3. Übungsblatt - Abgabe: 13.11.2012

In der folgenden Aufgabe (und auf manchen weiteren Übungsblättern) werden Sie mit dem Tiger-Corpus arbeiten. Das Tiger-Corpus enthält ca. 50000 deutsche Sätze, die mit morphologischen und syntaktischen Informationen wie zum Beispiel Wortarten annotiert sind. Sie können im Cip-Pool ein Such-Interface zum Tiger-Corpus öffnen, mit dem Sie, unter anderem auch unter Verwendung regulärer Ausdrücke, im Tiger-Corpus suchen können.

Aufruf von TIGERSearch im CIP-Pool:

- Öffnen Sie ein Terminal und geben Sie den folgenden Befehl ein:
  /proj/contrib/baumbank/TIGERSearch/bin/TIGERSearch
  Damit starten Sie das graphische User-Interface zur Suche in der TIGER Baumbank.
- Sollte kein Corpus geladen sein, laden Sie (durch Doppelklick mit der Maus auf TiGer v2.1 in der linken oberen Ecke des Fensters) das TIGER-Corpus.
- Machen Sie sich mit der Abfragesprache vertraut, indem Sie die folgenden Beispiele ausprobieren:

[word=„Fledermaus“]	→ findet alle Vorkommen des Wortes „Fledermaus“
[lemma=„Politiker“]	→ findet alle Wörter mit dem Lemma (Stamm) „Politiker“
[morph=„Dat\.Sg\.Fem“]	→ findet alle Wörter, die im Dativ Singular femininum stehen

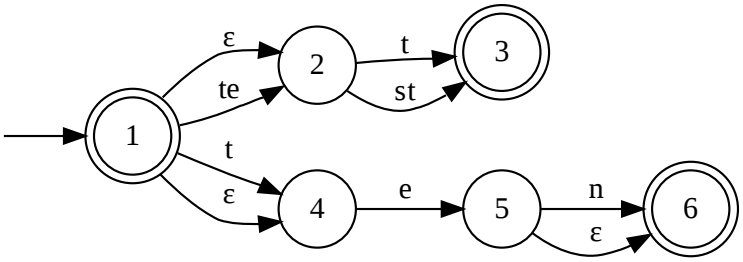
Die Tiger-Suche unterstützt außerdem folgende Notation für reguläre Ausdrücke:

.	irgendein beliebiges Zeichen
.*	kein oder beliebig viele Zeichen
[a-e]	a, b, c, d, e
[^a-e]	alle Zeichen außer a, b, c, d, e
(maus|hund)	Zeichenfolge maus oder hund
(ab)*	kein oder beliebig viele ab
(ab)+	mindestens ein oder beliebig viele ab
(ab)?	kein oder ein ab

Dabei werden reguläre Ausdrücke zwischen Slashes gesetzt. Die Suchanfrage [word=/.*tier/] würde also alle Wörter erkennen, die auf -tier enden, wie z.B. Haustier, Untier, Portier, Stier, etc., aber auch den String tier selbst.

Aufgabe 3.1

Im folgenden ist ein möglicher NEA angegeben, der die finiten Endungen eines schwachen Verbs wie „*stellen*“ akzeptiert.



a) Konstruieren Sie einen zum NEA äquivalenten buchstabierenden Automaten.

b) Erzeugen sie aus diesem Automaten durch Potenzautomatenkonstruktion einen äquivalenten DEA. Geben Sie auch die Übergangstabelle an.

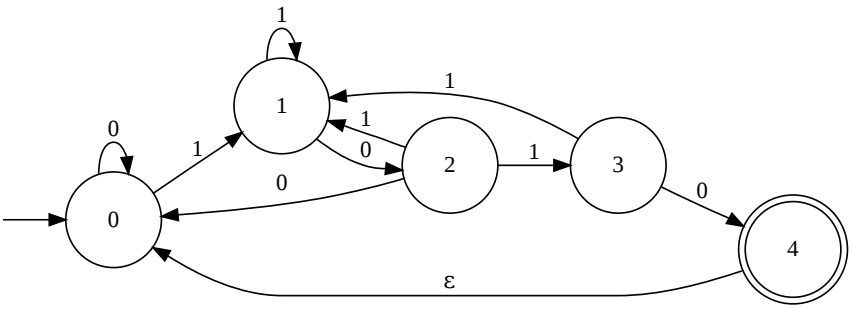
c) Benutzen Sie die Tigersuche, um alle Wörter mit dem Lemma „stellen“ im Tiger-Corpus zu finden. Wie viele sind es?

d) Versuchen Sie jetzt einen regulären Ausdruck zu formulieren, der dieselben Wortformen wie Ihr Automat erkennt. Suchen Sie alle Vorkommen im Tiger-Corpus, bei denen das Wort auf Ihren regulären Ausdruck passt. Wie viele Wörter finden Sie hier?

e) Wie erklären Sie sich den Unterschied zwischen den Zahlen? Suchen Sie Beispiele für Wörter, die von Ihrer Suchanfrage in Aufgabe c aber nicht von der in Aufgabe d gefunden wurden oder umgekehrt.

Aufgabe 3.2

Betrachten Sie den folgenden NEA A_1 über dem Alphabet $\Sigma = \{0,1\}$:

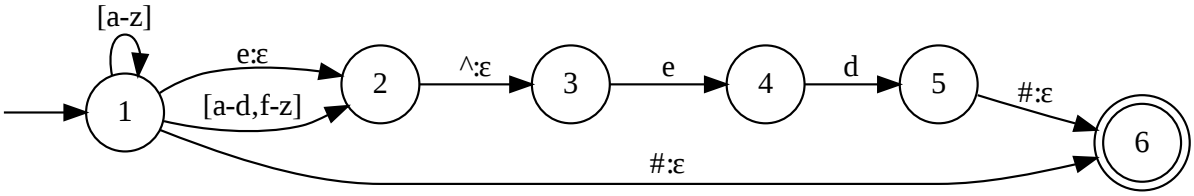


(a) Beschreiben Sie die Sprache, die der Automat erkennt.

(b) Erzeugen Sie durch Potenzautomatenkonstruktion einen DEA, der die gleiche Sprache wie A_1 erkennt. (Geben Sie ihre Zwischenschritte an und denken Sie an die Angabe der Übergangstabelle.)

Aufgabe 3.3

Ein endlicher Transduktor ist ein EA der jeder akzeptierten Eingabesequenz eine Menge von Ausgabesequenzen zuordnet. Zu jeder Kante des Zustandsdiagramms gehört deshalb neben dem Eingabe- ein zusätzliches Ausgabesymbol (notationell durch „:“ von einander getrennt).



Der abgebildete Transduktor übersetzt vorläufige Repräsentationen von Vergangenheitsformen englischer Verben, wie zum Beispiel

`bake^ed#`[1]

in die orthographisch korrekte Form:

[1] „`^`“ markiert eine Morphemgrenze und „`#`“ markiert das Wortende.

`baked`

Wie muss der Transduktor erweitert werden, damit auch die Formen der Verben *fry*, *envy* und *panic* richtig erzeugt werden (aus `fry^ed#`, `envy^ed#` und `panic^ed#`)?

Folgende Notation kann verwendet werden: Wird ein Symbol auf sich selbst abgebildet, muss es nur einmal an die Kante geschrieben werden; Mengen von Kanten, die Symbole auf sich selbst abbilden, können durch eine Kante dargestellt werden, an der die Menge der Symbole steht (z.B. [a-z]).

Aufgabe 3.4

a) Geben Sie die folgenden zwei Automaten über dem Alphabet $\Sigma = \{a, b\}$ an:
(1) den Automaten A_1, der alle Worte mit gerader Anzahl von a's (und beliebig vielen b's) akzeptiert,
(2) den Automaten A_2 ,der alle Worte mit gerader Anzahl von b's (und beliebig vielen a's) akzeptiert.

b) Zu zwei (entweder nicht-deterministischen oder deterministischen) Automaten $A_1 = \langle K_1, \Sigma, \Delta_1, s_1, F_1\rangle$ und $A_2 = \langle K_2, \Sigma, \Delta_2, s_2, F_2\rangle$ über dem gleichen Alphabet Σ lässt sich auf sehr einfache Weise ein NEA $A = \langle K, \Sigma, \Delta, s, F\rangle$ konstruieren, der genau die Vereinigung der beiden von A_1 und A_2 spezifizierten Sprachen akzeptiert, also: $L(A) = L(A_1) \cup L(A_2)$. Beschreiben Sie die allgemeine Konstruktion in Worten (oder durch eine Zeichnung) und formal, indem Sie die verschiedenen Komponenten des Automaten A unter Rückgriff auf A_1 und A_2 explizit spezifizieren.

c) Konstruieren Sie einen NEA, der alle Worte über dem Alphabet $\Sigma = \{a, b\}$ akzeptiert, die eine gerade Anzahl von a's oder eine gerade Anzahl von b's enthalten, indem Sie die beiden Automaten aus (a) durch das Verfahren aus (b) zu einem neuen Automaten verknüpfen.

d) Konstruieren Sie nach der Methode der NEA-DEA-Überführung einen zu diesem äquivalenten DEA A'.

Aufgabe 3.5

a) Wie kann man aus einem DEA A einen DEA A' konstruieren, der genau die Komplementsprache von $L(A)$ erkennt, also $L(A') = \Sigma^\star - L(A)$?
Hinweis: Die Konstruktion ist noch einfacher als die des Vereinigungs-Automaten!

b) Erzeugen Sie dementsprechend aus dem Resultat von 3.4. (d) den Automaten, der alle Worte über $\Sigma = \{a, b\}$ mit ungerader Zahl von a's und b's erkennt.

Aufgabe 3.6

Lesen Sie im Handbuch von Carstensen et al. den Abschnitt 5.1 (Korrekturprogramme), und zwar bis einschließlich 5.1.2 (Kontextabhängige Korrektur); das Papier befindet sich im Vorlesungsordner.

a) Eine ältere Version der MS Word-Rechtschreibprüfung hat vor Jahren als Alternativen für „semantisch“ die Wörter „seemännisch“ und „romantisch“ angegeben. Bestimmen Sie die im Text genannte Levenshtein-Distanz zwischen den drei Wörtern (also auch zwischen „seemännisch“ und „romantisch“).
Hinweis: Eine ausführlichere Beschreibung des Levenshtein-Algorithmus finden sie z.B. bei wikipedia.

b) Im Standard-Levenshtein-Algorithmus gibt es drei Operationen, die alle mit den gleiche Kosten verbunden sind. Auch innerhalb einer Operation kosten z.B. alle Ersetzungen gleich viel, egal, um welche Buchstaben es sich handelt.
Überlegen Sie sich mindestens 2 Möglichkeiten, wie man den Levenshtein-Algorithmus so abändern könnte, dass er für verschiedene Nutzergruppen (z.B nicht-Muttersprachler oder „flüchtige Tipper“) bessere Ergebnisse liefert, d.h. dass der Algorithmus besonders wahrscheinlichen Korrekturen einen niedrigeren Fehler-Score zuweist als unwahrscheinlichen Alternativen. Erklären Sie Ihre Verbesserungen anhand von Beispielen: Für welche Nutzergruppe sind welche Fehler wahrscheinlich und wieso weist Ihr Algorithmus diesem Fehler einen niedrigeren Score zu als anderen potenziellen Alternativen (die im Standardalgorithmus den gleichen Score hätten)?

c) Erklären Sie in an Hand von 2 Beispielen, wozu man bei der Rechtschreibprüfung kontextabhängige Korrektur benötigt.

Abgabe in Gruppen von bis zu drei Studierenden bis **13.11.2012** 10 Uhr entweder als Email im pdf-Format an **i2cl@coli.uni-sb.de** oder auf Papier im Briefkasten an der Tür von Raum 1.04 in C7.2.