Sei $G = (\{\mathrm{a}, \mathrm{b}\}, \{S, A\}, \{S\}, \{p_1, p_2, p_3\})$ eine LCFRS mit

$$p_1 = S \to \langle x_{1,1} x_{1,2} \rangle(A), \qquad \mu_G(p_1) = 0{,}6$$
$$p_2 = A \to \langle \varepsilon, \varepsilon \rangle(), \qquad \mu_G(p_2) = 0{,}4$$
$$p_3 = A \to \langle \mathrm{a} x_{1,1} \mathrm{b}, \mathrm{a} x_{1,2} \mathrm{b} \rangle(A), \text{ und} \qquad \mu_G(p_3) = 1$$

Wort w: abab

(Ruprecht, 2017)

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle(A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle(A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Implementierung und Evaluierung eines inkrementellen Parsers in Vanda

Komplexpraktikum - Endpräsentation

Niklas Wünsche

Fakultät Informatik
TU Dresden

20. Juli 2018

Übersicht

- Was bedeutet Parsen?
- Gewichtete LCFRS
- Gewichtete Deduktionssysteme
- Inkrementeller Parser
- Evaluation der Implementierung

Übersicht

- Was bedeutet Parsen?
- Gewichtete LCFRS
- Gewichtete Deduktionssysteme
- Inkrementeller Parser
- Evaluation der Implementierung

Übersicht

- Was bedeutet Parsen?
- Gewichtete LCFRS
- Gewichtete Deduktionssysteme
- Inkrementeller Parser
- Evaluation der Implementierung

Übersicht

- Was bedeutet Parsen?
- Gewichtete LCFRS
- Gewichtete Deduktionssysteme
- Inkrementeller Parser
- Evaluation der Implementierung

Übersicht

- Was bedeutet Parsen?
- Gewichtete LCFRS
- Gewichtete Deduktionssysteme
- Inkrementeller Parser
- Evaluation der Implementierung

Was bedeutet Parsen?

- Erkennen: Eingabe: w und G, Ausgabe: $w \in L(G)$
- Parsen: Eingabe: w und G, Ausgabe: Ableitungsbäume von w in G

Was bedeutet Parsen?

- Erkennen: Eingabe: w und G, Ausgabe: $w \in L(G)$
- Parsen: Eingabe: w und G, Ausgabe: Ableitungsbäume von w in G

Motivation von LCFRS

- Diskontinuitäten der natürlichen Sprache darstellen
  - z.B. „mer em **Hans** es huus **hälfed** aastriiche. “ im Schweizerdeutsch (Shieber, 1985)
- Wörter der Grammatik dennoch in polynomieller Zeit parsbar
- Lösung: *Mildly Context-Sensitive Formalisms* (Kallmeyer, 2010) vgl. (Joshi, 1985)
  - *String Linear Context-Free Rewriting Systems (LCFRS)*

(Vijay-Shanker, Weir & Joshi, 1987)

Motivation von LCFRS

- Diskontinuitäten der natürlichen Sprache darstellen
  - z.B. „mer em **Hans** es huus **hälfed** aastriiche. “ im Schweizerdeutsch (Shieber, 1985)
- Wörter der Grammatik dennoch in polynomieller Zeit parsbar
- Lösung: *Mildly Context-Sensitive Formalisms* (Kallmeyer, 2010) vgl. (Joshi, 1985)
  - *String Linear Context-Free Rewriting Systems (LCFRS)*

(Vijay-Shanker et al., 1987)

Motivation von LCFRS

- Diskontinuitäten der natürlichen Sprache darstellen
  - z.B. „mer em **Hans** es huus **hälfed** aastriiche. “ im Schweizerdeutsch (Shieber, 1985)
- Wörter der Grammatik dennoch in polynomieller Zeit parsbar
- Lösung: *Mildly Context-Sensitive Formalisms* (Kallmeyer, 2010) vgl. (Joshi, 1985)
  - *String Linear Context-Free Rewriting Systems (LCFRS)*

(Vijay-Shanker et al., 1987)

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathrm{a}x_{1,1}\mathrm{b}, \mathrm{a}x_{1,2}\mathrm{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathsf{a}x_{1,1}\mathsf{b}, \mathsf{a}x_{1,2}\mathsf{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathrm{a}x_{1,1}\mathrm{b}, \mathrm{a}x_{1,2}\mathrm{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathrm{a}x_{1,1}\mathrm{b}, \mathrm{a}x_{1,2}\mathrm{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathrm{a}x_{1,1}\mathrm{b}, \mathrm{a}x_{1,2}\mathrm{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Gewichtete LCFRS

(Vijay-Shanker et al., 1987), (Goodman, 1999)

- 4-Tupel $G = (\Sigma, N, P, S)$
- Besonderheit: Regeln der Form
  $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$,
- z.B. $A \to \langle \mathrm{a}x_{1,1}\mathrm{b}, \mathrm{a}x_{1,2}\mathrm{b} \rangle (A) \in P$
  - Variablen durch Komponenten der Argumente ersetzen
  - Nicht-löschend und linear
  - Regel wird Gewicht zugeordnet
- Regelgewicht $\mu_G \colon P \to [0, 1]$
  - Viterbi-Semiring $([0, 1], \max, \cdot, 0, 1)$
  - Gewicht einer Ableitungsbaums: Produkt der Gewichte der angewandten Regeln
  - Gewicht eines Wortes $w \in \Sigma^*$: Maximales Gewicht aller Ableitungsbäume von w

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente $ax_{1,1}b$ und Wort abab:
  $(0,1)x_{1,1}b$ für abab
  $(2,3)x_{1,1}b$ für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Ranges

(Kallmeyer, 2010), (Ruprecht, 2017)

- *Range*: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b und Wort abab:
  $(0,1)x_{1,1}$b für abab
  $(2,3)x_{1,1}$b für abab
- Konkatenation von Ranges: $(0,1) \cdot (1,2) = (0,2)$
  - Kann fehlschlagen: $(0,1) \cdot (2,3)$
  - e als neutrales Element: $e \cdot (0,1) = (0,1)$
- Rangevektor: Folge von nicht überlappenden Ranges
  Beispiel: $\langle (0,1), (2,3) \rangle$

Gewichtetes Deduktionssystem

nach Shieber, Schabes und Pereira (1994), Nederhof (2003)

- Tupel (I, R)
- Deduktionsregel leitet aus *Vorgängern* nichtleere Menge an *Konsequenzen* ab
- $$\frac{a_1, \ldots, a_n}{c_1, \ldots, c_m}$$
- Itemgewicht $\mu_D \colon I \to [0, 1]$
  - Teil in Item gespeichert (Angelov & Ljunglöf, 2014)
- Heuristik (Angelov & Ljunglöf, 2014)

Gewichtetes Deduktionssystem

nach Shieber et al. (1994), Nederhof (2003)

- Tupel (I, R)
- Deduktionsregel leitet aus *Vorgängern* nichtleere Menge an *Konsequenzen* ab
- $$\frac{a_1, \ldots, a_n}{c_1, \ldots, c_m}$$
- Itemgewicht $\mu_D \colon I \to [0, 1]$
  - Teil in Item gespeichert (Angelov & Ljunglöf, 2014)
- Heuristik (Angelov & Ljunglöf, 2014)

Gewichtetes Deduktionssystem

nach Shieber et al. (1994), Nederhof (2003)

- Tupel (I, R)
- Deduktionsregel leitet aus *Vorgängern* nichtleere Menge an *Konsequenzen* ab
- $$\frac{a_1, \ldots, a_n}{c_1, \ldots, c_m}$$
- Itemgewicht $\mu_D \colon I \to [0, 1]$
  - Teil in Item gespeichert

(Angelov & Ljunglöf, 2014)

- Heuristik

(Angelov & Ljunglöf, 2014)

Gewichtetes Deduktionssystem

nach Shieber et al. (1994), Nederhof (2003)

- Tupel (I, R)
- Deduktionsregel leitet aus *Vorgängern* nichtleere Menge an *Konsequenzen* ab
- $$\frac{a_1, \ldots, a_n}{c_1, \ldots, c_m}$$
- Itemgewicht $\mu_D \colon I \to [0, 1]$
  - Teil in Item gespeichert (Angelov & Ljunglöf, 2014)
- Heuristik (Angelov & Ljunglöf, 2014)

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt $(S' \in S)$.

Idee des inkrementellen Parser für LCFRS

nach Burden und Ljunglöf (2005), Kallmeyer (2010)

Für LCFRS $G = (\Sigma, N, P, S)$ und Wort $w \in \Sigma^*$:

- Ein Item für jede Komponente in $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$
- Terminale/Variablen durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items
- Falls alle Symbole in Komponente ersetzt: Nächste wählen
- Form eines *Zielitems*: Komponente u_1 von $S' \to \langle u_1 \rangle (A_1, \ldots, A_k)$ durch Range $(0, |w|)$ ersetzt ($S' \in S$).

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- $\phi : [\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- $u' : [\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- $\Gamma : [k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- $\zeta : [k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell]$ $\dashrightarrow$ Range - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell]$ $\dashrightarrow$ $(X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N}$ $\dashrightarrow$ Range - Variablenindizes $\to$ genutzte Range
- ζ: $[k]$ $\dashrightarrow$ $[0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- ζ: $[k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- $\phi : [\ell] \dashrightarrow \mathrm{Range}$ - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- $u' : [\ell] \dashrightarrow (X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- $\Gamma : [k] \times \mathbb{N} \dashrightarrow \mathrm{Range}$ - Variablenindizes $\to$ genutzte Range
- $\zeta : [k] \dashrightarrow [0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Ink. Parser: Aufbau eines Items

Item $[\phi, A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', u', \Gamma, \zeta]$ mit:

- ϕ: $[\ell]$ $\dashrightarrow$ Range - Komponentenindex $\to$ ermittelte Range
- $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ - Regel aus P
- i - Index aus $[\ell]$
- ρ - Range für Präfix von u_i
- u_i' - Suffix von u_i
- u': $[\ell]$ $\dashrightarrow$ $(X \times \Sigma)^*$ - Index $\to$ unbetrachtete Komponente
- Γ: $[k] \times \mathbb{N}$ $\dashrightarrow$ Range - Variablenindizes $\to$ genutzte Range
- ζ: $[k]$ $\dashrightarrow$ $[0, 1]$ - Nachfolgerindex $\to$ Gewicht des Vorgängers

Beispielhafte Deduktion (Initialisierung)

Items für Startregeln erzeugen

$[\quad \emptyset \qquad p_1 \quad 1 \quad e \bullet x_{1,1}x_{1,2} \qquad \emptyset \qquad \emptyset \qquad \emptyset \quad]$

Beispielhafte Deduktion (Initialisierung)

Items für Startregeln erzeugen

$[\quad \emptyset \qquad\qquad p_1 \quad 1 \quad e \bullet x_{1,1} x_{1,2} \qquad \emptyset \qquad\qquad \emptyset \qquad\qquad\qquad\qquad \emptyset \quad]$

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

$$[\quad \emptyset \qquad\qquad p_1 \quad 1 \quad e \bullet x_{1,1} x_{1,2} \qquad \emptyset \qquad\qquad \emptyset \qquad\qquad\qquad \emptyset \quad]$$

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

$$[\quad \emptyset \qquad\qquad p_1 \quad 1 \quad e \bullet x_{1,1} x_{1,2} \qquad \emptyset \qquad\qquad \emptyset \qquad\qquad\qquad \emptyset \quad]$$

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

[	$\emptyset$	p_1	1	$e \bullet x_{1,1} x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$e \bullet \mathrm{a} x_{1,1} \mathrm{b}$	$\{(2, \mathrm{a} x_{1,2} \mathrm{b})\}$	$\emptyset$	$\emptyset$	]

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	]

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	]

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

Beispielhafte Deduktion (Dynamische Initialisierung)

Items für Grammatikregeln generieren, sobald diese benötigt werden

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	]

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	]

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_3 & 1 & (0,1) \bullet x_{1,1}\mathsf{b} & \{(2, \mathsf{a}x_{1,2}\mathsf{b})\} & \emptyset & \emptyset \quad], \\
[& \emptyset & p_2 & 1 & e\bullet & \{(2,\varepsilon)\} & \emptyset & \emptyset \quad] \\
\hline
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathsf{a}x_{1,2}\mathsf{b})\} & \{((1,1), e)\} & \zeta_1 \quad],
\end{array}
$$

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_3 & 1 & (0,1) \bullet x_{1,1}\mathsf{b} & \{(2, \mathsf{a}x_{1,2}\mathsf{b})\} & \emptyset & \emptyset \quad], \\
[& \emptyset & p_2 & 1 & e\bullet & \{(2,\varepsilon)\} & \emptyset & \emptyset \quad] \\
\hline
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathsf{a}x_{1,2}\mathsf{b})\} & \{((1,1), e)\} & \zeta_1 \quad],
\end{array}
$$

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_3 & 1 & (0,1) \bullet x_{1,1}\mathrm{b} & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \emptyset & \emptyset \quad], \\
[& \emptyset & p_2 & 1 & e\bullet & \{(2,\varepsilon)\} & \emptyset & \emptyset \quad] \\
\hline
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \{((1,1),e)\} & \zeta_1 \quad], \\
[& \{(1,(0,2))\} & p_3 & 2 & (2,3) \bullet x_{1,2}\mathrm{b} & \emptyset & \{((1,1),e)\} & \zeta_1 \quad]
\end{array}
$$

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_3 & 1 & (0,1) \bullet x_{1,1}\mathrm{b} & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \emptyset & \emptyset \;], \\
[& \emptyset & p_2 & 1 & e\bullet & \{(2,\varepsilon)\} & \emptyset & \emptyset \;] \\
\hline
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \{((1,1),e)\} & \zeta_1 \;], \\
[& \{(1,(0,2))\} & p_3 & 2 & (2,3) \bullet x_{1,2}\mathrm{b} & \emptyset & \{((1,1),e)\} & \zeta_1 \;]
\end{array}
$$

mit $\zeta_1 = \{(1, \mu_G(p_2))\}$

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$(0,2)\bullet$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\{((1,1), e)\}$	ζ_1	],
[	$\{(1,(0,2))\}$	p_3	2	$(2,3) \bullet x_{1,2}\mathrm{b}$	$\emptyset$	$\{((1,1), e)\}$	ζ_1	]
[	$\{(1,(0,2))\}$	p_3	2	$(2,3) \bullet x_{1,2}\mathrm{b}$	$\emptyset$	$\{((1,1), e)\}$	ζ_1	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

mit $\zeta_1 = \{(1, \mu_G(p_2))\}$

Beispielhafte Deduktion (Kombinieren - 1)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_3	1	$(0,1) \bullet x_{1,1}\mathrm{b}$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_3	1	$(0,2)\bullet$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\{((1,1), e)\}$	ζ_1	],
[	$\{(1,(0,2))\}$	p_3	2	$(2,3) \bullet x_{1,2}\mathrm{b}$	$\emptyset$	$\{((1,1), e)\}$	ζ_1	]
[	$\{(1,(0,2))\}$	p_3	2	$(2,3) \bullet x_{1,2}\mathrm{b}$	$\emptyset$	$\{((1,1), e)\}$	ζ_1	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\{(1,(0,2))\}$	p_3	2	$(2,4)\bullet$	$\emptyset$	$\{((1,1), e), ((1,2), e)\}$	ζ_2	]

mit $\zeta_1 = \{(1, \mu_G(p_2))\} = \zeta_2$

Beispielhafte Deduktion (Kombinieren - 2)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_3	1	$(0,2)\bullet$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\{((1,1),e)\}$	ζ_1	]
[	$\emptyset$	p_1	1	$(0,2) \bullet x_{1,2}$	$\emptyset$	$\{((1,1),(0,2))\}$	ζ_3	]

mit $\zeta_1 = \{(1, \mu_G(p_2)))\}$

Beispielhafte Deduktion (Kombinieren - 2)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_1 & 1 & e \bullet x_{1,1}x_{1,2} & \emptyset & \emptyset & \emptyset \quad], \\
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \{((1,1), e)\} & \zeta_1 \quad] \\
\hline
[& \emptyset & p_1 & 1 & (0,2) \bullet x_{1,2} & \emptyset & \{((1,1),(0,2))\} & \zeta_3 \quad]
\end{array}
$$

mit $\zeta_1 = \{(1, \mu_G(p_2)))\}$
mit $\zeta_3 = \{(1, \mu_G(p_3) \qquad\qquad)\}$

Beispielhafte Deduktion (Kombinieren - 2)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_3	1	$(0,2)\bullet$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\{((1,1), e)\}$	ζ_1	]
[	$\emptyset$	p_1	1	$(0,2) \bullet x_{1,2}$	$\emptyset$	$\{((1,1),(0,2))\}$	ζ_3	]

mit $\zeta_1 = \{(1, \mu_G(p_2)))\}$
mit $\zeta_3 = \{(1, \mu_G(p_3) \cdot \prod_{(i,g)\in\zeta_1} g)\}$

Beispielhafte Deduktion (Kombinieren - 2)

Variablen durch Ranges ersetzen

$$
\begin{array}{llllllll}
[& \emptyset & p_1 & 1 & e \bullet x_{1,1}x_{1,2} & \emptyset & \emptyset & \emptyset \quad], \\
[& \emptyset & p_3 & 1 & (0,2)\bullet & \{(2, \mathrm{a}x_{1,2}\mathrm{b})\} & \{((1,1), e)\} & \zeta_1 \quad] \\
\hline
[& \emptyset & p_1 & 1 & (0,2) \bullet x_{1,2} & \emptyset & \{((1,1),(0,2))\} & \zeta_3 \quad]
\end{array}
$$

mit $\zeta_1 = \{(1, \mu_G(p_2)))\}$
mit $\zeta_3 = \{(1, \mu_G(p_3) \cdot \mu_G(p_2))\}$

Beispielhafte Deduktion (Kombinieren - 2)

Variablen durch Ranges ersetzen

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	],
[	$\emptyset$	p_3	1	$(0,2)\bullet$	$\{(2, \mathrm{a}x_{1,2}\mathrm{b})\}$	$\{((1,1),e)\}$	ζ_1	]
[	$\emptyset$	p_1	1	$(0,2) \bullet x_{1,2}$	$\emptyset$	$\{((1,1),(0,2))\}$	ζ_3	]
[	$\emptyset$	p_1	1	$(0,2) \bullet x_{1,2}$	$\emptyset$	$\{((1,1),(0,2))\}$	ζ_3	],
[	$\{(1,(0,2))\}$	p_3	2	$(2,4)\bullet$	$\emptyset$	$\{((1,1),e),((1,2),e)\}$	ζ_1	]
[	$\emptyset$	p_1	1	$(\mathbf{0},\mathbf{4})\bullet$	$\emptyset$	$\{((1,1),(0,2)),((1,2),(2,4))\}$	ζ_4	]

mit $\zeta_1 = \{(1, \mu_G(p_2)))\}$
mit $\zeta_3 = \{(1, \mu_G(p_3) \cdot \mu_G(p_2))\} = \zeta_4$

Evaluation der Implementierung

- Vergleich der Implementierung mit:
  - CYK-Parser, naiv-aktiven und aktiven Parser (*interne Parser*)

    (Ruprecht, 2017)
  - rparse und Grammatical Framework (*GF*) (*externe Parser*)

    (Kallmeyer & Maier, 2013), (Ranta, 2011)
- Vergleichsgrundlage: Ableitungsbäume des NeGra-Korpus

  (Skut, Brants, Krenn & Uszkoreit, 1998)
  - Trainingsmenge: Ableitungsbäume, aus denen Grammatik generiert wird
  - Testmenge: Sätze, die geparst werden, mit Ableitungsbäumen vergleichen
- Vergleich von:
  - Durchschnittlicher Zeit zum Parsen von Testmenge
  - Erfolgsquote - Quote gefundener Parsebäume
  - Präzision - Quote richtig bestimmter Labels

  (Ruprecht, 2017), (van Cranenburgh, Scha & Bod, 2016)

Evaluation der Implementierung

- Vergleich der Implementierung mit:
  - CYK-Parser, naiv-aktiven und aktiven Parser (*interne Parser*)

    (Ruprecht, 2017)
  - rparse und Grammatical Framework (*GF*) (*externe Parser*)

    (Kallmeyer & Maier, 2013), (Ranta, 2011)
- Vergleichsgrundlage: Ableitungsbäume des NeGra-Korpus (Skut et al., 1998)
  - Trainingsmenge: Ableitungsbäume, aus denen Grammatik generiert wird
  - Testmenge: Sätze, die geparst werden, mit Ableitungsbäumen vergleichen
- Vergleich von:
  - Durchschnittlicher Zeit zum Parsen von Testmenge
  - Erfolgsquote - Quote gefundener Parsebäume
  - Präzision - Quote richtig bestimmter Labels

    (Ruprecht, 2017), (van Cranenburgh et al., 2016)

Evaluation der Implementierung

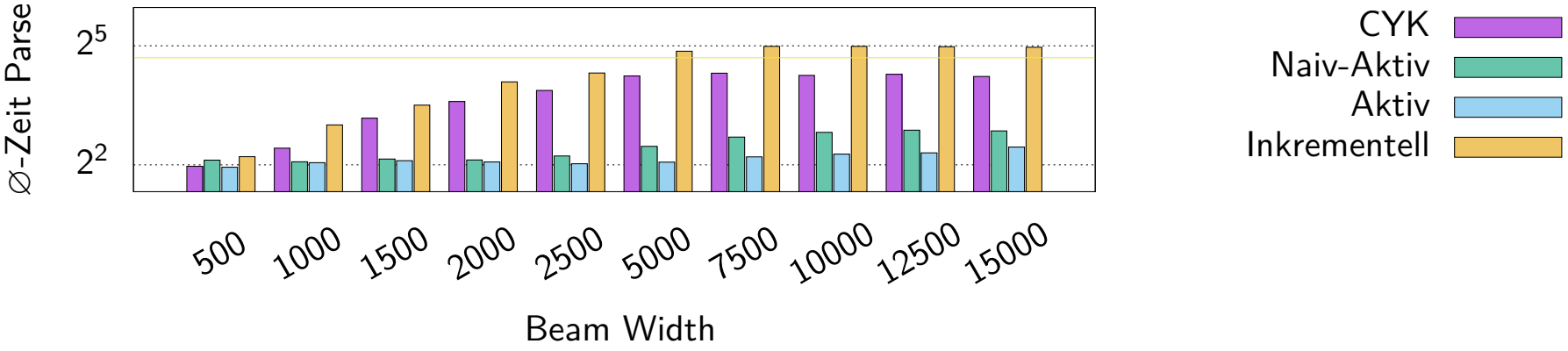
- Vergleich der Implementierung mit:
  - CYK-Parser, naiv-aktiven und aktiven Parser (*interne Parser*)

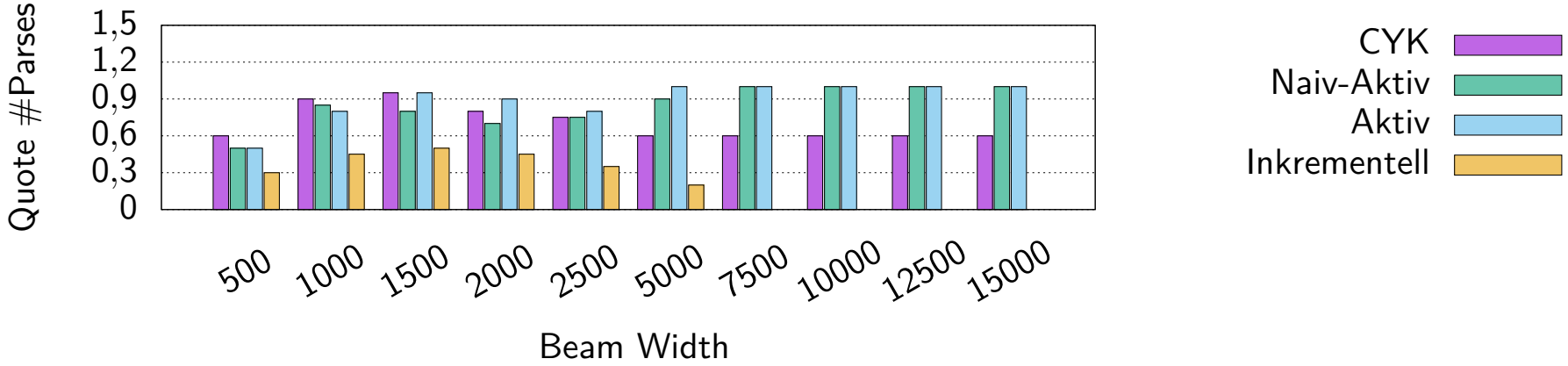
    (Ruprecht, 2017)
  - rparse und Grammatical Framework (*GF*) (*externe Parser*)

    (Kallmeyer & Maier, 2013), (Ranta, 2011)
- Vergleichsgrundlage: Ableitungsbäume des NeGra-Korpus (Skut et al., 1998)
  - Trainingsmenge: Ableitungsbäume, aus denen Grammatik generiert wird
  - Testmenge: Sätze, die geparst werden, mit Ableitungsbäumen vergleichen
- Vergleich von:
  - Durchschnittlicher Zeit zum Parsen von Testmenge
  - Erfolgsquote - Quote gefundener Parsebäume
  - Präzision - Quote richtig bestimmter Labels

    (Ruprecht, 2017), (van Cranenburgh et al., 2016)

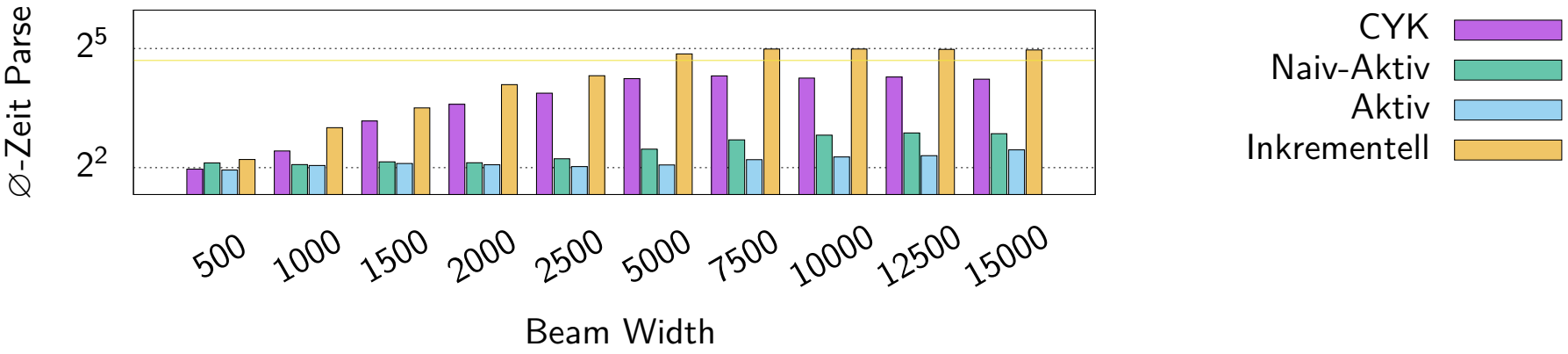
Untersuchung Beam Width (Intern)

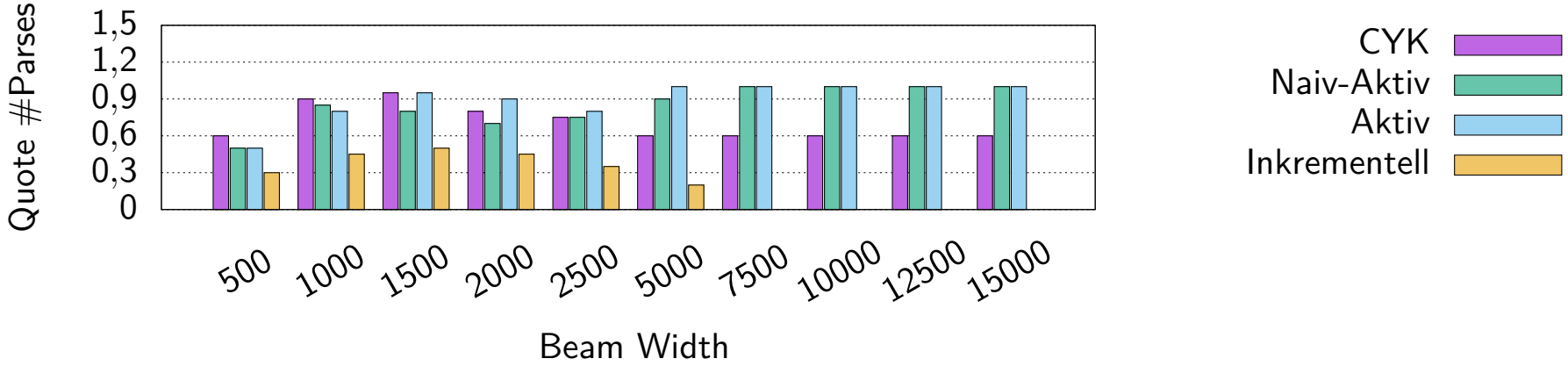




- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: NeGra-Korpus ohne Testmenge
- Timeout: 30 Sekunden
- Resultat: Ink. Parser für Beam Width 1500 beste Leistung
- Dennoch weiterhin Beam Width 2500 genutzt

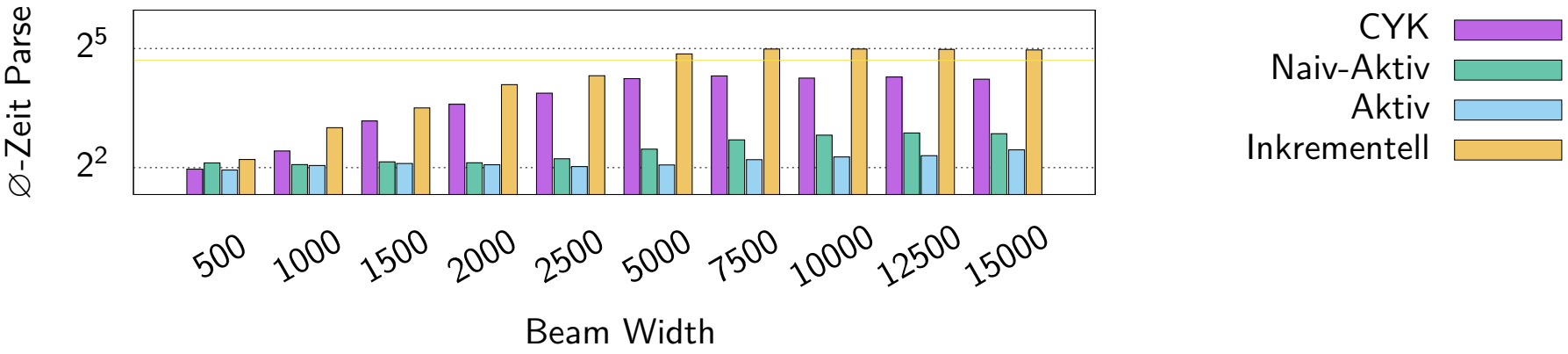
Untersuchung Beam Width (Intern)

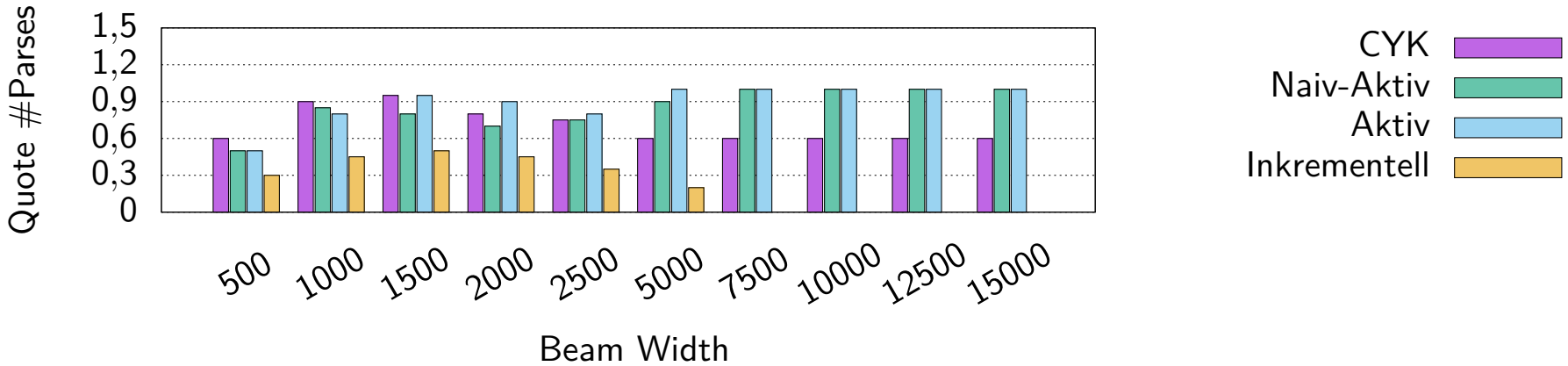




- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: NeGra-Korpus ohne Testmenge
- Timeout: 30 Sekunden
- Resultat: Ink. Parser für Beam Width 1500 beste Leistung
- Dennoch weiterhin Beam Width 2500 genutzt

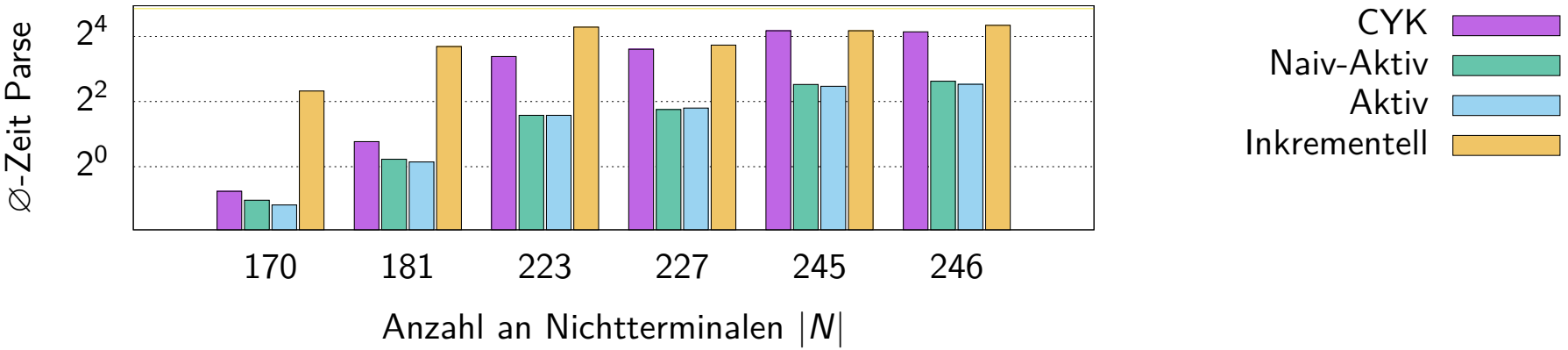
Untersuchung Beam Width (Intern)

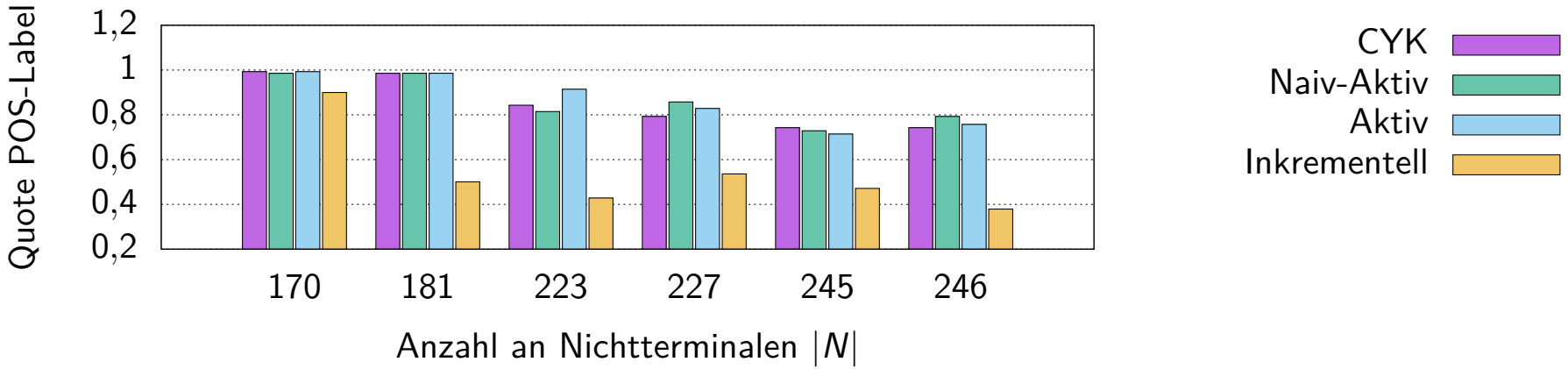




- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: NeGra-Korpus ohne Testmenge
- Timeout: 30 Sekunden
- Resultat: Ink. Parser für Beam Width 1500 beste Leistung
- Dennoch weiterhin Beam Width 2500 genutzt

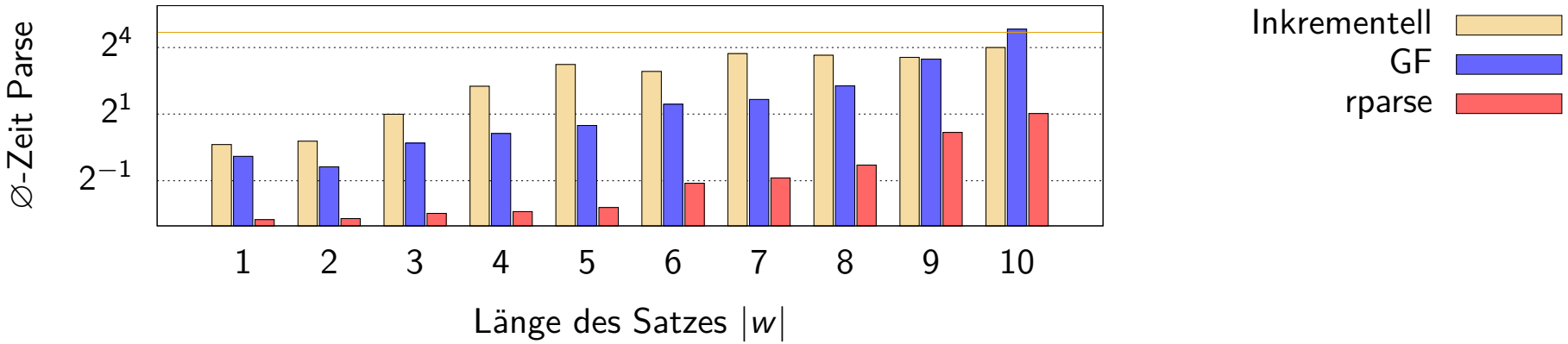
Untersuchung Grammatikgröße (Intern)

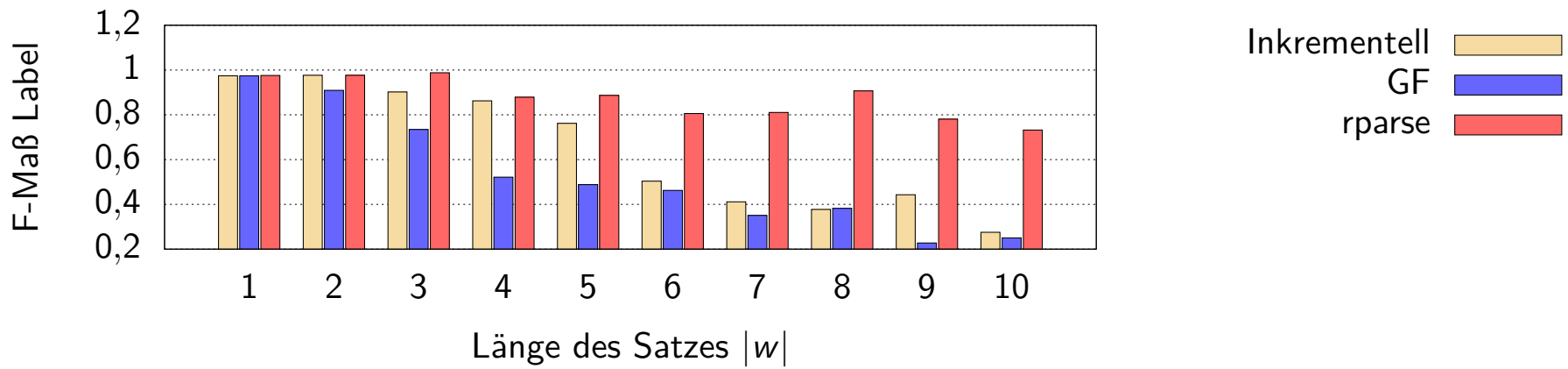




- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus
- Timeout: 30 Sekunden; Beam Width: 2500
- Quote der richtig vorhergesagten *POS-Label*

Untersuchung Satzlänge (Extern)





- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Timeout: 30 Sekunden (außer GF)
- rparse Testdaten mit POS-Labels
- F-Maß der richtig vorhergesagten *Label* ohne POS-Label

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Zusammenfassung

- Implementierung des inkrementellen Parsers für gewichtete LCFRS
  - Optimierungen: Beam Width, Heuristik, Anpassung Deduktionsregeln
- Vergleich mit anderen Parsern:
  - Verbesserungen an der Evaluation
  - Aussagen zum inkrementellen Parser:
    - Benötigt häufig die meiste Zeit zum Parsen
    - unpräziser als alle anderen Parser
  - Problem des inkrementellen Parsers:
    - leitet zu viele Items (auf einmal) ab
- Weitere Aufgaben:
  - Verbesserung der Implementierung
  - Ausweitung auf (P)MCFG

(Ruprecht, 2017)

Vielen Dank für Ihre Aufmerksamkeit!

Quellen I


Angelov, K. & Ljunglöf, P. (2014). Fast statistical parsing with parallel multiple context-free grammars. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics* (S. 368–376).


Burden, H. & Ljunglöf, P. (2005). Parsing Linear Context-free Rewriting Systems. In *Proceedings of the Ninth International Workshop on Parsing Technology* (S. 11–17). Parsing '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics. Zugriff unter http://dl.acm.org/citation.cfm?id=1654494.1654496


Goodman, J. (1999). Semiring Parsing. *Comput. Linguist. 25*(4), 573–605. Zugriff unter http://dl.acm.org/citation.cfm?id=973226.973230

Quellen II


Joshi, A. K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In D. R. Dowty, L. Karttunen & A. M. Zwicky (Hrsg.), *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives* (S. 206–250). Studies in Natural Language Processing. Cambridge University Press. doi:10.1017/CBO9780511597855.007


Kallmeyer, L. (2010). *Parsing Beyond Context-Free Grammars* (1st). Springer Publishing Company, Incorporated.


Kallmeyer, L. & Maier, W. (2013). Data-driven parsing using probabilistic linear context-free rewriting systems. *Computational Linguistics*, *39*(1), 87–119.


Nederhof, M.-J. (2003). Weighted Deductive Parsing and Knuth's Algorithm. *Comput. Linguist. 29*(1), 135–143. doi:10.1162/089120103321337467

Quellen III


Ranta, A. (2011). *Grammatical framework: Programming with multilingual grammars*. CSLI Publications, Center for the Study of Language und Information.


Ruprecht, T. (2017). *Parsing linear context-free rewriting systems*. Techn. Ber., Technische Universität Dresden.


Schroeder, M. (2017). *Vorlesung Intelligente Systeme - Einführung in die künstliche Intelligenz Teil 1*. http://www.biotec.tu-dresden.de/fileadmin/groups/schroeder/teaching/IntelligenteSysteme/slides/3_Entita__tenerkennung.pdf, zuletzt besucht am 05.07.2018.


Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, *8*(3), 333–343. doi:10.1007/BF00630917

Quellen IV


Shieber, S. M., Schabes, Y. & Pereira, F. C. N. (1994). Principles and Implementation of Deductive Parsing. *CoRR, abs/cmp-lg/9404008*. arXiv: cmp-lg/9404008. Zugriff unter http://arxiv.org/abs/cmp-lg/9404008


Skut, W., Brants, T., Krenn, B. & Uszkoreit, H. (1998). A linguistically interpreted corpus of German newspaper text. *arXiv preprint cmp-lg/9807008*.


van Cranenburgh, A., Scha, R. & Bod, R. (2016). Data-Oriented Parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, *4*(1), 57–111. Zugriff unter http://dx.doi.org/10.15398/jlm.v4i1.100

Quellen V

 Vijay-Shanker, K., Weir, D. J. & Joshi, A. K. (1987). Characterizing Structural Descriptions Produced by Various Grammatical Formalisms. In *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics* (S. 104–111). ACL '87. Stanford, California: Association for Computational Linguistics. doi:10.3115/981175.981190

Backup

Vergleich mit internen Parsern

- Untersuchung bei unterschiedlicher:
  - Beam Width
  - Größe der Grammatik (Anzahl Nichtterminale)
  - Satzlänge
- Timeout: 30 Sekunden
- Grammatik nach jedem Parse neu eingelesen
- Präzision: Quote der richtig vorhergesagten *POS-Label*

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)
    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle (A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)
    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle (A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)

    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle (A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)

    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle (A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)
    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle (A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)
    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle(A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Gewichtete Deduktionssysteme (Angelov & Ljunglöf, 2014)

- Itemgewicht $\mu_D : I \to [0, 1]$:
  - Gewicht der bisher angewandten Regeln
  - Gewicht eines durchlaufenen Items: Gewicht des Ableitungsbaum
  - Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht

    (Nederhof, 2003), (Ruprecht, 2017)
    - Problem: Ersetzung von zwei Variablen eines Nachfolgers kann zu falschem Gewicht führen ($S' \to \langle x_{1,1} x_{1,2} \rangle(A)$)
    - Festlegung auf Grammatikregel nicht vorgesehen

      (Burden & Ljunglöf, 2005)
  - Lösung:
    - Gewichte für Nachfolger in binärer Relation(BR) speichern
    - Aktualisierung bei Ersetzung von Variablen
    - Itemgewicht: Produkt aktuell gespeicherten Gewichte in BR und Grammatikregel-Gewicht
    - Keine Gewichte für Deduktionsregeln wie in Ruprecht (2017)

Dyn. Initialisierung - 2. Konsequenz

- Konsequenzen der Form $[\phi, p, i, \rho\bullet, u', \Gamma, \zeta]$ nur abgeleitet, wenn $u' = \emptyset$
  - $p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$
  - $p' = (A_m \to \langle v_1, \ldots, v_n \ldots, v_s \rangle (B_1, \ldots, B_t)) \in P$
  - Problem: Item für Kombinieren-Regel fehlt

$$r_K = \frac{[\phi, p, i, \rho \bullet x_{m,n} u_i', u', \Gamma, \zeta], [\phi', p', n, \rho'\bullet, v', \Gamma', \zeta']}{[\phi, p, i, \rho \cdot \rho' \bullet u_i', u', \Gamma[(m,n)/\rho'], \zeta[m/(\bar{\zeta}' \cdot \mu_G(p'))]]} \quad \forall h \in s\colon \forall \alpha \in \Gamma(m,h)\colon \alpha = \emptyset \vee \alpha = \phi'(h),$$

- Lösung: Alle Items der oben genannten Form zwischenspeichern, am Ende ausgeben

Ermitteln des Ableitungsbaum (1)

komplette Items der Form:

$$[\quad \phi \qquad p \quad i \quad \rho\bullet \qquad \emptyset \qquad \Gamma \qquad \zeta \quad]$$

werden in *Chart* gespeichert, falls:

- alle Ranges in $\phi \cup \{(i, \rho)\}$ in einen Rangevektor gespeichert werden können
- alle Ranges in Γ, geteilt nach den Indizes der Nachfolger, in Rangevektor umgewandelt werden können

Für Zielitem:

$$[\quad \emptyset \qquad p_1 \quad 1 \quad (0,4)\bullet \qquad \emptyset \qquad \{((1,1),(0,2)),((1,2),(2,4))\} \quad \zeta_4 \quad]$$

- $\{(0,4)\}$ kann in Rangevektor umgewandelt werden.
- $\{(0,2),(2,4)\}$ für Variablen des ersten Nachfolgers kann in Rangevektor umgewandelt werden.

Ermitteln des Ableitungsbaums(2)

$[\quad \phi \quad A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \quad i \quad \rho\bullet \quad \emptyset \quad \Gamma \quad \zeta \quad]$

- Item in Chart abgespeichert: (Ruprecht, 2017)
  - Nichtterminal A
  - Rangevektor aus $\phi \cup \{(i, \rho)\}$
  - *Backtrace*s:
    - $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$
    - $\mu_G(A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k))$
    - Folge von Rangevektoren für Nachfolger $A_1 \ldots A_k$ aus Γ
  - Itemgewicht
- Ausgehend von Zielitem Ranges Ableitungsbaum ermitteln

Ermitteln des Ableitungsbaums(3)

In Chart gespeichert:

[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	],
[	$\{(1, (0, 2))\}$	p_3	2	$(2, 4)\bullet$	$\emptyset$	$\{((1, 1), e), ((1, 2), e)\}$	ζ_2	],
[	$\emptyset$	p_1	1	$(0, 4)\bullet$	$\emptyset$	$\{((1, 1), (0, 2)), ((1, 2), (2, 4))\}$	ζ_4	]

Extraktion: $(S, p_1, \langle(0, 4)\rangle, \langle(0, 2), (2, 4)\rangle)$

$|$

$(A, p_3, \langle(0, 2), (2, 4)\rangle, \langle e, e\rangle)$

$|$

$(A, p_2, \langle e, e\rangle, \langle\rangle)$

Extraktion der Testdaten (Ruprecht, 2017)

- Sätze in NeGra aufteilen
  - Sätze, bei denen jedes Wort noch einmal auftritt
  - Sätze, bei denen mindestens ein Wort einzigartig ist
- Testdaten: Sätze, bei denen jedes Wort noch einmal in restlichen Sätzen auftritt

part. Fkt. Gewicht (1)

(Angelov & Ljunglöf, 2014)

Gewicht kompletter Items repräsentiert Gewicht des Ableitungsbaums

- Standard: Vorgänger-Gewichte Teil von Konsequenz-Gewicht
- Initialisierung bzw. Dyn. Initialisierung: Konsequenz Gewicht der Grammatikregel zuweisen
- Im Folgenden: $\zeta_c \in [0,1]$ für $c \in I$

Beispiel:

- $p_1 = S \to \langle x_{1,1} x_{1,2} \rangle (A)$
- $p_2 = A \to \langle \mathrm{a}, x_{1,1} \rangle (B)$
- $p_3 = B \to \langle \mathrm{b} \rangle ()$
- $w = \mathrm{ab}$

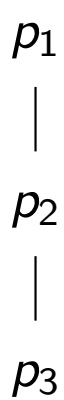
$\emptyset$	p_1	1	$\bullet x_{1,1} x_{1,2}$	$\emptyset$	$\emptyset$	$\mu_G(p_1)$
$\emptyset$	p_2	1	$(0,1)\bullet$	$\{(2, x_{1,1})\}$	$\emptyset$	$\mu_G(p_2)$
$\emptyset$	p_1	1	$(0,1) \bullet x_{1,2}$	$\emptyset$	$\emptyset$	$\mu_G(p_1) \cdot \mu_G(p_2)$
$\{(1,(0,1))\}$	p_2	2	$e \bullet x_{1,1}$	$\emptyset$	$\emptyset$	$\mu_G(p_2)$
$\emptyset$	p_3	1	$(1,2)\bullet$	$\emptyset$	$\emptyset$	$\mu_G(p_3)$
$\{(1,(0,1))\}$	p_2	2	$(1,2)\bullet$	$\emptyset$	$\emptyset$	$\mu_G(p_2) \cdot \mu_G(p_3)$
$\emptyset$	p_1	1	$(\mathbf{0},\mathbf{2})\bullet$	$\emptyset$	$\emptyset$	$\mu_G(p_1) \cdot \mu_G(p_2) \cdot \mu_G(p_2) \cdot \mu_G(p_3)$

part. Fkt. Gewicht (2)

Item kompletter Items repräsentiert Gewicht des Ableitungsbaums

- $p_1 = S \to \langle x_{1,1} x_{1,2} \rangle (A)$
- $p_2 = A \to \langle \mathrm{a}, x_{1,1} \rangle (B)$
- $p_3 = B \to \langle \mathrm{b} \rangle ()$
- $w = \mathrm{ab}$
- $[\quad \emptyset \quad p_1 \quad 1 \quad (0,2)\bullet \quad \emptyset \quad \emptyset \quad \mu_G(p_1) \cdot \mu_G(p_2) \cdot \mu_G(p_2) \cdot \mu_G(p_3) \quad]$

Parsebaum:



part. Fkt. Gewicht (3)

Item kompletter Items repräsentiert Gewicht des Ableitungsbaums

- $p_1 = S \to \langle x_{1,1} x_{1,2} \rangle (A)$
- $p_2 = A \to \langle \mathrm{a}, x_{1,1} \rangle (B)$
- $p_3 = B \to \langle \mathrm{b} \rangle ()$
- $w = \mathrm{ab}$
- Tatsächliches Gewicht Ableitungsbaum: $\mu_G(p_1) \cdot \mu_G(p_2) \cdot \mu_G(p_3)$
- Lösung: Gewicht in part. Fkt. speichern

$$
\begin{array}{llllllll}
[& \emptyset & p_1 & 1 & (0,1) \bullet x_{1,2} & \emptyset & \emptyset & \{(1, \mu_G(p_2))\} \quad] \\
[& \{(1,(0,1))\} & p_2 & 2 & (1,2) \bullet & \emptyset & \emptyset & \{(1, \mu_G(p_3))\} \quad] \\
[& \emptyset & p_1 & 1 & (0,2) \bullet & \emptyset & \emptyset & \{(1, \mu_G(p_2) \cdot \mu_G(p_3))\} \quad]
\end{array}
$$

- Zielitem-Gewicht: $\mu_G(p_1) \cdot \mu_G(p_2) \cdot \mu_G(p_3)$
- Stimmt mit Gewicht Parsebaum überein

Heuristik (1)

(Angelov & Ljunglöf, 2014)

- Agenda durch *Beam Width* größenbeschränkt
- Trigger-Items nacheinander aus *Agenda* nehmen
  - Reihenfolge der Auswahl?
  - Welche Konsequenzen in Agenda aufnehmen?
- Produkt aus Gewicht und *Heuristik* eines Items unterschätzt Gewicht, ein Zielitem abzuleiten
- Items aufsteigend in Agenda ordnen

Heuristik (1)

(Angelov & Ljunglöf, 2014)

- Agenda durch *Beam Width* größenbeschränkt
- Trigger-Items nacheinander aus *Agenda* nehmen
  - Reihenfolge der Auswahl?
  - Welche Konsequenzen in Agenda aufnehmen?
- Produkt aus Gewicht und *Heuristik* eines Items unterschätzt Gewicht, ein Zielitem abzuleiten
- Items aufsteigend in Agenda ordnen

Heuristik (1)

(Angelov & Ljunglöf, 2014)

- Agenda durch *Beam Width* größenbeschränkt
- Trigger-Items nacheinander aus *Agenda* nehmen
  - Reihenfolge der Auswahl?
  - Welche Konsequenzen in Agenda aufnehmen?
- Produkt aus Gewicht und *Heuristik* eines Items unterschätzt Gewicht, ein Zielitem abzuleiten
- Items aufsteigend in Agenda ordnen

Heuristik (1)

(Angelov & Ljunglöf, 2014)

- Agenda durch *Beam Width* größenbeschränkt
- Trigger-Items nacheinander aus *Agenda* nehmen
  - Reihenfolge der Auswahl?
  - Welche Konsequenzen in Agenda aufnehmen?
- Produkt aus Gewicht und *Heuristik* eines Items unterschätzt Gewicht, ein Zielitem abzuleiten
- Items aufsteigend in Agenda ordnen

Heuristik (1)

(Angelov & Ljunglöf, 2014)

- Agenda durch *Beam Width* größenbeschränkt
- Trigger-Items nacheinander aus *Agenda* nehmen
  - Reihenfolge der Auswahl?
  - Welche Konsequenzen in Agenda aufnehmen?
- Produkt aus Gewicht und *Heuristik* eines Items unterschätzt Gewicht, ein Zielitem abzuleiten
- Items aufsteigend in Agenda ordnen

Heuristik (2)

(Angelov & Ljunglöf, 2014), (Ruprecht, 2017)

- Nutzen: Items in größenbeschränken Agenda ordnen
- Produkt aus Gewicht und Heuristik des Items unterschätzt Gewicht, welches benötigt wird, um ein *Zielitem* abzuleiten
- Heuristik besteht aus: (Ruprecht, 2017)
  - *Innenheuristik* $\beta_D : I \to [0, 1]$
    - schätzt Gewicht, ein Item in komplettes Item umzuwandeln
  - *Außenheuristik* $\alpha_D : I \to [0, 1]$
    - schätzt Gewicht, ein Zielitem mit bereits abgeleiteten kompletten Items abzuleiten
    - Annahme: Jedes Item mit Startregel ist Zielitem (Range egal)
    - Annahme: Jedes Item ist bereits komplett

Heuristik (2)

(Angelov & Ljunglöf, 2014), (Ruprecht, 2017)

Für eine LCFRS $G = (\Sigma, N, P, S)$:

- Innengewicht $\beta_G \colon N \to [0, 1]$
  - Weißt jedem Nichtterminal A das größte Gewicht des Ableitungsbaums zu, welcher mit Regel der Form $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ beginnt
- Außengewicht $\alpha_G \colon N \to [0, 1]$
  - Weißt jedem Nichtterminal A das größte Gewicht des Ableitungsbaums zu, welcher Regel der Form $A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ beinhaltet, ohne das Gewicht des dort beginnenden Teilbaums

$[\quad \phi \quad A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \quad i \quad \rho \bullet u_i' \quad u' \quad \Gamma \quad \zeta \quad]$

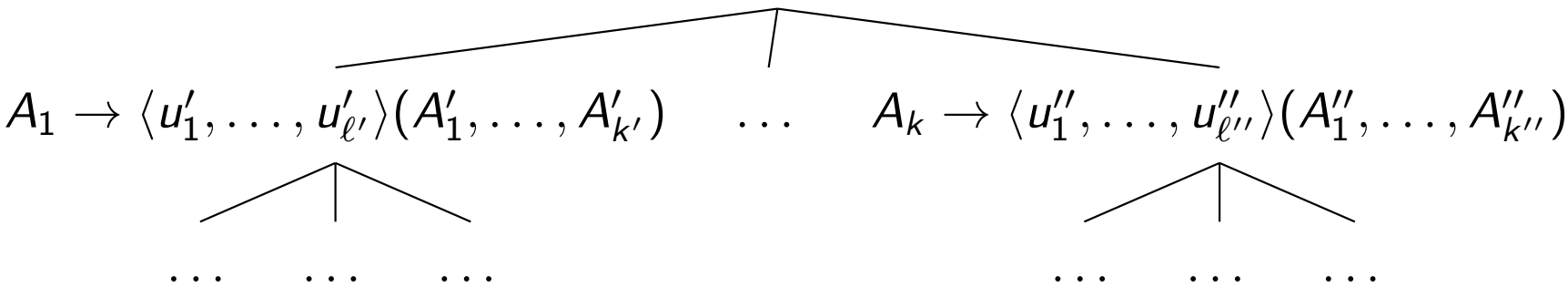
Berechnung von Innen- und Außenheuristik: (Ruprecht, 2017)

- Außenheuristik eines Items: Außengewicht von A
- Innenheuristik eines Items: Innengewicht aller Nachfolger $(A_1, \ldots, A_k)$, für die noch nicht alle Variablen ersetzt wurden
  - Analog Gewicht: In part. Fkt. in Item gespeichert

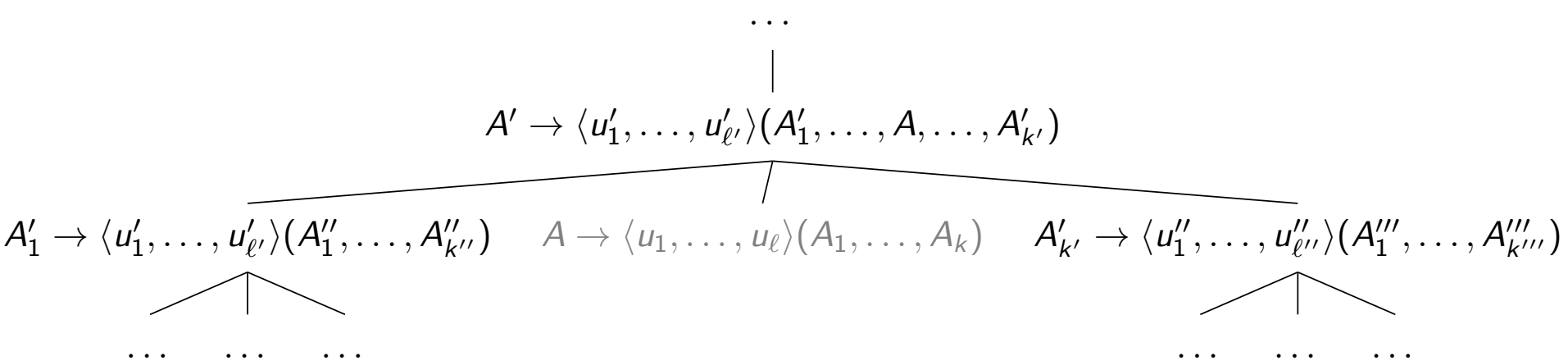
Heuristik (3)

(Angelov & Ljunglöf, 2014), (Ruprecht, 2017)

Innengewicht von A: Maximales Gewicht von (Ableitungs)bäumen der Form:



Außengewicht von A: Maximales Gewicht von (Ableitungs)bäumen der Form:



Heuristik (4)

(Angelov & Ljunglöf, 2014), (Ruprecht, 2017)

- Außenheuristik eines Items: Außengewicht von A
- Innenheuristik eines Items: Innengewicht aller Nachfolger $(A_1, \ldots, A_k)$, für die noch nicht alle Variablen ersetzt wurden

Außenheuristik:

- Fix, deshalb nicht in Item gespeichert

Innenheuristik(IH):

- Ändert sich bei Einsetzen einer Variable
- $S' \rightarrow \langle x_{1,1} x_{1,2} \rangle (A)$
- Item $c_{S'}$ hat initial $\beta_D(c_{S'}) = \beta_G(A)$
- Ersetzung von $x_{1,1}$ führt zu Änderung der IH der Konsequenz
  - Falls Regel für A fix: 1
  - Sonst: part. Fkt. nötig, um zu aktualisieren (Abhängig von Vorgänger zur Kombination)
- IH-BR elementweise mit Gewichts-BR eines Items verbinden
  - Item unvollständig: Nur Produkt Gewicht und IH nötig
  - Item vollständig: IH ist 1, Gewicht für *Chart* kann berechnet werden

Ansatz zur Reduzierung der Anzahl abgeleiteter Items

Wiederholung dynamische Initialisierung:

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	2	$e\bullet$	$\{(1, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(2, e)\}$	p_2	1	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

Ansatz zur Reduzierung der Anzahl abgeleiteter Items

Wiederholung dynamische Initialisierung:

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

[	$\emptyset$	p_1	1	$e \bullet x_{1,1}x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	2	$e\bullet$	$\{(1, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(2, e)\}$	p_2	1	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

Ansatz zur Reduzierung der Anzahl abgeleiteter Items

Wiederholung dynamische Initialisierung:

[	$\emptyset$	p_1	1	$e \bullet x_{1,1} x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	1	$e\bullet$	$\{(2, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(1, e)\}$	p_2	2	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

• • •

[	$\emptyset$	p_1	1	$(0, 2) \bullet x_{1,2}$	$\emptyset$	$\emptyset$	$\emptyset$	]
[	$\emptyset$	p_2	2	$e\bullet$	$\{(1, \varepsilon)\}$	$\emptyset$	$\emptyset$	],
[	$\{(2, e)\}$	p_2	1	$e\bullet$	$\emptyset$	$\emptyset$	$\emptyset$	]

Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

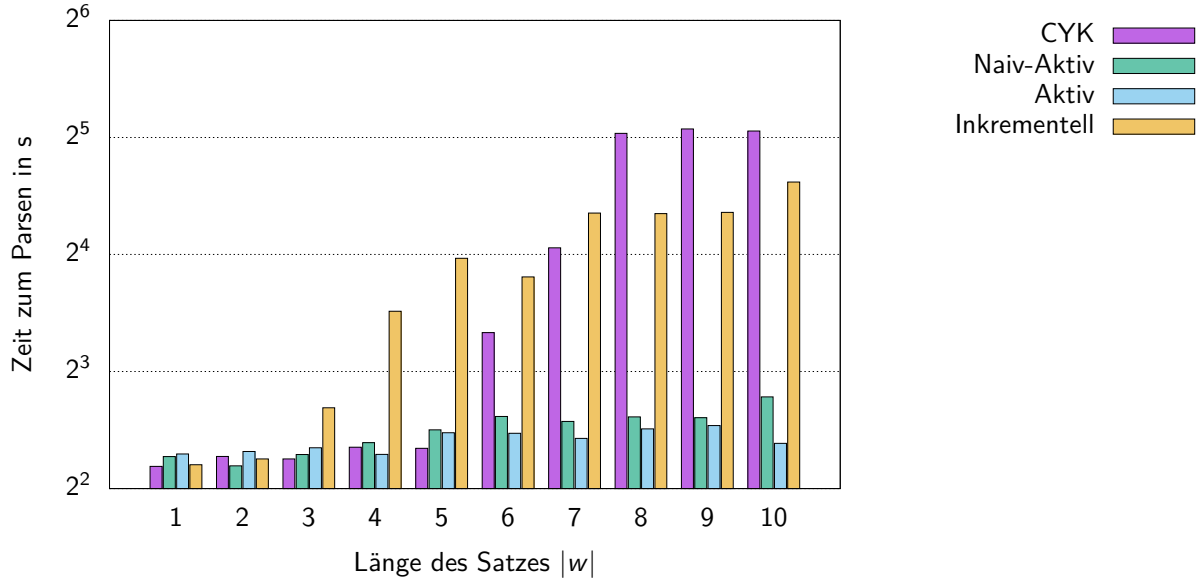
  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

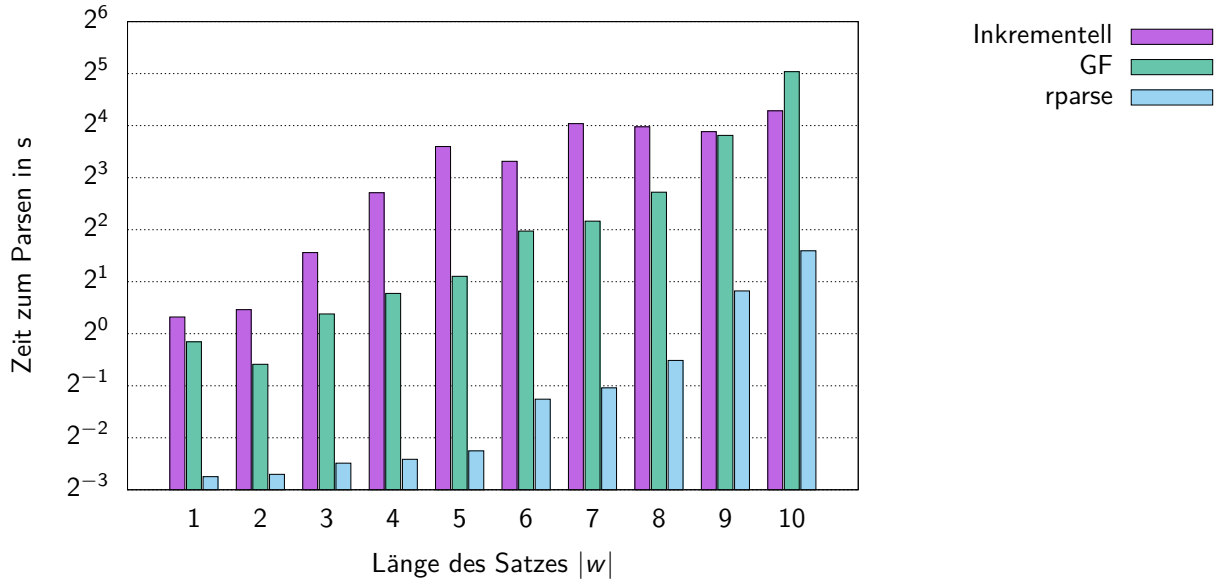
Implementierung eines Deduktionssystems (Kallmeyer, 2010)

- *Agenda*: Nicht betrachtete Konsequenzen (größenbeschränkt)

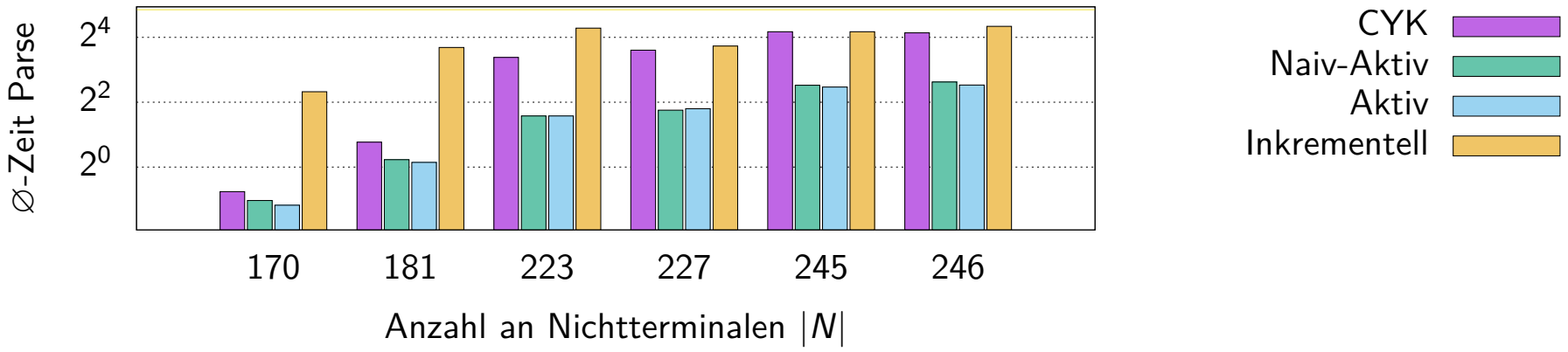
  (Ruprecht, 2017)
- *Trigger-Item*: Item aus Agenda
- *Container*: Menge von genutzten Trigger-Items
- *Deduktionsschritt*: Aktuelles Trigger-Item und Container als Vorgänger nutzen
- Beendet, wenn Agenda leer (Ruprecht, 2017)

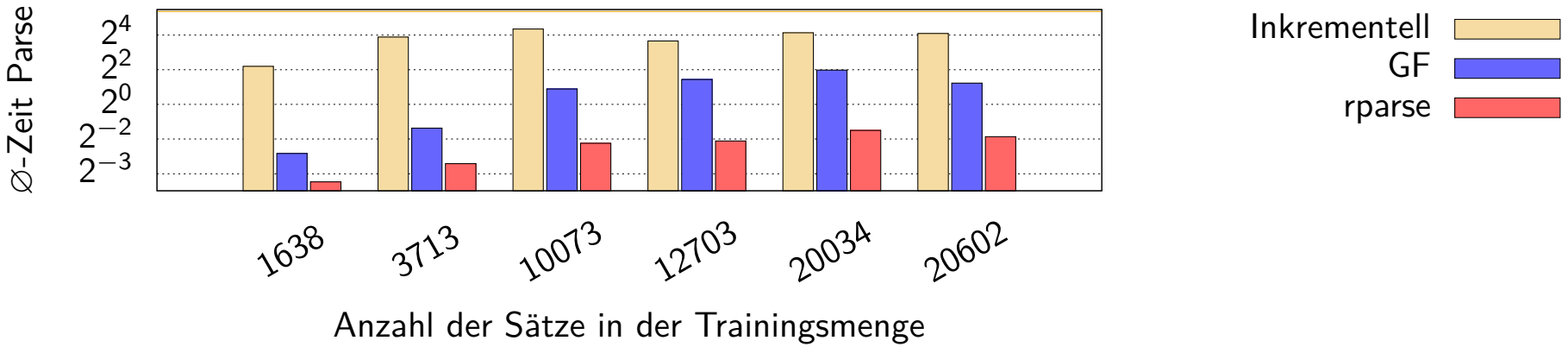
Unterschied Einlesen Grammatiken





Unterschied Einlesen Grammatiken





F-Maß

nach (Schroeder, 2017)

Label:

- *Trefferquote/Recall*: $\frac{\text{Anzahl Label richtig getroffen}}{\text{Anzahl tatsächlich Label}}$
- 0, wenn kein Ableitungsbaum gefunden
- *Genauigkeit/Precision*: $\frac{\text{Anzahl Label richtig getroffen}}{\text{Anzahl Label vergeben}}$
- 1, wenn kein Ableitungsbaum gefunden
- *F-Maß*: $\frac{2 \cdot T \cdot G}{T+G}$
- 0, wenn kein Ableitungsbaum gefunden

Ableitungsbaum:

- Jeweils arithmetisches Mittel aller Werte für alle Label

Besonderheiten inkrementeller Parser (Burden & Ljunglöf, 2005)

- Komponenten aus $A \rightarrow \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ in beliebiger Reihenfolge betrachtet
- Item muss nicht vollständig durchlaufen sein, um Komponente zu nutzen

Optimierungen nach Ruprecht (2017):

- Terminalsymbole ersetzen in andere Deduktionsregeln einbetten
- Auswahl neuer Komponente in andere Deduktionsregeln einbetten
- Dynamische Initialisierung

Besonderheiten inkrementeller Parser (Burden & Ljunglöf, 2005)

- Komponenten aus $A \rightarrow \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ in beliebiger Reihenfolge betrachtet
- Item muss nicht vollständig durchlaufen sein, um Komponente zu nutzen

Optimierungen nach Ruprecht (2017):

- Terminalsymbole ersetzen in andere Deduktionsregeln einbetten
- Auswahl neuer Komponente in andere Deduktionsregeln einbetten
- Dynamische Initialisierung

Besonderheiten inkrementeller Parser (Burden & Ljunglöf, 2005)

- Komponenten aus $A \rightarrow \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ in beliebiger Reihenfolge betrachtet
- Item muss nicht vollständig durchlaufen sein, um Komponente zu nutzen

Optimierungen nach Ruprecht (2017):

- Terminalsymbole ersetzen in andere Deduktionsregeln einbetten
- Auswahl neuer Komponente in andere Deduktionsregeln einbetten
- Dynamische Initialisierung

Besonderheiten inkrementeller Parser (Burden & Ljunglöf, 2005)

- Komponenten aus $A \rightarrow \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k)$ in beliebiger Reihenfolge betrachtet
- Item muss nicht vollständig durchlaufen sein, um Komponente zu nutzen

Optimierungen nach Ruprecht (2017):

- Terminalsymbole ersetzen in andere Deduktionsregeln einbetten
- Auswahl neuer Komponente in andere Deduktionsregeln einbetten
- Dynamische Initialisierung

Deduktionsregeln (1)

Initialisierung: Für jedes Startsymbol $S' \in S$ und jede Startregel der Grammatik $p = S' \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k) \in P$ und jedes $(i, u_i) \in u$ gilt:

$$r_{\mathsf{Init}} = \frac{}{[\emptyset, p, i, e \bullet u_i, u \backslash \{(i, u_i)\}, \emptyset, \{(m, \beta_G(A_m)) \mid m \in [k]\}]} \,,$$

Deduktionsregeln (2)

Dynamische Initialisierung: Für jedes Item $[\phi, A \to \langle v_1, \ldots, v_o \rangle (A_1, \ldots, A_m, \ldots, A_k), j, \rho \bullet x_{m,n} v'_j, v', \Gamma, \zeta] \in I$ und $A_m \in N$ wurde noch nicht dynamisch initialisiert, gilt für jede Grammatikregel $p = A_m \to \langle u_1, \ldots, u_n, \ldots, u_\ell \rangle (A'_1, , \ldots, A'_s) \in P$ und jedes Tupel $(n, u_n) \in u$:

$$r_{\text{DInit}} = \frac{[\phi, A \to \langle v_1, \ldots, v_o \rangle (A_1, \ldots, A_m, \ldots, A_k), j, \rho \bullet x_{m,n} v'_j, v', \Gamma, \zeta]}{[\emptyset, p, n, e \bullet u_n, u \backslash \{(n, u_n)\}, \{(t, \beta_G(A'_t)) \mid t \in [s]\}]} ,$$

Deduktionsregeln (3)

Kombinieren: Für alle $(\mathbb{N} \times \mathrm{Range})$-Maps ϕ und ϕ',
$p = A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$,
$p' = (A_m \to \langle v_1, \ldots, v_n \ldots, v_s \rangle (B_1, \ldots, B_t)) \in P$, $(i, u_i) \in u$, $(n, v_n) \in v$,
$\rho, \rho' \in \mathrm{Range}$, $x_{m,n} \in X$, $u'_i \in (\Sigma \cup X)^*$, $(\mathbb{N} \times (\Sigma \times X)^*)$-Maps u', v'
und $((\mathbb{N} \times \mathbb{N}) \times \mathrm{Range})$-Maps Γ, Γ' sowie $(\mathbb{N} \times [0,1])$-Maps ζ, ζ' gilt:

$$r_{\mathsf{K}} = \frac{[\phi, p, i, \rho \bullet x_{m,n} u'_i, u', \Gamma, \zeta], [\phi', p', n, \rho' \bullet, v', \Gamma', \zeta']}{[\phi'', p, j, \rho'' \bullet u'_j, u'', \Gamma[(m,n)/\rho'], \zeta[m/(\bar{\zeta}' \cdot \mu_G(p'))]], \alpha_{EK_1}, \ldots, \alpha_{EK_z}}$$
$$\forall h \in s \colon \forall \alpha \in \Gamma(m,h) \colon \alpha = \emptyset \vee \alpha = \phi'(h), \left(\bigcup_{s \in [z]} \alpha_{EK_s}\right) = \boldsymbol{\alpha}_{EK},$$

Tabelle Intern (1)

$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Präzision	Trefferquote	Genauigkeit	F-Maß
13649	170	17	1,35	0,58	0,99	1,00	0,99	0,99	0,99
24916	181	16	1,37	1,69	0,99	1,00	0,99	0,98	0,98
50573	223	18	1,45	10,67	0,84	0,85	0,85	0,99	0,91
59070	227	18	1,46	12,46	0,79	0,80	0,80	0,99	0,88
80290	245	40	1,50	18,60	0,74	0,75	0,75	0,99	0,85
81800	246	40	1,50	18,17	0,74	0,75	0,75	0,99	0,85

Tabelle Intern (2)

$\|P\|$	$\|N\|$	Fanout	$\varnothing$-Fanout	Parsezeit in s	Erfolgsquote	Präzision	Trefferquote	Genauigkeit	F-Maß
13649	170	17	1,35	0,47	0,99	1,00	0,99	0,99	0,99
24916	181	16	1,37	1,15	0,99	1,00	0,99	0,99	0,99
50573	223	18	1,45	2,98	0,81	0,85	0,82	0,97	0,89
59070	227	18	1,46	3,38	0,86	0,90	0,86	0,96	0,91
80290	245	40	1,50	5,81	0,73	0,80	0,75	0,94	0,83
81800	246	40	1,50	6,23	0,79	0,85	0,80	0,95	0,87

Tabelle Intern (3)

$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Präzision	Trefferquote	Genauigkeit	F-Maß
13649	170	17	1,35	0,43	0,99	1,00	0,99	0,99	0,99
24916	181	16	1,37	1,09	0,99	1,00	0,99	0,98	0,98
50573	223	18	1,45	2,99	0,91	0,95	0,92	0,97	0,94
59070	227	18	1,46	3,49	0,83	0,85	0,83	0,98	0,90
80290	245	40	1,50	5,60	0,71	0,75	0,72	0,96	0,82
81800	246	40	1,50	5,83	0,76	0,80	0,77	0,96	0,85

Tabelle Intern (4)

$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Präzision	Trefferquote	Genauigkeit	F-Maß
13649	170	17	1,35	5,04	0,90	0,90	0,90	1,00	0,95
24916	181	16	1,37	13,20	0,50	0,50	0,50	1,00	0,67
50573	223	18	1,45	20,15	0,43	0,45	0,43	0,98	0,60
59070	227	18	1,46	13,64	0,54	0,55	0,54	0,99	0,70
80290	245	40	1,50	18,58	0,47	0,50	0,48	0,98	0,64
81800	246	40	1,50	20,93	0,38	0,40	0,39	0,98	0,56

Tabelle Extern (1)

$\|K\|$	$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Trefferquote	Genauigkeit	F-Maß
1638	13649	170	17	1,35	4,54	0,90	0,83	0,93	0,88
3713	24916	181	16	1,37	12,40	0,50	0,47	0,92	0,63
10073	50573	223	18	1,45	16,38	0,45	0,29	0,55	0,38
12703	59070	227	18	1,46	10,86	0,55	0,43	0,72	0,54
20034	80290	245	40	1,50	14,36	0,50	0,32	0,56	0,40
20602	81800	246	40	1,50	14,02	0,45	0,28	0,52	0,36

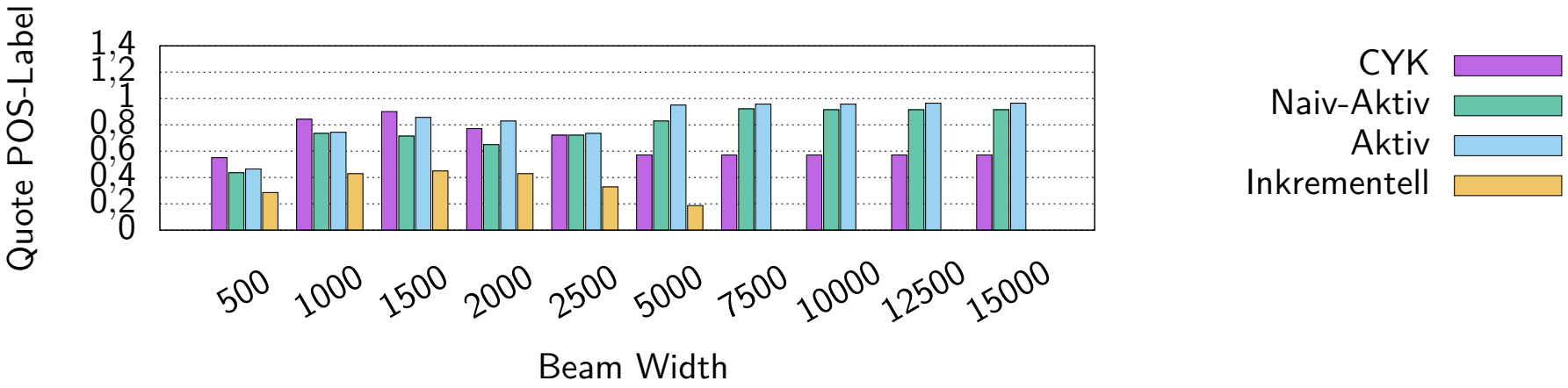
Tabelle Extern (2)

$\|K\|$	$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Trefferquote	Genauigkeit	F-Maß
1638	3230	83	8	1,83	0,23	1,00	0,84	0,74	0,79
3713	6071	92	9	1,89	0,54	1,00	0,75	0,64	0,69
10073	12385	107	9	1,99	2,08	1,00	0,57	0,47	0,51
12703	14682	113	9	2,02	2,87	1,00	0,59	0,42	0,49
20034	20361	122	9	2,07	3,99	1,00	0,62	0,48	0,54
20602	20754	123	9	2,07	2,54	1,00	0,49	0,36	0,41

Tabelle Extern (3)

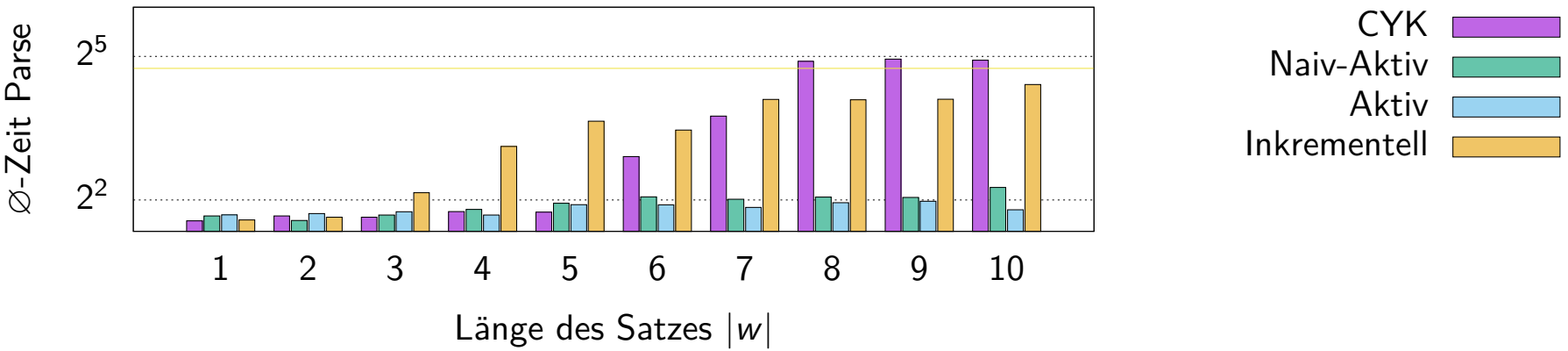
$\|K\|$	$\|P\|$	$\|N\|$	Fanout	∅-Fanout	Parsezeit in s	Erfolgsquote	Trefferquote	Genauigkeit	F-Maß
1638	9240	3860	8	/	0,08	1,00	0,83	0,85	0,84
3713	16097	6033	9	/	0,16	1,00	0,84	0,93	0,88
10073	29788	9928	9	/	0,32	1,00	0,80	0,87	0,84
12703	34522	11145	9	/	0,35	1,00	0,79	0,85	0,82
20034	45577	13997	9	/	0,50	1,00	0,82	0,90	0,86
20602	46220	14119	9	/	0,40	1,00	0,83	0,90	0,86

Untersuchung Beam Width (Intern)



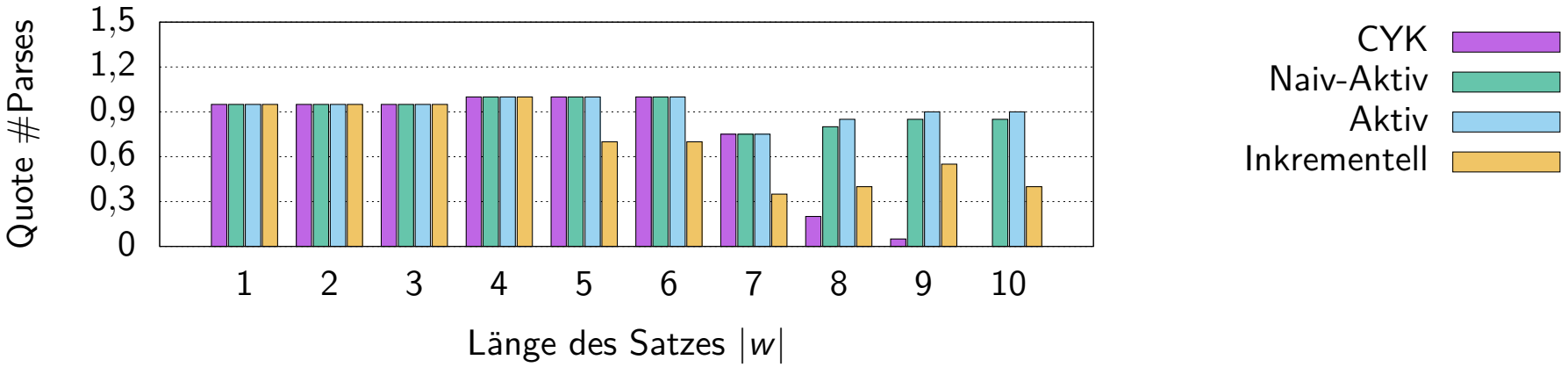
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



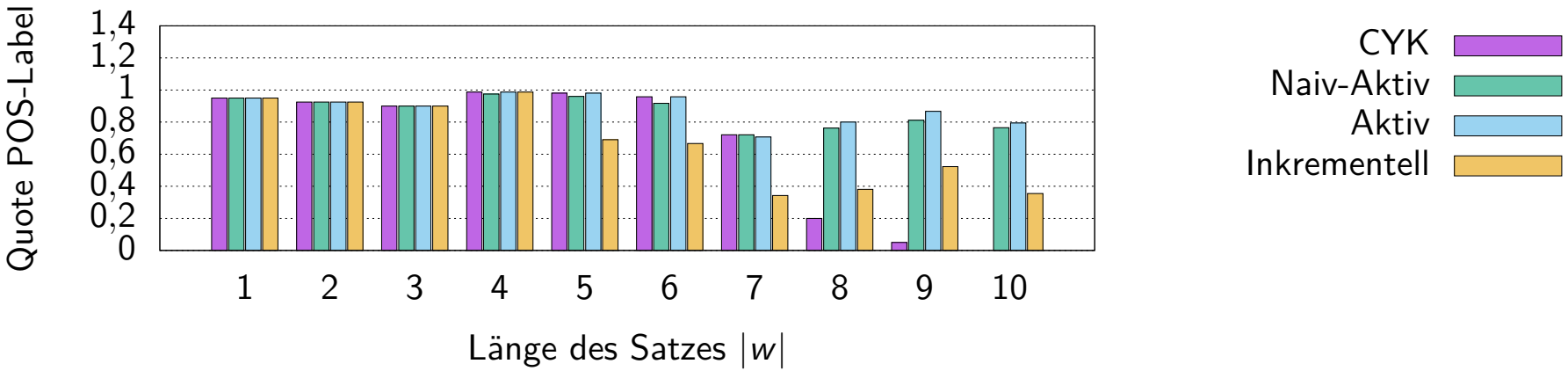
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



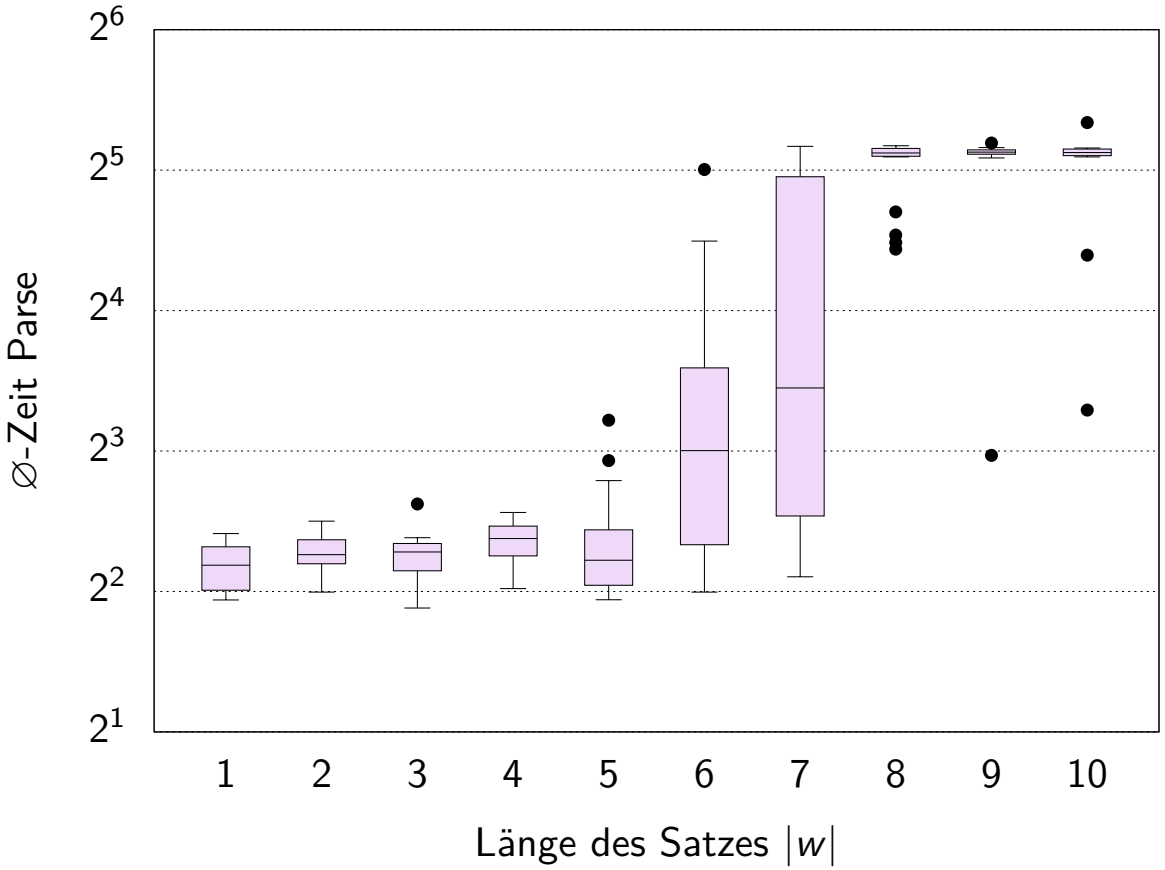
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



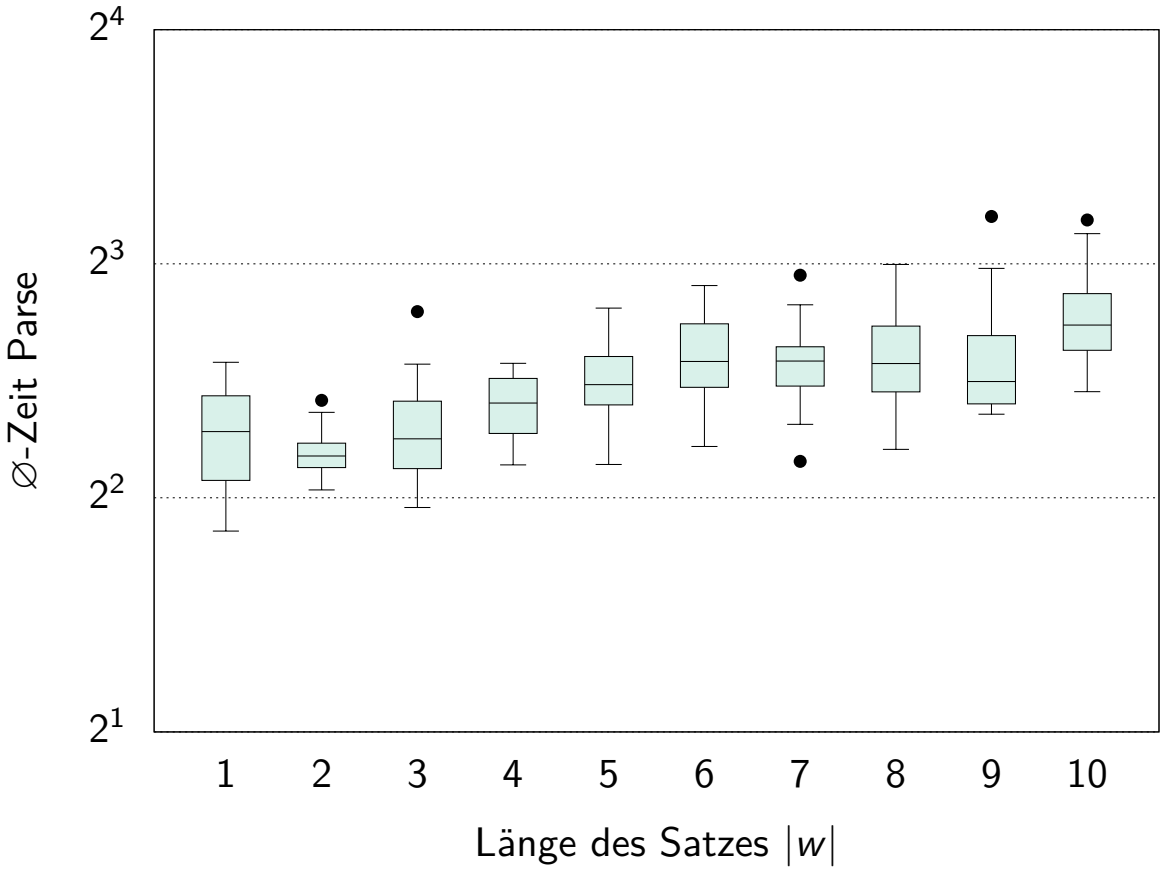
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



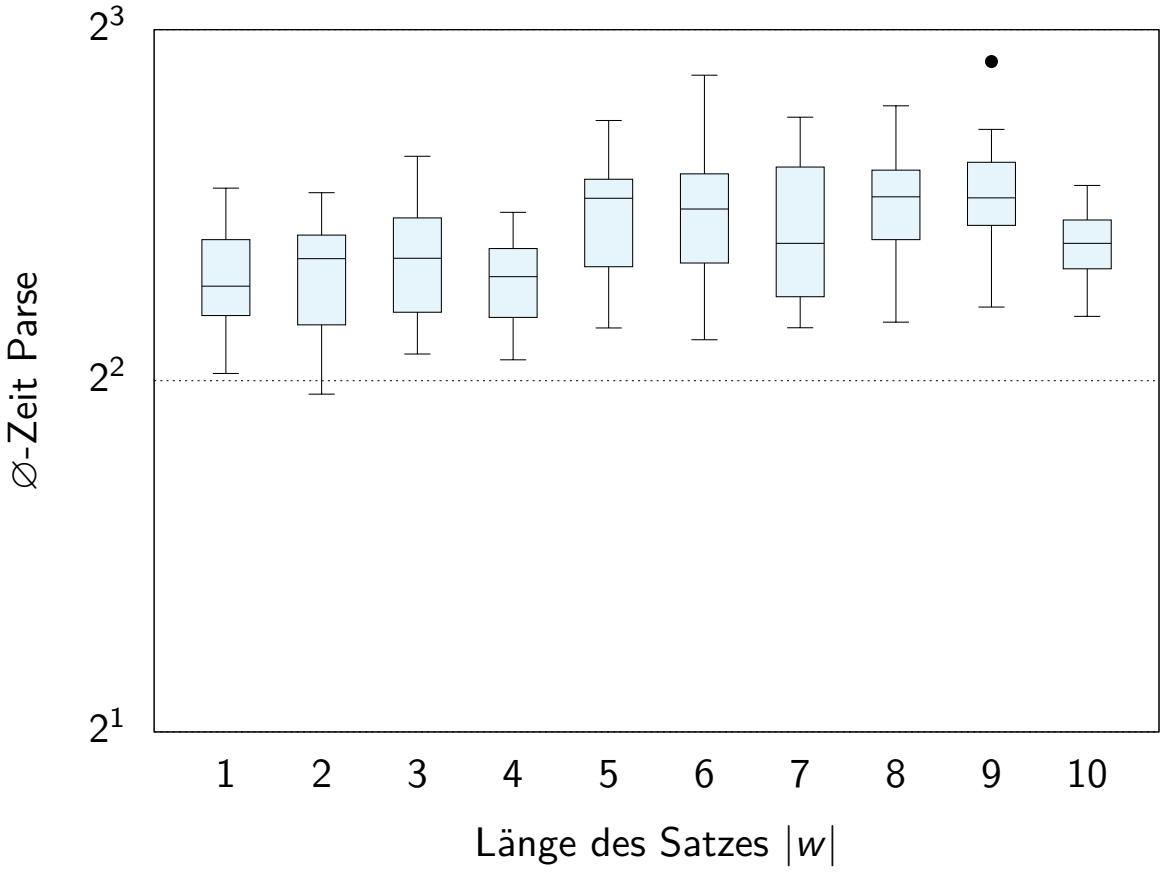
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



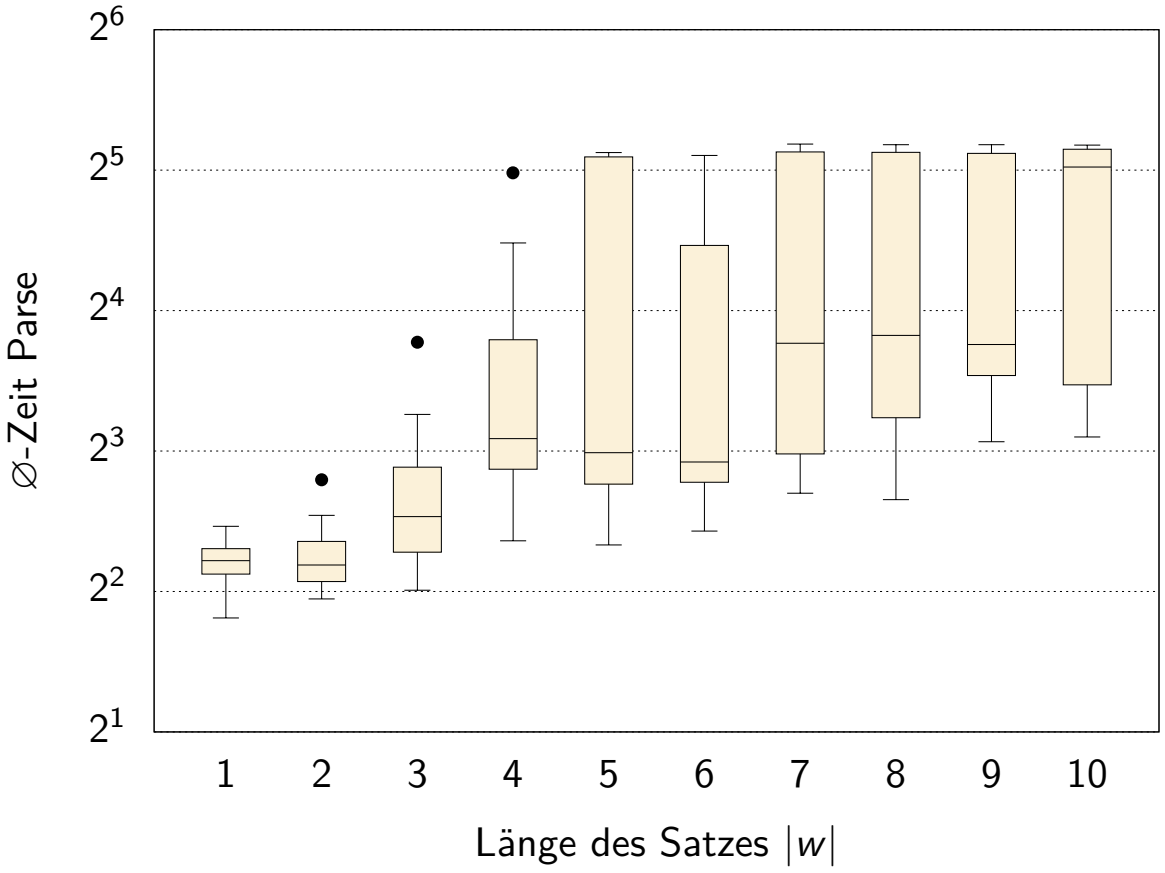
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



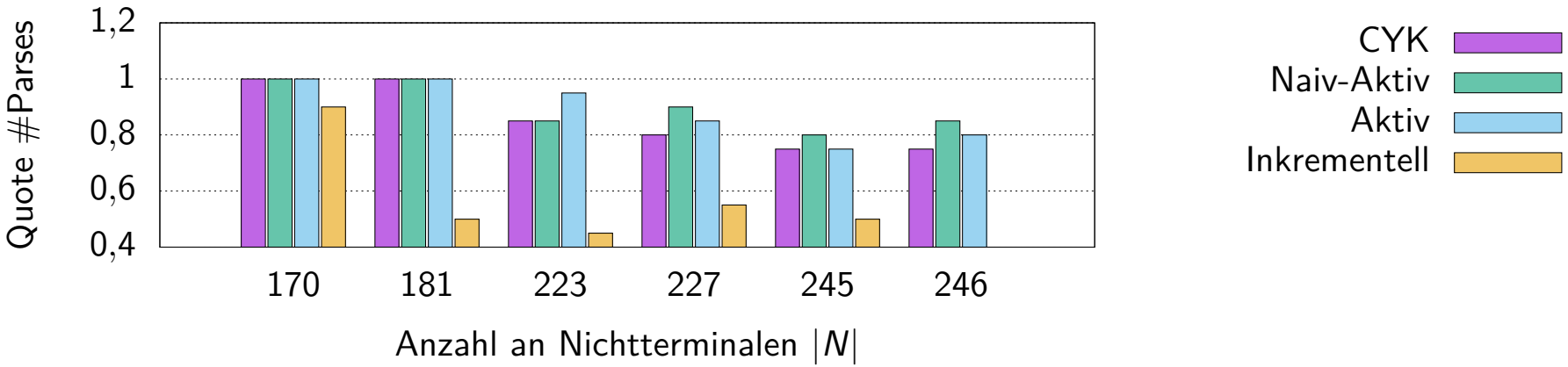
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Satzlänge (Intern)



- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

Untersuchung Grammatikgröße (Intern)



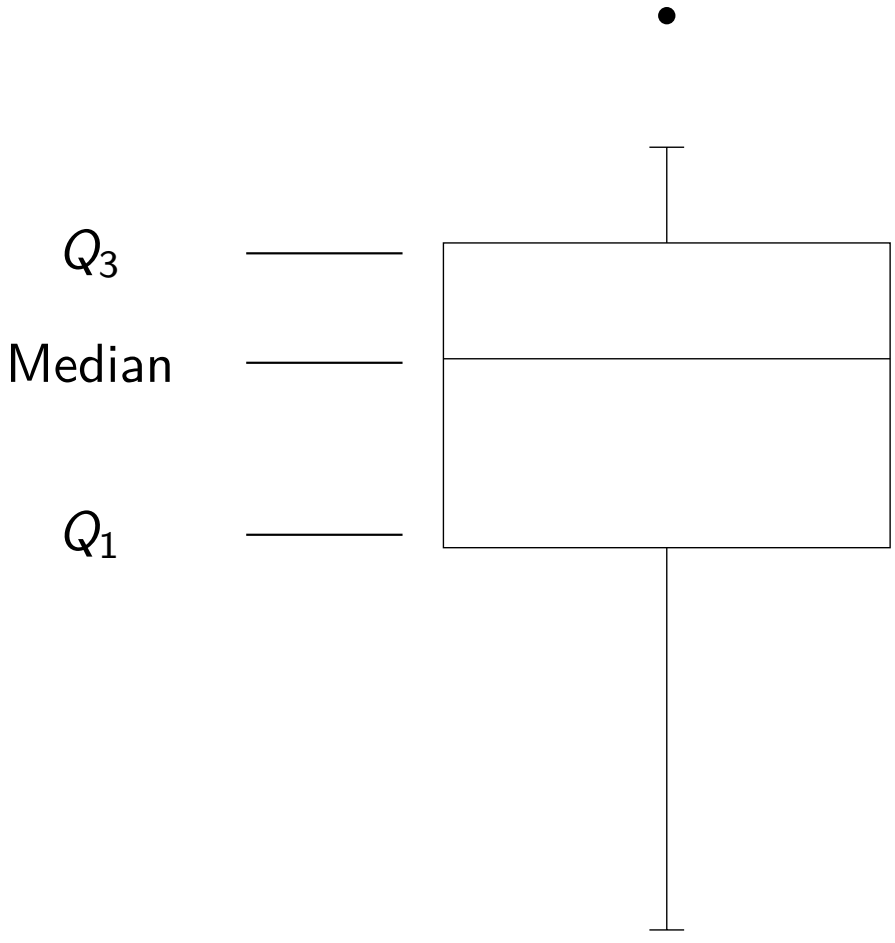
- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus
- Beam Width: 2500

Vergleich mit externen Parsern

- Untersuchung bei unterschiedlicher:
  - Satzlänge
  - Größe der Grammatik
- Beam Width inkrementeller Parser: 2500
- Timeout: 30 Sekunden (außer GF)
- Grammatik einmal pro Testmenge eingelesen
- rparse benötigt Testdaten mit POS-Labels
- Präzision: u.a. F-Maß der richtig vorhergesagten *Label* ohne POS-Label

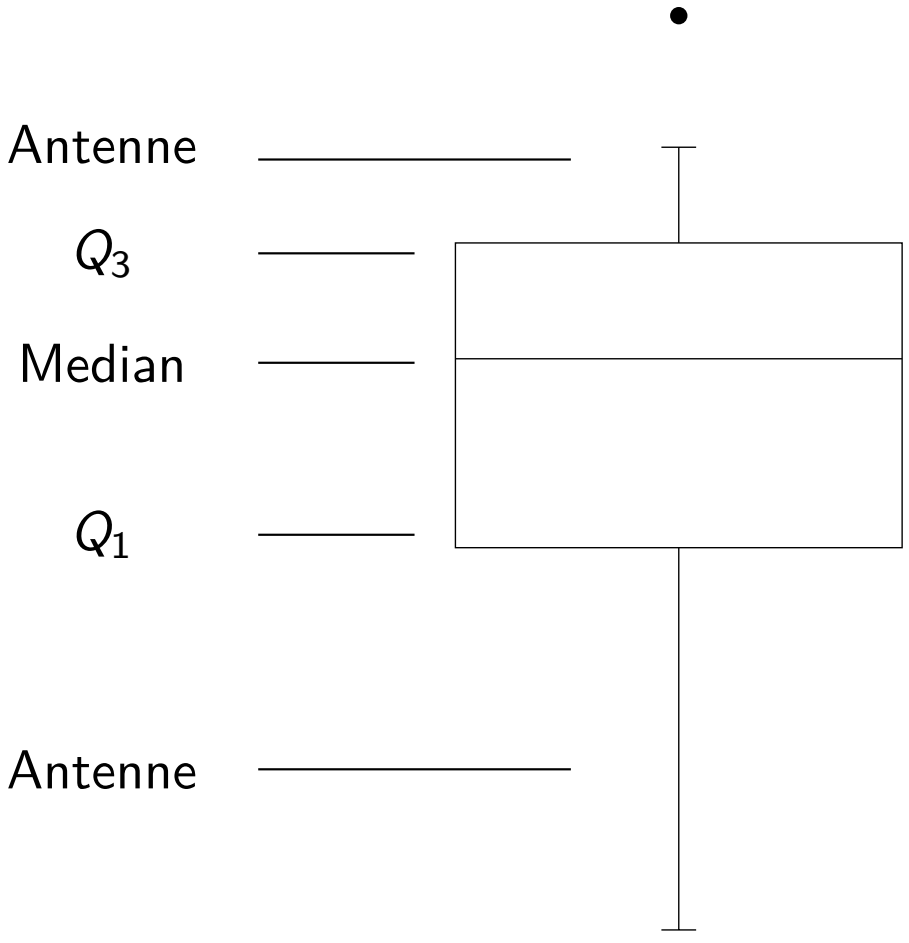
Boxplot

nach Ruprecht (2017)



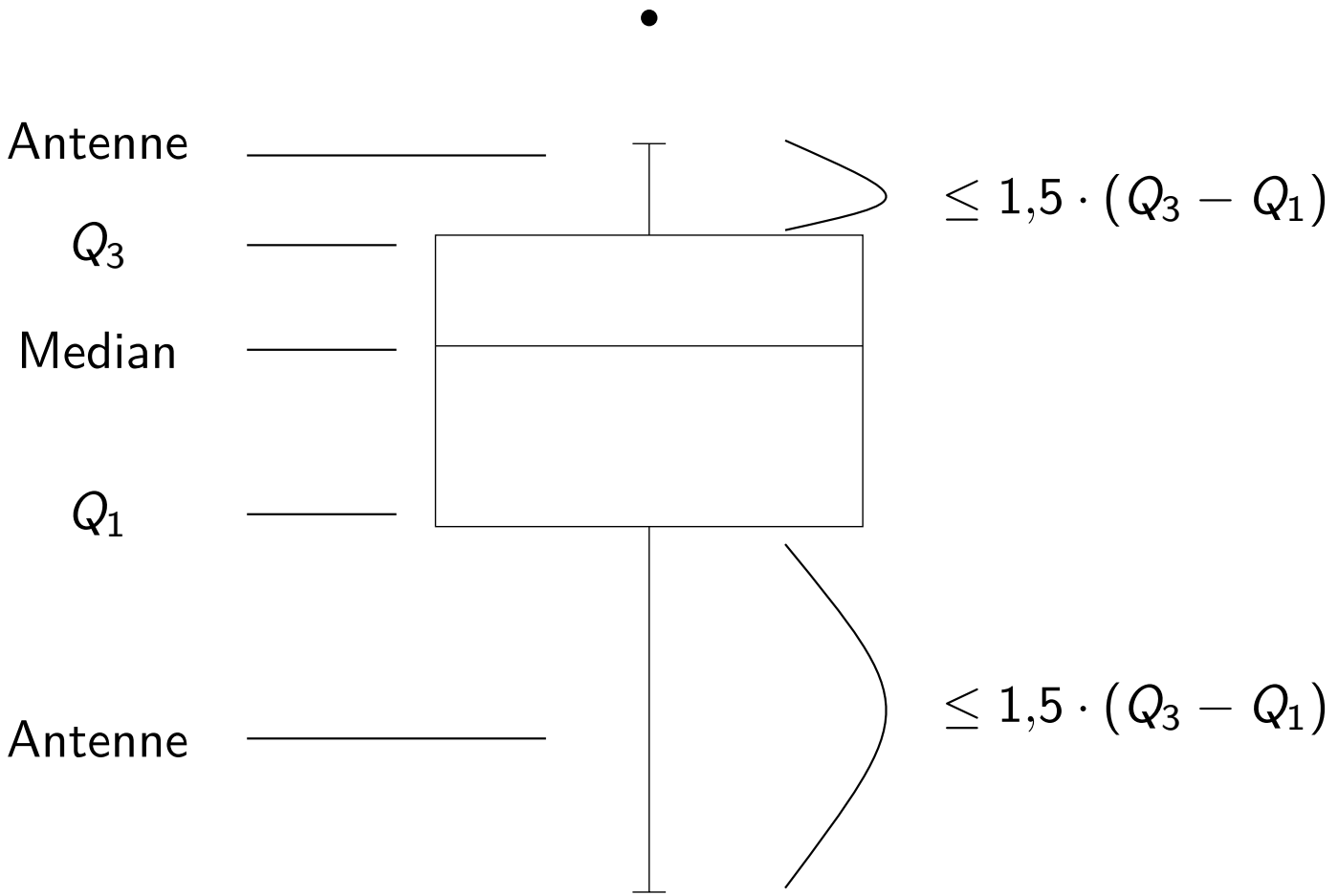
Boxplot

nach Ruprecht (2017)



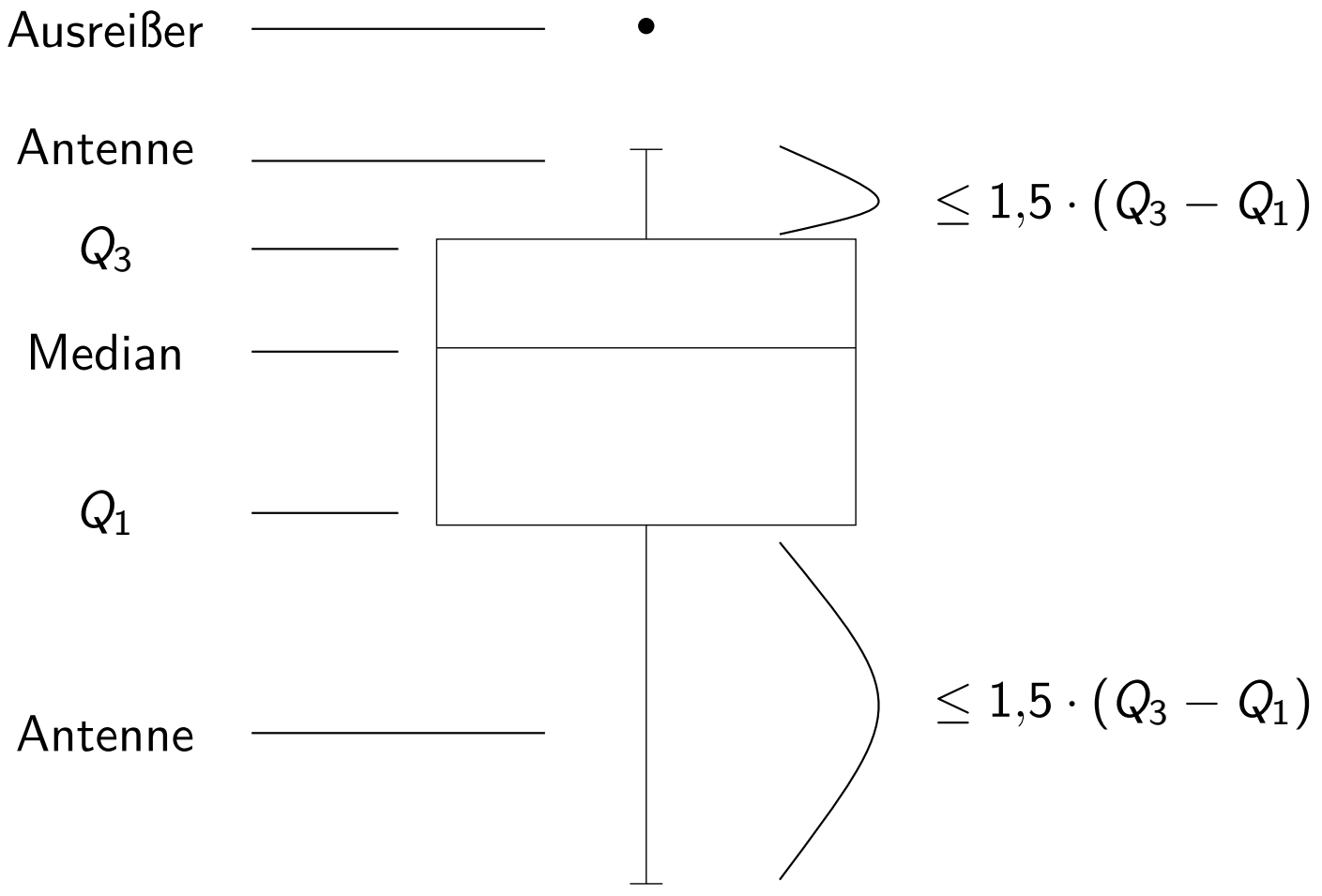
Boxplot

nach Ruprecht (2017)



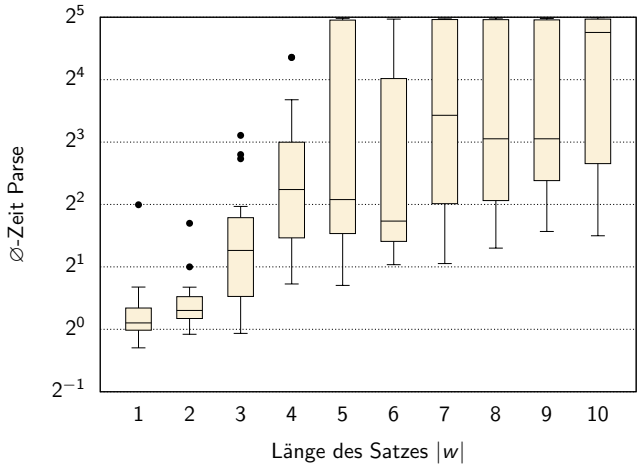
Boxplot

nach Ruprecht (2017)

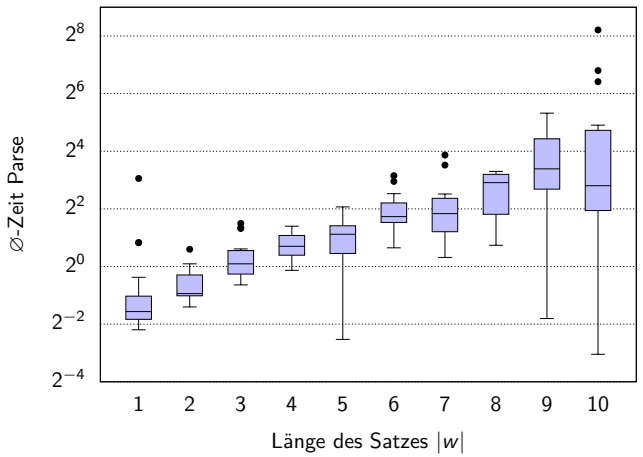


Untersuchung Satzlänge (Extern)

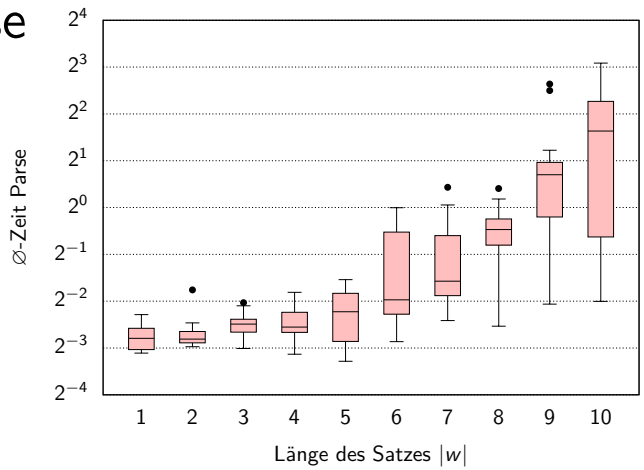
Ink



GF

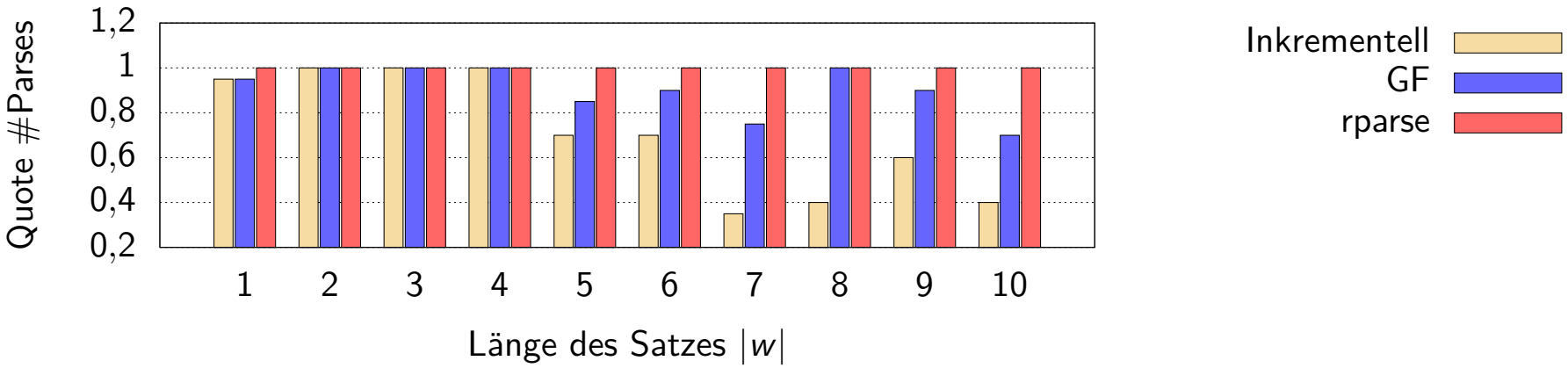


rparse



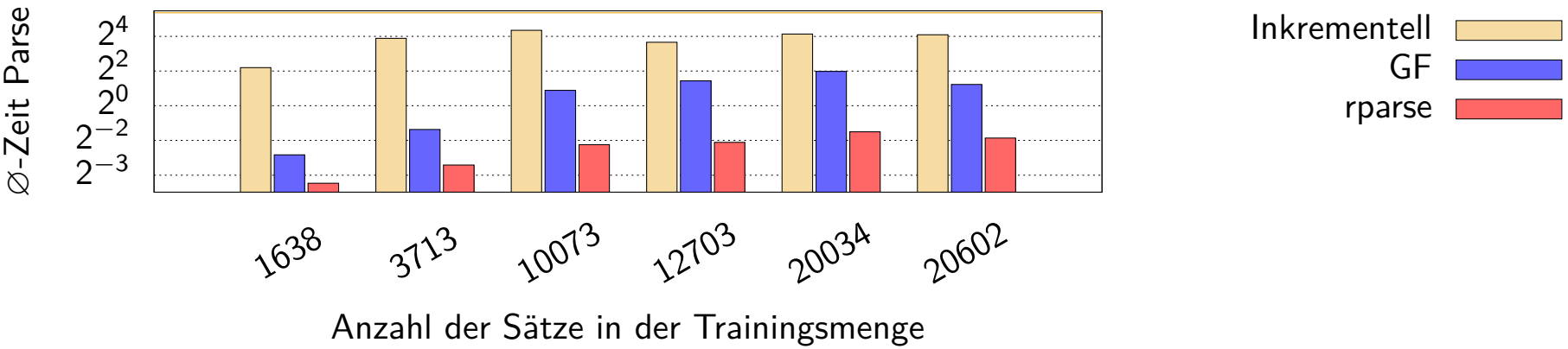
- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge

Untersuchung Satzlänge (Extern)



- Testmenge: Je 20 Sätze der Längen 1 bis 10
- Trainingsmenge: NeGra-Korpus ohne gesamte Testmenge
- Beam Width: 2500

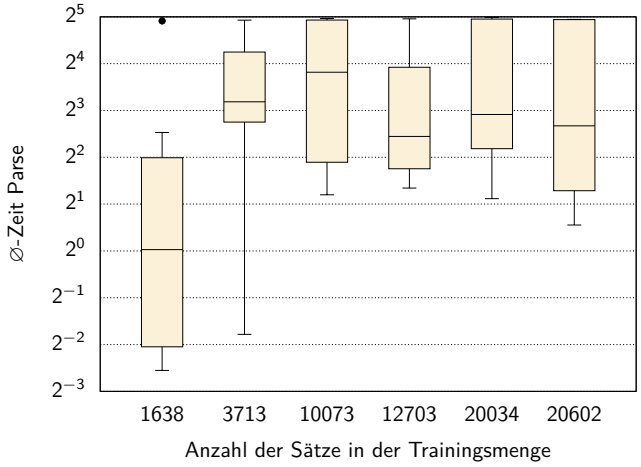
Untersuchung Grammatikgröße (Extern)



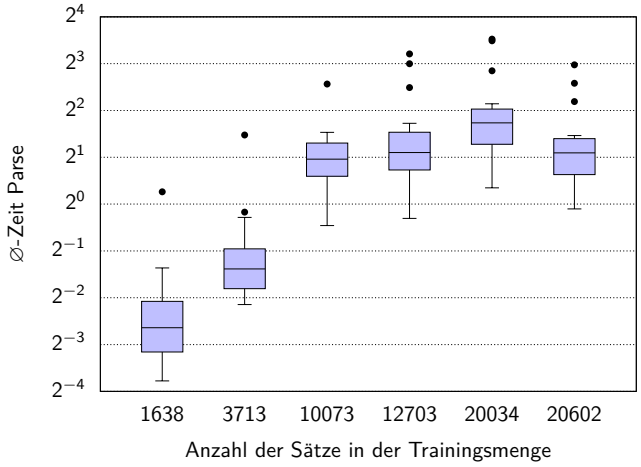
- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus

Untersuchung Grammatikgröße (Extern)

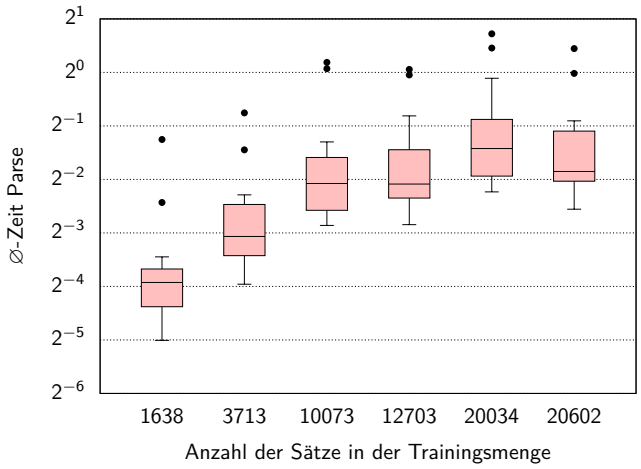
Ink



GF

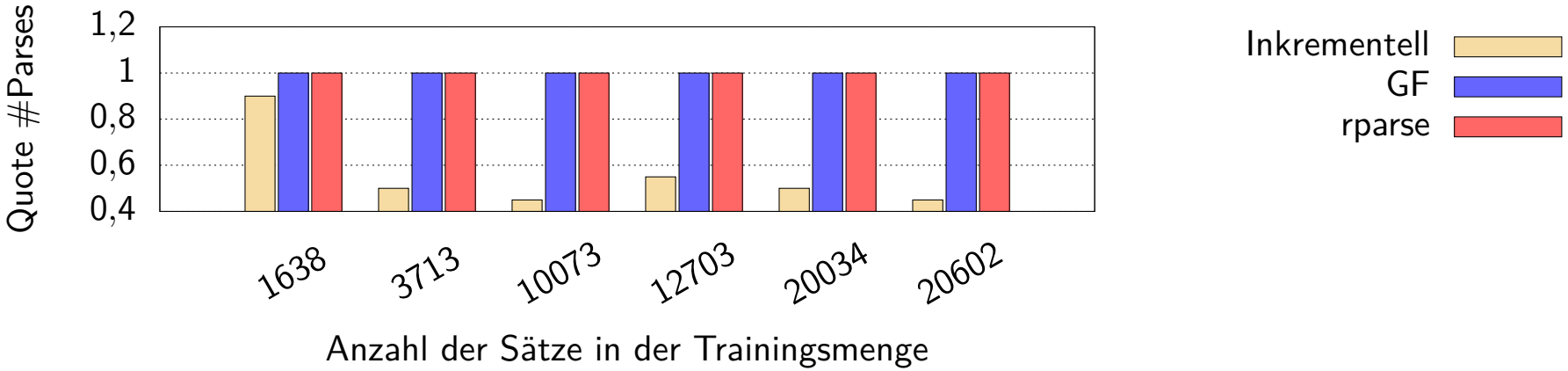


rparse



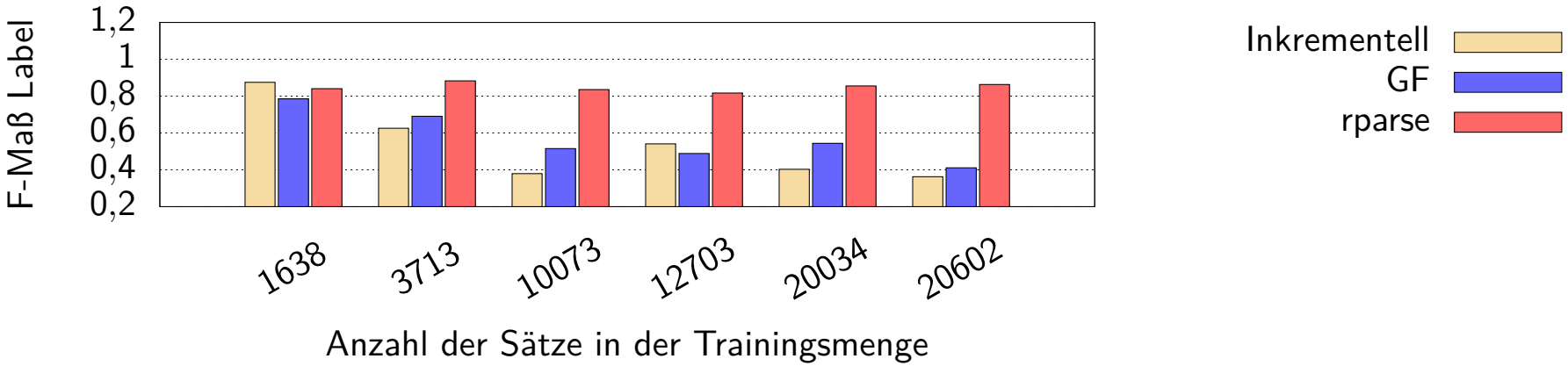
- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus

Untersuchung Grammatikgröße (Extern)



- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus

Untersuchung Grammatikgröße (Extern)



- Testmenge: 20 Sätze der Länge 7
- Trainingsmenge: 5 beliebige Teilmengen des NeGra-Korpus inklusive Testmenge; gesamter NeGra-Korpus