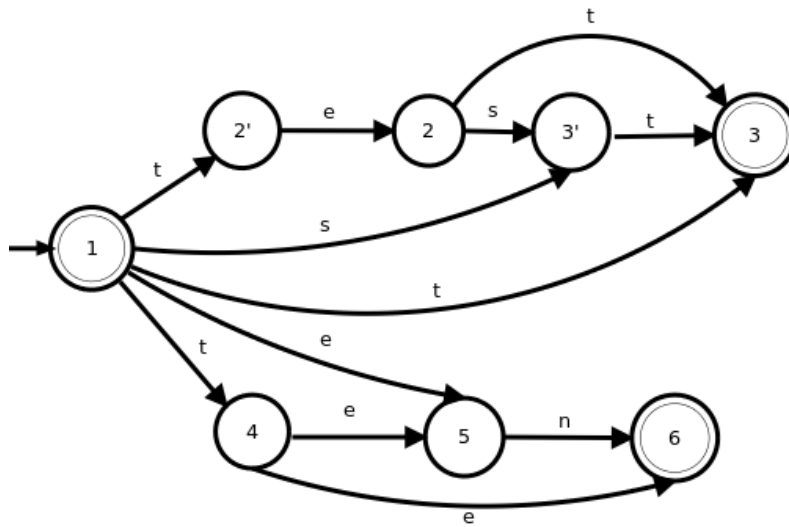


Einführung in die Computerlinguistik

3. Übungsblatt

Aufgabe 3.1

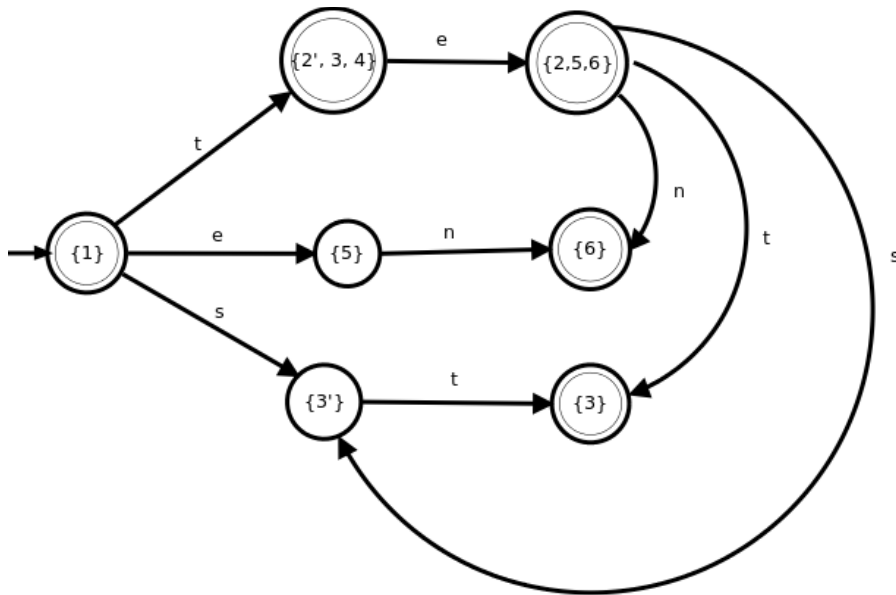
A.



B.

$S = \{1\}$, $F = \{\{1\}, \{3\}, \{6\}, \{2', 3, 4\}, \{2, 5, 6\}\}$

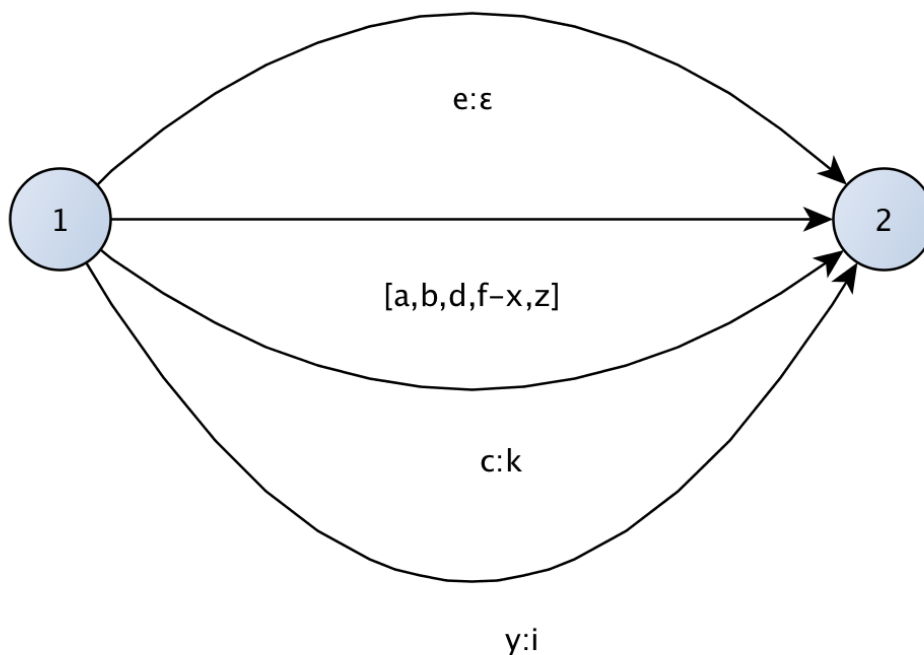
	t	e	s	n
{1}	{2', 3, 4}	{5}	{3'}	{}
{2', 3, 4}	{}	{2, 5, 6}	{}	{}
{5}	{}	{}	{}	{6}
{3'}	{3}	{}	{}	{}
{2, 5, 6}	{3}	{}	{3'}	{6}
{6}	{}	{}	{}	{}
{3}	{}	{}	{}	{}



- C. Anfrage: `grep -E "(\\skaufen)$" tiger2.txt | grep -E -o "\\S*"`
 72 Wörter.
- D. Ausdruck: `(t | te | test | ten | tet | en | st)?$`
 Anfrage: `grep -o -E "^((kauf(t|te|test|ten|tet|en|st)?\\s))" tiger2.txt`
- E. Der Automat ist strenger. Beispielsweise werden Präfixe wie "ge-" nicht akzeptiert.
 Beispiele: gekauft, kaufe

Aufgabe 3.2

Die Übergänge zwischen den Zuständen eins und zwei müssen angepasst werden. Der Rest bleibt unverändert.



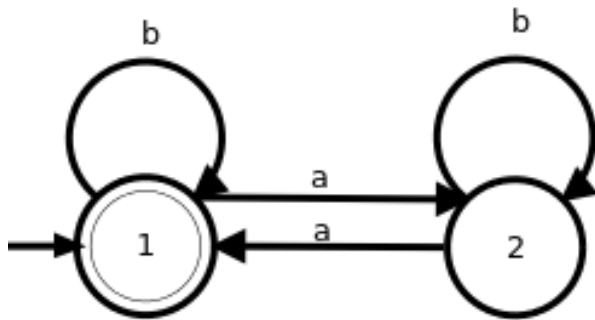
Aufgabe 3.3

Befehl: `sed s/'e\^/'^/g Verben.txt | sed s/'^ed#/'ed'/g`

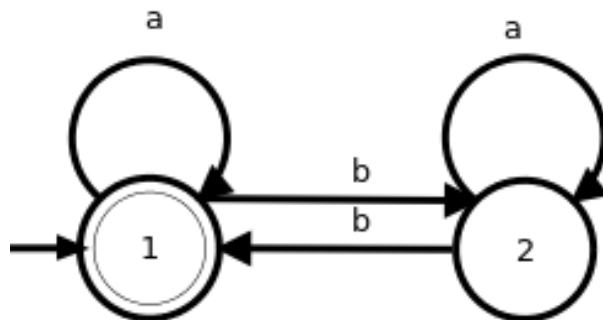
Semantik: Entfernt zunächst 'e's die vor ^ auftreten um Doppelaufreten von 'e's in der Ausgabe vorzubeugen. Im Ergebnis kann dann einfach der '^ed#' Suffix durch 'ed' ersetzt werden.

Aufgabe 3.4

A. 1.)

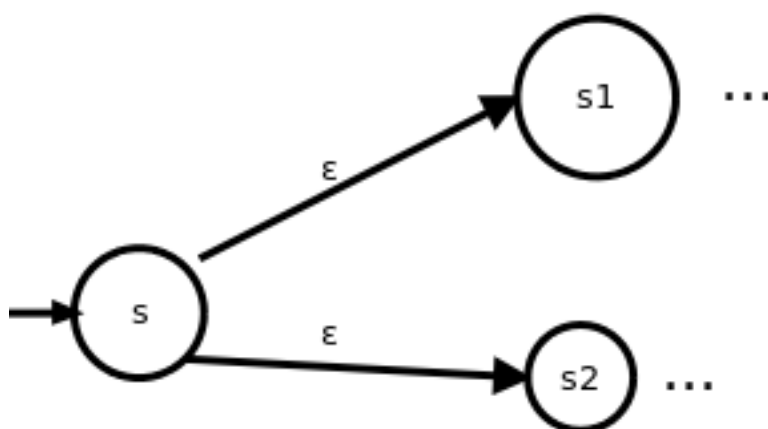


2.)

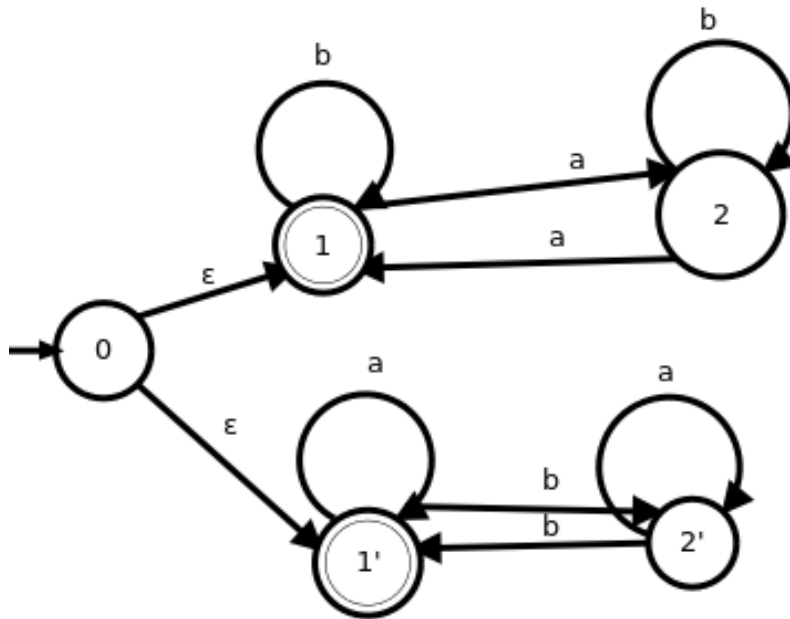


B. Der Automat A, der genau die Vereinigung der Automaten A1 und A2 erkennt, lässt sich konstruieren, indem man einen neuen Startzustand s einführt, welcher mit ϵ -Transitionen in die Startzustände von A1, also s1, und A2, also s2 geht.

Zeichnung:

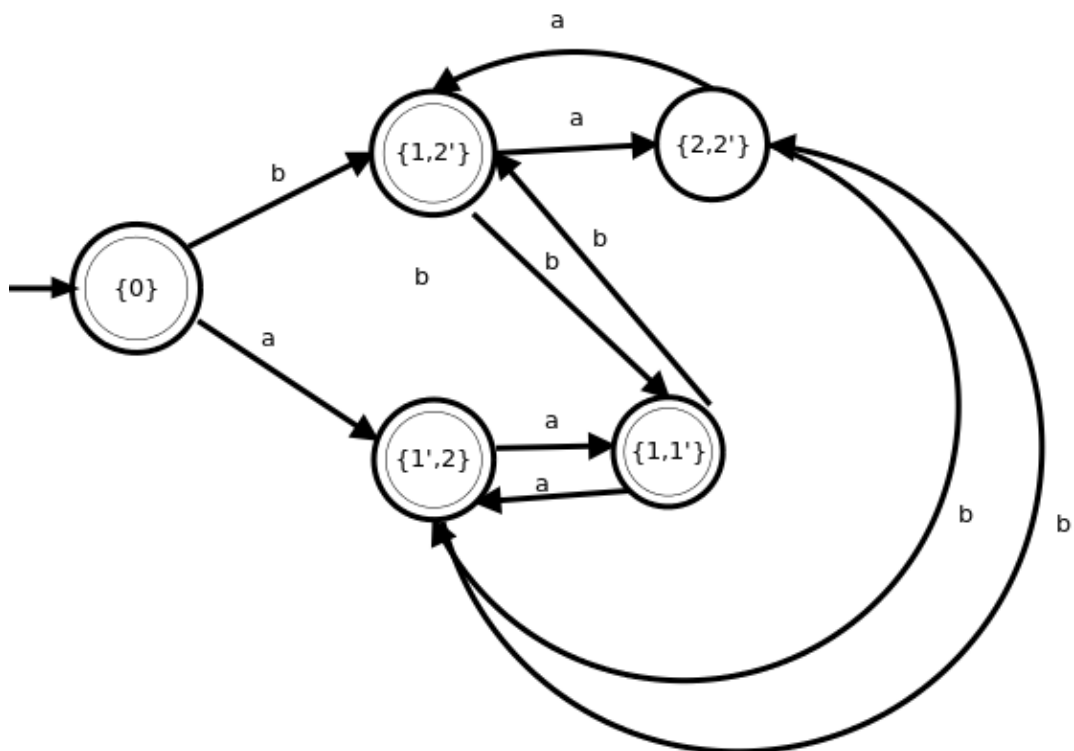


C.



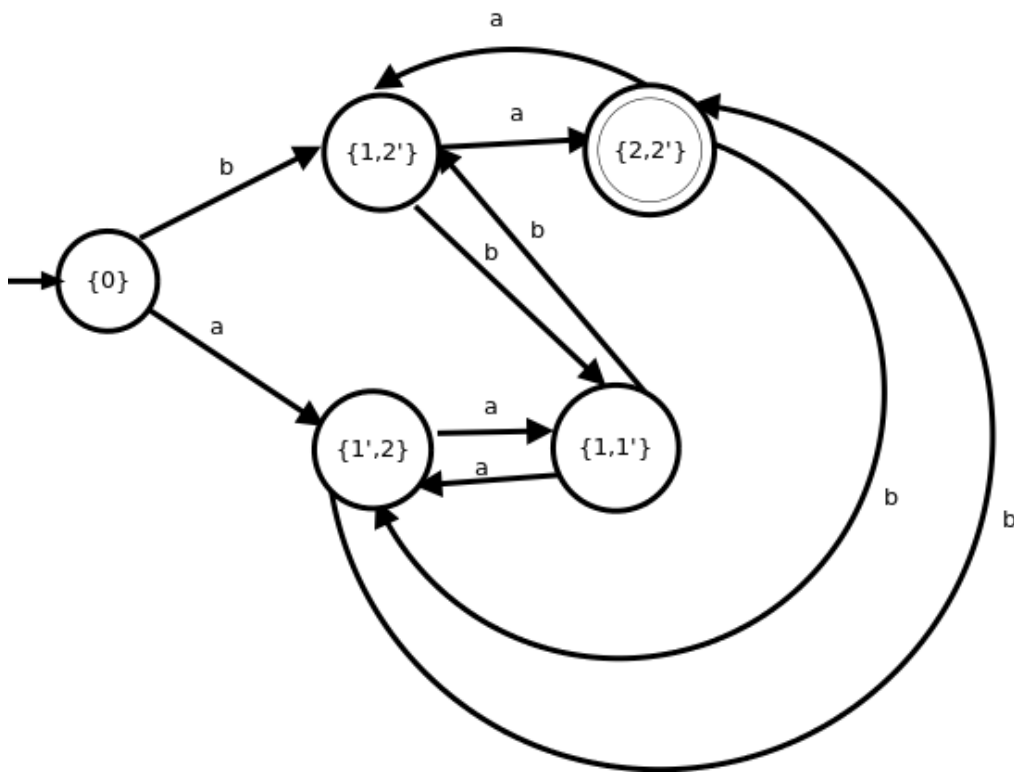
D. $S=\{0\}$, $F=\{\{0\},\{1',2\},\{1,1'\},\{1,2'\}\}$

	a	b
$\{0\}$	$\{1',2\}$	$\{1,2'\}$
$\{1',2\}$	$\{1',1\}$	$\{2,2'\}$
$\{1,2'\}$	$\{2,2'\}$	$\{1,1'\}$
$\{1,1'\}$	$\{1',2\}$	$\{1,2'\}$
$\{2,2'\}$	$\{1,2'\}$	$\{1',2\}$



Aufgabe 3.5

- A. Man muss nur alle Endzustände in Nichtendzustände wandeln und alle Nichtendzustände in Endzustände.
- B.



Aufgabe 3.6

- A. semantisch - seemännisch: 3
semantisch - romantisch: 2
seemännisch - romantisch: 5
- B. 1.) Flüchtige Tipper haben oft Verdreher in ihrem Text. Eine Lösung wäre die Wörter vor Anwendung des Algorithmus so abzuändern, dass Buchstaben, die an einer beliebigen Position in beiden Wörtern auftreten, aus beiden Wörtern entfernt werden. Beispielsweise würde man aus den Wörtern "Anwendung" und "Anwednung" dann zwei mal das leere Wort generieren. Laut dem Levenshtein Algorithmus wäre die Distanz dann null. Das würde die Genauigkeit der Rechtschreibprüfung erhöhen.
- 2.) Besoffene tippen oft wirres Zeug. Ausdrücke wie "Ich liebe dich" haben oft persönliche Konsequenzen und sollten durch Warnmeldungen ersetzt werden. Wir definieren die Levenshtein-Distanz zum Wort "Hallo" als null und zu jedem anderen Wort als eins. Dann wählt das Rechtschreibprogramm immer "Hallo" und die Nachricht wird dementsprechend entschärft.
- C. 1.) "Ich singe ein Ljue." Die Korrektur des falschen Wortes wird eindeutiger durch die Einschränkung des Themengebiets, aus dem das Wort stammt. Analysiert man den Kontext, steigt die Wahrscheinlichkeit des Wortes "Lied" als möglicher Ersatz im Vergleich zu Wörtern wie "Laus."
- 2.) "Ich zahle mit Glet." Analog zum ersten Beispiel ist es sinnvoller "Glet" durch "Geld" zu ersetzen, als durch beispielsweise "Gott".