Fixpunktsemantik am Beispiel

Einleitung:

Die Fixpunktsemantik wird benutzt um Schritt für Schritt die, durch eine EBNF-Definition gegebene, Sprache zu entwickeln. Dazu beginnt man mit der leeren Menge und fängt an, einzelne Durchläufe durch die Regeln der syntaktischen Variablen zu simulieren.
Man beschreibt jeweils die Sprache, die durch eine bestimmte syntaktische Regel beim aktuellen Durchlauf erzeugt werden kann. Kommt in dieser Regel eine syntaktische Variable vor, so "füllt"man diese mit der Sprache des vorherigen Schrittes. Dieses Schema führt man solange aus, bis man die Sprache erraten kann und sooft die Aufgabenstellung es verlangt.
Soviel zur trockenen Theorie ...

AGS 2.38

Sei $\mathcal{E} = (V, \Sigma, S, R)$ mit $V = \{S, A\}$, $\Sigma = \{a, b\}$ und $R = \{S ::= [aaA]., A ::= (Sbb|bb).\}$.
Berechnen Sie die syntaktischen Kategorien $W(\mathcal{E}, S)$ und $W(\mathcal{E}, A)$ mit Hilfe der Fixpunktsemantik.

Schritt 1: Aller Anfang ist leicht ... - Die Aufgabe

$$\begin{pmatrix} W(\mathcal{E}, S) \\ W(\mathcal{E}, A) \end{pmatrix}$$

Schritt 2: Am Anfang war das Nichts ... - Der Induktionsanfang

Als Anfang für die Induktion benutzt man das kleinstmögliche (bzw. -nötige) Element.
In diesem Fall also die leere Menge.

$$\begin{pmatrix} W(\mathcal{E}, S) \\ W(\mathcal{E}, A) \end{pmatrix} \rightsquigarrow \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix}$$

Schritt 3: Immer einen Fuß vor den anderen ... - Der Induktionsschritt

Als nächstes wendet man die Funktion f auf das Tupel der Sprachen an.
Da in den Regeln der gesuchten Sprachen syntaktische Variablen auftauchen, nehmen wir die jeweiligen Sprachen aus dem vorherigen Schritt zu Hilfe.

$$\begin{pmatrix} W(\mathcal{E}, S) \\ W(\mathcal{E}, A) \end{pmatrix} \rightsquigarrow \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \quad \not\searrow \quad \begin{pmatrix} \quad \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \quad \end{pmatrix} \overset{f}{\mapsto} \dots$$

Zunächst berechnen wir also $\mathbf{W(\mathcal{E},S)}$:

Laut EBNF-Defintion ergibt sich W($\mathcal{E}$,S) aus Anwendung der Regel $S ::= [aaA]$. Also entweder erzeugt man Wörter der Form “aaA“ oder man lässt es weg... Lässt man es weg, so erhält man einfach das leere Wort ϵ.
Um die Wörter der Form “aaA“ zu erzeugen, benötigt man die Wörter, die man durch A bekommt. Wir nehmen dazu $W(\mathcal{E}, A)$ aus dem **vorherigen** Schritt zu Hilfe, also die leere Menge $\emptyset$. Unsere gesuchte Menge setzt sich also aus $\{\epsilon\}$ und allen Wörtern der Form “$aa\cdot\emptyset$“ zusammen. Da ein Komplexprodukt mit einer leeren Menge stets die leere Menge zurückgibt, bekommt man bei diesem Schritt keine neuen Wörter zu der Sprache.
Die Sprache $W(\mathcal{E}, S)$ ergibt sich somit aus $\{\epsilon\} \cup \emptyset = \{\epsilon\}$.

Als Nächstes berechnen wir $\mathbf{W(\mathcal{E},A)}$:

Laut EBNF-Defintion ergibt sich W($\mathcal{E}$,A) aus Anwendung der Regel $A ::= (Sbb|bb)$. Also entweder erzeugt man Wörter der Form "Sbb" oder das Wort "bb"... Teil der gesuchten Sprache ist also auf jeden Fall $\{bb\}$.
Um die Wörter der Form "Sbb" zu erzeugen, benötigt man die Wörter, die man durch S bekommt. Wir nehmen dazu $W(\mathcal{E}, S)$ aus dem **vorherigen** Schritt zu Hilfe, also die leere Menge Ø. Unsere gesuchte Menge setzt sich also aus $\{bb\}$ und allen Wörtern der Form "Ø$\cdot bb$" zusammen. Da ein Komplexprodukt mit einer leeren Menge stets die leere Menge zurückgibt, bekommt man bei diesem Schritt keine neuen Wörter zu der Sprache.
Die Sprache $W(\mathcal{E}, A)$ ergibt sich somit aus $\{bb\} \cup \emptyset = \{bb\}$.
Wir erhalten deshalb:

$$\begin{pmatrix} W(\mathcal{E},S) \\ W(\mathcal{E},A) \end{pmatrix} \rightsquigarrow \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon\} \\ \{bb\} \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \\ \end{pmatrix} \overset{f}{\mapsto} \dots$$

Analog geht es mit dem zweiten Iterationsschritt weiter:

$W(\mathcal{E}, S)$ ergibt sich aus dem leeren Wort und allen Wörtern der Form "aaA". Wir nehmen wiederum $W(\mathcal{E}, A)$ des **vorherigen** Schrittes zu Hilfe, in diesem Fall also $\{bb\}$.
Die Menge der Wörter "aa$\cdot\{bb\}$" ist die Menge $\{aabb\}$.
Die neue Menge ergibt sich also aus $\{\epsilon\} \cup \{aabb\} = \{\epsilon, aabb\}$.

$W(\mathcal{E}, A)$ ergibt sich aus dem Wort "bb" und allen Wörtern der Form "Sbb". Wir nehmen wiederum $W(\mathcal{E}, S)$ des **vorherigen** Schrittes zu Hilfe, in diesem Fall also $\{\epsilon\}$.
Die Menge der Wörter "$\{\epsilon\} \cdot bb$" ist die Menge $\{bb\}$.
Die neue Menge ergibt sich also aus $\{bb\} \cup \{bb\} = \{bb\}$.

$$\begin{pmatrix} W(\mathcal{E},S) \\ W(\mathcal{E},A) \end{pmatrix} \rightsquigarrow \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon\} \\ \{bb\} \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon, aabb\} \\ \{bb\} \end{pmatrix} \overset{f}{\mapsto} \dots$$

Nach dem selben Schema erzeugt man nun die restlichen Iterationsschritte.

$$\begin{pmatrix} W(\mathcal{E},S) \\ W(\mathcal{E},A) \end{pmatrix} \rightsquigarrow \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon\} \\ \{bb\} \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon\} \cup \{aabb\} \\ \{bb\} \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon, aabb\} \\ \{bb\} \cup \{aabbbb\} \end{pmatrix} \overset{f}{\mapsto}$$

$$\begin{pmatrix} \{\epsilon, aabb\} \cup \{aaaabbbb\} \\ \{bb, aabbbb\} \end{pmatrix} \overset{f}{\mapsto} \begin{pmatrix} \{\epsilon, aabb, aaaabbbb\} \\ \{bb, aabbbb\} \cup \{aaaabbbbbb\} \end{pmatrix}$$

Schritt 4: So long, and thanks for all the fish ... - Die allgemeine Form $f^i(\perp)$

Als Nächstes will man nun eine allgemeine Formel für den i-ten Schritt der Funktion herausbekommen.

W($\mathcal{E}$,S):

Als Erstes erkennt man die Struktur der Wörter. Diese haben die Form $\{a^k b^k \dots\}$. Desweiteren fällt auf, dass die Potenzen nur in geradzahliger Form auftreten $\Longrightarrow \{a^{2k} b^{2k} \dots\}$.
Nun muss man nur noch Grenzwerte für k angeben. Da ϵ in der Sprache W($\mathcal{E}$,S) enthalten ist, muss k auch den Wert 0 annehmen können.
Bei jedem zweiten Schritt der Iteration nimmt das maximale k um eins zu ($f_2 \rightarrow k = 1$, $f_4 \rightarrow k = 2, \dots$).
Sieht man nun nicht sofort den Wert der Potenzen im Zusammenhang mit i, schafft eine kleine Tabelle meist Abhilfe:

i	1	2	3	4	5
k_{max} in S	0	1	1	2	2
k_{max} in A	0	0	1	1	2

Man sieht, dass die Potenzen immer im "Zweierpack" auftreten. Das maximale k lässt sich also mit Hilfe von $\frac{i}{2}$ darstellen. Bei geradzahligem i hat man damit seine Aufgabe bereits erfüllt.
Nun wendet man sich den ungeraden "i"s zu. Am Einfachsten ist es hierbei ein Beispiel zu bemühen. Wir geben i jetzt also den Wert 3 und bekommen damit für $k_{max} = 1,5$. Da das korrekte k_{max} allerdings laut Tabelle immernoch 1 ist, weiß man nun, dass man den Wert bei ungeradem i noch abrunden muss. Leicht kommt man zu der Erkenntnis, dass dies auch bei allen anderen ungeraden "i"s der Fall ist.
Beim i-ten Schritt nimmt k demnach einen Wert zwischen 0 und $\lfloor \frac{i}{2} \rfloor$ an.
$\Rightarrow \{a^{2k} b^{2k} | 0 \leq k \leq \lfloor \frac{i}{2} \rfloor\}$

W($\mathcal{E}$,A):
Die Potenzen der Wörter sind zwar immernoch geradzahlig, aber nun nichtmehr gleich.
Es gibt immer zwei "b"s mehr als "a"s $\Rightarrow \{a^{2k} b^{2k+2} \dots\}$
Auch die Grenzwerte der Potenzen beim i-ten Iterationsschritt ändern sich nur geringfügig: Die "Zweierpacks" sind nur um einen Schritt nach rechts verschoben.

$\Rightarrow \{a^{2k} b^{2k+2} | 0 \leq k \leq \lfloor \frac{\mathbf{i-1}}{2} \rfloor\}$

$$f^i \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix} = \begin{pmatrix} \{a^{2k} b^{2k} | 0 \leq k \leq \lfloor \frac{i}{2} \rfloor\} \\ \{a^{2k} b^{2k+2} | 0 \leq k \leq \lfloor \frac{i-1}{2} \rfloor\} \end{pmatrix}, i \geq 1$$

Schritt 5: Der Lohn der Arbeit... - Die Mengenschreibweise der Sprachen

Da die Sprache keine Begrenzung bezüglich der Länge der Wörter hat, i also jeden Wert annehmen kann, gibt es keine Begrenzung der Sprachen nach oben.

$$\lfloor \tfrac{\infty}{2} \rfloor = \infty \qquad \lfloor \tfrac{\infty - 1}{2} \rfloor = \infty$$

$W(\mathcal{E}, S) = \{a^{2n}b^{2n} | n \geq 0\}$ und
$W(\mathcal{E}, A) = \{a^{2n}b^{2n+2} | n \geq 0\}$.