POSIX Lexing with Derivatives of Regular Expressions (Proof Pearl)

Fahad Ausaf[1], Roy Dyckhoff[2], and Christian Urban[1]

[1] King's College London, United Kingdom
[2] St Andrews

Abstract. Brzozowski introduced the notion of derivatives of regular expressions that can be used for very simple regular expression matching algorithms. BLA BLA Sulzmann and Lu [1]

Keywords:

1 Introduction

Regular exprtessions

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

Values

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

The language of a regular expression

$$\begin{array}{lcl}
L\ NULL & \stackrel{\text{def}}{=} & \varnothing \\
L\ EMPTY & \stackrel{\text{def}}{=} & \{[]\} \\
L\ (CHAR\ c) & \stackrel{\text{def}}{=} & \{[c]\} \\
L\ (SEQ\ r_1\ r_2) & \stackrel{\text{def}}{=} & (L\ r_1)\ @\ (L\ r_2) \\
L\ (ALT\ r_1\ r_2) & \stackrel{\text{def}}{=} & (L\ r_1) \cup (L\ r_2) \\
L\ (STAR\ r) & \stackrel{\text{def}}{=} & (L\ r)\star
\end{array}$$

The nullable function

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

The derivative function for characters and strings

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

The *flat* function for values

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

The *mkeps* function

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

The *inj* function

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

The inhabitation relation:

$$\frac{\vdash v_1 : r_1 \qquad \vdash v_2 : r_2}{\vdash \mathit{Seq}\ v_1\ v_2 : \mathit{SEQ}\ r_1\ r_2}$$

$$\frac{\vdash v_1 : r_1}{\vdash (\mathit{Left}\ v_1) : \mathit{ALT}\ r_1\ r_2} \qquad \frac{\vdash v_2 : r_1}{\vdash (\mathit{Right}\ v_2) : \mathit{ALT}\ r_2\ r_1}$$

$$\frac{}{\vdash \mathit{Void} : \mathit{EMPTY}} \qquad \frac{}{\vdash (\mathit{Char}\ c) : \mathit{CHAR}\ c}$$

$$\frac{}{\vdash \mathit{Stars}\ [] : \mathit{STAR}\ r} \qquad \frac{\vdash v : r \qquad \vdash \mathit{Stars}\ vs : \mathit{STAR}\ r}{\vdash \mathit{Stars}\ (v{::}vs) : \mathit{STAR}\ r}$$

We have also introduced a slightly restricted version of this relation where the last rule is restricted so that $|v| \neq []$. This relation for *non-problematic* is written $\models v : r$.

Our Posix relation $s \in r \to v$

$$\frac{}{[] \in \mathit{EMPTY} \to \mathit{Void}} \qquad \frac{}{[c] \in \mathit{CHAR}\ c \to (\mathit{Char}\ c)}$$

$$\frac{s \in r_1 \to v}{s \in \mathit{ALT}\ r_1\ r_2 \to (\mathit{Left}\ v)} \qquad \frac{s \in r_2 \to v \qquad s \notin (L\ r_1)}{s \in \mathit{ALT}\ r_1\ r_2 \to (\mathit{Right}\ v)}$$

$$\frac{\begin{array}{c} s_1 \in r_1 \to v_1 \qquad s_2 \in r_2 \to v_2 \\ \nexists s_3\ s_4.\ s_3 \neq [] \wedge s_3 @ s_4 = s_2 \wedge s_1 @ s_3 \in (L\ r_1) \wedge s_4 \in (L\ r_2) \end{array}}{(s_1 @ s_2) \in \mathit{SEQ}\ r_1\ r_2 \to \mathit{Seq}\ v_1\ v_2}$$

$$\frac{|v| \neq [] \qquad \begin{array}{c} s_1 \in r \to v \qquad s_2 \in \mathit{STAR}\ r \to \mathit{Stars}\ vs \\ \nexists s_3\ s_4.\ s_3 \neq [] \wedge s_3 @ s_4 = s_2 \wedge s_1 @ s_3 \in (L\ r) \wedge s_4 \in (L\ (\mathit{STAR}\ r)) \end{array}}{(s_1 @ s_2) \in \mathit{STAR}\ r \to \mathit{Stars}\ (v{::}vs)}$$

$$\frac{}{[] \in \mathit{STAR}\ r \to \mathit{Stars}\ []}$$

Our version of Sulzmann's ordering relation

$$\frac{v_1 \succ r_1\ v_1{}' \qquad v_1 \neq v_1{}'}{Seq\ v_1\ v_2 \succ SEQ\ r_1\ r_2\ Seq\ v_1{}'\ v_2{}'} \qquad \frac{v_2 \succ r_2\ v_2{}'}{Seq\ v_1\ v_2 \succ SEQ\ r_1\ r_2\ Seq\ v_1\ v_2{}'}$$

$$\frac{len\ (|v_1|) \leq len\ (|v_2|)}{(Left\ v_2) \succ ALT\ r_1\ r_2\ (Right\ v_1)} \qquad \frac{len\ (|v_2|) < len\ (|v_1|)}{(Right\ v_1) \succ ALT\ r_1\ r_2\ (Left\ v_2)}$$

$$\frac{v_2 \succ r_2\ v_2{}'}{(Right\ v_2) \succ ALT\ r_1\ r_2\ (Right\ v_2{}')} \qquad \frac{v_1 \succ r_1\ v_1{}'}{(Left\ v_1) \succ ALT\ r_1\ r_2\ (Left\ v_1{}')}$$

$$\frac{}{Void \succ EMPTY\ Void} \qquad \frac{}{(Char\ c) \succ CHAR\ c\ (Char\ c)}$$

$$\frac{|Stars\ (v{::}vs)| = []}{Stars\ []\ \succ STAR\ r\ Stars\ (v{::}vs)} \qquad \frac{|Stars\ (v{::}vs)| \neq []}{Stars\ (v{::}vs) \succ STAR\ r\ Stars\ []}$$

$$\frac{v_1 \succ r\ v_2 \qquad v_1 \neq v_2}{Stars\ (v_1{::}vs_1) \succ STAR\ r\ Stars\ (v_2{::}vs_2)}$$

$$\frac{Stars\ vs_1 \succ STAR\ r\ Stars\ vs_2}{Stars\ (v{::}vs_1) \succ STAR\ r\ Stars\ (v{::}vs_2)} \qquad \frac{}{Stars\ []\ \succ STAR\ r\ Stars\ []}$$

A prefix of a string s

$$s_1 \sqsubseteq s_2 \stackrel{def}{=} \exists\, s3.\ s_1\ @\ s3 = s_2$$

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:

$(L\ r) = \{|v| \mid \vdash v : r\}$
$(L\ r) = \{|v| \mid \models v : r\}$
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* $v_1 = v_2$.

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

$$\begin{array}{ll}(IH1) & \forall\, s\, v.\ \text{if}\ s \in der\ c\ r_1 \to v\ \text{then}\ (c{::}s) \in r_1 \to (inj\ r_1\ c\ v)\\ (IH2) & \forall\, s\, v.\ \text{if}\ s \in der\ c\ r_2 \to v\ \text{then}\ (c{::}s) \in r_2 \to (inj\ r_2\ c\ v)\end{array}$$

and have

$$s \in ALT\ (SEQ\ (der\ c\ r_1)\ r_2)\ (der\ c\ r_2) \to v$$

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

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

$$s_1 \in der\ c\ r_1 \to v_1 \qquad \text{and} \qquad s_2 \in r_2 \to v_2$$

and also

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

and have to prove

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

The two requirements $(c{::}s_1) \in r_1 \to (inj\ r_1\ c\ v_1)$ and $s_2 \in r_2 \to v_2$ can be proved by the induction hypothese (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 $s_1\ @\ s_3 \in (L\ (der\ c\ r_1))$

(2) This case is similar.

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

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

$$Void \lhd \epsilon \qquad Char\,c \lhd Lit\,c$$

$$\frac{v_1 \lhd r_1}{Left\,v_1 \lhd r_1 + r_2} \qquad \frac{v_2 \lhd r_2 \qquad |v_2| \notin L(r_1)}{Right\,v_2 \lhd r_1 + r_2}$$

$$\frac{v_1 \lhd r_1 \qquad v_2 \lhd r_2 \qquad s \in L(r_1 \backslash |v_1|) \wedge |v_2| \backslash s \,\epsilon\, L(r_2) \Rightarrow s = []}{(v_1, v_2) \lhd r_1 \cdot r_2}$$

$$\frac{v \lhd r \qquad vs \lhd r^* \qquad |v| \neq []}{(v :: vs) \lhd r^*} \qquad [] \lhd r^*$$