Transducer und Bimaschinen als Finite-State Technologien mit Ein- und Ausgabe

Vertiefende Einführung in die Computerlinguistik

Marcel Braasch

Ludwig-Maximilians-Universität

München, den 04.02.2021

Inhalt

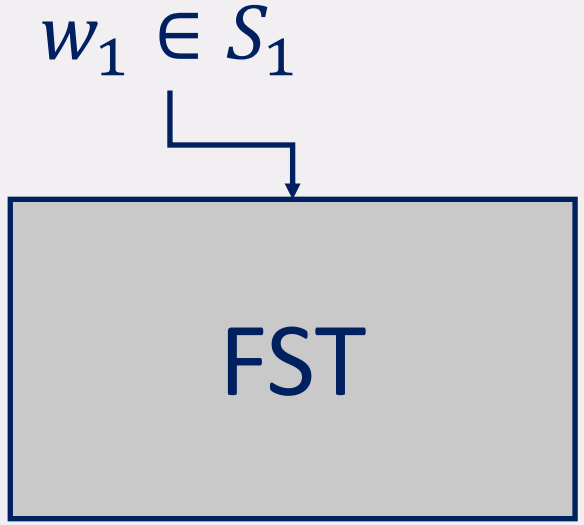
- Einführung
- Definition
- Beispiel
- Anwendungen

Transducer

Einführung

Ein Transducer (FST)

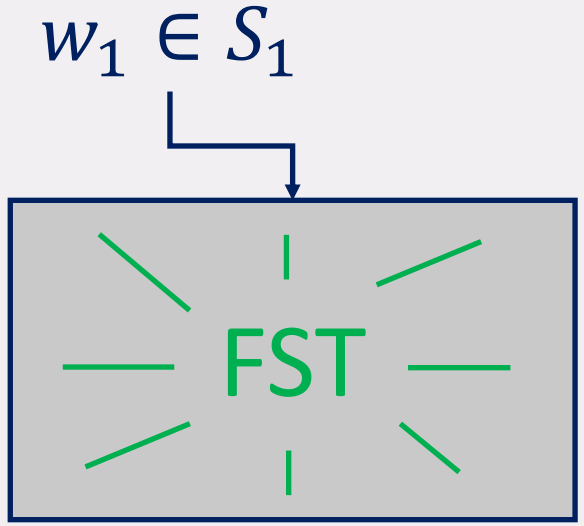
- ist ein endlicher Automat
- erwartet eine Eingabe der Sprache S_1



Einführung

Ein Transducer (FST)

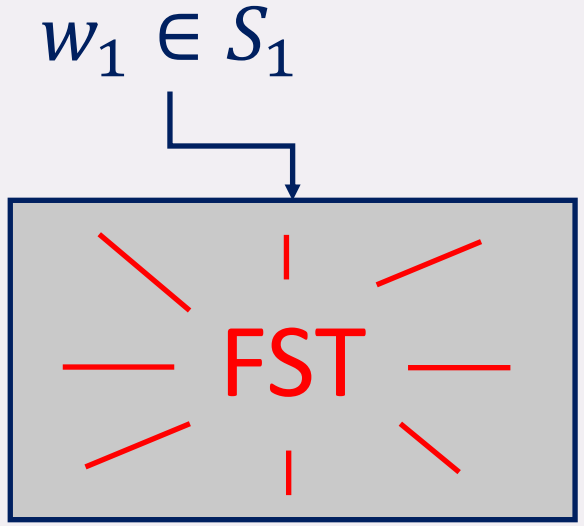
- ist ein endlicher Automat
- erwartet eine Eingabe der Sprache S_1
- akzeptiert oder lehnt ab



Einführung

Ein Transducer (FST)

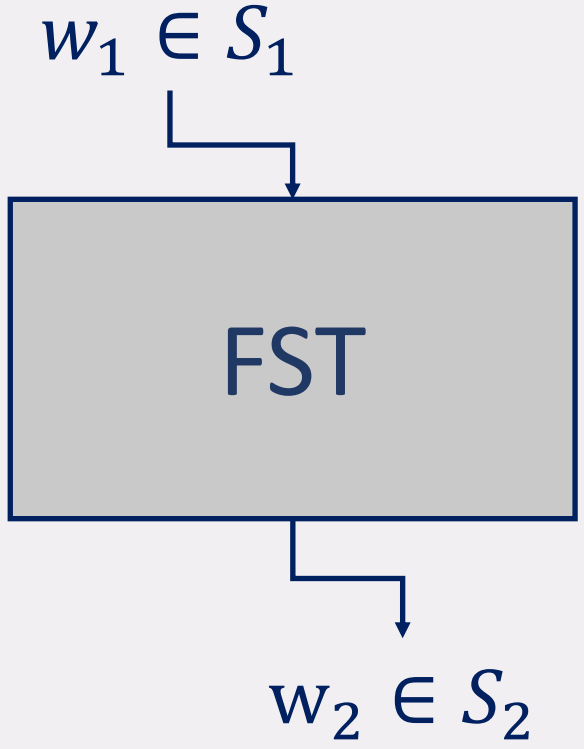
- ist ein endlicher Automat
- erwartet eine Eingabe der Sprache S_1
- akzeptiert oder lehnt ab



Einführung

Ein Transducer (FST)

- ist ein endlicher Automat
- erwartet eine Eingabe der Sprache S_1
- akzeptiert oder lehnt ab
- produziert eine Ausgabe der Sprache S_2
- beschreibt **Relationen** zwischen formalen Sprachen
- ist isomorph zu regulären Relationen
- ermöglicht Vielzahl algebraischer Operationen



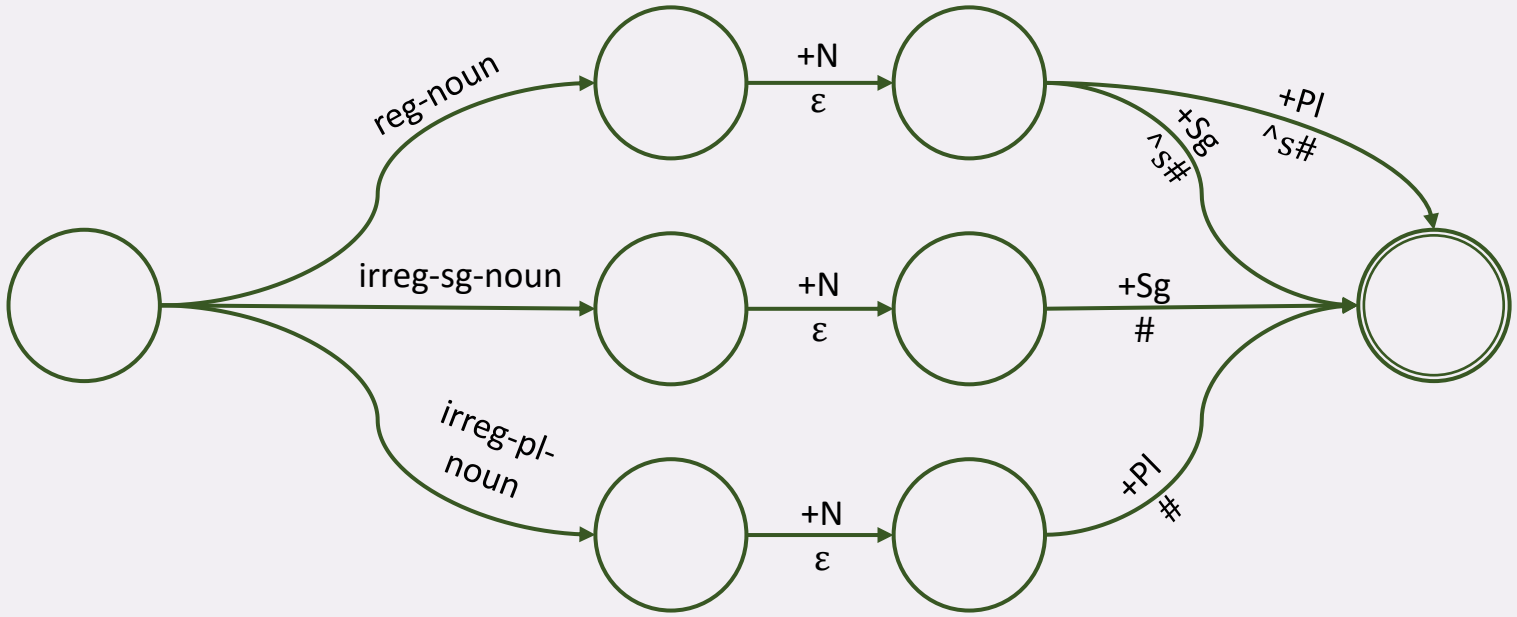
Formale Definition

Ein FST ist ein 7-Tupel $A = (\Sigma, Q, q_0, \delta, \omega, F, \Gamma)$ wobei

$Q \neq \emptyset$	die **Menge der Zustände**,
$q_0 \in Q$	der **Startzustand**,
$\delta: Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$	die **Zustandsübergangsfunktion**,
$\omega: Q \times \Sigma \cup \{\varepsilon\} \times Q \rightarrow \Gamma^*$	die **Ausgabefunktion**,
$\Sigma \neq \emptyset$ und $\lvert\Sigma\rvert < \infty$	das **Eingabealphabet**,
$F \subseteq Q$	die **Menge der akzept. Zustände**,
$\Gamma \neq \emptyset$ und $\lvert\Gamma\rvert < \infty$	das **Ausgabealphabet** ist.

Beispiel I

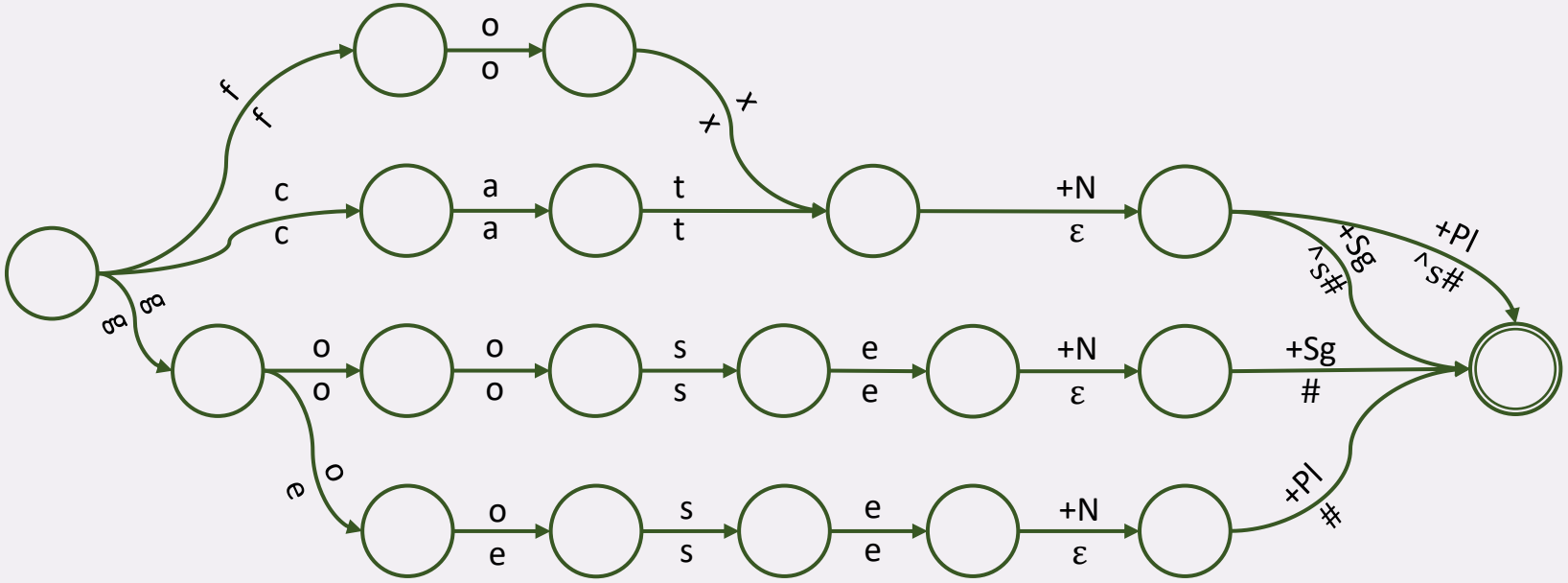
Schematischer Aufbau eines Transducers für Englische Numerusflexion.

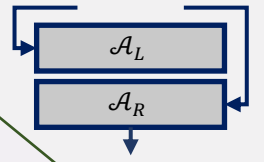


Beispiel II

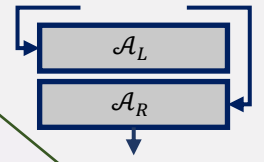
Expansion von `reg-noun`, `irreg-sg-noun` und `irreg-pl-noun`

Mapping	
Reg-noun	fox
	cat
Irreg-sg-noun	goose
	sheep
Irreg-pl-noun	goose
	sheep





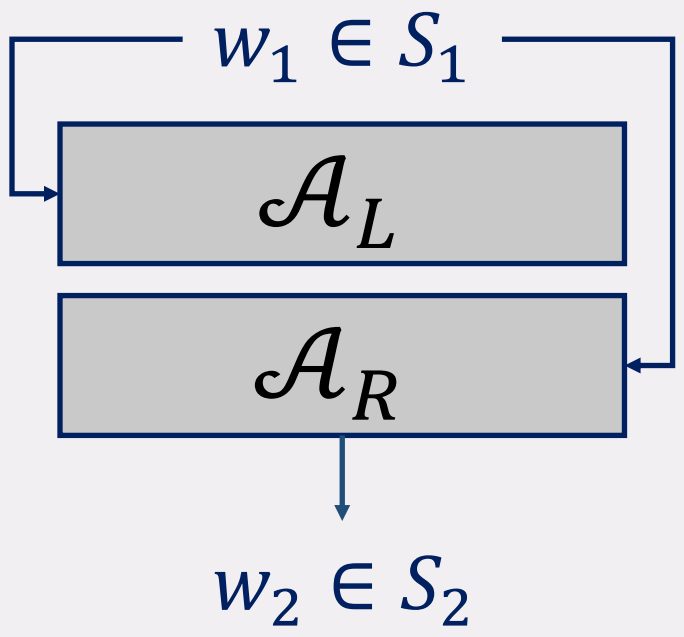
Bimaschinen

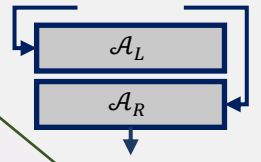


Einführung

Eine Bimaschine

- ist eine Kombination aus zwei endlichen Automaten (DFA)
- erwartet und liest eine Eingabe der Sprache S_1 bidirektional
- produziert eine Ausgabe der Sprache S_2
- wird meist für effizientes “Rule Rewriting“ verwendet





Formale Definition

Eine Bimaschine ist ein 4-Tupel $\mathcal{B} = (\mathcal{M}, \mathcal{A}_L, \mathcal{A}_R, \psi)$ wobei

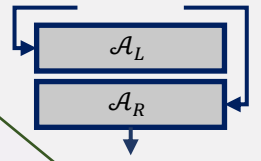
$\mathcal{M} \neq \emptyset$ das **Ausgabealphabet**,

$\mathcal{A}_L = (\Sigma, Q_L, s_L, Q_L, \delta_L)$ der **linke DFA der Bimaschine**,

$\mathcal{A}_R = (\Sigma, Q_R, s_R, Q_R, \delta_R)$ der **rechte DFA der Bimaschine** mit

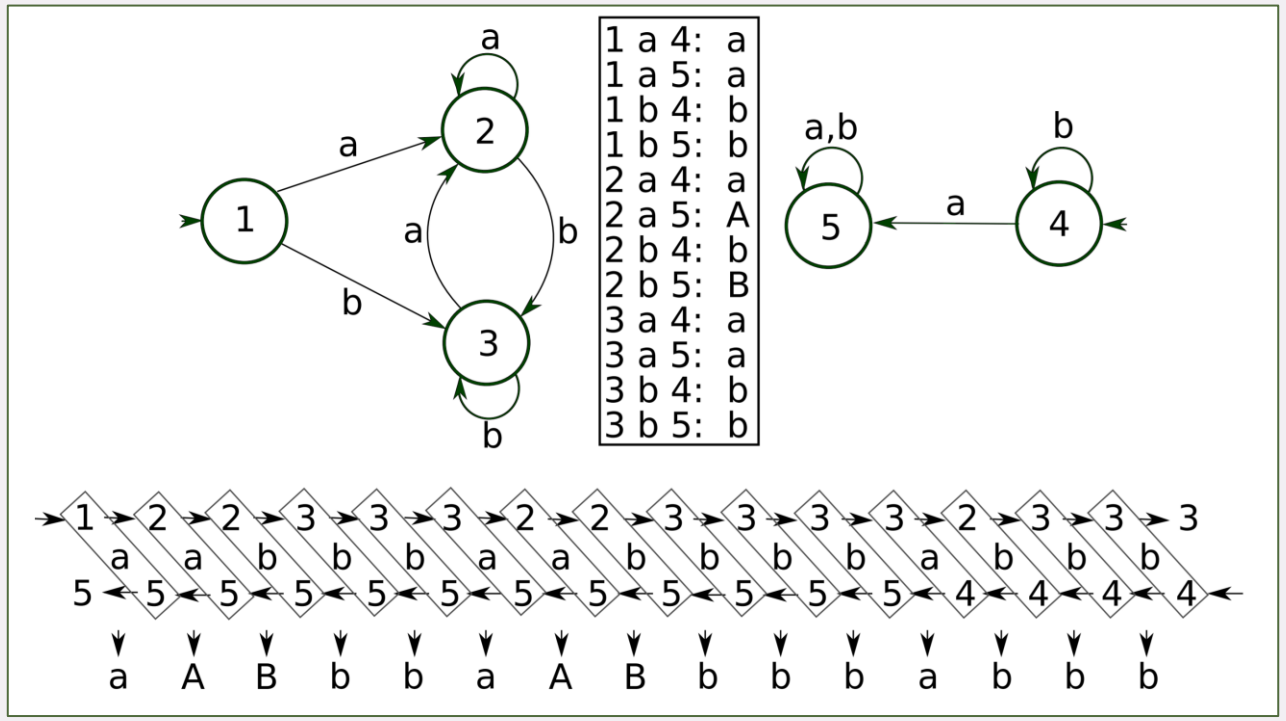
- $\Sigma \neq \emptyset$ und $|\Sigma| < \infty$ dem **Eingabealphabet**
- Q_L und Q_R mit $Q \neq \emptyset$ den **(akzeptierenden) Zuständen**,
- $q_L \subseteq Q_L$ und $q_R \subseteq Q_R$ den **Startzuständen**,
- δ_R und δ_L: $Q \times \Sigma \rightarrow 2^Q$ der **Ausgabefunktion der DFAs**.

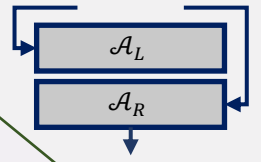
ψ: $(Q_L \times \Sigma \times Q_R)$ der **Ausgabefunktion der Bimaschine**.



Beispiel I

Die Bimaschine übersetzt Buchstaben a und b in deren Großbuchstaben, falls der linke Nachbar ein a ist und im späteren Verlauf des Texts ein weiteres a erscheint.





Anwendungsbeispiele

Doychinova et al. (2004) benutzen Bimaschinen für einen High-Performance Part-Of-Speech Tagger für Ungarisch

- Regeln hinzufügen in real-time ohne auf Kompilierung zu warten
- Programm verarbeitet 35.000 Wörter/Sek mit 98.4% Precision

- Vielfältige Anwendungsmöglichkeiten im Bereich Tokenization, Word Sense Disambiguation, Unkown Word Guessing, Morphem-extraktion, etc.
- Beispiel: Unknown-Word-Guesser kann 73 Regeln mit einer Bimaschine darstellen

Referenzen

S. 3 - 10

Jurafsky, D. (2000). *Speech & language processing*. Chapter 3. Pearson Education India.

S 11 - 13

Mihov, S., & Schulz, K. U. (2019). *Finite-State Techniques* (Vol. 60). Seite 122. Cambridge University Press.

S. 14

Veselka Doychinova, Stoyan Mihov. High Performance Part-of-Speech Tagging of Bulgarian. Proceedings of AIMSA-2004, LNAI #3192, pp. 246-255, 2004 zitiert in

Christoph Ringlstetter, Klaus U. Schulz, Florian Schiel. Seminar Dialogsysteme. Vorlesung 3. 2005.