

Proof

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

$$\begin{array}{l|l}
 r & ::= \\
 \hline
 & \emptyset \\
 & \epsilon \\
 & c \\
 & r_1 \cdot r_2 \\
 & r_1 + r_2 \\
 & r^*
 \end{array}
 \qquad
 \begin{array}{l}
 L(\emptyset) \stackrel{\text{def}}{=} \emptyset \\
 L(\epsilon) \stackrel{\text{def}}{=} \{\epsilon\} \\
 L(c) \stackrel{\text{def}}{=} \{c\} \\
 L(r_1 \cdot r_2) \stackrel{\text{def}}{=} L(r_1) @ L(r_2) \\
 L(r_1 + r_2) \stackrel{\text{def}}{=} L(r_1) \cup L(r_2) \\
 L(r^*) \stackrel{\text{def}}{=} \bigcup_{n \geq 0} L(r)^n
 \end{array}$$

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

$$\begin{array}{l}
 \text{der } c(\emptyset) \stackrel{\text{def}}{=} \emptyset \\
 \text{der } c(\epsilon) \stackrel{\text{def}}{=} \emptyset \\
 \text{der } c(d) \stackrel{\text{def}}{=} \text{if } c = d \text{ then } \epsilon \text{ else } \emptyset \\
 \text{der } c(r_1 + r_2) \stackrel{\text{def}}{=} (\text{der } c r_1) + (\text{der } c r_2) \\
 \text{der } c(r_1 \cdot r_2) \stackrel{\text{def}}{=} \text{if } \text{nullable}(r_1) \\
 \qquad \qquad \qquad \text{then } ((\text{der } c r_1) \cdot r_2) + (\text{der } c r_2) \\
 \qquad \qquad \qquad \text{else } (\text{der } c r_1) \cdot r_2 \\
 \text{der } c(r^*) \stackrel{\text{def}}{=} (\text{der } c r) \cdot (r^*)
 \end{array}$$

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. $P(\emptyset)$, $P(\epsilon)$ 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

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

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

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

by induction over the regular expression r .

Proof

According to 1. above we need to prove $P(\emptyset)$, $P(\epsilon)$ and $P(d)$. Lets do this in turn.

- First Case: $P(\emptyset)$ is $L(\text{der } c \emptyset) = \text{Der } c(L(\emptyset))$ (a). We have $\text{der } c \emptyset = \emptyset$ and $L(\emptyset) = \emptyset$. We also have $\text{Der } c \emptyset = \emptyset$. Hence we have $\emptyset = \emptyset$ in (a).
- Second Case: $P(\epsilon)$ is $L(\text{der } c \epsilon) = \text{Der } c(L(\epsilon))$ (b). We have $\text{der } c \epsilon = \emptyset$, $L(\emptyset) = \emptyset$ and $L(\epsilon) = \{\epsilon\}$. We also have $\text{Der } c \{\epsilon\} = \emptyset$. Hence we have $\emptyset = \emptyset$ in (b).
- Third Case: $P(d)$ is $L(\text{der } c d) = \text{Der } c(L(d))$ (c). We need to treat the cases $d = c$ and $d \neq c$.

$d = c$: We have $\text{der } c c = \epsilon$ and $L(\epsilon) = \{\epsilon\}$. We also have $L(c) = \{c\}$ and $\text{Der } c \{c\} = \{\epsilon\}$. Hence we have $\{\epsilon\} = \{\epsilon\}$ in (c).

$d \neq c$: We have $\text{der } c d = \emptyset$. We also have $\text{Der } c \{d\} = \emptyset$. Hence we have $\emptyset = \emptyset$ in (c).
- Fourth Case: $P(r_1 + r_2)$ is $L(\text{der } c(r_1 + r_2)) = \text{Der } c(L(r_1 + r_2))$ (d). This is what we have to show. We can assume already:

$$P(r_1): \quad L(\text{der } c r_1) = \text{Der } c(L(r_1)) \quad \text{(I)}$$

$$P(r_2): \quad L(\text{der } c r_2) = \text{Der } c(L(r_2)) \quad \text{(II)}$$

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

$$\text{Der } c(A \cup B) = (\text{Der } c A) \cup (\text{Der } c B)$$

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

- Fifth Case:
- Sixth Case: