Schlussprüfung Finite-State-Methoden in der Sprachtechnologie FS 13

Aufgabenstellung: Simon Clematide

Prüfung vom 3. Juni 2013
Institut für Computerlinguistik
Universität Zürich

Vorname _____ Matrikelnummer _____

Nachname _____

Für Studierende der folgenden Studiengänge:

☐ BA - Studiengang Computerlinguistik (Phil. Fakultät)

☐ BA - Studiengang Computerlinguistik und Sprachtechnologie (Phil. Fakultät)

☐ BA-Studierende (Wirtschaftswissenschaftliche Fakultät)

☐ Studierende des Nebenfachs Informatik mit Studienbeginn ab WS 04/05

☐ Andere:

Nur für Lizentiatsstudierende der Computerlinguistik als ein Fach aus der Phil. Fakultät:

Strasse: _____ Hauptfach: _____

PLZ/Ort: _____ E-Mail: _____

Aufgabe Nr.:	1	2	3	4	5	6	7	8	**Summe**
Punktzahl:	5	4	5	5	8	11	10	12	60
Davon erreicht:									

Note SU: _____ Note SP: _____

Endnote: _____ Bestanden: ☐ Ja ☐ Nein

Bitte auf jedes separate, d.h. nicht angeheftete Blatt mit Lösungen den Nachnamen schreiben. ***Viel Erfolg!***

Wichtige Hinweise

Punkte-Maximum: 60 (pro Minute 1 Punkt)
Hinweis: Bitte schreiben Sie in einem überlegten und knappen, aber verbalen Stil (keine Stichwortsammlungen). Bei inhaltlichen Auswahlsendungen, wo einfach mal alles spontan hingeschrieben wird und Falsches wie Korrektes munter vermischt sind, behalte ich mir Abzüge vor. Das Zeitbudget ist so berechnet, dass man vor dem Schreiben kurz überlegen darf.

5 **1. Lemmatisierung (5 Punkte)** Wieso macht man überhaupt **Lemmatisierung** mit morphologischer Analyse in der NLP? Geben Sie 1 gutes und motivierendes Beispiel für die **Vorteile**. Für welche **Typen von Sprachen** ist sie sinnvoll (Stichworte)? Welche **Alternativen zur Lemmatisierung** gibt es (Stichworte)?

Lemmas (d.h. Grundformen) sind eine linguistisch gut akzeptierte Repräsentation von den verschiedenen Wortformen eines Lexems. Lexikalische Ressourcen (mehrsprachige Wörterbücher) oder Ontologien enthalten oft nur die Grundform. Im Information Retrieval möchte man meist für einen Query-Term Resultate mit allen möglichen Flexionsformen bekommen. Z.B. soll eine Suche nach "Matterhorn" auch Genitivformen wie "Matterhorns" finden.

Sinnvoll ist Lemmatisierung vor allem für flektierende, agglutinierende oder polysynthetische Sprachen. D.h. für alle ausser die isolierenden. Eine verbreitete Alternative ist Stemming oder allenfalls auch Trunkierung.

4 **2. Morphologische Prozesse (4 Punkte)** Nennen Sie **2 unterschiedliche morphologische Prozesse** und geben Sie je **1 gutes Beispiel** dafür.

Siehe Skript Kapitel "Was ist Morphologie? II".

5 **3. lexc (5 Punkte)** Was sind die **wesentlichen** Eigenschaften des LEXC-Formalismus? Worin liegen seine sprachtechnologischen **Vorteile**? Inwiefern unterstützt der lexc-Formalismus auch den **Umgang mit Suppletiv-Formen**?

Siehe Skript Kapitel "Klassische Zwei-Ebenen-Morphologie".

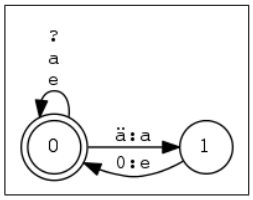
5 **4. (A) Frage zu gewichteten Endliche Automaten oder (B) Alternativfrage (5 Punkte)**
(A) Was sind die wesentlichen **Unterschiede** zwischen normalen EA und gewichteten EA? Geben Sie eine mögliche **Anwendung** von gewichteten Automaten in der NLP.
(B) Alternativfrage: Was versteht man unter morphologischer **Über- bzw. Unteranalyse**? Geben Sie realistische Beispiele! (Beantworten Sie entweder A oder B!)

(A) Siehe Vortrag von Roger Wechsler. (B) Siehe Skript Kapitel "Morphologisches Engineering"

8 **5. Zwei-Ebenen-Regeln vs. Ersetzen und Komponieren (8 Punkte)** Geben Sie die wesentlichen Eigenheiten, Unterschiede und Gemeinsamkeiten von beiden Ansätzen zur Bildung von Morphologiesystemen an. Geben Sie auch eine schematische Zeichnung.

Siehe Skript Kapitel "Klassische Zwei-Ebenen-Morphologie".

6. Reguläre Ausdrücke und Übergangsdiagramme (11 Punkte)



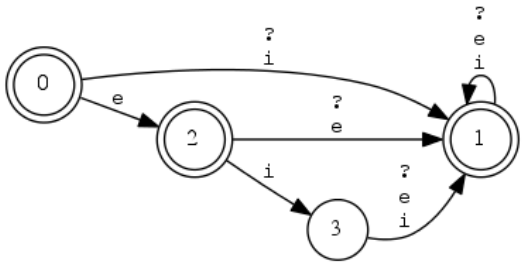
3 (a) Schreiben Sie einen **regulären Ausdruck**, der in XFST den rechts abgebildeten Automaten ergibt:

Zwei mögliche Lösungen mit xfst-Operatoren:

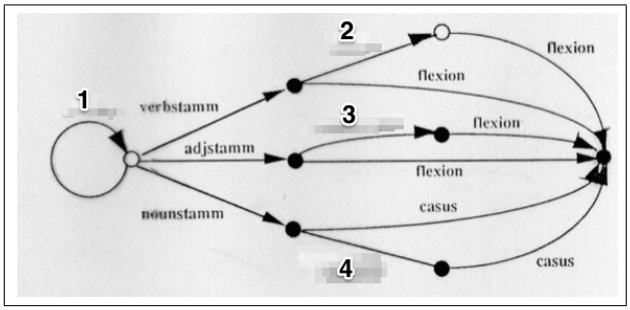
```
read regex [ ä -> {ae} ];
read regex [ \ä | ä:a 0:e ]*;
```

8 (b) Zeichnen Sie das **Zustandsdiagramm** eines möglichst kleinen Automaten, der die Sprache eines in XFST mittels `read regex ?* - {ei};` definierten regulären Ausdrucks akzeptiert. Geben Sie auch das **Sigma** Ihres Automaten an. Hinweis: Bitte benutzen Sie "?" als Kantenbeschriftung für das UNKNOWN-Symbol wie im Automaten von Teilaufgabe (a).

Bepunktung: 4 Punkte für einen Automaten, der zumindest "ei" nicht erkennt. Mehr Punkte, falls der Automat auch Fälle behandeln kann, wo "ei" irgendwo als echter Substring eines zulässigen Worts vorkommt.



7. Modell für Deutsch (10 Punkte)



1 (a) Wie heisst der Fachbegriff, den obige Abbildung exemplifiziert? Morphotaktik, allenfalls auch Fortsetzungslexika

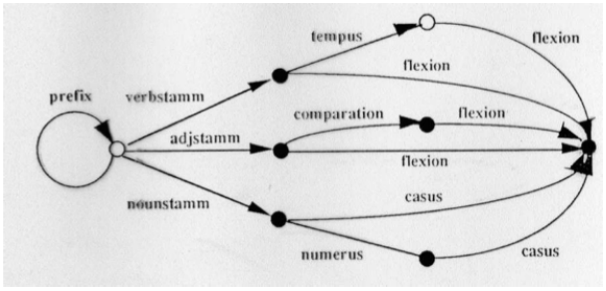
8 (b) Ergänzen Sie die **4 fehlenden nummerierten Beschriftungen** mit der für die deutsche Sprache passende morphologische Kategorie. Geben Sie je **1 Beispiel**, welche Morphe durch die 4 nummerierten angesprochen werden und **markieren Sie die Morphe** im Wort.

1 Präfix, oder auch als korrekt gezählt: Komposition(svorderglied)

2 Tempus oder Tempus/Modus

3 Steigerung

4 Numerus, allenfalls auch Derivationssuffix ("Metzger-ei")



1 (c) Geben Sie **1 Beispiel**, wo dieses Modell nicht passt.

Einige Beispiele, welche als korrekt betrachtet werden: Bei Umlautpluralbildung bei Nomen kann Numerus nicht als separater Infix betrachtet werden. Bei starken Verben ist Tempus nicht als Suffix realisiert. Suppletivformen werden nicht beschrieben. Komposition ist nicht drin im Modell (falls der Zirkel am Anfang als Präfigierung verstanden wird).

8. Ein Problem (12 Punkte)

Das "Translation Bureau" der kanadischen Regierung hat folgende Regel publiziert auf ihrer Homepage:

Double the final consonant before y or before a suffix beginning with a vowel in a word of one syllable that ends in a single consonant preceded by a single vowel:
bed → bedded, dip → dipped,s fat → fatty, fit → fitted, flit → flitting, gum → gummy, log → logged, mad → madden, rot → rotted

Exceptions

Do not double the final consonant in a word of one syllable if the vowel sound is long:

boat → boating, light → lighten, stoop → stooped, read → reading

3 (a) Wie lautet der Fachbegriff für solche Phänomene? Worum geht es dabei?

Orthographische oder phonologische Alternanz (Alternation)

9 (b) Ihr Praktikant Peter sollte ursprünglich einen Transduktor bauen, der auf der Unterseite die orthographisch korrekt derivierten Wortformen und auf der Oberseite ihre morphologische Struktur enthält, welche durch obige Suffigierungen entstehen können (z.B. `{bed+ed}:{bedded}`. Er hat aus einem Aussprachewörterbuch alle einsilbigen Wörter extrahiert und jeweils vor dem Wort markiert, ob der Vokal kurz ("=") oder lang "==" ist. Leider endete Peters Praktikum, bevor er fertig implementiert hatte. Programmieren Sie seine Aufgabe fertig und verwenden Sie wie im Beispiel das Zeichen + als Trennsymbol. Hinweis: Übergenerierung (z.B. "ready") ist erlaubt, da ja keine Part-Of-Speech-Information vorhanden ist.

```
define OneSyllable [ "="[{bed}|{dip}|{fat}|{fit}|{gum}|{log}|{mad}]
                   |["==" [{boat}|{light}||{stoop}|{read}] ]];
```

```
# Filter short and long syllables into separate lexicon
define ShortStems [[OneSyllable - $"=="] .o. ["=" -> 0]].l  ;
define LongStems  [[OneSyllable - $"=" ] .o. ["==" -> 0]].l ;

# Suffix lexicon
define Suffix [{ing}|{en}|{ed}|{y}] ;

# The phonological rules at morph boundaries
define PhonoRule [{d+}->{dd}, {g+}->{gg}, {m+}->{mm}, {p+}->{pp},
                  {t+}->{tt}] ;

# Putting things together
```

```
define Transducer [[ShortStems "+" Suffix ] .o. PhonoRule ]
                  | LongStems "+":0  Suffix ;
```