,X):- mensch(X).
2  mensch(anna).
3  mensch(paul).
4  hund(hasso).

```

27. Geben Sie eine kontextfreie Grammatik zu der Sprache $L(a^nca^n)$ mit $n \geq 1$ an. Zeichnen Sie den Ableitungsbaum für das Wort `aaacaaa`.

28. Gegeben sei die kontextfreie Grammatik

$$G_1 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow bSb, S \rightarrow A, A \rightarrow aA, A \rightarrow a\})$$

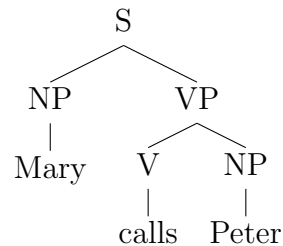
- Welche Sprache generiert die Grammatik G_1 ?
- Zeichnen Sie den Ableitungsbaum für das Wort $bbabb$.
- Zeichnen Sie einen Kellerautomaten, der die Sprache $L(G_1)$ akzeptiert.
- Zeigen Sie, wie der von ihnen konstruierte Kellerautomat das Wort bab abarbeitet, indem Sie eine Tabelle, wie unten dargestellt, ausfüllen. Geben Sie die Übergänge bitte als 5-Tupel an (alter Zustand, gelesenes Symbol der Eingabekette, vom Keller entferntes oberstes Symbol [pop up], auf den Keller gelegtes Wort [push down], neuer Zustand). Beschreiben Sie, warum der Kellerautomat das Wort bab akzeptiert.

(noch) zu lesende Kette	aktueller Zustand	aktueller Keller	anzuwendender Übergang
bab	q_0	ϵ	?

29. Geben Sie eine kontextfreie Grammatik an, die Sätze der folgenden Art generiert: He likes the meal on the flight to New York on Monday. Zu den Regeln sollten $NP \rightarrow NP PP$ und $VP \rightarrow VP PP$ gehören.

- Zeichnen Sie 4 verschiedene Derivationsbäume, die sich für den obigen Satz aus Ihrer Grammatik ergeben.
- Welche Parsingstrategie (top-down oder bottom-up) ist für Ihre Grammatik geeignet? Begründen Sie Ihre Entscheidung und führen Sie einen Parse durch. Beachten Sie, daß Sie unter Umständen Backtracking einsetzen müssen.

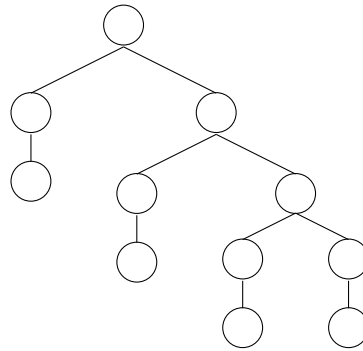
30. Geben Sie den entsprechenden Klammerausdruck zu dem folgenden Derivationsbaum an und erstellen Sie eine Liste der Regeln, die mindestens in einer Grammatik sein müssen, damit dieser Baum ein Derivationsbaum der Grammatik ist:



31. Geben Sie den Derivationsbaum zu dem folgenden Klammersausdruck an:

$$[s[_{NP}[_{D}die][_{N}Frau]][_{VP}[_{V}sieht][_{NP}[_{EN}Peter]]]]$$

32. Angenommen, der folgende Baum wäre ein durch einen left-to-right-top-down-breadth-first-Parser entstandener Derivationsbaum. In welcher Reihenfolge hat der Parser die Knoten generiert? Nummerieren Sie die Knoten entsprechend. Wenn zwei Knoten in demselben Schritt entstanden sind, so sollen sie dieselbe Nummer tragen.



33. Angenommen, der folgende Baum wäre ein durch einen right-to-left-bottom-up-depth-first-Parser entstandener Derivationsbaum. In welcher Reihenfolge hat der Parser die Knoten generiert? Nummerieren Sie die Knoten entsprechend. Wenn zwei Knoten in demselben Schritt entstanden sind, so sollen sie dieselbe Nummer tragen.

