Finite State Automata

Hans Uszkoreit

**including some slides by Julia Hockenmaier (Upenn)
and Reinhard Völler (U. Hamburg)**

Idea

- A finite state automaton is a mathematical model of a sysem with in- and outpout
- Such a system is always in one of several possible states
- Examples:
  - A circuit with n gates always is in a certain state
  - Text editors or lexical analyzers of compilers can be modelled as FSA
  - Every computer is a FSA. However, this view is not very useful because of the large number of states.

Automata

- Automata are used, to test, analyze and process sequences of symbols or actions
- E.g. for the processing of user input
- With the help of automata one can test, whether
  - The input is correct
  - it ‘s components come in the right order
  - functions have the right number an type of arguments
  - Etc

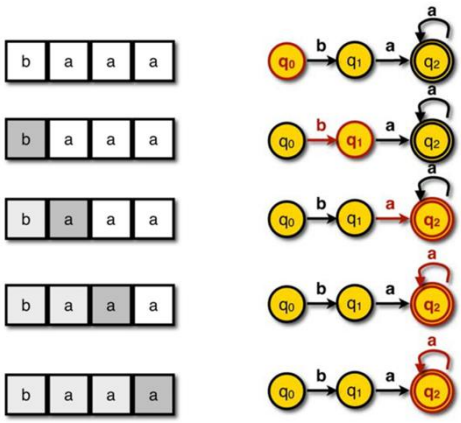
Components of an automaton

- Finite set of states
- One start state, one or more final states
- input: word of the language
- output: word of the language
- rules for all types of inputs
- Important! These rules have to be complete, all possible types of input have to be accounted for.
- An error state may be be defined

Formal Definition

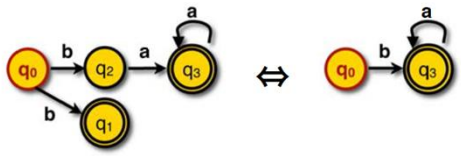
- A deterministic finite state automaton is a quintuple
  $A = (Z, E, \delta, z_0, F)$
- Z Set of States
- E Set of input symbols
- δ: Z x E $\rightarrow$ Z state transition functions
- $z_0 \in Z$ start state
- $F \subseteq Z$ Set of final states

Example

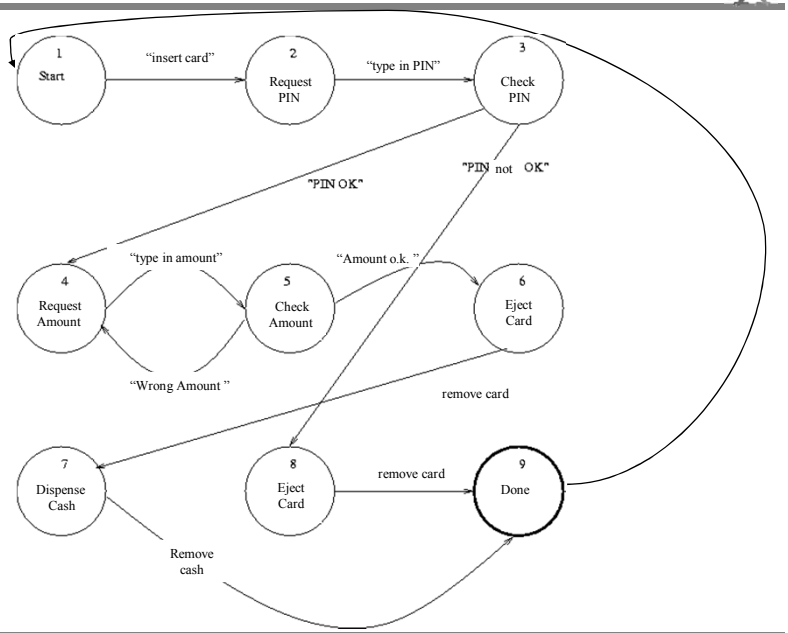


Finite State Automata (FSAs)

- Every NFA can be transformed into an equivalent DFA:



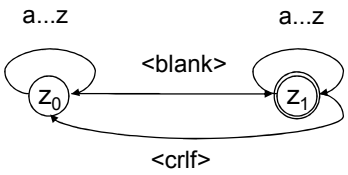
- Recognition of a string w with a DFA is linear in the length of w
- Finite-state automata define the class of regular languages
  - L1 = { anbm } = {ab, aab, abb, aaab, abb,… } is a regular language,
  - L2 = { anbn } = {ab, aabb, aaabbb,…} is not (it ‘s context-free).
- You can‘t construct an FSA that accepts all the strings in L2 and nothing else.



Example 1

- Name List:

Peter Müller
.
.
.
Doris Steckler



	z_0	z_1
a	z_0	z_1
.	z_0	z_1
.	z_0	z_1
.	z_0	z_1
z	z_0	z_1
<blank>	z_1	
<crlf>		z_0

Definition

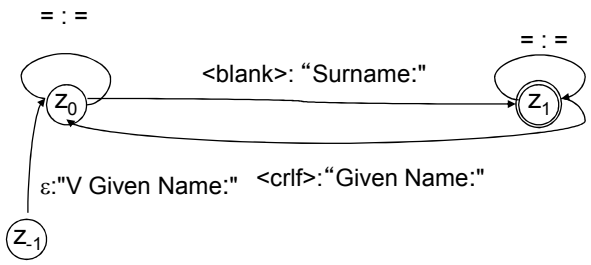
- A nondeterministic finite state automaton is a quintuple
  $A= (Z, E, \delta, z_0, F)$
- Z — Set of States
- E — Set of input symbols
- δ: Z x E $\rightarrow 2^Z$ — state transition functions
- $z_0 \in Z$ — start state
- $F \subseteq Z$ — Set of final states

Automata with output

- Ein Mealey-Automaton is a sixtuple A= (Z, E, A, δ, z_0, λ)
- Z — Set of States
- E — Set of input symbols
- A — Set of output symbols
- δ: Z x E $\rightarrow 2^Z$ — state transition functions
- λ: Z x E $\rightarrow$ A — output function
- $z_0 \in Z$ — start state
- $F \subseteq Z$ — Set of final states

Example 1a

	z_{-1}	z_0	z_1
ε : "Given Name:"	z_0		
= : =		z_0	z_1
<blank>: "Surname:"		z_1	
<crlf> : "Given Name:"			z_0



Example 2

- Telefonliste

Abbecker, Marion	0681/343239
Becker, Klaus	0631/4457328
Bruck, Günter	06897/45322
.	
.	
.	
Zeltmacher, Doris	06898/3496

Example 2

Given Name: Marion Surname: Abbecker

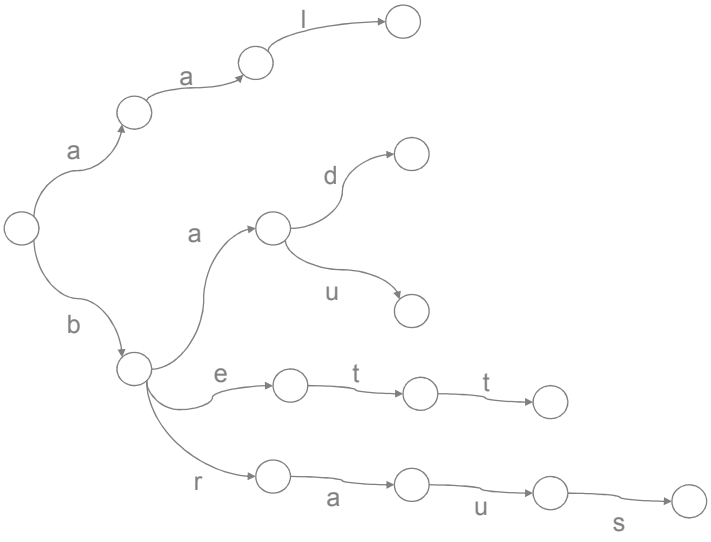
Given Name: Klaus Surname: Becker

Given Name: Günter Surname: Bruck

Morphology

- Inflection creates different forms of the same word:
  - Verbs: to be, being, I am, you are, he is, I was,
  - Nouns: one book, two books
- Derivation creates different words from the same lemma:
  - Grace, disgrace, disgraceful, disgracefully
- Compounding combines two words into a new word:
  - Cream, ice cream, ice cream cone, ice cream cone bakery
  - Eis, Speiseeis, Erdbeerspeiseeis Erdbeersspeiseeiskugel, Erdbeersspeiseeiskugelverkäufer
- Word formation is productive: New words are subject to all of these processes:
  - Google: Googler, to google, to ungoogle, to misgoogle, googlification,
  - ungooglification, googlified, Google Maps, Google Maps service,...

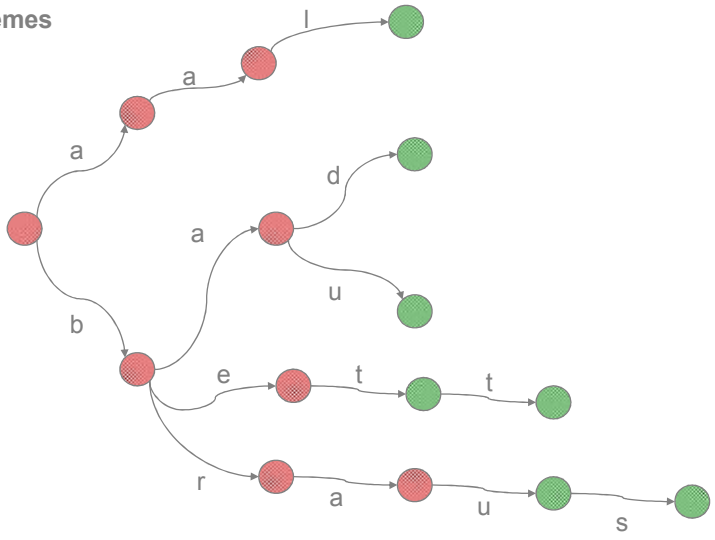
Letter Tree



Letter Tree

Verb stem-morphemes

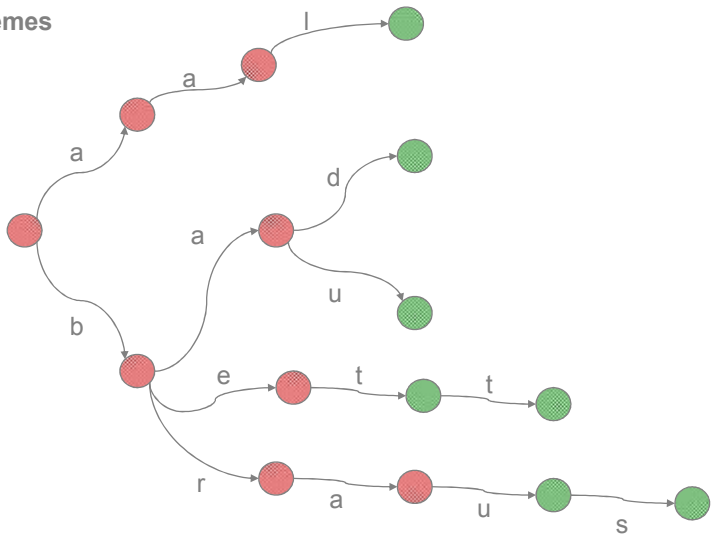
aal
bad
bau
bet
bett
brau
braus



VSTEM: aal

Verb stem-morphemes

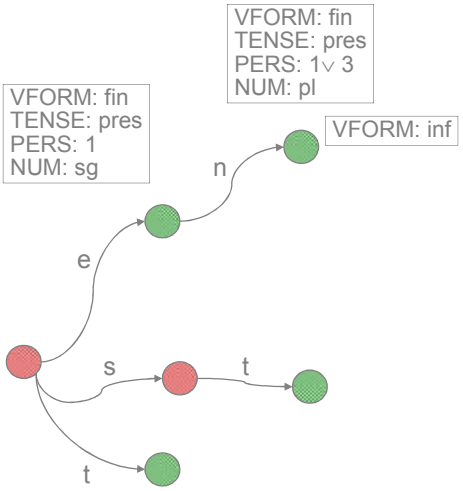
aal
bad
bau
bet
bett
brau
braus



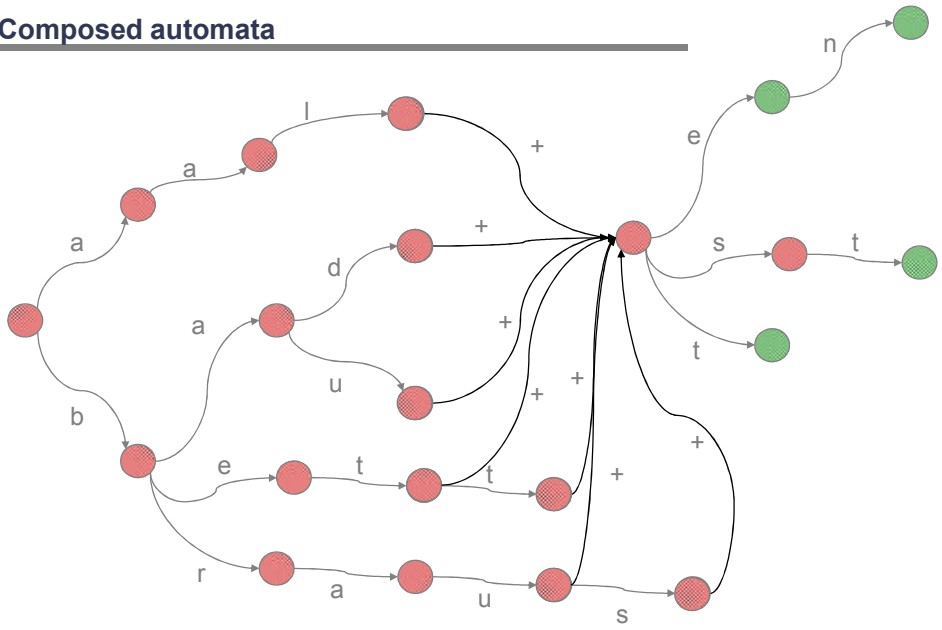
Letter Tree

Verb suffix - morpheme

-e
-en
-st
-t



Composed automata



Examples

- a a l + s t a a l + s t
- a a l s t a a l e s t
- b a d + s t b a d + s t
- b a d s t b a d e s t

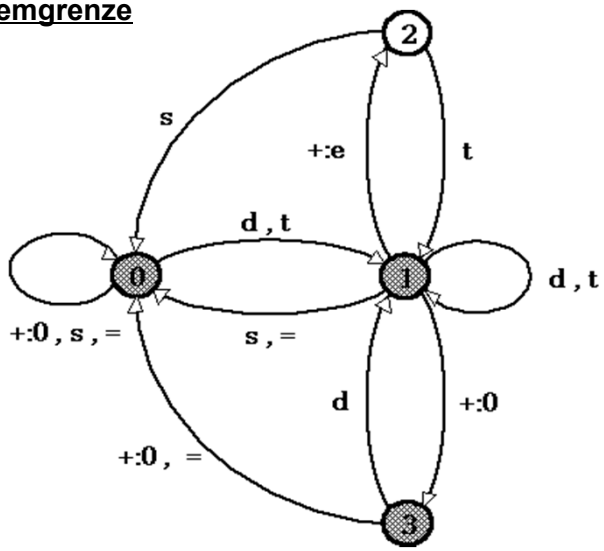
Zwei-Ebenen Morphologie

e-Epenthese an der Morphemgrenze
(vereinfacht)

Regel: +:e <=> {d, t} _ {s, t}

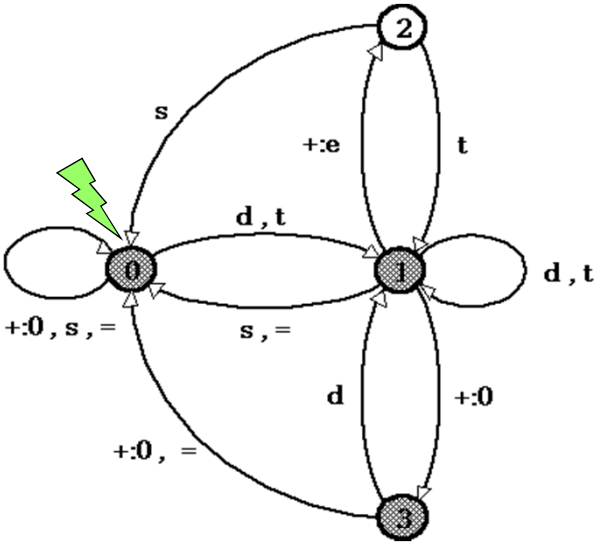
Übergangstabelle:

	0	1	2	3
+:e	-	2	-	-
+:0	0	3	-	0
d:d	1	1	-	1
s:s	0	0	0	-
t:t	1	1	1	-
=:=	0	0	-	0

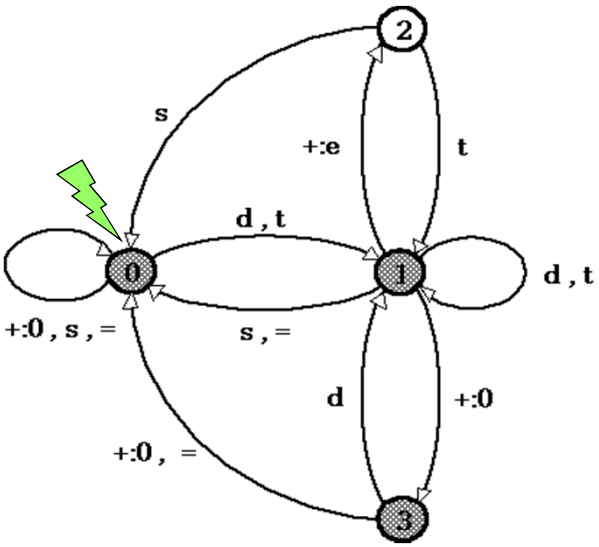


Abgewandeltes Beispiel von Prof. Dr. Harald Trost (Wien)

b a d + s t
b a d s t

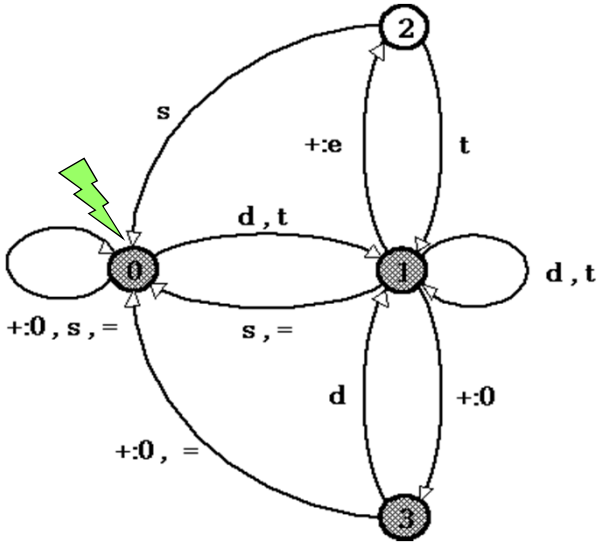


b a d + s t
b a d s t

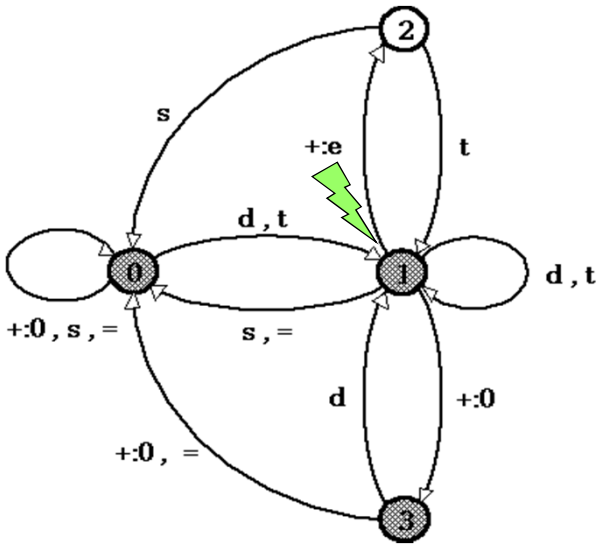


Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d s t

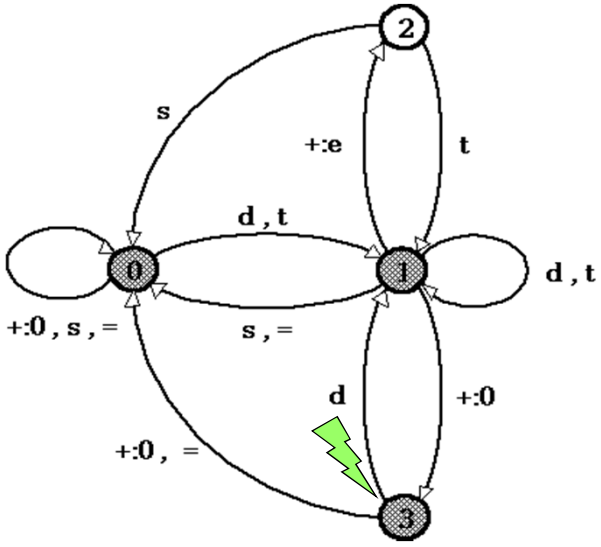


b a d + s t
b a d s t

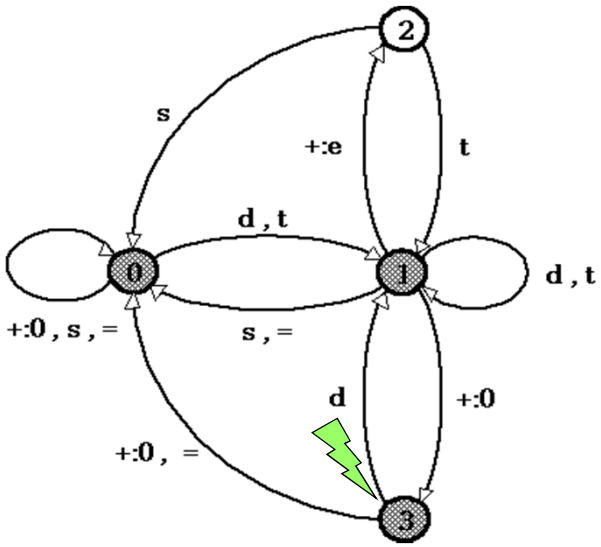


Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d s t

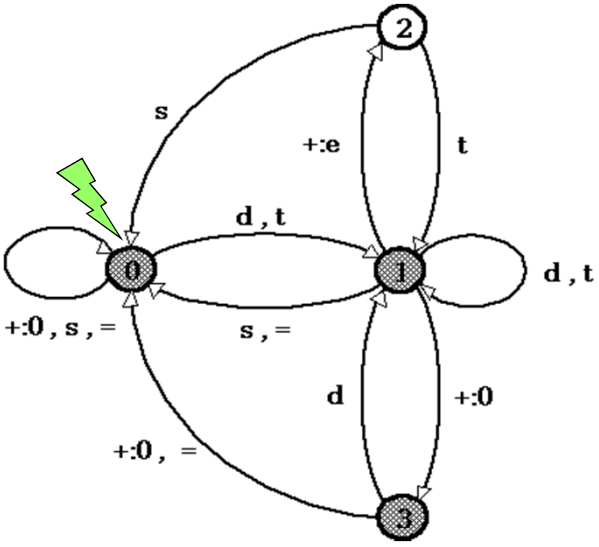


b a d + s t
b a d s t

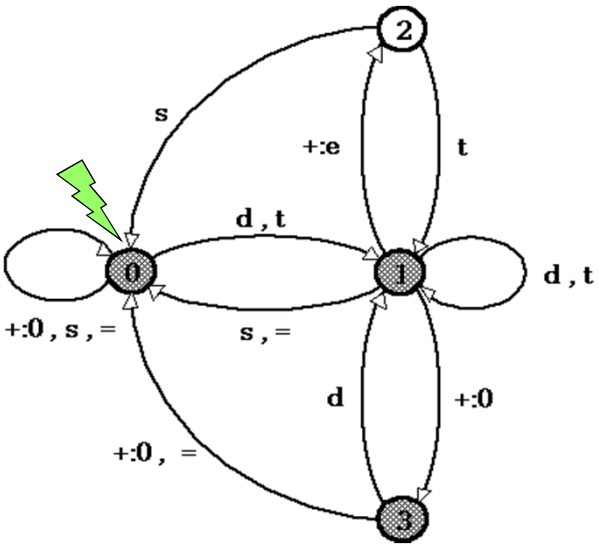


Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d e s t

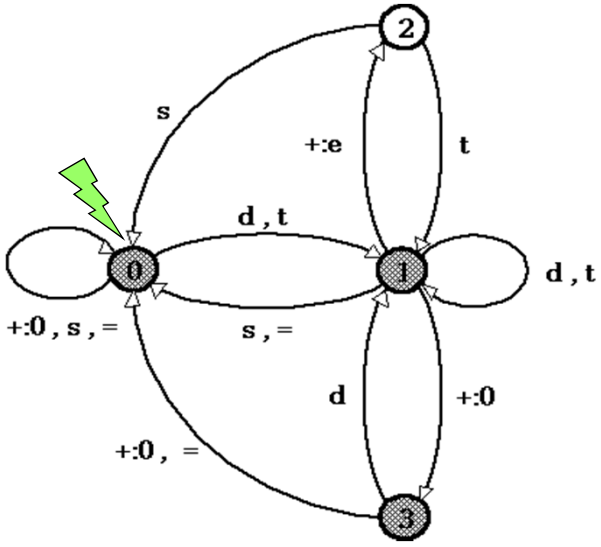


b a d + s t
b a d e s t

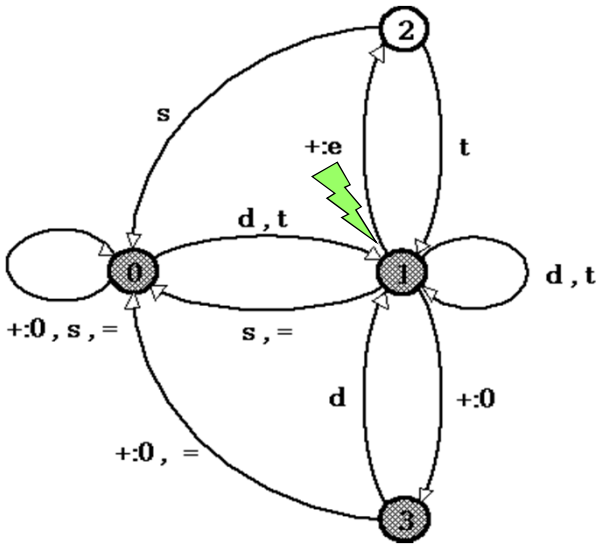


Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d e s t

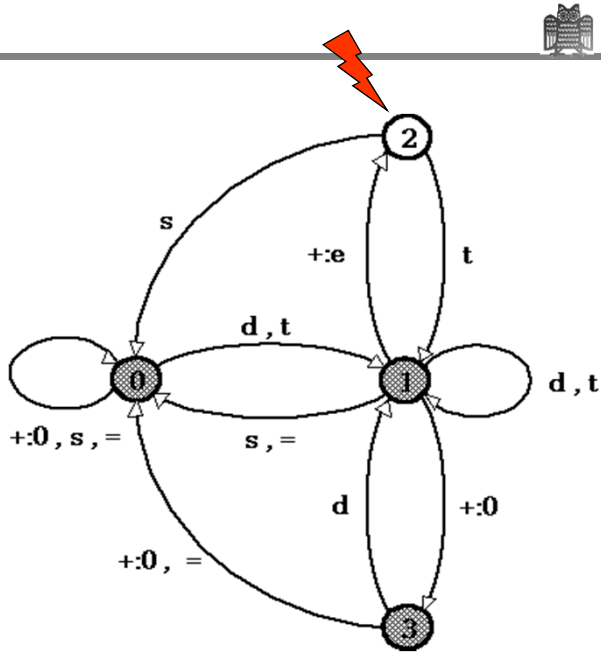


b a d + s t
b a d e s t

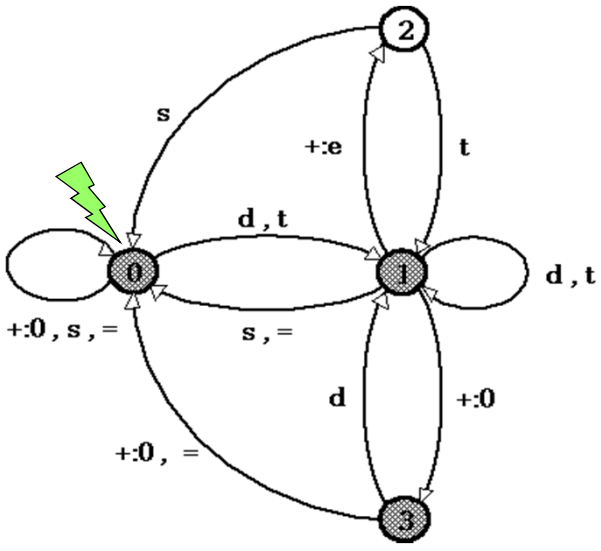


Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d e s t

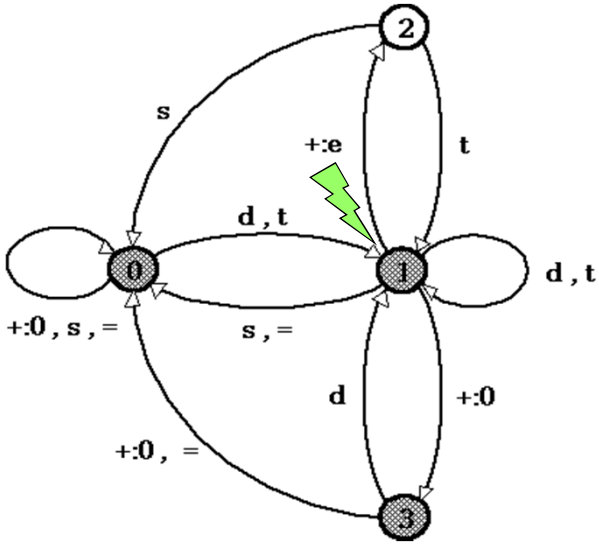


b a d + s t
b a d e s t



Foundations of Language Science and Technology • M.Sc. Program in LST • Winter Semester 13/14

b a d + s t
b a d e s t



Exercises

1. Design a transductor, that prevents a triple consonant before a vowel at the morpheme boundary (old German orthography rule: Schiffahrt vs. Balletttruppe)!

2. Design a transductor that prevents the plural form “kisss” but allows “kisses” and also prevents “wishs” but not “wishes”!

3. Design a transdurctor that converts a number of Euro cents into a proper German Euro amount (“3200000” → “EUR 32.000,00”).

4. a) Build a transductor that abbreviates first and middle names:
   “William Brown” → “W. Brown”
   “Peter Albert Tucker” → “P. A. Tucker”

   b) Can you build a transductor that does the following transformations
   “William Brown” → “Brown, W.”
   “Kate Plumsfield” → “Plumsfield, K.”
   and what about “Peter Albert Tucker” → “Tucker, P. A.”