Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Beispiel: Match

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei $\Sigma = \{a, b\}, w = bbaaabbb \in \Sigma^*$.

$$\underset{0}{|}\ \text{b}\ \underset{1}{|}\ \text{b}\ \underset{2}{|}\ \text{a}\ \underset{3}{|}\ \text{a}\ \underset{4}{|}\ \text{a}\ \underset{5}{|}\ \text{b}\ \underset{6}{|}\ \text{b}\ \underset{7}{|}\ \text{b}\ \underset{8}{|}$$

Beispiel: Match

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei $\Sigma = \{a, b\}, w = bbaaabbb \in \Sigma^*$.

$$\underset{0}{|}\;b\;\underset{1}{|}\;b\;\underset{2}{|}\;a\;\underset{3}{|}\;a\;\underset{4}{|}\;a\;\underset{5}{|}\;b\;\underset{6}{|}\;b\;\underset{7}{|}\;b\;\underset{8}{|}$$

Bspw. ist $\{(0,0),(0,8),(3,7)\} \in \mathcal{M}$ eine Menge von Matches mit

$(0,0) \approx ||bbaaabbb \approx \varepsilon$
$(0,8) \approx |bbaaabbb| \approx w$
$(3,7) \approx bba|aabb|b \approx aabb$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun $\mathit{Parser} := \mathcal{M}^{\mathcal{M}}$. Für $f \in \mathit{Parser}$ ist also $f : \mathcal{M} \to \mathcal{M}$.

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun *Parser* $:= \mathcal{M}^{\mathcal{M}}$. Für $f \in$ *Parser* ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \ M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\}$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun *Parser* $:= \mathcal{M}^{\mathcal{M}}$. Für $f \in$ *Parser* ist also $f : \mathcal{M} \to \mathcal{M}$.

$$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$$
$$\texttt{readChar}_w(\sigma) = \underbrace{M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\}}_{\in \mathit{Parser}}$$

Beispiel: `readChar`

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\}$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$.

$$\underset{0}{|}\ \mathsf{b}\ \underset{1}{|}\ \mathsf{b}\ \underset{2}{|}\ \mathsf{a}\ \underset{3}{|}\ \mathsf{a}\ \underset{4}{|}\ \mathsf{a}\ \underset{5}{|}\ \mathsf{b}\ \underset{6}{|}\ \mathsf{b}\ \underset{7}{|}\ \mathsf{b}\ \underset{8}{|}$$

Beispiel: `readChar`

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\}$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_a(\{(0,0),(2,2),(2,4)\}) = \{(2,3),(2,5)\}$$

Beispiel: `readChar`

$\texttt{readChar}_w : \Sigma \to \textit{Parser}$
$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\}$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_a(\{(0,0),(2,2),(2,4)\}) = \{(2,3),(2,5)\}$$

$$\approx f_a(\{||bbaaabbb, bb||aaabbb, bb|aa|abbb\})$$

$$= \{bb|a|aabbb, bb|aaa|bbb\}$$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun $\mathit{Parser} := \mathcal{M}^{\mathcal{M}}$. Für $f \in \mathit{Parser}$ ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun *Parser* $:= \mathcal{M}^{\mathcal{M}}$. Für $f \in$ *Parser* ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i, j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

Beispiel: `concatenate`

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{ab} := \texttt{concatenate}_w(f_a, f_b)$.

$$\underset{0}{|}\ \mathsf{b}\ \underset{1}{|}\ \mathsf{b}\ \underset{2}{|}\ \mathsf{a}\ \underset{3}{|}\ \mathsf{a}\ \underset{4}{|}\ \mathsf{a}\ \underset{5}{|}\ \mathsf{b}\ \underset{6}{|}\ \mathsf{b}\ \underset{7}{|}\ \mathsf{b}\ \underset{8}{|}$$

Beispiel: `concatenate`

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{ab} := \texttt{concatenate}_w(f_a, f_b)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_{ab}(\{(0,0),(2,2),(2,4)\}) = f_b(f_a(\{(0,0),(2,2),(2,4)\}))$$
$$= f_b(\{(2,3),(2,5)\}) = \{(2,6)\}$$

Beispiel: `concatenate`

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{ab} := \texttt{concatenate}_w(f_a, f_b)$.

$$\begin{array}{ccccccccccccccccc} | & \mathrm{b} & | & \mathrm{b} & | & \mathrm{a} & | & \mathrm{a} & | & \mathrm{a} & | & \mathrm{b} & | & \mathrm{b} & | & \mathrm{b} & | \\ 0 & & 1 & & 2 & & 3 & & 4 & & 5 & & 6 & & 7 & & 8 \end{array}$$

$$f_{ab}(\{(0,0),(2,2),(2,4)\}) = f_b(f_a(\{(0,0),(2,2),(2,4)\}))$$
$$= f_b(\{(2,3),(2,5)\}) = \{(2,6)\}$$

$$\approx f_{ab}(\{||bbaaabbb, bb||aaabbb, bb|aa|abbb\})$$
$$= f_b(\{bb|a|aabbb, bb|aaa|bbb\}) = \{bb|aaab|bb\}$$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun *Parser* $:= \mathcal{M}^{\mathcal{M}}$. Für $f \in$ *Parser* ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun $\mathit{Parser} := \mathcal{M}^{\mathcal{M}}$. Für $f \in \mathit{Parser}$ ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$

Beispiel: `alternate`

$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{a|b} := \texttt{alternate}_w(f_a, f_b)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

Beispiel: `alternate`

$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{a|b} := \texttt{alternate}_w(f_a, f_b)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_{ab}(\{(0,0),(2,2),(2,4)\}) = f_a(\{\cdots\}) \cup f_b(\{\cdots\})$$
$$= \{(2,3),(2,5)\} \cup \{(0,1)\} = \{(0,1),(2,3),(2,5)\}$$

Beispiel: `alternate`

$$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$$
$$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_a := \texttt{readChar}_w(a)$
$f_b := \texttt{readChar}_w(b)$, $f_{a|b} := \texttt{alternate}_w(f_a, f_b)$.

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_{ab}(\{(0,0),(2,2),(2,4)\}) = f_a(\{\cdots\}) \cup f_b(\{\cdots\})$$
$$= \{(2,3),(2,5)\} \cup \{(0,1)\} = \{(0,1),(2,3),(2,5)\}$$

$$f_{a|b}(\{||bbaaabbb, bb||aaabbb, bb|aa|abbb\}) =$$
$$= \{|b|baaabbb, bb|a|aabbb, bb|aaa|bbb\}$$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun $\mathit{Parser} := \mathcal{M}^{\mathcal{M}}$. Für $f \in \mathit{Parser}$ ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to \mathit{Parser}$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

$\texttt{alternate}_w : \mathit{Parser} \times \mathit{Parser} \to \mathit{Parser}$
$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$

$\texttt{iterate}_w : \mathit{Parser} \to \mathit{Parser}$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M) = M \mapsto M \cup f(M) \cup f(f(M)) \cdots$$

Beispiel: `iterate`

$$\texttt{iterate}_w : \mathit{Parser} \to \mathit{Parser}$$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M)$$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_{a^*} := \texttt{iterate}_w(f_a)$

$$\underset{0}{|}\ \text{b}\ \underset{1}{|}\ \text{b}\ \underset{2}{|}\ \text{a}\ \underset{3}{|}\ \text{a}\ \underset{4}{|}\ \text{a}\ \underset{5}{|}\ \text{b}\ \underset{6}{|}\ \text{b}\ \underset{7}{|}\ \text{b}\ \underset{8}{|}$$

Beispiel: `iterate`

$$\texttt{iterate}_w : \mathit{Parser} \to \mathit{Parser}$$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M)$$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_{a^*} := \texttt{iterate}_w(f_a)$

$$\underset{0}{|}\ \mathsf{b}\ \underset{1}{|}\ \mathsf{b}\ \underset{2}{|}\ \mathsf{a}\ \underset{3}{|}\ \mathsf{a}\ \underset{4}{|}\ \mathsf{a}\ \underset{5}{|}\ \mathsf{b}\ \underset{6}{|}\ \mathsf{b}\ \underset{7}{|}\ \mathsf{b}\ \underset{8}{|}$$

$$f_{a^*}(\{(0,0),(2,2),(2,4)\}) = \{\cdots\} \cup f_a(\{\cdots\}) \cup f_a(f_a(\{\cdots\}) \cup \cdots$$

Beispiel: `iterate`

$$\texttt{iterate}_w : Parser \to Parser$$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M)$$

Sei $\Sigma = \{a, b\},\ w = bbaaabbb \in \Sigma^*,\ f_{a^*} := \texttt{iterate}_w(f_a)$

$$\underset{0}{|}\ \mathsf{b}\ \underset{1}{|}\ \mathsf{b}\ \underset{2}{|}\ \mathsf{a}\ \underset{3}{|}\ \mathsf{a}\ \underset{4}{|}\ \mathsf{a}\ \underset{5}{|}\ \mathsf{b}\ \underset{6}{|}\ \mathsf{b}\ \underset{7}{|}\ \mathsf{b}\ \underset{8}{|}$$

$$f_{a^*}(\{(0,0),(2,2),(2,4)\}) = \{\cdots\} \cup f_a(\{\cdots\}) \cup f_a(f_a(\{\cdots\}) \cup \cdots$$
$$= \{(0,0),(2,2),(2,4)\} \cup \{(2,3),(2,5)\} \cup \{(2,4)\} \cup \{(2,5)\} \cup \varnothing \cdots$$

Beispiel: `iterate`

$$\texttt{iterate}_w : \mathit{Parser} \to \mathit{Parser}$$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M)$$

Sei $\Sigma = \{a, b\}$, $w = bbaaabbb \in \Sigma^*$, $f_{a^*} := \texttt{iterate}_w(f_a)$

$$\underset{0}{|}\ \mathsf{b}\ \underset{1}{|}\ \mathsf{b}\ \underset{2}{|}\ \mathsf{a}\ \underset{3}{|}\ \mathsf{a}\ \underset{4}{|}\ \mathsf{a}\ \underset{5}{|}\ \mathsf{b}\ \underset{6}{|}\ \mathsf{b}\ \underset{7}{|}\ \mathsf{b}\ \underset{8}{|}$$

$$f_{a^*}(\{(0,0),(2,2),(2,4)\}) = \{\cdots\} \cup f_a(\{\cdots\}) \cup f_a(f_a(\{\cdots\}) \cup \cdots$$
$$= \{(0,0),(2,2),(2,4)\} \cup \{(2,3),(2,5)\} \cup \{(2,4)\} \cup \{(2,5)\} \cup \varnothing \cdots$$
$$= \{(0,0),(2,2),(2,3),(2,4),(2,5)\}$$

Beispiel: `iterate`

$$\texttt{iterate}_w : \mathit{Parser} \to \mathit{Parser}$$
$$\texttt{iterate}_w(f) = M \mapsto \bigcup_{i \in \mathbb{N}_0} f^i(M)$$

Sei $\Sigma = \{a, b\},\ w = bbaaabbb \in \Sigma^*,\ f_{a^*} := \texttt{iterate}_w(f_a)$

$$\underset{0}{|}\ \mathrm{b}\ \underset{1}{|}\ \mathrm{b}\ \underset{2}{|}\ \mathrm{a}\ \underset{3}{|}\ \mathrm{a}\ \underset{4}{|}\ \mathrm{a}\ \underset{5}{|}\ \mathrm{b}\ \underset{6}{|}\ \mathrm{b}\ \underset{7}{|}\ \mathrm{b}\ \underset{8}{|}$$

$$f_{a^*}(\{(0,0),(2,2),(2,4)\}) = \{\cdots\} \cup f_a(\{\cdots\}) \cup f_a(f_a(\{\cdots\}) \cup \cdots$$
$$= \{(0,0),(2,2),(2,4)\} \cup \{(2,3),(2,5)\} \cup \{(2,4)\} \cup \{(2,5)\} \cup \varnothing \cdots$$
$$= \{(0,0),(2,2),(2,3),(2,4),(2,5)\}$$

$$f_{a^*}(\{||bbaaabb, bb||aaabbb, bb|aa|abbb\})$$
$$= \{||bbaaabb, bb||aaabbb, bb|a|aabbb, bb|aa|abbb, bb|aaa|bbb\}$$

Definitionen

Gegeben: Eingabealphabet Σ, Eingabestring $w \in \Sigma^*$.

Bezeichne $m \in \mathbb{N}_0 \times \mathbb{N}_0$ als Match. (startIndex, endIndex)
Definiere $\mathcal{M} := \mathcal{P}(\mathbb{N}_0 \times \mathbb{N}_0)$ als Menge der Mengen von Matches.

Sei nun $Parser := \mathcal{M}^{\mathcal{M}}$. Für $f \in Parser$ ist also $f : \mathcal{M} \to \mathcal{M}$.

$\texttt{readChar}_w : \Sigma \to Parser$
$\texttt{readChar}_w(\sigma) = \quad M \mapsto \{(i, j+1) \mid (i,j) \in M \wedge w_j = c\}$

$\texttt{concatenate}_w : Parser \times Parser \to Parser$
$\texttt{concatenate}_w(f_1, f_2) = f_2 \circ f_1$

$\texttt{alternate}_w : Parser \times Parser \to Parser$
$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup f_2(M)$

$\texttt{iterate}_w : Parser \to Parser$
$$\texttt{iterate}_w(f) = \quad M \mapsto \bigcup_{i \in \mathbb{N}_0} \mathtt{p}^i(M)$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$$
\begin{aligned}
&f := \texttt{concatenate}_w(\\
&\quad \texttt{alternate}_w(\texttt{readChar}_w(a),\ \texttt{readChar}_w(b)), \\
&\quad \texttt{iterate}_w(\texttt{readChar}_w(c)))
\end{aligned}
$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\},\ w = xabccx$. Regex $(a \mid b)c^*$.

$$
\begin{aligned}
f := {} & \texttt{concatenate}_w(\\
& \quad \texttt{alternate}_w(\texttt{readChar}_w(a),\ \texttt{readChar}_w(b)), \\
& \quad \texttt{iterate}_w(\texttt{readChar}_w(c)))
\end{aligned}
$$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$f :=$ $\texttt{concatenate}_w($
    $\texttt{alternate}_w(\texttt{readChar}_w(a),\ \texttt{readChar}_w(b)),$
    $\texttt{iterate}_w(\texttt{readChar}_w(c)))$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$S_w = \{(0,0), (1,1), (2,2), (3,3), (4,4), (5,5)\}$$

$$\approx \{\textcolor{red}{||}xabccx, x\textcolor{red}{||}abccx, xa\textcolor{red}{||}bccx, xab\textcolor{red}{||}ccx, xabc\textcolor{red}{||}cx, xabcc\textcolor{red}{||}x\}$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$f :=$ `concatenate`$_w($
    `alternate`$_w($`readChar`$_w(a)$, `readChar`$_w(b))$,
    `iterate`$_w($`readChar`$_w(c)))$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$\underset{0}{|}\ \mathsf{x}\ \underset{1}{|}\ \mathsf{a}\ \underset{2}{|}\ \mathsf{b}\ \underset{3}{|}\ \mathsf{c}\ \underset{4}{|}\ \mathsf{c}\ \underset{5}{|}\ \mathsf{x}\ \underset{6}{|}$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$$f := \texttt{concatenate}_w(\\ \quad \texttt{alternate}_w(\texttt{readChar}_w(a),\ \texttt{readChar}_w(b)),\\ \quad \texttt{iterate}_w(\texttt{readChar}_w(c)))$$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$\underset{0}{|}\ \text{x}\ \underset{1}{|}\ \text{a}\ \underset{2}{|}\ \text{b}\ \underset{3}{|}\ \text{c}\ \underset{4}{|}\ \text{c}\ \underset{5}{|}\ \text{x}\ \underset{6}{|}$$

$$f(S_w) = f_{c*}(f_{a|b}(S_w)) = f_{c*}(f_a(S_w) \cup f_b(S_w))$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$$f := \texttt{concatenate}_w(\\ \quad \texttt{alternate}_w(\texttt{readChar}_w(a),\ \texttt{readChar}_w(b)),\\ \quad \texttt{iterate}_w(\texttt{readChar}_w(c)))$$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$\underset{0}{|}\ \text{x}\ \underset{1}{|}\ \text{a}\ \underset{2}{|}\ \text{b}\ \underset{3}{|}\ \text{c}\ \underset{4}{|}\ \text{c}\ \underset{5}{|}\ \text{x}\ \underset{6}{|}$$

$$f(S_w) = f_{c*}(f_{a|b}(S_w)) = f_{c*}(f_a(S_w) \cup f_b(S_w))$$

$$= f_{c*}(\{(1, 2)\} \cup f_b(S_w)) = f_{c*}(\{(1, 2), (2, 3)\})$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$f :=$ `concatenate`$_w($
  `alternate`$_w($`readChar`$_w(a)$, `readChar`$_w(b))$,
  `iterate`$_w($`readChar`$_w(c)))$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$\underset{0}{|}\ \mathrm{x}\ \underset{1}{|}\ \mathrm{a}\ \underset{2}{|}\ \mathrm{b}\ \underset{3}{|}\ \mathrm{c}\ \underset{4}{|}\ \mathrm{c}\ \underset{5}{|}\ \mathrm{x}\ \underset{6}{|}$$

$$f(S_w) = f_{c*}(f_{a|b}(S_w)) = f_{c*}(f_a(S_w) \cup f_b(S_w))$$

$$= f_{c*}(\{(1,2)\} \cup f_b(S_w)) = f_{c*}(\{(1,2),(2,3)\})$$

$$= \bigcup_{i \in \mathbb{N}_0} f_c^i(\{(1,2),(2,3)\}) = \{(1,2),(2,3)\} \cup \{(2,4)\} \cup \{(2,5)\} \cup \varnothing \cdots$$

Komplettbeispiel

Sei $\Sigma = \{a, b, c, x\}$, $w = xabccx$. Regex $(a \mid b)c^*$.

$f :=$ `concatenate`$_w($
  `alternate`$_w($`readChar`$_w(a)$, `readChar`$_w(b))$,
  `iterate`$_w($`readChar`$_w(c)))$

Definiere Startmenge $S_w \in \mathcal{M}$ durch $S_w := \{(i, i) \mid i \in [0, |w| - 1]\}$.

$$\begin{array}{ccccccccccccc} | & x & | & a & | & b & | & c & | & c & | & x & | \\ 0 & & 1 & & 2 & & 3 & & 4 & & 5 & & 6 \end{array}$$

$$f(S_w) = f_{c*}(f_{a|b}(S_w)) = f_{c*}(f_a(S_w) \cup f_b(S_w))$$

$$= f_{c*}(\{(1, 2)\} \cup f_b(S_w)) = f_{c*}(\{(1, 2), (2, 3)\})$$

$$= \bigcup_{i \in \mathbb{N}_0} f_c^i(\{(1, 2), (2, 3)\}) = \{(1, 2), (2, 3)\} \cup \{(2, 4)\} \cup \{(2, 5)\} \cup \varnothing \cdots$$

$$= \{(1, 2), (2, 3), (2, 4), (2, 5)\}$$

$$\approx \{x|a|bccx, xa|b|ccx, xa|bc|cx, xa|bcc|x\}$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\mathtt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\mathtt{ext}_w(r, M) := \left\{ \vphantom{\begin{array}{c} \\ \\ \\ \\ \\ \end{array}} \right.$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\mathtt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\mathtt{ext}_w(r, M) := \begin{cases} \{(i, j+1) \mid (i, j) \in M \wedge w_j = \sigma\} & , r = \sigma \in \Sigma \\ & \\ & \end{cases}$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\mathtt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\mathtt{ext}_w(r, M) := \begin{cases} \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\} & , r = \sigma \in \Sigma \\ \mathtt{ext}_w(r_2, \mathtt{ext}_w(r_1, M)) & , r = (r_1 r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ & \\ & \end{cases}$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\texttt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\texttt{ext}_w(r, M) := \begin{cases} \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\} & , r = \sigma \in \Sigma \\ \texttt{ext}_w(r_2, \texttt{ext}_w(r_1, M)) & , r = (r_1 r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \texttt{ext}_w(r_1, M) \cup \texttt{ext}_w(r_2, M) & , r = (r_1 \mid r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \end{cases}$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\mathtt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\mathtt{ext}_w(r, M) := \begin{cases} \{(i, j+1) \mid (i,j) \in M \land w_j = \sigma\} & , r = \sigma \in \Sigma \\ \mathtt{ext}_w(r_2, \mathtt{ext}_w(r_1, M)) & , r = (r_1 r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \mathtt{ext}_w(r_1, M) \cup \mathtt{ext}_w(r_2, M) & , r = (r_1 \mid r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \mathtt{iter}_w(t, M) & , r = t^*, t \in \mathcal{R}_\Sigma \end{cases}$$

$$\mathtt{iter}_w(t, M) := \begin{cases} \varnothing & \text{falls } M = \varnothing \\ M \cup \mathtt{iter}_w(t, \mathtt{ext}_w(t, M)) & \text{sonst} \end{cases}$$

Rekursiv gedacht

Σ Alphabet, $\mathcal{R}_\Sigma$ als Menge der Regexe über Σ. Weiter $w \in \Sigma^*$.

$\mathtt{ext}_w : \mathcal{R}_\Sigma \times \mathcal{M} \to \mathcal{M}$

$$\mathtt{ext}_w(r, M) := \begin{cases} \{(i, j+1) \mid (i,j) \in M \wedge w_j = \sigma\} & , r = \sigma \in \Sigma \\ \mathtt{ext}_w(r_2, \mathtt{ext}_w(r_1, M)) & , r = (r_1 r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \mathtt{ext}_w(r_1, M) \cup \mathtt{ext}_w(r_2, M) & , r = (r_1 \mid r_2), r_1, r_2 \in \mathcal{R}_\Sigma \\ \mathtt{iter}_w(t, M) & , r = t^*, t \in \mathcal{R}_\Sigma \end{cases}$$

$$\mathtt{iter}_w(t, M) := \begin{cases} \varnothing & \text{falls } M = \varnothing \\ M \cup \mathtt{iter}_w(t, \mathtt{ext}_w(t, M)) & \text{sonst} \end{cases}$$

Aufruf wäre dann:

$$\mathtt{ext}_w((a \mid b)c^*, S_w)$$

Erweiterung: Priorität bei Alternation

0 höchste, ∞ niedrigste Priorität.

Match Element von $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$. Entsprechend $\mathcal{M}$ anpassen.

$S_w := \{(i, i, 0) \mid i \in [0, |w| - 1]\}$

Erweiterung: Priorität bei Alternation

0 höchste, ∞ niedrigste Priorität.

Match Element von $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$. Entsprechend $\mathcal{M}$ anpassen.

$S_w := \{(i, i, 0) \mid i \in [0, |w| - 1]\}$

$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j + 1, p) \mid (i, j, p) \in M \wedge w_j = \sigma\}$

Erweiterung: Priorität bei Alternation

0 höchste, ∞ niedrigste Priorität.

Match Element von $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$. Entsprechend $\mathcal{M}$ anpassen.

$S_w := \{(i, i, 0) \mid i \in [0, |w| - 1]\}$

$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1, p) \mid (i, j, p) \in M \wedge w_j = \sigma\}$

$\texttt{decrPrio}\colon \mathcal{M} \to \mathcal{M}$
$\texttt{decrPrio}(M) = \{(i, j, p+1) \mid (i, j, p) \in M\}$

Erweiterung: Priorität bei Alternation

0 höchste, ∞ niedrigste Priorität.

Match Element von $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$. Entsprechend $\mathcal{M}$ anpassen.

$S_w := \{(i, i, 0) \mid i \in [0, |w| - 1]\}$

$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1, p) \mid (i, j, p) \in M \wedge w_j = \sigma\}$

$\texttt{decrPrio}\colon \mathcal{M} \to \mathcal{M}$
$\texttt{decrPrio}(M) = \{(i, j, p+1) \mid (i, j, p) \in M\}$

$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup \texttt{decrPrio}(f_2(M))$

Erweiterung: Priorität bei Alternation

0 höchste, ∞ niedrigste Priorität.

Match Element von $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$. Entsprechend $\mathcal{M}$ anpassen.

$S_w := \{(i, i, 0) \mid i \in [0, |w| - 1]\}$

$\texttt{readChar}_w(\sigma) = M \mapsto \{(i, j+1, p) \mid (i, j, p) \in M \wedge w_j = \sigma\}$

$\texttt{decrPrio}\colon \mathcal{M} \to \mathcal{M}$
$\texttt{decrPrio}(M) = \{(i, j, p+1) \mid (i, j, p) \in M\}$

$\texttt{alternate}_w(f_1, f_2) = M \mapsto f_1(M) \cup \texttt{decrPrio}(f_2(M))$

$$\text{nicht leer} > \text{startIndex} > \text{Priorität} > \text{Länge}$$

$$(0,3,0) > (1,5,0) > (1,8,1) > (1,5,1) > (0,0,0)$$

Erweiterung: Priorität bei Alternation (cont.)

Unerwartetes Verhalten bei Iteration:
$\Sigma = \{a, b\}$, $w = ab$. $f \approx (a \mid b)^*$.

Erweiterung: Priorität bei Alternation (cont.)

Unerwartetes Verhalten bei Iteration:
$\Sigma = \{a, b\},\ w = ab.\ f \approx (a \mid b)^*.$

$$f(S_w) = f(\{(0,0,0),(1,1,0)\})$$

$$= \{(0,0,0),(1,1,0)\} \cup f_{a|b}(\{\cdots\}) \cup f_{a|b}(f_{a|b}(\{\cdots\}))\cdots$$
$$= \{(0,0,0),(1,1,0)\} \cup \underbrace{\{(0,1,0)\}}_{f_a(\{\cdots\})} \cup \underbrace{\{(1,2,1)\}}_{f_b(\{\cdots\})} \cup \underbrace{\{(0,2,1)\}}_{f_b(f_{a|b}(\{\cdots\}))}$$

Erweiterung: Priorität bei Alternation (cont.)

Unerwartetes Verhalten bei Iteration:
$\Sigma = \{a, b\}$, $w = ab$. $f \approx (a \mid b)^*$.

$$f(S_w) = f(\{(0,0,0),(1,1,0)\})$$

$$= \{(0,0,0),(1,1,0)\} \cup f_{a|b}(\{\cdots\}) \cup f_{a|b}(f_{a|b}(\{\cdots\}))\cdots$$
$$= \{(0,0,0),(1,1,0)\} \cup \underbrace{\{(0,1,0)\}}_{f_a(\{\cdots\})} \cup \underbrace{\{(1,2,1)\}}_{f_b(\{\cdots\})} \cup \underbrace{\{(0,2,1)\}}_{f_b(f_{a|b}(\{\cdots\}))}$$

$$\Longrightarrow \underbrace{(0,1,0) > (0,2,1)}_{|a|b \quad > \quad |ab|} > (1,1,0) > (1,2,1) > (0,0,0)$$

Implementierung in Java

java.util.function.Function<A,B>

Function<A,B>.apply : A $\to$ B

Function<A,B>.compose : Function<B,C> $\to$
Function<A,C>

Implementierung in Java

`java.util.function.Function<A,B>`

`Function<A,B>.apply` : `A` $\to$ `B`

`Function<A,B>.compose` : `Function<B,C>` $\to$ `Function<A,C>`

Syntax mit Java-Lambdas, ähnlich zum Syntax hier:

```
Function<String, String> f = s -> s+"!";
System.out.println(f.apply("ok"));
```

Ausgabe: `ok!`