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

$$p_1 = S \to \langle x_{1,1} x_{1,2} \rangle (A)$$
$$p_2 = A \to \langle \epsilon, \epsilon \rangle ()$$
$$p_3 = A \to \langle \mathsf{a} x_{1,1} \mathsf{b}, \mathsf{a} x_{1,2} \mathsf{b} \rangle (A)$$

Wort w: abab (nach (Ruprecht, 2017))

Arten von Items:

- Aktiv: $[\phi, A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k), i, \rho \bullet u_i', \Gamma]$
  - ϕ: partielle Funktion $\mathbb{N} \to P$ (Welche Regel für Nichtterminale)
  - i: Index der betrachteten Komponente
  - ρ: Range
  - u_i': Verbleibende Komponente
  - Γ: partielle Funktion $(\mathbb{N} \times \mathbb{N}) \to$ Range (Ranges der Variablen)
- Passiv: $(A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_k), \zeta, \eta)$
  - ζ: Rangevektor aller Komponenten (der Länge ℓ)
  - η: Rangevektoren aller eingesetzten Nichtterminale

Implementierung und Evaluierung eines inkrementellen Parsers in Vanda

Komplexpraktikum - Zwischenpräsentation

Niklas Wünsche

Fakultät Informatik
TU Dresden

8. Juni 2018

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Übersicht

- Was bedeutet Parsen?
- Vorteile des Formalismus LCFRS
- Deduktionssysteme
- Inkrementeller Parser
- Aktueller Stand der Implementierung
- Weitere Aufgaben

Was bedeutet Parsen?

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

Was bedeutet Parsen?

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

Bedeutung von LCFRS

Problem:

- CFGs können keine Cross-Serial Dependencies darstellen (Shieber, 1985)
  - Parsen von CFGs in polynomieller Zeit möglich
- CSGs können Cross-Serial Dependencies darstellen
  - Parsingproblem von CSGs ist *PSPACE*-vollständig
- **Verbinden der positiven Eigenschaften möglich?**

Bedeutung von LCFRS

Problem:

- CFGs können keine Cross-Serial Dependencies darstellen (Shieber, 1985)
  - Parsen von CFGs in polynomieller Zeit möglich
- CSGs können Cross-Serial Dependencies darstellen
  - Parsingproblem von CSGs ist *PSPACE*-vollständig
- **Verbinden der positiven Eigenschaften möglich?**

Bedeutung von LCFRS

Problem:

- CFGs können keine Cross-Serial Dependencies darstellen (Shieber, 1985)
  - Parsen von CFGs in polynomieller Zeit möglich
- CSGs können Cross-Serial Dependencies darstellen
  - Parsingproblem von CSGs ist *PSPACE*-vollständig
- **Verbinden der positiven Eigenschaften möglich?**

Bedeutung von LCFRS

Problem:

- CFGs können keine Cross-Serial Dependencies darstellen (Shieber, 1985)
  - Parsen von CFGs in polynomieller Zeit möglich
- CSGs können Cross-Serial Dependencies darstellen
  - Parsingproblem von CSGs ist *PSPACE*-vollständig
- **Verbinden der positiven Eigenschaften möglich?**

Mildly Context-Sensitive Formalisms nach (Kallmeyer, 2010) vgl. (Joshi, 1985)

- Formalismen mit Ausdrucksstärke zwischen CFGs und CSGs
- Können Cross-Serial Dependencies darstellen
- In polynomieller Zeit parsbar
- Viele verschiedene Formalismen definiert
- **String - Linear Context-Free Rewriting Systems (LCFRS)**

  nach (Vijay-Shanker, Weir & Joshi, 1987)

Mildly Context-Sensitive Formalisms nach (Kallmeyer, 2010) vgl. (Joshi, 1985)

- Formalismen mit Ausdrucksstärke zwischen CFGs und CSGs
- Können Cross-Serial Dependencies darstellen
- In polynomieller Zeit parsbar
- Viele verschiedene Formalismen definiert
- **String - Linear Context-Free Rewriting Systems (LCFRS)**

  nach (Vijay-Shanker et al., 1987)

Mildly Context-Sensitive Formalisms nach (Kallmeyer, 2010) vgl. (Joshi, 1985)

- Formalismen mit Ausdrucksstärke zwischen CFGs und CSGs
- Können Cross-Serial Dependencies darstellen
- In polynomieller Zeit parsbar
- Viele verschiedene Formalismen definiert
- **String - Linear Context-Free Rewriting Systems (LCFRS)**

  nach (Vijay-Shanker et al., 1987)

Mildly Context-Sensitive Formalisms nach (Kallmeyer, 2010) vgl. (Joshi, 1985)

- Formalismen mit Ausdrucksstärke zwischen CFGs und CSGs
- Können Cross-Serial Dependencies darstellen
- In polynomieller Zeit parsbar
- Viele verschiedene Formalismen definiert
- **String - Linear Context-Free Rewriting Systems (LCFRS)**

  nach (Vijay-Shanker et al., 1987)

Mildly Context-Sensitive Formalisms nach (Kallmeyer, 2010) vgl. (Joshi, 1985)

- Formalismen mit Ausdrucksstärke zwischen CFGs und CSGs
- Können Cross-Serial Dependencies darstellen
- In polynomieller Zeit parsbar
- Viele verschiedene Formalismen definiert
- **String - Linear Context-Free Rewriting Systems (LCFRS)**

  nach (Vijay-Shanker et al., 1987)

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Beispiel einer LCFRS (Definiert nach (Vijay-Shanker et al., 1987))

- Menge von Regeln der Form
  $A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_i, \ldots, A_k)$
  - **Komponente**: u_i
  - Nichtterminal: A_i
  - Regeln nicht-löschend und linear

- Eigenschaften von G:
  - Regel p_3 mit Komponenten $\mathrm{a}x_{1,1}\mathrm{b}$ und $\mathrm{a}x_{1,2}\mathrm{b}$
  - Komponente $\mathrm{a}x_{1,1}\mathrm{b}$ enthält **Variable** $x_{1,1}$
  - Erzeugt die Sprache $L(G) = \{\mathrm{a}^n\mathrm{b}^n\mathrm{a}^n\mathrm{b}^n | n \in \mathbb{N}\}$

Ranges nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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 nach (Kallmeyer, 2010), (Ruprecht, 2017)

- **Range**: Paar von Indizes
  - Bezieht sich auf ein Wort
- Instanziierung von a in Komponente a$x_{1,1}$b von p_3:
  $(0,1)x_{1,1}$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: $\langle (0,1), (2,3) \rangle$

Deduktionssysteme nach (Shieber, Schabes & Pereira, 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Deduktionssysteme nach (Shieber et al., 1995)

Ein Deduktionssystem $D = (I, R)$ besteht aus

- Menge von Items I
- Menge von Deduktionsregeln $R \subseteq I^* \rightarrow I$ (partielle Funktionen)
- Deduktionsregel leitet aus **Vorgängern** eine **Konsequenz** ab

Notation Deduktionsregeln:

- $\frac{a_1, \ldots, a_n}{c}$ Nebenbedingung
- $a_1, \ldots, a_n \vdash c$, wenn $\exists r \in R : r(a_1, \ldots, a_n) = c$

Der inkrementelle Parser für LCFRS nach (Burden & Ljunglöf, 2005), (Kallmeyer, 2010)

Idee:

- Items für jede Regel und Komponente
- Terminale und Variablen in Items durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items

Der inkrementelle Parser für LCFRS nach (Burden & Ljunglöf, 2005), (Kallmeyer, 2010)

Idee:

- Items für jede Regel und Komponente
- Terminale und Variablen in Items durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items

Der inkrementelle Parser für LCFRS nach (Burden & Ljunglöf, 2005), (Kallmeyer, 2010)

Idee:

- Items für jede Regel und Komponente
- Terminale und Variablen in Items durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items

Der inkrementelle Parser für LCFRS nach (Burden & Ljunglöf, 2005), (Kallmeyer, 2010)

Idee:

- Items für jede Regel und Komponente
- Terminale und Variablen in Items durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items

Der inkrementelle Parser für LCFRS nach (Burden & Ljunglöf, 2005), (Kallmeyer, 2010)

Idee:

- Items für jede Regel und Komponente
- Terminale und Variablen in Items durch Ranges ersetzen
  - Terminale durch Instanziierung
  - Variablen durch Vorgänger-Items

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

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

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a}x_{1,1}\mathrm{b}, \emptyset]$

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

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

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

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

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

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

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

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

Initialisierung

Nutzen: Items ohne Vorgänger erzeugen

Für:

- Jede Regel $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$:

$$r_{Init} = \frac{}{[\emptyset, p, i, e \bullet u_i, \emptyset]}$$

Bereits abgeleitet: $\emptyset$

Neu abgeleitet:

$\vdash [\emptyset, p_2, 1, e\bullet, \emptyset]$

$\vdash [\emptyset, p_2, 2, e\bullet, \emptyset]$

$\vdash [\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

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

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Scannen

Nutzen: Terminalsymbole durch Ranges ersetzen

Für:

- $p = (A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_k)) \in P$
- $w = \delta_1 \ldots \delta_{b+1} \ldots \delta_n$ für $b, n \in \mathbb{N}, \delta_1, \ldots, \delta_n \in \Sigma$

$$r_{Scannen} = \frac{[\phi, p, i, (a, b) \bullet \delta_{b+1} u_i', \Gamma]}{[\phi, p, i, (a, b+1) \bullet u_i', \Gamma]}$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, e \bullet \mathrm{a} x_{1,1} \mathrm{b}, \emptyset]$

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

Neu abgeleitet:

$1 \vdash [\emptyset, p_3, 1, (0, 1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$1 \vdash [\emptyset, p_3, 1, (2, 3) \bullet x_{1,1} \mathrm{b}, \emptyset]$

$2 \vdash [\emptyset, p_3, 2, (2, 3) \bullet x_{1,2} \mathrm{b}, \emptyset]$

Kombinieren

Nutzen: Variablen durch Ranges ersetzen

Für:

- $p = A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$
- $p' = A_m \to \langle v_1, \ldots, v_n, \ldots v_j \rangle (B_1, \ldots, B_o) \in P$

$$\frac{[\phi, p, i, \rho \bullet x_{m,n} u'_i, \Gamma], [\phi', p', n, \rho' \bullet, \Gamma']}{[\phi \cup \{(m, p')\}, p, i, \rho \cdot \rho' \bullet u'_i, \Gamma \cup \{((m, n), \rho')\}]} \; \phi(m) = \emptyset \vee \phi(m) = p'$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, (0,1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

2: $[\emptyset, p_2, 1, e \bullet, \emptyset]$

Neu abgeleitet:

$1,2 \vdash [\{x_1 \to p_2\}, p_3, 1, (0,1) \bullet \mathrm{b}, \{x_{1,1} \to e\}]$

Kombinieren

Nutzen: Variablen durch Ranges ersetzen

Für:

- $p = A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$
- $p' = A_m \to \langle v_1, \ldots, v_n, \ldots v_j \rangle (B_1, \ldots, B_o) \in P$

$$\frac{[\phi, p, i, \rho \bullet x_{m,n} u'_i, \Gamma], [\phi', p', n, \rho' \bullet, \Gamma']}{[\phi \cup \{(m, p')\}, p, i, \rho \cdot \rho' \bullet u'_i, \Gamma \cup \{((m, n), \rho')\}]} \; \phi(m) = \emptyset \vee \phi(m) = p'$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, (0,1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

2: $[\emptyset, p_2, 1, e \bullet, \emptyset]$

Neu abgeleitet:

$1,2 \vdash [\{x_1 \to p_2\}, p_3, 1, (0,1) \bullet \mathrm{b}, \{x_{1,1} \to e\}]$

Kombinieren

Nutzen: Variablen durch Ranges ersetzen

Für:

- $p = A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$
- $p' = A_m \to \langle v_1, \ldots, v_n, \ldots v_j \rangle (B_1, \ldots, B_o) \in P$

$$\frac{[\phi, p, i, \rho \bullet x_{m,n} u'_i, \Gamma], [\phi', p', n, \rho' \bullet, \Gamma']}{[\phi \cup \{(m, p')\}, p, i, \rho \cdot \rho' \bullet u'_i, \Gamma \cup \{((m, n), \rho')\}]} \; \phi(m) = \emptyset \vee \phi(m) = p'$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, (0,1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

2: $[\emptyset, p_2, 1, e \bullet, \emptyset]$

Neu abgeleitet:

$1,2 \vdash [\{x_1 \to p_2\}, p_3, 1, (0,1) \bullet \mathrm{b}, \{x_{1,1} \to e\}]$

Kombinieren

Nutzen: Variablen durch Ranges ersetzen

Für:

- $p = A \to \langle u_1, \ldots, u_i, \ldots u_\ell \rangle (A_1, \ldots, A_m, \ldots, A_k) \in P$
- $p' = A_m \to \langle v_1, \ldots, v_n, \ldots v_j \rangle (B_1, \ldots, B_o) \in P$

$$\frac{[\phi, p, i, \rho \bullet x_{m,n} u'_i, \Gamma], [\phi', p', n, \rho' \bullet, \Gamma']}{[\phi \cup \{(m, p')\}, p, i, \rho \cdot \rho' \bullet u'_i, \Gamma \cup \{((m, n), \rho')\}]} \; \phi(m) = \emptyset \vee \phi(m) = p'$$

Bereits abgeleitet:

1: $[\emptyset, p_3, 1, (0,1) \bullet x_{1,1} \mathrm{b}, \emptyset]$

2: $[\emptyset, p_2, 1, e \bullet, \emptyset]$

Neu abgeleitet:

$1{,}2 \vdash [\{x_1 \to p_2\}, p_3, 1, (0,1) \bullet \mathrm{b}, \{x_{1,1} \to e\}]$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$
$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

$1,2 \vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
$3 \vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$

$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

1,2 $\vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
3 $\vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$

$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha, \quad \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

$1,2 \vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
$3 \vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$

$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$

$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

1,2 $\vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
3 $\vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$

$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

$1,2 \vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
$3 \vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$
$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

$1,2 \vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
$3 \vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$
$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

$1,2 \vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
$3 \vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Vervollständigen

Nutzen: Items gleicher Funktion zusammenführen

Für:

- $p = (A \to \langle u_1, \ldots, u_\ell \rangle (A_1, \ldots, A_m, \ldots A_k)) \in P$
- $p' = (A_m \to \langle u'_1, \ldots, u'_m \rangle (A_1, \ldots, A_k)) \in P$

$$r_{Vervoll} = \frac{[\phi_1, p, 1, \rho_1 \bullet, \Gamma_1], \ldots, [\phi_\ell, p, \ell, \rho_\ell \bullet, \Gamma_\ell]}{(p, \langle \rho_1, \ldots, \rho_\ell \rangle, \langle \rho'_1, \ldots, \rho'_k \rangle)}$$
$$(\Gamma := \bigcup_{\alpha \in [\ell]} \Gamma_\alpha,\ \forall i \in [k] : \rho'_i := \langle \Gamma(i,1), \ldots, \Gamma(i,m) \rangle,$$
$$\forall y, z \in [\ell] : \forall v \in [k] : \phi_y(v) = \phi_z(v) \vee \phi_y(v) = \emptyset \vee \phi_z(v) = \emptyset)$$

Bereits abgeleitet:

1: $[\{x_1 \to p_2\}, p_3, 1, (0,2)\bullet, \{x_{1,1} \to e\}]$
2: $[\{x_1 \to p_2\}, p_3, 2, (2,4)\bullet, \{x_{1,2} \to e\}]$
3: $[\{x_1 \to p_3\}, p_1, 1, (0,4)\bullet, \{x_{1,1} \to (0,2), x_{1,2} \to (2,4)\}]$

Neu abgeleitet:

1,2 $\vdash (p_3, \langle (0,2), (2,4) \rangle, \langle \langle e, e \rangle \rangle)$
3 $\vdash (p_1, \langle (0,4) \rangle, \langle \langle (0,2), (2,4) \rangle \rangle)$

Übersicht Deduktionsregeln

- Initialisierung: Items ohne Vorgänger erzeugen
- Scannen: Terminalsymbole durch Ranges ersetzen
- Kombinieren: Variablen durch Ranges ersetzen
- Vervollständigen: Items der gleichen Funktion zusammenführen

- Zielitem $(p_s, \langle(0, |w|)\rangle, \rho')$ abgeleitet $\Leftrightarrow w \in L(G)$

$(p_1, \langle(0, 4)\rangle, \langle\langle(0, 2), (2, 4)\rangle\rangle)$ abgeleitet $\Rightarrow$ abab $\in L(G)$

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Stand des Praktikums

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

Aktueller Stand:

- Inkrementeller Parser in Vanda-Haskell implementiert
- Generierung von Ableitungsbäumen

Ausblick:

- Kompatibilität implementieren
- Quelltext der Implementation optimieren
- Gewichtete LCFRS und Heuristiken einführen
- Vergleich von verschiedenen LCFRS-Parsern

Vielen Dank für Ihre Aufmerksamkeit!

Quellen I


Burden, H. & Ljunglöf, P. (2005). Parsing linear context-free rewriting systems. In *Proceedings of the ninth international workshop on parsing technology* (pp. 11–17). Parsing '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1654494.1654496


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 (Eds.), *Natural language parsing: Psychological, computational, and theoretical perspectives* (pp. 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.

Quellen II

- Ruprecht, T. (2017). *Parsing linear context-free rewriting systems*. Techn. Ber., Technische Universität Dresden.
- Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, *8*(3), 333–343. doi:10.1007/BF00630917
- Shieber, S. M., Schabes, Y. & Pereira, F. C. (1995). Principles and implementation of deductive parsing. *The Journal of Logic Programming*, *24*(1), 3–36. Computational Linguistics and Logic Programming. doi:https://doi.org/10.1016/0743-1066(95)00035-I
- Vijay-Shanker, K., Weir, D. J. & Joshi, A. K. (1987). Characterizing structural descriptions produced by various grammatical formalisms. In *Proc. of the 25th annual meeting on acl*.