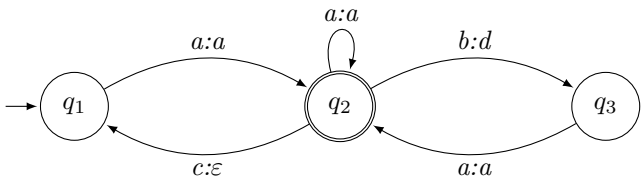
Einführung in die Computerlinguistik
Hausaufgabe zu FST und Morphologie, Abgabe 03.05.2021, 8.30 Uhr

Laura Kallmeyer

Sommer 2022, Heinrich-Heine-Universität Düsseldorf

Aufgabe 1 *Betrachten Sie folgenden FST:*



1. *Auf welche Strings bildet der FST die folgenden Eingaben ab? Akzeptiert er dabei die jeweilige Eingabe?*

   (a) aaa *(b) acabaab* *(c) acaaba* *(d) aac*

2. *Welche Strings abkezptiert dieser FST und wie transformiert er sie? (Eine Beschreibung der Transformation in Worten genügt.)*

Lösung:

1. (a) *aaa*, Akzeptanz
   (b) *aadaad*, keine Akzeptanz, da kein Endzustand erreicht wird
   (c) *aaada*, Akzeptanz
   (d) *aa*, keine Akzeptanz, da kein Endzustand erreicht wird

2. Akzeptiert wird die von $a(ca|a|ba)^*$ denotierte Sprache. Jedes c wird gelöscht, jedes a in die Ausgabe kopiert, jedes b durch d ersetzt.

Aufgabe 2 *Erstellen Sie einen Finite State Transducer, der die von $a^*(ab|c)^*$ denotierte Sprache akzeptiert, und der jedes a, das vor einem b steht, durch b ersetzt, jedes a, das vor einem c steht, durch c ersetzt, und alle anderen as in die Ausgabe kopiert.*

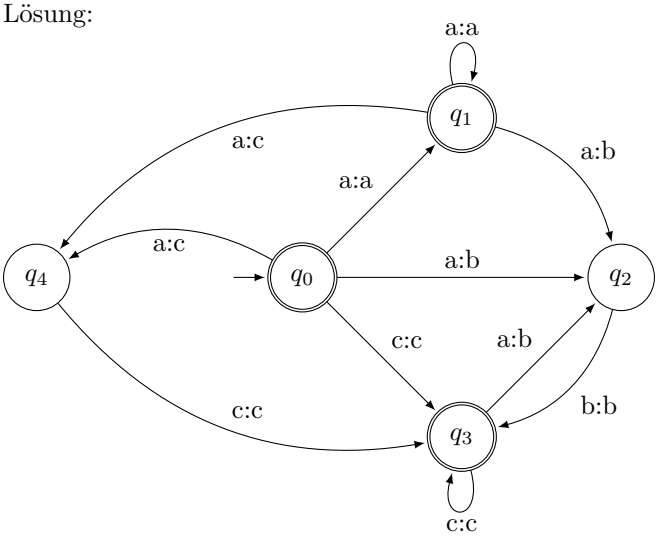
Z.B.

Eingabe	ε	*aaa*	*abab*	*cc*	*aaccab*
Ausgabe	ε	*aaa*	*bbbb*	*cc*	*acccbb*

Tipp: Sie brauchen (vermutlich)

- *einen Zustand, in dem man gerade ein a gesehen hat, das in die Ausgabe kopiert wurde, das also zu den ersten der initialen a entweder bis zum vorletzten, falls danach b oder c kommen, sonst bis zum letzten, gehört;*
- *einen Zustand, in dem man gerade ein a gesehen hat, das durch b ersetzt wurde, als nächstes wird also ein b erwartet;*
- *einen Zustand, in dem man gerade ein a gesehen hat, das durch c ersetzt wurde, als nächstes wird also ein c erwartet;*
- *einen Zustand, in dem man gerade ein c gesehen hat, es sind also weitere c's erlaubt oder a's, die durch b ersetzt werden müssen;*
- *und den Startzusatnd (hier gut überlegen, was am Anfang alles stehen kann).*

Lösung:



Aufgabe 3

Diminutiva Tor - Törchen, Tür - Türchen, Hut - Hütchen, Kind - Kindchen

Erstellen Sie einen Finite State Transducer, der Diminutivformen, gebildet mit "-chen" für deutsche Nomen erkennt, die aus einer Silbe mit einem einzigen Vokal und beliebig vielen Konsonanten davor und danach bestehen. Der FST sollte dabei davon ausgehen, dass Vokale a, o und u im Diminutiv zu den entsprechenden Umlauten werden, während sich e und i und auch Umlaute nicht ändern. Der FST sollte eine entsprechende Analyse ausgeben.

Wir nehmen an, dass vorher schon alles in Kleinbuchstaben umgewandelt wurde.

Ausgabe: lexikalischen Ebene	*Eingabe: Oberflächenebene*
tor Dim	*törchen*
tür Dim	*türchen*
schnur Dim	*schnürchen*
hut Dim	*hütchen*
hund Dim	*hündchen*
stuhl Dim	*stühlchen*
hand Dim	*händchen*
band Dim	*bändchen*
arm Dim	*ärmchen*
hemd Dim	*hemdchen*
kind Dim	*kindchen*
tisch Dim	*tischchen*

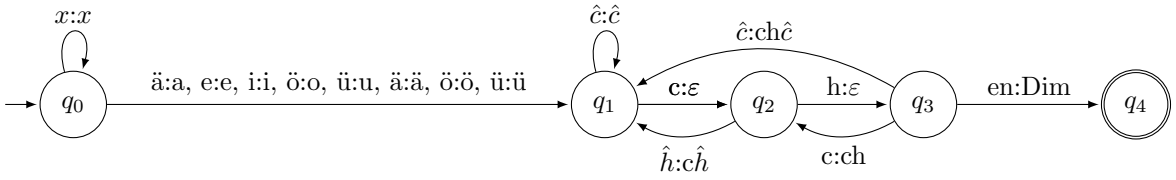
Sie dürfen mehrere aufeinander folgende Buchstaben zusammenfassen.

Beachten Sie, dass keine Erkennung existierender deutscher Nomen vorgenommen werden soll. Anders gesagt: Wenn der Diminutiv "Türchen" ist, weiß man nicht, ob die Grundform "Tür" oder "Tur" ist, beide sollten ausgegeben werden. Ihr Automat sollte auch mit nicht existierenden Wörtern, sofern sie dem Muster entsprechen, klar kommen. Z.B. für "börgchen" die Analysen "borg DIM" und "börg Dim" ausgeben.

Tipp: Sie können Variablen verwenden, z.B. x als Variable für beiebige Konsonanten, und dann Kanten beispielsweise mit $x : x$ labeln.

Lösung: (es gibt natürlich mehrere, die Lösung sollte in einem gewissen Maße generalisieren)

x steht für einen beliebigen Konsonanten, $\hat{c}$ steht für alle Konsonanten außer "c", $\hat{h}$ für alle außer "h".


x:x
$\hat{c}$:$\hat{c}$
$\hat{c}$:ch$\hat{c}$
ä:a, e:e, i:i, ö:o, ü:u, ä:ä, ö:ö, ü:ü
c:ε
h:ε
en:Dim
q_0
q_1
q_2
q_3
q_4
$\hat{h}$:c$\hat{h}$
c:ch