TECHNISCHE UNIVERSITÄT DRESDEN

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 3: EXTENDED BACKUS-NAUR-FORM

Eric Kunze

eric.kunze@tu-dresden.de

TU Dresden, 26. Oktober 2021

EBNF und ihre Semantik

EBNF-DEFINITION

$$\mathcal{E} = (\underbrace{V}_{\text{Nichtterminalsymbole}}, \underbrace{\Sigma}_{\text{Terminalsymbole}}, \underbrace{S}_{\text{Startsymbol}}, \underbrace{R}_{\text{Regelmenge}})$$

Jede **EBNF-Regel** besteht aus einer linken und einer rechten Seite, die rechte Seite ist ein **EBNF-Term**.

$$\textit{Nichtterminalsymbol} ::= \textit{EBNF-Term}$$

EBNF-DEFINITION

$$\mathcal{E} = (\ V,\ \Sigma,\ S,\ R\ )$$

Nichtterminalsymbole Terminalsymbole Startsymbol Regelmenge

Jede **EBNF-Regel** besteht aus einer linken und einer rechten Seite, die rechte Seite ist ein **EBNF-Term**.

$$\textit{Nichtterminalsymbol} ::= \textit{EBNF-Term}$$

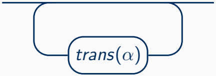
Definition (EBNF-Terme): Seien V (syntaktische Variablen) und Σ (Terminalsymbole) endliche Mengen mit $V \cap \Sigma = \emptyset$. Die Menge der EBNF-Terme über V und Σ (notiere: $T(\Sigma, V)$), ist die *kleinste* Menge $T \subseteq \left(V \cup \Sigma \cup \left\{\hat{\{}, \hat{\}}, \hat{[}, \hat{]}, \hat{(}, \hat{)}, \hat{|}\right\}\right)$ mit $V \subseteq T$, $\Sigma \subseteq T$ und

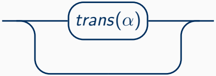
- Wenn $\alpha \in T$, so auch $\hat{(}\ \alpha\ \hat{)} \in T$, $\hat{\{}\ \alpha\ \hat{\}} \in T$, $\hat{[}\ \alpha\ \hat{]} \in T$.
- Wenn $\alpha_1, \alpha_2 \in T$, so auch $\hat{(}\ \alpha_1\ \hat{|}\ \alpha_2\ \hat{)} \in T$, $\alpha_1\alpha_2 \in T$.

ÜBERSETZUNG EBNF $\leftrightarrow$ SYNTAXDIAGRAMME

Sei $v \in V$ und $w \in \Sigma$. $\mathit{trans}(v) = \boxed{v}$; $\mathit{trans}(w) = w$

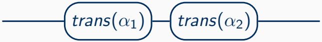
Sei $\alpha \in T(\Sigma, V)$ ein EBNF-Term.

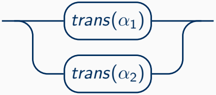
- $\mathit{trans}(\ \hat{\{}\ \alpha\ \hat{\}}\ ) =$ 

- $\mathit{trans}(\ \hat{[}\ \alpha\ \hat{]}\ ) =$ 

- $\mathit{trans}(\ \hat{(}\ \alpha\ \hat{)}\ ) = \mathit{trans}(\alpha)$

Seien $\alpha_1, \alpha_2 \in T(\Sigma, V)$ zwei EBNF-Terme.

- $\mathit{trans}(\ \alpha_1\alpha_2\ ) =$ 

- $\mathit{trans}(\ \hat{(}\ \alpha_1\ \hat{|}\ \alpha_2\ \hat{)}\ ) =$ 

SEMANTIK VON EBNF-TERMEN

Ziel: Ordne einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ ihre Sprache zu

- $W(\mathcal{E}, v)$ bezeichnet von $v \in V$ beschriebene Objektsprache
- $\rho\colon V \to \mathcal{P}(\Sigma^*)$ ordnet jeder syntaktischen Variable $v \in V$ eine Sprache zu
- Vorstellung: $\rho(v)$ ist bestes Wissen über die von v beschriebene Sprache

SEMANTIK VON EBNF-TERMEN

Ziel: Ordne einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ ihre Sprache zu

- $W(\mathcal{E}, v)$ bezeichnet von $v \in V$ beschriebene Objektsprache
- $\rho \colon V \to \mathcal{P}(\Sigma^*)$ ordnet jeder syntaktischen Variable $v \in V$ eine Sprache zu
- Vorstellung: $\rho(v)$ ist bestes Wissen über die von v beschriebene Sprache

Problem: Wie bekomme ich aus einem EBNF-Term eine Sprache?

Semantik $[\![\cdot]\!] : \underbrace{T(\Sigma, V)}_{\text{EBNF-Term } \alpha} \to ((\underbrace{V \to \mathcal{P}(\Sigma^*)}_{\rho}) \to \mathcal{P}(\Sigma^*))$

SEMANTIK VON EBNF-TERMEN

$$[\![\cdot]\!] : \underbrace{T(\Sigma, V)}_{\text{EBNF-Term } \alpha} \to (\underbrace{(V \to \mathcal{P}(\Sigma^*))}_{\rho} \to \mathcal{P}(\Sigma^*))$$

Sei $\alpha \in T(\Sigma, V)$ ein EBNF-Term. Die Semantik $[\![\alpha]\!](\rho)$ von α ist definiert als:

- Wenn $\alpha = v \in V$, dann gilt $[\![\alpha]\!](\rho) = \rho(v)$.
- Wenn $\alpha = w \in \Sigma$, dann gilt $[\![\alpha]\!](\rho) = \{w\}$.
- Wenn $\alpha = \hat{\{}\ \alpha_1\ \hat{\}}$, dann gilt $[\![\alpha]\!](\rho) = ([\![\alpha_1]\!](\rho))^*$.
- Wenn $\alpha = \hat{[}\ \alpha_1\ \hat{]}$, dann gilt $[\![\alpha]\!](\rho) = [\![\alpha_1]\!](\rho) \cup \{\varepsilon\}$.
- Wenn $\alpha = \hat{(}\ \alpha_1\ \hat{)}$, dann gilt $[\![\alpha]\!](\rho) = [\![\alpha_1]\!](\rho)$.
- Wenn $\alpha = \alpha_1\alpha_2$, dann gilt $[\![\alpha]\!](\rho) = [\![\alpha_1]\!](\rho) \cdot [\![\alpha_2]\!](\rho)$.
- Wenn $\alpha = \hat{(}\ \alpha_1 \ \hat{|}\ \alpha_2\ \hat{)}$, dann gilt $[\![\alpha]\!](\rho) = [\![\alpha_1]\!](\rho) \cup [\![\alpha_2]\!](\rho)$.

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g\colon \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g : \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

Methode: Newtonverfahren — definiere $\Phi(x) := x - \frac{g(x)}{g'(x)}$.

- Starte mit „beliebigem“ Startwert $x_0 \in \mathbb{R}$.
- Berechne stets $x_{i+1} = \Phi(x_i)$.

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g : \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

Methode: Newtonverfahren — definiere $\Phi(x) := x - \frac{g(x)}{g'(x)}$.

- Starte mit „beliebigem" Startwert $x_0 \in \mathbb{R}$.
- Berechne stets $x_{i+1} = \Phi(x_i)$.

Beobachtung:
x_i nähert sich der Nullstelle $\overline{x}$ an

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g : \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

Methode: Newtonverfahren — definiere $\Phi(x) := x - \frac{g(x)}{g'(x)}$.

- Starte mit „beliebigem" Startwert $x_0 \in \mathbb{R}$.
- Berechne stets $x_{i+1} = \Phi(x_i)$.

Beobachtung:
x_i nähert sich der Nullstelle $\overline{x}$ an

Ein *Fixpunkt* von Φ ist ein Punkt x mit $\Phi(x) = x$.

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g : \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

Methode: Newtonverfahren — definiere $\Phi(x) := x - \frac{g(x)}{g'(x)}$.

- Starte mit „beliebigem" Startwert $x_0 \in \mathbb{R}$.
- Berechne stets $x_{i+1} = \Phi(x_i)$.

Beobachtung:

x_i nähert sich der Nullstelle $\overline{x}$ an

Ein *Fixpunkt* von Φ ist ein Punkt x mit $\Phi(x) = x$.

Die Nullstelle $\overline{x}$ ist ein Fixpunkt von Φ, da

$$\Phi(\overline{x}) = \overline{x} - \frac{g(\overline{x})}{g'(\overline{x})} = \overline{x}.$$

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

Gegeben sei eine Funktion $g : \mathbb{R} \to \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\overline{x} \in \mathbb{R}$ mit $g(\overline{x}) = 0$.

Methode: Newtonverfahren — definiere $\Phi(x) := x - \frac{g(x)}{g'(x)}$.

- Starte mit „beliebigem" Startwert $x_0 \in \mathbb{R}$.
- Berechne stets $x_{i+1} = \Phi(x_i)$.

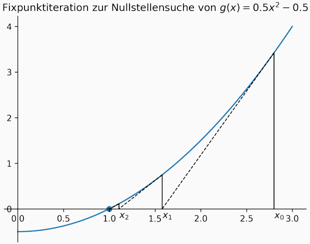
Beobachtung:

x_i nähert sich der Nullstelle $\overline{x}$ an

Ein *Fixpunkt* von Φ ist ein Punkt x mit $\Phi(x) = x$.

Die Nullstelle $\overline{x}$ ist ein Fixpunkt von Φ, da

$$\Phi(\overline{x}) = \overline{x} - \frac{g(\overline{x})}{g'(\overline{x})} = \overline{x}.$$



FIXPUNKTITERATION FÜR EBNF

Ziel: berechne Sprache $W(\mathcal{E}, v)$ für alle $v \in V$ einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$.

FIXPUNKTITERATION FÜR EBNF

Ziel: berechne Sprache $W(\mathcal{E}, v)$ für alle $v \in V$ einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$.

Iterierende Funktion:

$$f\colon \underbrace{(V \to \mathcal{P}(\Sigma^*))}_{\rho} \to (V \to \mathcal{P}(\Sigma^*))$$

- Starte mit bisherigen Kenntnis $\rho(v) = \emptyset$ für alle $v \in V$.
(Nichtswissen)
- Berechne stets neues Wissen $\rho_{\text{neu}} = f(\rho_{\text{alt}})$.
(Generiere neues Wissen)

FIXPUNKTITERATION FÜR EBNF

Ziel: berechne Sprache $W(\mathcal{E}, v)$ für alle $v \in V$ einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$.

Iterierende Funktion:

$$f: \underbrace{(V \to \mathcal{P}(\Sigma^*))}_{\rho} \to (V \to \mathcal{P}(\Sigma^*))$$

- Starte mit bisherigen Kenntnis $\rho(v) = \emptyset$ für alle $v \in V$. (Nichtswissen)
- Berechne stets neues Wissen $\rho_{\text{neu}} = f(\rho_{\text{alt}})$. (Generiere neues Wissen)

Ende: erreiche einen Fixpunkt ρ mit $f(\rho) = \rho$

Dann gilt $\rho(v) = W(\mathcal{E}, v)$ für alle $v \in V$.

FIXPUNKTITERATION FÜR EBNF

Da V endlich ist, ist $f(\rho)\colon V \to \mathcal{P}(\Sigma^*)$ nur auf endlich vielen Argumenten definiert, deren Bilder wir nun als Spaltenvektor schreiben:

$$\begin{pmatrix} f(\rho)(v_1) \\ f(\rho)(v_2) \\ \vdots \\ f(\rho)(v_n) \end{pmatrix} \begin{matrix} \in \mathcal{P}(\Sigma^*) \\ \in \mathcal{P}(\Sigma^*) \\ \vdots \\ \in \mathcal{P}(\Sigma^*) \end{matrix}$$

FIXPUNKTITERATION FÜR EBNF

Da V endlich ist, ist $f(\rho)\colon V \to \mathcal{P}(\Sigma^*)$ nur auf endlich vielen Argumenten definiert, deren Bilder wir nun als Spaltenvektor schreiben:

$$\begin{pmatrix} f(\rho)(v_1) \\ f(\rho)(v_2) \\ \vdots \\ f(\rho)(v_n) \end{pmatrix} \begin{matrix} \in \mathcal{P}(\Sigma^*) \\ \in \mathcal{P}(\Sigma^*) \\ \vdots \\ \in \mathcal{P}(\Sigma^*) \end{matrix}$$

Ein Iterationsprozess lässt sich dann wie folgt notieren:

$$\begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \overset{f}{\mapsto}^{1} \begin{pmatrix} f(\rho)(v_1) \\ f(\rho)(v_2) \end{pmatrix} \overset{f}{\mapsto}^{2} \begin{pmatrix} f(f(\rho))(v_1) \\ f(f(\rho))(v_2) \end{pmatrix} \overset{f}{\mapsto}^{3} \dots$$

$$\overset{f}{\mapsto}^{n} \begin{pmatrix} f^n(\rho)(v_1) \\ f^n(\rho)(v_2) \end{pmatrix} \overset{f}{\mapsto}^{n+1} \dots$$

Übungsblatt 3

AUFGABE 1 — TEIL (A)

Gesucht ist eine EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ mit $\Sigma = \{a, b, c, d\}$, sodass

$$W(\mathcal{E}, S) = \left\{ a^k b^\ell c^{2k} c^m : k \geq 1, \ell \geq m \geq 0 \right\}.$$

AUFGABE 1 — TEIL (A)

Gesucht ist eine EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ mit $\Sigma = \{a, b, c, d\}$, sodass

$$W(\mathcal{E}, S) = \left\{ a^k b^\ell c^{2k} c^m : k \geq 1, \ell \geq m \geq 0 \right\}.$$

Wir zerlegen und strukturieren die Sprache

$$\begin{aligned} W(\mathcal{E}, S) &= \left\{ a^k b^\ell c^m c^{2k} : k \geq 1, \ell \geq m \geq 0 \right\} \quad (\ell = m + n, n \geq 0) \\ &= \left\{ a^k b^{m+n} c^m c^{2k} : k \geq 1, m \geq 0, n \geq 0 \right\} \\ &= \left\{ a^k b^m b^n c^m c^{2k} : k \geq 1, m \geq 0, n \geq 0 \right\} \end{aligned}$$

AUFGABE 1 — TEIL (A)

Gesucht ist eine EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ mit $\Sigma = \{a, b, c, d\}$, sodass

$$W(\mathcal{E}, S) = \left\{ a^k b^\ell c^{2k} c^m : k \geq 1, \ell \geq m \geq 0 \right\}.$$

Wir zerlegen und strukturieren die Sprache

$$\begin{aligned} W(\mathcal{E}, S) &= \left\{ a^k b^\ell c^m c^{2k} : k \geq 1, \ell \geq m \geq 0 \right\} \quad (\ell = m + n, n \geq 0) \\ &= \left\{ a^k b^{m+n} c^m c^{2k} : k \geq 1, m \geq 0, n \geq 0 \right\} \\ &= \left\{ a^k b^m b^n c^m c^{2k} : k \geq 1, m \geq 0, n \geq 0 \right\} \end{aligned}$$

Damit ergibt sich

$$\begin{aligned} V &= \{S, A\} \\ R &= \left\{ S ::= a \,\hat{(}\, S \,\hat{|}\, A \,\hat{)} cc, \quad A ::= \hat{(}\, bAc \,\hat{|}\, \hat{\{}\, b \,\hat{\}}\, \hat{)} \right\} \end{aligned}$$

AUFGABE 1 — TEIL (B)

Sei $\Sigma' = \{a, b\}$ und $\mathcal{E}' = (V', \Sigma', X, R')$ eine EBNF-Definition mit $V' = \{X, Y\}$ sowie

$$R' = \left\{\; X ::= \hat{(}\; aXa \;\hat{|}\; Y \;\hat{)}, \quad Y ::= \hat{[}\; bY \;\hat{]} \;\right\}.$$

Wir brauchen die Semantik der EBNF-Terme:

$$\begin{aligned}
\left[\!\left[\hat{(}\; aXa \;\hat{|}\; Y \;\hat{)}\right]\!\right](\rho) &= [\![aXa]\!](\rho) \cup [\![Y]\!](\rho) \\
&= [\![a]\!](\rho) \cdot [\![X]\!](\rho) \cdot [\![a]\!](\rho) \cup [\![Y]\!](\rho) \\
&= \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y) \\
\left[\!\left[\hat{[}\; bY \;\hat{]}\right]\!\right](\rho) &= [\![bY]\!](\rho) \cup \{\varepsilon\} \\
&= [\![b]\!](\rho) \cdot [\![Y]\!](\rho) \cup \{\varepsilon\} \\
&= \{b\} \cdot \rho(Y) \cup \{\varepsilon\}
\end{aligned}$$

$$\left[\!\left[\hat{(}\ aXa\ \hat{|}\ Y\ \hat{)} \right]\!\right](\rho) = \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y)$$
$$\left[\!\left[\hat{[}\ bY\ \hat{]} \right]\!\right](\rho) = \{b\} \cdot \rho(Y) \cup \{\varepsilon\}$$

Die zu iterierende Funktion $f\colon (V \to \mathcal{P}(\Sigma^*)) \to (V \to \mathcal{P}(\Sigma^*))$ ist dann gegeben durch

$$f(\rho) = \begin{pmatrix} f(\rho)(X) \\ f(\rho)(Y) \end{pmatrix} = \begin{pmatrix} \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y) \\ \{b\} \cdot \rho(Y) \cup \{\varepsilon\} \end{pmatrix}$$

4 Iterationen durch Anwendung der Funktion f:

$$\begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \mapsto^1 \begin{pmatrix} \emptyset \\ \{\varepsilon\} \end{pmatrix} \mapsto^2 \begin{pmatrix} \{\varepsilon\} \\ \{\varepsilon, b\} \end{pmatrix} \mapsto^3 \begin{pmatrix} \{aa, \varepsilon, b\} \\ \{\varepsilon, b, bb\} \end{pmatrix}$$
$$\mapsto^4 \begin{pmatrix} \{aaaa, aa, ab, \varepsilon, b\} \\ \{\varepsilon, b, bb, bbb\} \end{pmatrix}$$

AUFGABE 1 — TEIL (C)

$$\begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \mapsto^1 \dots \mapsto^4 \begin{pmatrix} \{aaaa, aa, ab, \varepsilon, b\} \\ \{\varepsilon, b, bb, bbb\} \end{pmatrix}$$

AUFGABE 1 — TEIL (C)

$$\begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \mapsto^1 \ldots \mapsto^4 \begin{pmatrix} \{aaaa, aa, ab, \varepsilon, b\} \\ \{\varepsilon, b, bb, bbb\} \end{pmatrix}$$

Mithilfe der Ergebnisse der Iterationen und der Intuition anhand der Regelmenge können wir vermuten, dass die syntaktische Kategorie von X ist gegeben durch

$$W(\mathcal{E}', X) = \{a^n\ b^m\ a^n : n \geq 0, m \geq 0\}\,.$$

AUFGABE 2

Sei $\mathcal{E} = (V, \Sigma, S, R)$ mit $V = \{S\}$, $\Sigma = \{a, b\}$ und $R = \left\{ S ::= \hat{(}\ aSa\ \hat{|}\ \hat{[}\ b\ \hat{]}\ \hat{)} \right\}$. Außerdem sei $\rho\colon V \to \mathcal{P}(\Sigma^*)$ mit

$$\rho(S) = \{a^n w a^n : n \geq 0, w \in \{\varepsilon, b\}\}\,.$$

zu zeigen: $\left[\!\left[\hat{(}\ aSa\ \hat{|}\ \hat{[}\ b\ \hat{]}\ \hat{)} \right]\!\right](\rho) = \rho(S)$ (d.h. $f(\rho) = \rho$)

AUFGABE 2

Sei $\mathcal{E} = (V, \Sigma, S, R)$ mit $V = \{S\}$, $\Sigma = \{a, b\}$ und $R = \left\{ S ::= \hat{(}\ aSa \ \hat{|}\ \hat{[}\ b\ \hat{]}\ \hat{)} \right\}$. Außerdem sei $\rho\colon V \to \mathcal{P}(\Sigma^*)$ mit

$$\rho(S) = \{a^n w a^n : n \geq 0, w \in \{\varepsilon, b\}\}.$$

zu zeigen: $\left[\!\left[\hat{(}\ aSa \ \hat{|}\ \hat{[}\ b\ \hat{]}\ \hat{)} \right]\!\right](\rho) = \rho(S)$ (d.h. $f(\rho) = \rho$)

$$\begin{aligned}
&\left[\!\left[\hat{(}\ aSa \ \hat{|}\ \hat{[}\ b\ \hat{]}\ \hat{)} \right]\!\right](\rho) \\
=\ & \{a\}\,\rho(S)\,\{a\} \cup (\{b\} \cup \{\varepsilon\}) \\
=\ & \{a\}\,\{a^n w a^n : n \geq 0, w \in \{\varepsilon, b\}\}\,\{a\} \cup \{\varepsilon, b\} \\
=\ & \left\{a^{n+1} w a^{n+1} : n \geq 0, w \in \{\varepsilon, b\}\right\} \cup \{\varepsilon, b\} \\
=\ & \{a^n w a^n : n \geq 1, w \in \{\varepsilon, b\}\} \cup \{\varepsilon, b\} \\
=\ & \{a^n w a^n : n \geq 1, w \in \{\varepsilon, b\}\} \cup \{a^n w a^n : n = 0, w \in \{\varepsilon, b\}\} \\
=\ & \{a^n w a^n : n \geq 0, w \in \{\varepsilon, b\}\} \\
=\ & \rho(S)
\end{aligned}$$