

# RegEx

Fahad Ausaf<sup>1</sup>, Roy Dyckhoff<sup>2</sup>, and Christian Urban<sup>1</sup>

<sup>1</sup> King's College London, United Kingdom  
<sup>2</sup> St Andrews

**Abstract.** BLA BLA Sulzmann and Lu [1]

**Keywords:**

## 1 Introduction

Regular expressions

$$r := \text{NULL} \mid \text{EMPTY} \mid \text{CHAR } c \mid \text{ALT } r_1 r_2 \mid \text{SEQ } r_1 r_2 \mid \text{STAR } r$$

Values

$$v := \text{Void} \mid \text{Char } c \mid \text{Left } v \mid \text{Right } v \mid \text{Seq } v_1 v_2 \mid \text{Stars } vs$$

The language of a regular expression

$$\begin{aligned} L \text{ NULL} &\stackrel{\text{def}}{=} \emptyset \\ L \text{ EMPTY} &\stackrel{\text{def}}{=} \{\emptyset\} \\ L (\text{CHAR } c) &\stackrel{\text{def}}{=} \{[c]\} \\ L (\text{SEQ } r_1 r_2) &\stackrel{\text{def}}{=} (L r_1) @ (L r_2) \\ L (\text{ALT } r_1 r_2) &\stackrel{\text{def}}{=} (L r_1) \cup (L r_2) \\ L (\text{STAR } r) &\stackrel{\text{def}}{=} (L r)^* \end{aligned}$$

The nullable function

$$\begin{aligned} \text{nullable } \text{NULL} &\stackrel{\text{def}}{=} \text{False} \\ \text{nullable } \text{EMPTY} &\stackrel{\text{def}}{=} \text{True} \\ \text{nullable } (\text{CHAR } c) &\stackrel{\text{def}}{=} \text{False} \\ \text{nullable } (\text{ALT } r_1 r_2) &\stackrel{\text{def}}{=} \text{nullable } r_1 \vee \text{nullable } r_2 \\ \text{nullable } (\text{SEQ } r_1 r_2) &\stackrel{\text{def}}{=} \text{nullable } r_1 \wedge \text{nullable } r_2 \\ \text{nullable } (\text{STAR } r) &\stackrel{\text{def}}{=} \text{True} \end{aligned}$$

The derivative function for characters and strings

$$\begin{aligned}
\text{der } c \text{ } \mathit{NULL} &\stackrel{\text{def}}{=} \mathit{NULL} \\
\text{der } c \text{ } \mathit{EMPTY} &\stackrel{\text{def}}{=} \mathit{NULL} \\
\text{der } c \text{ } (\mathit{CHAR } c') &\stackrel{\text{def}}{=} \text{if } c = c' \text{ then } \mathit{EMPTY} \text{ else } \mathit{NULL} \\
\text{der } c \text{ } (\mathit{ALT } r_1 r_2) &\stackrel{\text{def}}{=} \mathit{ALT} (\text{der } c r_1) (\text{der } c r_2) \\
\text{der } c \text{ } (\mathit{SEQ } r_1 r_2) &\stackrel{\text{def}}{=} \text{if nullable } r_1 \text{ then } \mathit{ALT} (\mathit{SEQ} (\text{der } c r_1) r_2) (\text{der } c r_2) \\
&\quad \text{else } \mathit{SEQ} (\text{der } c r_1) r_2 \\
\text{der } c \text{ } (\mathit{STAR } r) &\stackrel{\text{def}}{=} \mathit{SEQ} (\text{der } c r) (\mathit{STAR } r) \\
\text{ders } [] r &\stackrel{\text{def}}{=} r \\
\text{ders } (c::s) r &\stackrel{\text{def}}{=} \text{ders } s (\text{der } c r)
\end{aligned}$$

The *flat* function for values

$$\begin{aligned}
|\mathit{Void}| &\stackrel{\text{def}}{=} [] \\
|\mathit{Char } c| &\stackrel{\text{def}}{=} [c] \\
|\mathit{Left } v| &\stackrel{\text{def}}{=} |v| \\
|\mathit{Right } v| &\stackrel{\text{def}}{=} |v| \\
|\mathit{Seq } v_1 v_2| &\stackrel{\text{def}}{=} |v_1| @ |v_2| \\
|\mathit{Stars } []| &\stackrel{\text{def}}{=} [] \\
|\mathit{Stars } (v::vs)| &\stackrel{\text{def}}{=} |v| @ |\mathit{Stars } vs|
\end{aligned}$$

The *mkeps* function

$$\begin{aligned}
\text{mkeps } \mathit{EMPTY} &\stackrel{\text{def}}{=} \mathit{Void} \\
\text{mkeps } (\mathit{SEQ } r_1 r_2) &\stackrel{\text{def}}{=} \mathit{Seq} (\text{mkeps } r_1) (\text{mkeps } r_2) \\
\text{mkeps } (\mathit{ALT } r_1 r_2) &\stackrel{\text{def}}{=} \text{if nullable } r_1 \text{ then } \mathit{Left} (\text{mkeps } r_1) \text{ else } \mathit{Right} (\text{mkeps } r_2) \\
\text{mkeps } (\mathit{STAR } r) &\stackrel{\text{def}}{=} \mathit{Stars } []
\end{aligned}$$

The *inj* function

$$\begin{aligned}
\text{inj } (\mathit{CHAR } d) c \text{ } \mathit{Void} &\stackrel{\text{def}}{=} \mathit{Char } d \\
\text{inj } (\mathit{ALT } r_1 r_2) c \text{ } (\mathit{Left } v_1) &\stackrel{\text{def}}{=} \mathit{Left} (\text{inj } r_1 c v_1) \\
\text{inj } (\mathit{ALT } r_1 r_2) c \text{ } (\mathit{Right } v_2) &\stackrel{\text{def}}{=} \mathit{Right} (\text{inj } r_2 c v_2) \\
\text{inj } (\mathit{SEQ } r_1 r_2) c \text{ } (\mathit{Seq } v_1 v_2) &\stackrel{\text{def}}{=} \mathit{Seq} (\text{inj } r_1 c v_1) v_2 \\
\text{inj } (\mathit{SEQ } r_1 r_2) c \text{ } (\mathit{Left} (\mathit{Seq } v_1 v_2)) &\stackrel{\text{def}}{=} \mathit{Seq} (\text{inj } r_1 c v_1) v_2 \\
\text{inj } (\mathit{SEQ } r_1 r_2) c \text{ } (\mathit{Right } v_2) &\stackrel{\text{def}}{=} \mathit{Seq} (\text{mkeps } r_1) (\text{inj } r_2 c v_2) \\
\text{inj } (\mathit{STAR } r) c \text{ } (\mathit{Seq } v (\mathit{Stars } vs)) &\stackrel{\text{def}}{=} \mathit{Stars} ((\text{inj } r c v)::vs)
\end{aligned}$$

The inhabitation relation:



$$\begin{array}{c}
\frac{v_1 \succeq_{r_1} v_1' \quad v_1 \neq v_1'}{Seq\ v_1\ v_2 \succeq_{SEQ\ r_1\ r_2} Seq\ v_1'\ v_2'} \quad \frac{v_2 \succeq_{r_2} v_2'}{Seq\ v_1\ v_2 \succeq_{SEQ\ r_1\ r_2} Seq\ v_1\ v_2'} \\
\frac{len(|v_1|) \leq len(|v_2|)}{Left\ v_2 \succeq_{ALT\ r_1\ r_2} Right\ v_1} \quad \frac{len(|v_2|) < len(|v_1|)}{Right\ v_1 \succeq_{ALT\ r_1\ r_2} Left\ v_2} \\
\frac{v_2 \succeq_{r_2} v_2'}{Right\ v_2 \succeq_{ALT\ r_1\ r_2} Right\ v_2'} \quad \frac{v_1 \succeq_{r_1} v_1'}{Left\ v_1 \succeq_{ALT\ r_1\ r_2} Left\ v_1'} \\
\\
\frac{}{Void \succeq_{EMPTY} Void} \quad \frac{}{Char\ c \succeq_{CHAR} c\ Char\ c} \\
\frac{|Stars\ (v::vs)| = []}{Stars\ [] \succeq_{STAR\ r} Stars\ (v::vs)} \quad \frac{|Stars\ (v::vs)| \neq []}{Stars\ (v::vs) \succeq_{STAR\ r} Stars\ []} \\
\frac{v_1 \succeq_r v_2 \quad v_1 \neq v_2}{Stars\ (v_1::vs_1) \succeq_{STAR\ r} Stars\ (v_2::vs_2)} \\
\frac{Stars\ vs_1 \succeq_{STAR\ r} Stars\ vs_2}{Stars\ (v::vs_1) \succeq_{STAR\ r} Stars\ (v::vs_2)} \quad \frac{}{Stars\ [] \succeq_{STAR\ r} Stars\ []}
\end{array}$$

A prefix of a string  $s$

$$s_1 \sqsubseteq s_2 \stackrel{def}{=} \exists s_3. s_1 @ s_3 = s_2$$

Values and non-problematic values

$$Values\ r\ s \stackrel{def}{=} \{v \mid \vdash v : r \wedge (|v|) \sqsubseteq s\}$$

$$NValues\ r\ s \stackrel{def}{=} \{v \mid \models v : r \wedge (|v|) \sqsubseteq s\}$$

The point is that for a given  $s$  and  $r$  there are only finitely many non-problematic values.

Some lemmas we have proved:

$$\begin{array}{l}
(L\ r) = \{|v| \mid \vdash v : r\} \\
(L\ r) = \{|v| \mid \models v : r\} \\
If\ nullable\ r\ then\ \vdash\ mkeps\ r : r. \\
If\ nullable\ r\ then\ |mkeps\ r| = []. \\
If\ \vdash\ v : der\ c\ r\ then\ \vdash\ (inj\ r\ c\ v) : r. \\
If\ \vdash\ v : der\ c\ r\ then\ |inj\ r\ c\ v| = c::(|v|). \\
If\ nullable\ r\ then\ [] \in r \rightarrow mkeps\ r. \\
If\ s \in r \rightarrow v\ then\ |v| = s. \\
If\ s \in r \rightarrow v\ then\ \models v : r. \\
If\ s \in r \rightarrow v_1\ and\ s \in r \rightarrow v_2\ then\ v_1 = v_2.
\end{array}$$

This is the main theorem that lets us prove that the algorithm is correct according to  $s \in r \rightarrow v$ :

$$If\ s \in der\ c\ r \rightarrow v\ then\ (c::s) \in r \rightarrow (inj\ r\ c\ v).$$

**Proof** The proof is by induction on the definition of *der*. Other inductions would go through as well. The interesting case is for *SEQ*  $r_1 r_2$ . First we analyse the case where *nullable*  $r_1$ . We have by induction hypothesis

$$\begin{aligned} (IH1) \quad & \forall s v. \text{ if } s \in \text{der } c r_1 \rightarrow v \text{ then } (c::s) \in r_1 \rightarrow (\text{inj } r_1 c v) \\ (IH2) \quad & \forall s v. \text{ if } s \in \text{der } c r_2 \rightarrow v \text{ then } (c::s) \in r_2 \rightarrow (\text{inj } r_2 c v) \end{aligned}$$

and have

$$s \in \text{ALT } (\text{SEQ } (\text{der } c r_1) r_2) (\text{der } c r_2) \rightarrow v$$

There are two cases what  $v$  can be: (1) *Left*  $v'$  and (2) *Right*  $v'$ .

- (1) We know  $s \in \text{SEQ } (\text{der } c r_1) r_2 \rightarrow v'$  holds, from which we can infer that there are  $s_1, s_2, v_1, v_2$  with

$$s_1 \in \text{der } c r_1 \rightarrow v_1 \quad \text{and} \quad s_2 \in r_2 \rightarrow v_2$$

and also

$$\nexists s_3 s_4. s_3 \neq [] \wedge s_3 @ s_4 = s_2 \wedge s_1 @ s_3 \in (L (\text{der } c r_1)) \wedge s_4 \in (L r_2)$$

and have to prove

$$(c::s_1 @ s_2) \in \text{SEQ } r_1 r_2 \rightarrow \text{Seq } (\text{inj } r_1 c v_1) v_2$$

The two requirements  $(c::s_1) \in r_1 \rightarrow (\text{inj } r_1 c v_1)$  and  $s_2 \in r_2 \rightarrow v_2$  can be proved by the induction hypotheses (IH1) and the fact above.

This leaves to prove

$$\nexists s_3 s_4. s_3 \neq [] \wedge s_3 @ s_4 = s_2 \wedge c::s_1 @ s_3 \in (L r_1) \wedge s_4 \in (L r_2)$$

which holds because  $c::s_1 @ s_3 \in (L r_1)$  implies  $s_1 @ s_3 \in (L (\text{der } c r_1))$

- (2) This case is similar except that in the last step we have to instantiate the existential quantifier with  $\text{Seq } (\text{proj } r_1 c v) v'$

The final case is that  $\neg \text{nullable } r_1$  holds. This case again similar to the cases above.

Things we have proved about our version of the Sulzmann ordering

$$\text{If } \vdash v : r \text{ then } v \succeq_r v.$$

Things we like to prove, but cannot:

$$\text{If } s \in r \rightarrow v_1, \vdash v_2 : r, \text{ then } v_1 \succeq_r v_2$$

## References

1. M. Sulzmann and K. Lu. POSIX Regular Expression Parsing with Derivatives. In *Proc. of the 12th International Conference on Functional and Logic Programming (FLOPS)*, volume 8475 of *LNCS*, pages 203–220, 2014.

## 2 Roy's Rules

$$\begin{array}{c}
 \text{Void } \triangleleft \epsilon \quad \text{Char } c \triangleleft \text{Lit } c \\
 \\
 \frac{v_1 \triangleleft r_1}{\text{Left } v_1 \triangleleft r_1 + r_2} \quad \frac{v_2 \triangleleft r_2 \quad |v_2| \notin L(r_1)}{\text{Right } v_2 \triangleleft r_1 + r_2} \\
 \\
 \frac{v_1 \triangleleft r_1 \quad v_2 \triangleleft r_2 \quad s \in L(r_1 \setminus |v_1|) \wedge |v_2| \setminus s \in L(r_2) \Rightarrow s = \square}{(v_1, v_2) \triangleleft r_1 \cdot r_2} \\
 \\
 \frac{v \triangleleft r \quad vs \triangleleft r^* \quad |v| \neq \square}{(v :: vs) \triangleleft r^*} \quad \square \triangleleft r^*
 \end{array}$$