Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Compilerbau I
SS 2020

Übungsblatt 4

Aufgabe 1 Gegeben seien folgende Scannerregeln über dem Alphabet {a,b}:

$$\begin{array}{rl} a & \{\text{action}_1\} \\ a^*b & \{\text{action}_2\} \end{array}$$

Bestimmen Sie den Scanner (Folien 51 und 52).

Lösung:
Sei $\Sigma = \{a, b\}$, $e_1 = a$ und $e_2 = a^*b$. Zunächst müssen wir den Berry-Sethi-Automat zu $e := e_1 \mid e_2$ konstruieren. Wir erhalten

$$e' := \text{num}(e) = [1, a] \mid [2, a]^*[3, b].$$

Für empty, first, last und next ergibt sich:

- $\text{empty}(e') = \text{empty}([1, a]) \vee \text{empty}([2, a]^*[3, b]) = f$
  - $\text{empty}([1, a]) = f$
  - $\text{empty}([2, a]^*[3, b]) = \text{empty}([2, a]^*) \wedge \text{empty}([3, b]) = f$
    - $\text{empty}([2, a]^*) = t$
      - $\text{empty}([2, a]) = f$
    - $\text{empty}([3, b]) = f$
- $\text{first}(e') = \text{first}([1, a]) \cup \text{first}([2, a]^*[3, b]) = \{[1, a], [2, a], [3, b]\}$
  - $\text{first}([1, a]) = \{[1, a]\}$
  - $\text{first}([2, a]^*[3, b]) = \text{first}([2, a]^*) \cup \text{first}([3, b]) = \{[2, a], [3, b]\}$ [1]
    - $\text{first}([2, a]^*) = \text{first}([2, a]) = \{[2, a]\}$
      - $\text{first}([2, a]) = \{[2, a]\}$
    - $\text{first}([3, b]) = \{[3, b]\}$

[1] weil $\text{empty}([2, a]^*) = t$

$\text{last}(e') = \text{last}([1,a]) \cup \text{last}([2,a]^*[3,b]) = \{[1,a],[3,b]\}$

- $\text{last}([1,a]) = \{[1,a]\}$
- $\text{last}([2,a]^*[3,b]) = \text{last}([3,b]) = \{[3,b]\}$, weil $\text{empty}([3,b]) = f$
  - $\text{last}([2,a]^*) = \text{last}([2,a]) = \{[2,a]\}$
    - $\text{last}([2,a]) = \{[2,a]\}$
  - $\text{last}([3,b]) = \{[3,b]\}$

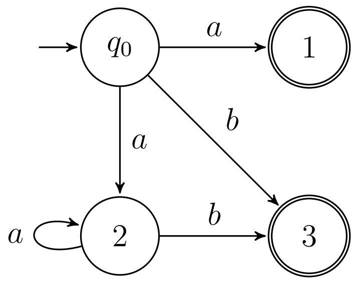
$\text{next}(e') = \emptyset$

- $\text{next}([1,a]) = \text{next}(e') = \emptyset$
- $\text{next}([2,a]^*[3,b]) = \text{next}(e') = \emptyset$
  - $\text{next}([2,a]^*) = \text{first}([3,b]) = \{[3,b]\}$, weil $\text{empty}([3,b]) = f$
    - $\text{next}([2,a]) = \text{first}([2,a]) \cup \text{next}([2,a]^*) = \{[2,a],[3,b]\}$
  - $\text{next}([3,b]) = \text{next}([2,a]^*[3,b]) = \emptyset$

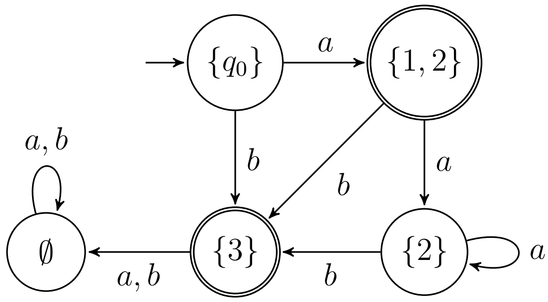
Der Berry-Sethi-Automat ist dann $A = (Q, \Sigma, \Delta, I, F)$ mit

$$
\begin{aligned}
Q &= \{q_0\} \cup \{1, \ldots, \ell(e')\} = \{q_0, 1, 2, 3\},\\
\Sigma &= \{a, b\},\\
\Delta &= \{(q_0, x, i) \mid [i, x] \in \text{first}(e')\}\\
&\quad \cup \{(i, x, j) \mid \exists y.[j, x] \in \text{next}([i, y])\}\\
&\quad = \{(q_0, a, 1), (q_0, a, 2), (q_0, b, 3), (2, a, 2), (2, b, 3)\}\\
I &= \{q_0\},\\
F &= \{i \mid \exists x.[i, x] \in \text{last}(e')\}\\
&\quad \cup \{q_0 \mid \text{empty}(e') = t\}\\
&\quad = \{1, 3\}.
\end{aligned}
$$

Grafisch:



Der erreichbare Teil des Potenzmengenautomat $\mathcal{P}(A)$ ist dann



Wir bezeichnen obigen Automat (also den erreichbaren Teil von $\mathcal{P}(A)$) mit $(Q', \Sigma, \delta, I', F')$. D.h. der erreichbare Teil der Endzustände von $\mathcal{P}(A)$ ist $F' = \{\{1,2\}, \{3\}\}$. Da wir zunächst die Alphabetsymbole in last stehen lassen, ergibt sich eine leicht andere Definition der Endzustandsmengen F_i. Zum Löschen der Alphabetsymbole sei $f\colon \Sigma_n \to \mathbb{N}$ definiert als $f([j,a]) = j$ und für eine Menge $M \subseteq \Sigma_n$ schreiben wir $f(M) = \{f(z) \mid z \in M\}$. Dann ist F_i (für $1 \leq i \leq n$ bei n regulären Ausdrücken $e_1, \ldots, e_n$) definiert als

$$F_i = \{q \in (F' \setminus (F_1 \cup \cdots \cup F_{i-1})) \mid q \cap \mathrm{last}(f(e_i)) \neq \emptyset\}.$$

Für e_1 und e_2 ergibt sich

$$\begin{aligned} F_1 &= \{q \in \{\{1,2\},\{3\}\} \mid q \cap \{1\} \neq \emptyset\} = \{\{1,2\}\}, \\ F_2 &= \{q \in \{\{3\}\} \mid q \cap \{3\} \neq \emptyset\} = \{\{3\}\}. \end{aligned}$$

Aufgabe 2 Verwenden Sie den Scanner aus Aufgabe 1. Führen Sie jeweils den Tokenizing-Algorithmus (Folie 54), gefolgt von Reps' Maximal-Munch-Algorithmus (Folie 57) für die folgenden Wörter durch:

Lösung:
Zu einem Wort $w \in \Sigma^*$ sei $P_w = \{1, \ldots, |w|+1\}$ die Menge seiner Positionen. Ein Zustand des Tokenizing-Algorithmus in w besteht aus einem Paar $(q,k) \in Q' \times P_w$ und einem Element $\ell \in (Q' \times P_w) \cup \{\perp\}$. Hier ist q der aktuelle Zustand und k die Position in w. Die Position k bedeutet, dass man *vor* dem k-ten Symbol steht, wobei $k = |w| + 1$ bedeutet, dass man am Ende des Worts angelangt ist. Der Wert ℓ ist die Position und der Wert des zuletzt gesehenen Endzustands. Wenn $\ell = \perp$, so wurde noch kein Endzustand gesehen. Wenn $\ell = (p, j)$, dann wurde zuletzt der Endzustand p vor dem j-ten Symbol gesehen. Außerdem merken wir uns mit $s \in P_w$, von wo wir das nächste Mal ein Token abschneiden müssen, was zu Anfang $s = 1$ ist. Wenn wir zu Reps' Algorithmus übergehen, muss der Algorithmus noch zusätzlich die Menge fv $\subseteq Q' \times P_w$ verwalten, die sagt, welche Konfigurationen nicht mehr zu einem Endzustand führen.

(a) $w_1 = aababa$

Lösung:
Wir notieren Zustände wie folgt: Der Startzustand ist $|\{q_0\}aababa$. Dies bedeutet, dass $k = 1$, $q = \{q_0\}$ und $\ell = \bot$ (was wir nicht explizit repräsentieren). Das Symbol $|$ wird benutzt, um s zu markieren, d.h. $s = 1$. Der erste Übergang ist $\delta(\{q_0\}, a) = \{1, 2\}$, also wäre der nächste Zustand $|a\{1, 2\}aababa$. Dass $\{1, 2\}$ der zuletzt gesehene Endzustand ist, müssen wir allerdings auch noch markieren, was wir tun, indem wir $|a\underline{\{1, 2\}}\{1, 2\}aababa$ schreiben, d.h. $\ell = (\{1, 2\}, 2)$. Es geht weiter mit

$$|a\underline{\{1,2\}}a\{2\}ababa,\ |a\underline{\{1,2\}}aa\{2\}baba,$$
$$|aaab\underline{\{3\}}\{3\}aba \text{ und } |aaab\underline{\{3\}}a\emptyset ba.$$

Da wir $\emptyset$ erreicht haben, muss nun das Token zwischen $|$ und $\underline{\{3\}}$ abgeschnitten werden. Wir rufen also write($aaab$) und action_2 auf, weil $\{3\} \in F_2$. Der nächste Zustand ist $aaab|\{q_0\}aba$. Es geht weiter mit

$$aaab|a\underline{\{1,2\}}\{1,2\}ba,\ aaab|ab\underline{\{3\}}\{3\}a \text{ und } aaab|ab\underline{\{3\}}a\emptyset.$$

Wir rufen write(ab) und action_2 auf und machen weiter mit $aaabab|\{q_0\}a$, gefolgt von $aaabab|a\underline{\{1, 2\}}\{1, 2\}$. Weil hier das Wortende erreicht wurde und wir einen Endzustand gesehen haben, rufen wir write(a) und action_1 auf, da $\{1, 2\} \in F_1$. Da s nun am Ende ist, stoppt der Algorithmus.

Reps' Algorithmus muss zusätzlich die Menge fv verwalten. Wenn wir $\emptyset$ oder das Wortende erreichen, müssen alle Konfigurationen in fv aufgenommen werden, die wir seit dem letzten Endzustand gesehen haben. Wir starten zunächst wieder in $|\{q_0\}aababa$ und erreichen nach ein paar Schritten $|aaab\underline{\{3\}}a\emptyset ba$. Die zuletzt gesehene Konfiguration, $|aaab\underline{\{3\}}\{3\}aba$, ist in einem Endzustand, also wird nichts in fv eingetragen. Ebenso wird in $aaab|ab\underline{\{3\}}a\emptyset$ und in $aaabab|a\underline{\{1, 2\}}\{1, 2\}$ nichts in fv eintragen.

(b) $w_2 = aaaaa$

Lösung:
Wir beginnen in $|\{q_0\}aaaaa$. Die folgenden Zustände sind

$$|a\underline{\{1,2\}}\{1,2\}aaaa,\ |a\underline{\{1,2\}}a\{2\}aaa,\ |a\underline{\{1,2\}}aa\{2\}aa,$$
$$|a\underline{\{1,2\}}aaa\{2\}a \text{ und } |a\underline{\{1,2\}}aaaa\emptyset.$$

Es wird also write(a) mit action_1 aufgerufen. Danach geht es weiter in $a|\{q_0\}aaaa$, gefolgt von

$a|a\underline{\{1,2\}}\{1,2\}aaa$, $a|a\underline{\{1,2\}}a\{2\}aa$, $a|a\underline{\{1,2\}}aa\{2\}a$ und $a|a\underline{\{1,2\}}aaa\emptyset$.

Es wird wieder write(a) mit action_1 aufgerufen. Danach geht es weiter mit $aa|\{q_0\}aaa$, $aa|a\underline{\{1,2\}}\{1,2\}aa$, $aa|a\underline{\{1,2\}}a\{2\}a$ und $aa|a\underline{\{1,2\}}aa\emptyset$. Es wird wieder write($a$) mit action_1 aufgerufen. Die nächste Iteration ist $aaa|\{q_0\}aa$, $aaa|a\underline{\{1,2\}}\{1,2\}a$ und $aaa|a\underline{\{1,2\}}a\{2\}$. Da das Wortende erreicht wurde, rufen wir write(a) und action_1 auf. Zuletzt werden noch $aaaa|\{q_0\}a$ und $aaaa|a\underline{\{1,2\}}\{1,2\}$ durchlaufen, wonach wir wieder write(a) und action_1 aufrufen.

Mit Reps' Algorithmus kürzen wir einige Schritte ab. Beim ersten Erreichen von $\emptyset$ sind wir in $|a\underline{\{1,2\}}aaaa\emptyset$. Alle seit dem letzten Endzustand gesehenen Konfigurationen werden nun in fv eingetragen, also

$$|a\underline{\{1,2\}}a\{2\}aaa,\ |a\underline{\{1,2\}}aa\{2\}aa \text{ und } |a\underline{\{1,2\}}aaa\{2\}a.$$

Dabei spielen der Wert von s, also der Marker $|$, und die zuletzt gesehenen Endzustände keine Rolle und werden weggelassen. Wir haben also

$$\text{fv} = \{(\{2\},3),(\{2\},4),(\{2\},5)\}.$$

Im nächsten Durchlauf starten wir wieder in $a|\{q_0\}aaaa$. Nach zwei Schritten erreichen wir $a|a\underline{\{1,2\}}a\{2\}aa$. Da $(\{2\},4)$ in fv eingetragen wurde, brechen wir hier ab und es geht mit $aa|\{q_0\}aaa$ weiter. Nach zwei Schritten erreichen wir $aa|a\underline{\{1,2\}}a\{2\}a$. Da $(\{2\},5)$ in fv eingetragen wurde, brechen wir auch hier ab. Wir starten nun in $aaa|\{q_0\}aa$. Nach zwei Schritten erreichen wir $aaa|a\underline{\{1,2\}}a\emptyset$. Zuletzt starten wir in $aaaa|\{q_0\}a$ und erreichen $aaaa|a\underline{\{1,2\}}\{1,2\}$.