Modeling Regex Operators for Solving Regex Crossword Puzzles

No Author Given

No Institute Given

1 Appendix

1.1 The Function $\texttt{len}(r, k)$

We define the function $\texttt{len}(r, k)$ to calculate all the possible lengths of string s satisfying $s \in \mathcal{L}(r) \wedge |s| \leq k$, where r is a regex and $k \in \mathbb{N}_+$, the output of $\texttt{len}(r, k)$ is a set of integers. In the process of solving puzzle, the function $\texttt{len}$ is used to calculate the number of variables that affected by r in the variable sequence V. The recursive calculation of the function $\texttt{len}(r, k)$ is shown in Fig. 1. The function $\texttt{sum}$ is an auxiliary function of $\texttt{len}(r, k)$, which is defined as $\texttt{sum}(N_1, N_2, k) = \{x + y \mid x \in N_1, y \in N_2, x + y \leq k\}$, where N_1 and N_2 are two sets of positive integers, and k is a positive integer. In particular, when $N_1 = N_2 = N$, we use the symbol $\texttt{sum}^t$ to represent calling the function $\texttt{sum}$ t times recursively. For example, $\texttt{sum}^1(N, N, k) = \texttt{sum}(N, N, k)$, $\texttt{sum}^2(N, N, k) = \texttt{sum}(\texttt{sum}^1(N, N, k), N, k)$, $\texttt{sum}^3(N, N, k) = \texttt{sum}(\texttt{sum}^2(N, N, k), N, k)$.

We define the calculation method of function $\texttt{len}(r, k)$ according to the language of REs or the semantics of extended operators. The Equ. (1) — Equ. (7) are trivial. We illustrate Equ. (5) by Example 1. Equ. (8) and Equ. (9) indicate that non-capturing group and capturing group operations do not affect the calculation of the possible length of string s. According to the semantics of backreference, $\backslash i$ matches the exact same text that matched by the i-th capturing group, and considering the initial value of backreferences is ε in this paper, Equ. (10) holds immediately. Similarly, lookarounds and anchors do not consume characters. Therefore, Equ. (11) — Equ. (18) are defined as calculating $\texttt{len}(r, k)$ for lookarounds-free and anchors-free regexes. In addition, it should be noted that Equ. (13) — Equ. (18) obtain an over-approximate set of possible lengths of string, as shown in Example 2 and Example 3.

Example 1. For a regex $r = \texttt{a*}$ and $k = 3$, $\texttt{len}(r, k) \xlongequal{\text{Equ. (5), m=0}} \{0\} \cup \{1\} \cup \bigcup_{1 \leqslant t \leqslant +\infty} \texttt{sum}^t(\{1\}, \{1\}, 3) = \{0, 1\} \cup \{2\}^{t=1} \cup \{3\}^{t=2} \cup \varnothing^{t \geqslant 3} = \{0, 1, 2, 3\}$.

Example 2. For a regex $r = \texttt{(?=a\{2,\})a*}$ and $k \geqslant 2$, $\texttt{len}(r, k) = \texttt{len}(\texttt{a*}, k) = \{0, 1, \ldots, k\}$. However, the shortest string matching r is $s = \texttt{aa}$, $|s| = 2$, the string $s' \in \mathcal{L}(r)$ with length 0 or 1 does not exist. Therefore, Equ. (13) is over-approximate.

Example 3. For a regex $r = \texttt{a*\textbackslash ba*}$ and $k \geqslant 1$, $\texttt{len}(r, k) = \{0, 1, \ldots, k\}$. However, the shortest string matching r is $s = \texttt{a}$, $|s| = 1$, there is no string $s' \in \mathcal{L}(r)$ with $|s'| = 0$. Therefore, Equ. (17) is over-approximate.

$$\begin{aligned}
\texttt{len}(\varepsilon, k) &= \{0\} && (1)\\
\texttt{len}(a, k) &= \{1\} \quad (a \in \Sigma) && (2)\\
\texttt{len}(r_1 r_2, k) &= \texttt{sum}(\texttt{len}(r_1, k), \texttt{len}(r_2, k), k) && (3)\\
\texttt{len}(r_1\texttt{|}r_2, k) &= \texttt{len}(r_1, k) \cup \texttt{len}(r_2, k) && (4)\\
\texttt{len}(r\texttt{\{m,n\}}, k) &= \begin{cases} \{0\} \cup \texttt{len}(r,k) \cup \bigcup\limits_{1 \leqslant t \leqslant \texttt{n}-1} \texttt{sum}^t(\texttt{len}(r,k), \texttt{len}(r,k), k) & \texttt{m} = 0 \\ \texttt{len}(r,k) \cup \bigcup\limits_{1 \leqslant t \leqslant \texttt{n}-1} \texttt{sum}^t(\texttt{len}(r,k), \texttt{len}(r,k), k) & \texttt{m} = 1 \\ \bigcup\limits_{\texttt{m}-1 \leqslant t \leqslant \texttt{n}-1} \texttt{sum}^t(\texttt{len}(r,k), \texttt{len}(r,k), k) & \texttt{m} \geqslant 2 \end{cases} && (5)\\
\texttt{len}(r\texttt{\{m,n\}?}, k) &= \texttt{len}(r\texttt{\{m,n\}}, k) && (6)\\
\texttt{len}(\texttt{[}C\texttt{]}, k) &= \{1\} && (7)\\
\texttt{len}(\texttt{(?:}r\texttt{)}, k) &= \texttt{len}(r, k) && (8)\\
\texttt{len}(\texttt{(}r\texttt{)}_i, k) &= \texttt{len}(r, k) && (9)\\
\texttt{len}(\backslash i, k) &= \texttt{len}(\texttt{(}r\texttt{)}_i, k) \cup \{0\} && (10)\\
\texttt{len}(\hat{}\,r, k) &= \texttt{len}(r, k) && (11)\\
\texttt{len}(r\texttt{\$}, k) &= \texttt{len}(r, k) && (12)\\
\texttt{len}(\texttt{(?=}r_1\texttt{)}r_2, k) &= \texttt{len}(r_2, k) && (13)\\
\texttt{len}(\texttt{(?!}r_1\texttt{)}r_2, k) &= \texttt{len}(r_2, k) && (14)\\
\texttt{len}(r_1\texttt{(?<=}r_2\texttt{)}, k) &= \texttt{len}(r_1, k) && (15)\\
\texttt{len}(r_1\texttt{(?<!}r_2\texttt{)}, k) &= \texttt{len}(r_1, k) && (16)\\
\texttt{len}(r_1\backslash\texttt{b}r_2, k) &= \texttt{sum}(\texttt{len}(r_1, k), \texttt{len}(r_2, k), k) && (17)\\
\texttt{len}(r_1\backslash\texttt{B}r_2, k) &= \texttt{sum}(\texttt{len}(r_1, k), \texttt{len}(r_2, k), k) && (18)
\end{aligned}$$

Fig. 1. The calculation of the function $\texttt{len}(r, k)$.