

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 3: EXTENDED BACKUS-NAUR-FORM

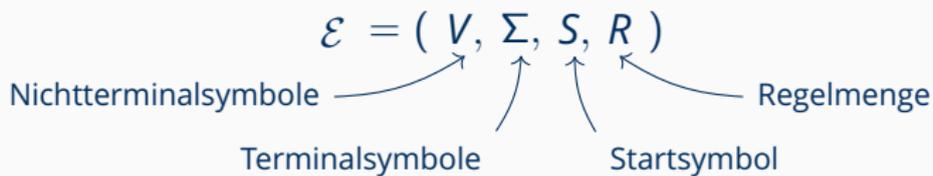
Eric Kunze

`eric.kunze@tu-dresden.de`

TU Dresden, 27. Oktober 2021

EBNF und ihre Semantik

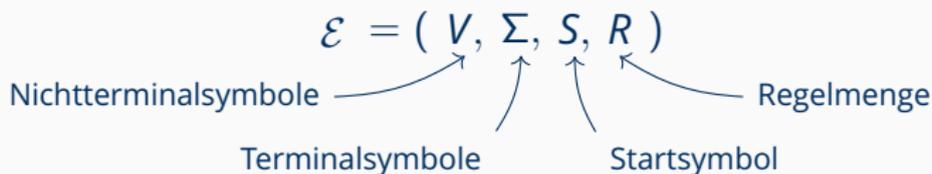
EBNF-DEFINITION



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

Nichtterminalsymbol ::= EBNF-Term

EBNF-DEFINITION



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

Nichtterminalsymbol ::= EBNF-Term

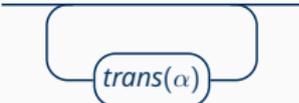
Definition (EBNF-Terme): Seien V (syntaktische Variablen) und Σ (Terminalsymbole) endliche Mengen mit $V \cap \Sigma = \emptyset$. Die Menge der EBNF-Terme über V und Σ (notiere: $T(\Sigma, V)$), ist die *kleinste* Menge $T \subseteq (V \cup \Sigma \cup \{\hat{\{ \}}, \hat{\} \}, \hat{[\]}, \hat{] \},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

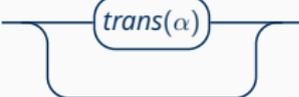
- ▶ Wenn $\alpha \in T$, so auch $\hat{(\alpha)} \in T, \hat{\{\alpha\}} \in T, \hat{[\alpha]} \in T$.
- ▶ Wenn $\alpha_1, \alpha_2 \in T$, so auch $\hat{(\alpha_1 \mid \alpha_2)} \in T, \alpha_1 \alpha_2 \in T$.

ÜBERSETZUNG EBNF \leftrightarrow SYNTAXDIAGRAMME

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

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

► $trans(\hat{\{\alpha\}}) =$  A horizontal line with a rounded rectangle labeled $trans(\alpha)$ connected to it by a U-shaped line above and below.

► $trans(\hat{[\alpha]}) =$  A horizontal line with a rounded rectangle labeled $trans(\alpha)$ connected to it by a U-shaped line below and a hook-like line above.

► $trans(\hat{(\alpha)}) = trans(\alpha)$

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

► $trans(\alpha_1\alpha_2) =$  A horizontal line with two rounded rectangles labeled $trans(\alpha_1)$ and $trans(\alpha_2)$ connected in sequence.

► $trans(\hat{(\alpha_1 \mid \alpha_2)}) =$  A horizontal line with two rounded rectangles labeled $trans(\alpha_1)$ and $trans(\alpha_2)$ connected in parallel by a U-shaped line below and a hook-like line above.

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
- ▶ $\rho: V \rightarrow \mathcal{P}(\Sigma^*)$ ordnet jeder syntaktischen Variable $v \in V$ eine Sprache zu
- ▶ Vorstellung: $\rho(v)$ ist bestes Wissen über die von v beschriebene Sprache

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
- ▶ $\rho: V \rightarrow \mathcal{P}(\Sigma^*)$ 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 $\llbracket \cdot \rrbracket : \underbrace{T(\Sigma, V)}_{\text{EBNF-Term } \alpha} \rightarrow \underbrace{((V \rightarrow \mathcal{P}(\Sigma^*)))}_{\rho} \rightarrow \mathcal{P}(\Sigma^*)$

SEMANTIK VON EBNF-TERMEN

$$\llbracket \cdot \rrbracket : \underbrace{T(\Sigma, V)}_{\text{EBNF-Term } \alpha} \rightarrow \underbrace{((V \rightarrow \mathcal{P}(\Sigma^*)) \rightarrow \mathcal{P}(\Sigma^*))}_{\rho}$$

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

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

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

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

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

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

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 $g: \mathbb{R} \rightarrow \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\bar{x} \in \mathbb{R}$ mit $g(\bar{x}) = 0$.

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 \bar{x} an

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

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

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 \bar{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 $g: \mathbb{R} \rightarrow \mathbb{R}$, von der wir eine Nullstelle suchen, d.h. ein $\bar{x} \in \mathbb{R}$ mit $g(\bar{x}) = 0$.

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 \bar{x} an

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

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

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

FIXPUNKTITERATION – EINE ANALOGIE

Ausblick: Fixpunktiteration zur Nullstellenbestimmung

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

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:

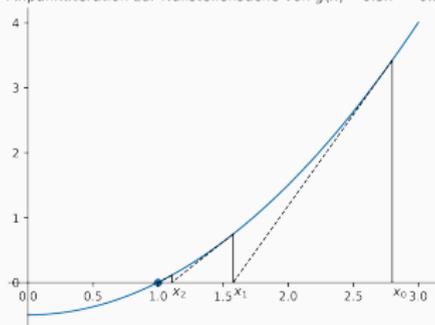
x_i nähert sich der Nullstelle \bar{x} an

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

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

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

Fixpunktiteration zur Nullstellensuche von $g(x) = 0.5x^2 - 0.5$



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:

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

- ▶ Starte mit bisherigem 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:

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

- ▶ Starte mit bisherigem 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 $f(\rho): V \rightarrow \mathcal{P}(\Sigma^*)$ 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 $f(\rho): V \rightarrow \mathcal{P}(\Sigma^*)$ 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:

$$\begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \xrightarrow{f_1} \begin{pmatrix} f(\rho)(v_1) \\ f(\rho)(v_2) \end{pmatrix} \xrightarrow{f_2} \begin{pmatrix} f(f(\rho))(v_1) \\ f(f(\rho))(v_2) \end{pmatrix} \xrightarrow{f_3} \dots \\ \xrightarrow{f_n} \begin{pmatrix} f^n(\rho)(v_1) \\ f^n(\rho)(v_2) \end{pmatrix} \xrightarrow{f_{n+1}} \dots$$

Übungsblatt 3

AUFGABE 1 — TEIL (A)

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

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

AUFGABE 1 — TEIL (A)

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

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

Wir zerlegen und strukturieren die Sprache

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

AUFGABE 1 — TEIL (A)

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

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

Wir zerlegen und strukturieren die Sprache

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

Damit ergibt sich

$$V = \{S, A\}$$

$$R = \left\{ S ::= a \hat{ } (S \hat{ } A) c c, \quad A ::= \hat{ } (b A c \hat{ } \{ \hat{ } b \hat{ } \}) \right\}$$

AUFGABE 1 — TEIL (B)

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

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

AUFGABE 1 — TEIL (B)

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

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

Wir brauchen die Semantik der EBNF-Terme:

$$\begin{aligned} \llbracket \hat{(aXa \hat{Y})} \rrbracket (\rho) &= \llbracket aXa \rrbracket (\rho) \cup \llbracket Y \rrbracket (\rho) \\ &= \llbracket a \rrbracket (\rho) \cdot \llbracket X \rrbracket (\rho) \cdot \llbracket a \rrbracket (\rho) \cup \llbracket Y \rrbracket (\rho) \\ &= \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y) \end{aligned}$$

$$\begin{aligned} \llbracket \hat{bY} \hat{ } \rrbracket (\rho) &= \llbracket bY \rrbracket (\rho) \cup \{\varepsilon\} \\ &= \llbracket b \rrbracket (\rho) \cdot \llbracket Y \rrbracket (\rho) \cup \{\varepsilon\} \\ &= \{b\} \cdot \rho(Y) \cup \{\varepsilon\} \end{aligned}$$

$$\llbracket \hat{a} X a \hat{Y} \rrbracket (\rho) = \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y)$$

$$\llbracket \hat{b} Y \rrbracket (\rho) = \{b\} \cdot \rho(Y) \cup \{\varepsilon\}$$

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

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

$$\llbracket \hat{a} X a \hat{a} \uparrow Y \hat{a} \rrbracket (\rho) = \{a\} \cdot \rho(X) \cdot \{a\} \cup \rho(Y)$$

$$\llbracket \hat{b} Y \hat{b} \rrbracket (\rho) = \{b\} \cdot \rho(Y) \cup \{\varepsilon\}$$

Die zu iterierende Funktion $f: (V \rightarrow \mathcal{P}(\Sigma^*)) \rightarrow (V \rightarrow \mathcal{P}(\Sigma^*))$ 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 :

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

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)

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

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 $R = \{S ::= \hat{(} aSa \hat{)} \hat{[} \hat{b} \hat{]}\hat{)}$. Außerdem sei $\rho: V \rightarrow \mathcal{P}(\Sigma^*)$ mit

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

zu zeigen: $\llbracket \hat{(} aSa \hat{)} \hat{[} \hat{b} \hat{]}\hat{)} \rrbracket (\rho) = \rho(S)$ (d.h. $f(\rho) = \rho$)

AUFGABE 2

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

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

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

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