Proof

Recall the definitions for regular expressions and the language associated with a regular expression:

$$
\begin{array}{rcl}
r & ::= & \mathbf{0} \\
& | & \mathbf{1} \\
& | & c \\
& | & r_1 \cdot r_2 \\
& | & r_1 + r_2 \\
& | & r^*
\end{array}
\qquad
\begin{array}{rcl}
L(\mathbf{0}) & \stackrel{\text{def}}{=} & \varnothing \\
L(\mathbf{1}) & \stackrel{\text{def}}{=} & \{\texttt{""}\} \\
L(c) & \stackrel{\text{def}}{=} & \{\texttt{"}c\texttt{"}\} \\
L(r_1 \cdot r_2) & \stackrel{\text{def}}{=} & L(r_1) \,@\, L(r_2) \\
L(r_1 + r_2) & \stackrel{\text{def}}{=} & L(r_1) \cup L(r_2) \\
L(r^*) & \stackrel{\text{def}}{=} & \bigcup_{n \geq 0} L(r)^n
\end{array}
$$

We also defined the notion of a derivative of a regular expression (the derivative with respect to a character):

$$
\begin{array}{lcl}
der\, c\, (\mathbf{0}) & \stackrel{\text{def}}{=} & \mathbf{0} \\
der\, c\, (\mathbf{1}) & \stackrel{\text{def}}{=} & \mathbf{0} \\
der\, c\, (d) & \stackrel{\text{def}}{=} & \text{if } c = d \text{ then } \mathbf{1} \text{ else } \mathbf{0} \\
der\, c\, (r_1 + r_2) & \stackrel{\text{def}}{=} & (der\, c\, r_1) + (der\, c\, r_2) \\
der\, c\, (r_1 \cdot r_2) & \stackrel{\text{def}}{=} & \text{if } nullable(r_1) \\
& & \text{then } ((der\, c\, r_1) \cdot r_2) + (der\, c\, r_2) \\
& & \text{else } (der\, c\, r_1) \cdot r_2 \\
der\, c\, (r^*) & \stackrel{\text{def}}{=} & (der\, c\, r) \cdot (r^*)
\end{array}
$$

With our definition of regular expressions comes an induction principle. Given a property P over regular expressions. We can establish that $\forall r.\ P(r)$ holds, provided we can show the following:

1. $P(\mathbf{0})$, $P(\mathbf{1})$ and $P(c)$ all hold,
2. $P(r_1 + r_2)$ holds under the induction hypotheses that $P(r_1)$ and $P(r_2)$ hold,
3. $P(r_1 \cdot r_2)$ holds under the induction hypotheses that $P(r_1)$ and $P(r_2)$ hold, and
4. $P(r^*)$ holds under the induction hypothesis that $P(r)$ holds.

Let us try out an induction proof. Recall the definition

$$Der\, c\, A \stackrel{\text{def}}{=} \{s \mid c :: s \in A\}$$

whereby A is a set of strings. We like to prove

$$P(r) \stackrel{\text{def}}{=} \quad L(der\, c\, r) = Der\, c\, (L(r))$$

by induction over the regular expression r.

Proof
According to 1. above we need to prove $P(\mathbf{0})$, $P(\mathbf{1})$ and $P(d)$. Lets do this in turn.

- First Case: $P(\mathbf{0})$ is $L(der\,c\,\mathbf{0}) = Der\,c\,(L(\mathbf{0}))$ (a). We have $der\,c\,\mathbf{0} = \mathbf{0}$ and $L(\mathbf{0}) = \mathbf{0}$. We also have $Der\,c\,\mathbf{0} = \mathbf{0}$. Hence we have $\mathbf{0} = \mathbf{0}$ in (a).

- Second Case: $P(\mathbf{1})$ is $L(der\,c\,\mathbf{1}) = Der\,c\,(L(\mathbf{1}))$ (b). We have $der\,c\,\mathbf{1} = \mathbf{0}$, $L(\mathbf{0}) = \mathbf{0}$ and $L(\mathbf{1}) = \{\texttt{""}\}$. We also have $Der\,c\,\{\texttt{""}\} = \mathbf{0}$. Hence we have $\mathbf{0} = \mathbf{0}$ in (b).

- Third Case: $P(d)$ is $L(der\,c\,d) = Der\,c\,(L(d))$ (c). We need to treat the cases $d = c$ and $d \neq c$.

  $d = c$: We have $der\,c\,c = \mathbf{1}$ and $L(\mathbf{1}) = \{\texttt{""}\}$. We also have $L(c) = \{\texttt{"}c\texttt{"}\}$ and $Der\,c\,\{\texttt{"}c\texttt{"}\} = \{\texttt{""}\}$. Hence we have $\{\texttt{""}\} = \{\texttt{""}\}$ in (c).

  $d \neq c$: We have $der\,c\,d = \mathbf{0}$. We also have $Der\,c\,\{\texttt{"}d\texttt{"}\} = \mathbf{0}$. Hence we have $\mathbf{0} = \mathbf{0}$ in (c).

These were the easy base cases. Now come the inductive cases.

- Fourth Case: $P(r_1 + r_2)$ is $L(der\,c\,(r_1 + r_2)) = Der\,c\,(L(r_1 + r_2))$ (d). This is what we have to show. We can assume already:

  $$\begin{array}{ll} P(r_1): & L(der\,c\,r_1) = Der\,c\,(L(r_1)) \text{ (I)} \\ P(r_2): & L(der\,c\,r_2) = Der\,c\,(L(r_2)) \text{ (II)} \end{array}$$

  We have that $der\,c\,(r_1 + r_2) = (der\,c\,r_1) + (der\,c\,r_2)$ and also $L((der\,c\,r_1) + (der\,c\,r_2)) = L(der\,c\,r_1) \cup L(der\,c\,r_2)$. By (I) and (II) we know that the left-hand side is $Der\,c\,(L(r_1)) \cup Der\,c\,(L(r_2))$. You need to ponder a bit, but you should see that

  $$Der\,c(A \cup B) = (Der\,c\,A) \cup (Der\,c\,B)$$

  holds for every set of strings A and B. That means the right-hand side of (d) is also $Der\,c\,(L(r_1)) \cup Der\,c\,(L(r_2))$, because $L(r_1 + r_2) = L(r_1) \cup L(r_2)$. And we are done with the fourth case.

- Fifth Case: $P(r_1 \cdot r_2)$ is $L(der\,c\,(r_1 \cdot r_2)) = Der\,c\,(L(r_1 \cdot r_2))$ (e). We can assume already:

  $$\begin{array}{ll} P(r_1): & L(der\,c\,r_1) = Der\,c\,(L(r_1)) \text{ (I)} \\ P(r_2): & L(der\,c\,r_2) = Der\,c\,(L(r_2)) \text{ (II)} \end{array}$$

  Let us first consider the case where $nullable(r_1)$ holds. Then

  $$der\,c\,(r_1 \cdot r_2) = ((der\,c\,r_1) \cdot r_2) + (der\,c\,r_2).$$

The corresponding language of the right-hand side is

$$(L(der\,c\,r_1)\,@\,L(r_2)) \cup L(der\,c\,r_2).$$

By the induction hypotheses (I) and (II), this is equal to

$$(Der\,c\,(L(r_1))\,@\,L(r_2)) \cup (Der\,c\,(L(r_2)).\ \ (**)$$

We also know that $L(r_1 \cdot r_2) = L(r_1)\,@\,L(r_2)$. We have to know what $Der\,c\,(L(r_1)\,@\,L(r_2))$ is.

Let us analyse what $Der\,c\,(A\,@\,B)$ is for arbitrary sets of strings A and B. If A does *not* contain the empty string, then every string in $A\,@\,B$ is of the form $s_1\,@\,s_2$ where $s_1 \in A$ and $s_2 \in B$. So if s_1 starts with c then we just have to remove it. Consequently, $Der\,c\,(A\,@\,B) = (Der\,c\,(A))\,@\,B$. This case does not apply here though, because we already proved that if r_1 is nullable, then $L(r_1)$ contains the empty string. In this case, every string in $A\,@\,B$ is either of the form $s_1\,@\,s_2$, with $s_1 \in A$ and $s_2 \in B$, or s_3 with $s_3 \in B$. This means $Der\,c\,(A\,@\,B) = ((Der\,c\,(A))\,@\,B) \cup Der\,c\,B$. But this proves that (**) is $Der\,c\,(L(r_1)\,@\,L(r_2))$.

Similarly in the case where r_1 is *not* nullable.

- Sixth Case: $P(r^*)$ is $L(der\,c\,(r^*)) = Der\,c\,L(r^*)$. We can assume already:

  $$P(r): \quad L(der\,c\,r) = Der\,c\,(L(r))\ \text{(I)}$$

  We have $der\,c\,(r^*) = der\,c\,r \cdot r^*$. Which means $L(der\,c\,(r^*)) = L(der\,c\,r \cdot r^*)$ and further $L(der\,c\,r)\,@\,L(r^*)$. By induction hypothesis (I) we know that is equal to $(Der\,c\,L(r))\,@\,L(r^*)$. (*)

Let us now analyse $Der\,c\,L(r^*)$, which is equal to $Der\,c\,((L(r))^*)$. Now $(L(r))^*$ is defined as $\bigcup_{n\geq 0} L(r)^n$. We can write this as $L(r)^0 \cup \bigcup_{n\geq 1} L(r)^n$, where we just separated the first union and then let the "big-union" start from 1. Form this we can already infer

$$Der\,c\,(L(r^*)) = Der\,c\,(L(r)^0 \cup \textstyle\bigcup_{n\geq 1} L(r)^n) =$$
$$(Der\,c\,L(r)^0) \cup Der\,c\,(\textstyle\bigcup_{n\geq 1} L(r)^n)$$

The first union "disappears" since $Der\,c\,(L(r)^0) = \mathbf{0}$.