

Maschinelles Übersetzen natürlicher Sprachen

7. Übungsblatt

2015-12-03

Aufgabe 1

Gegeben seien die Alphabete

$$\Gamma = \{S, NP, VP, ADJ, NN, VB, DT\},$$

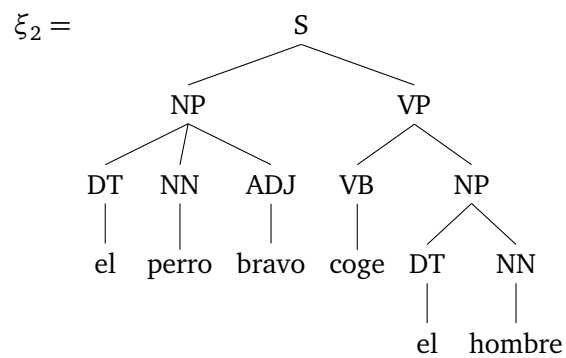
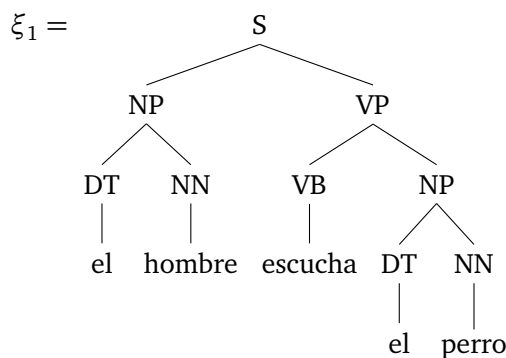
$$\Sigma = \Gamma \cup \{el, perro, bravo, escucha, coge, hombre\},$$

$$\Delta = \Gamma \cup \{der, den, Hund, wilde, wilden, Mann, fängt, hört\}.$$

Wir nehmen einen extended tree transducer $\mathcal{M} = (Q, \Sigma, \Delta, q, R)$ an mit der Zustandsmenge $Q = \{q, n, a\}$ und der Regelmenge R wie folgt:

$$\begin{array}{ll} q(S(x_1 : NP, VP(x_2 : VB, x_3 : NP))) \xrightarrow{\rho_1} S(n(x_1), VP(q(x_2), a(x_3))) & q(NN(perro)) \xrightarrow{\rho_6} NN(Hund) \\ n(NP(DT(el), x_1 : NN, x_2 : ADJ)) \xrightarrow{\rho_2} NP(DT(der), n(x_2), q(x_1)) & q(NN(hombre)) \xrightarrow{\rho_7} NN(Mann) \\ a(NP(DT(el), x_1 : NN, x_2 : ADJ)) \xrightarrow{\rho_3} NP(DT(den), a(x_2), q(x_1)) & q(VB(escucha)) \xrightarrow{\rho_8} VB(hört) \\ n(NP(DT(el), x_1 : NN)) \xrightarrow{\rho_4} NP(DT(der), q(x_1)) & q(VB(coge)) \xrightarrow{\rho_9} VB(fängt) \\ a(NP(DT(el), x_1 : NN)) \xrightarrow{\rho_5} NP(DT(den), q(x_1)) & a(ADJ(bravo)) \xrightarrow{\rho_{10}} ADJ(wilden) \\ & n(ADJ(bravo)) \xrightarrow{\rho_{11}} ADJ(wilde) \end{array}$$

Bestimmen Sie, unter Angabe der entsprechenden Ableitungen, die Übersetzungen der folgenden Parsebäume:



Vollziehen Sie für eine dieser Ableitungen die Arbeitsweise der Projektionsfunktionen $\pi_\Sigma : D_{\mathcal{M}} \rightarrow T_\Sigma$ sowie $\pi_\Delta : D_{\mathcal{M}} \rightarrow T_\Delta$ nach! Wie müssten die Regeln von \mathcal{M} geändert werden, damit der Eingabe- bzw. Ausgabebaum einer Ableitung nicht eindeutig bestimmt ist?

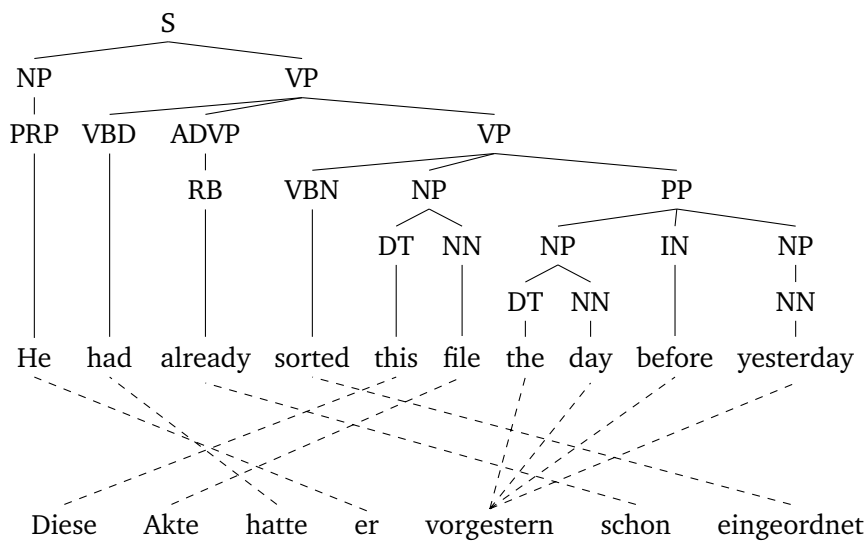
Aufgabe 2

Wir betrachten noch einmal die Übersetzung von arithmetischen Ausdrücken in Infix-Notation in umgekehrte polnische Notation (vgl. 6. Übungsblatt, Aufgabe 1). Entwerfen Sie einen XTT \mathcal{M} , welcher einen Parsebaum $\xi \in PT_G$ eines Infix-Ausdrucks in den entsprechenden Parsebaum einer CFG G' für RPN-Ausdrücke übersetzt!

Zusatzaufgabe: Entwerfen Sie außerdem einen yXTT \mathcal{M}' , welcher direkt den Yield des abgeleiteten Baumes ausgibt!

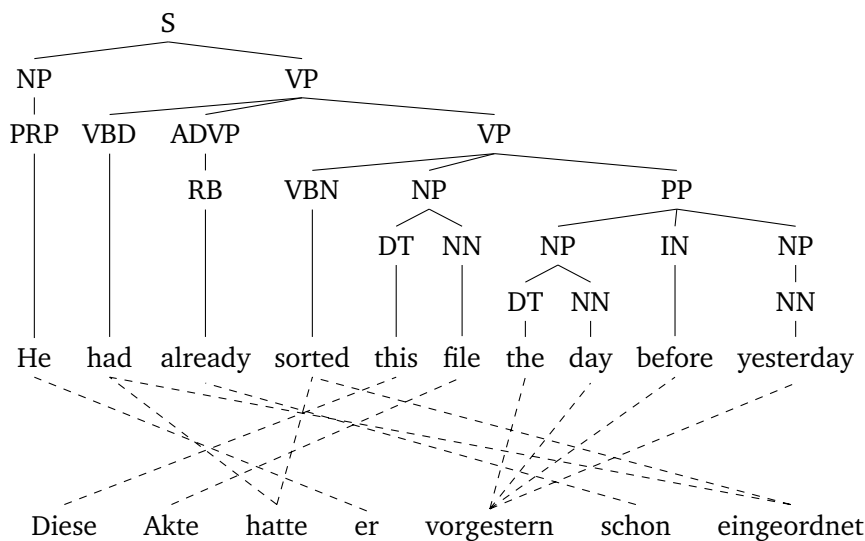
Aufgabe 3 (Regelextraktion für yXTT)

Gegeben ist der folgende Alignment-Graph $G(\xi, f, A)$:



Nutzen Sie die in der Vorlesung vorgestellte Technik, um Regeln für einen yXTT zu erzeugen. Kennzeichnen Sie dabei die *frontier nodes* des Graphen!

Zusatzaufgabe: Inwiefern verändern sich die Regeln, wenn wir zum Extrahieren den folgenden Alignment-Graphen $G(\xi, f, A')$ verwenden? Welche Regeln versprechen eine bessere Übersetzung?



Aufgabe 4 (Konstruktion von RTG)

Gegeben sei das Alphabet $\Sigma = \{\sigma, \tau, \gamma, \alpha\}$. Konstruieren Sie eine RTG \mathcal{G} , so dass $\xi \in L(\mathcal{G})$ genau dann, wenn

- die Symbole σ und τ nur mit Rang 2 in ξ vorkommen, γ nur mit Rang 1 und α mit Rang 0, und außerdem
- das Symbol τ genau einmal in ξ enthalten ist.

Geben Sie für den Baum $\sigma(\alpha, \tau(\alpha, \gamma(\alpha)))$ eine Ableitung von \mathcal{G} an!

Aufgabe 5 (Top-Down-Determinismus)

Wir nehmen das Alphabet $\Sigma = \{\sigma, \alpha, \beta\}$ an. Warum existiert für die Baumsprache $L = \{\sigma(\alpha, \beta), \sigma(\beta, \alpha)\}$ keine top-down-deterministische RTG \mathcal{G} mit $L = L(\mathcal{G})$?

Wie stehen die Sprachklassen REG und REG_{det} daher also in Beziehung, wenn REG die Klasse aller durch reguläre Baumgrammatiken, und REG_{det} die Klasse aller durch top-down-deterministische reguläre Baumgrammatiken erzeugten Baumsprachen bezeichnet?

Welchen Vorteil bieten deterministische Grammatiken jedoch im Vergleich zu nicht deterministischen?

Aufgabe 6 (Input- and Output-Product)

Sei $\Sigma = \{\sigma, \gamma, \alpha, \beta\}$. Wir nehmen einen XTT $\mathcal{M} = (\{q, p\}, \Sigma, \Sigma, q, R)$ an mit den Regeln

$$\begin{array}{ll} q(\sigma(x_1, x_2)) \xrightarrow{\rho_1} \sigma(q(x_2), p(x_1)) & q(\alpha) \xrightarrow{\rho_3} \alpha \\ p(\gamma(x_1)) \xrightarrow{\rho_2} \gamma(\gamma(\gamma(p(x_1)))) & p(\alpha) \xrightarrow{\rho_4} \beta, \end{array}$$

sowie RTGs $\mathcal{G}_1 = (\{S, A\}, \Sigma, S, R_1)$ und $\mathcal{G}_2 = (\{S, B, C\}, \Sigma, S, R_2)$, wobei die Regeln aus R_1 durch

$$\begin{array}{ll} S \rightarrow \sigma(A, S) & A \rightarrow \gamma(A) \\ S \rightarrow \alpha & A \rightarrow \alpha \end{array}$$

und die aus R_2 durch

$$\begin{array}{ll} S \rightarrow \sigma(S, B) & S \rightarrow \alpha \\ B \rightarrow \gamma(C) & B \rightarrow \beta \\ C \rightarrow \gamma(B) & \end{array}$$

gegeben sind.

Bilden Sie mittels der in der Vorlesung vorgestellten Konstruktion das *Input- and Output-Product* $\text{Prod}(\mathcal{G}_1, \mathcal{M}, \mathcal{G}_2)$.

Fleißaufgabe: Berechnen Sie $\text{Prod}(\mathcal{G}'_1, \mathcal{M}, \mathcal{G}_2)$ für $\mathcal{G}'_1 = (\{S, A, B, C\}, \Sigma, S, R'_1)$ mit den Regeln

$$\begin{array}{lll} S \rightarrow \sigma(A, S) & S \rightarrow \alpha & A \rightarrow \alpha \\ A \rightarrow \gamma(B) & B \rightarrow \gamma(C) & C \rightarrow \gamma(A). \end{array}$$