Structural induction and DFA union lecture summary

M. George

October 24, 2014

1 Proof of correctness of union construction

Last time we constructed a machine that we claimed recognized the union of two regular languages. That is, given $M_1 = (Q_1, \Sigma, \delta_1, F_1, q_{01})$ and $M_2 = (Q_2, \Sigma, \delta_2, F_2, q_{02})$, we constructed a machine M and claimed that $L(M) = L(M_1) \cup L(M_2)$. In this lecture, we proved that the construction is correct.

Here is the construction: $M = (Q_1 \times Q_2, \Sigma, \delta, F, (q_{01}, q_{02}))$ where

$$\begin{aligned} \delta((q_1, q_2), a) &= (\delta_1(q_1, a), \delta_2(q_2, a)) \\ F &= \{(q_1, q_2) \mid q_1 \in F_1 \text{ or } q_2 \in F_2\} \end{aligned}$$

To prove that this construction is correct, we first proved that the extended transition function behaves as expected.

Claim: for all $x \in \Sigma^*$, $q_1 \in Q_1$ and $q_2 \in Q_2$, $\hat{\delta}((q_1, q_2), x) = (\hat{\delta}_1(q_1, x), \hat{\delta}_2(q_2, x))$.

Proof: by induction on the structure of x. If $x = \epsilon$, then the left-hand side is just (q_1, q_2) (by definition of $\hat{\delta}$). Similarly, plugging in the definition of $\hat{\delta}$ into the right hand side also yields (q_1, q_2). So in the $x = \epsilon$ case the claim holds.

If $x = ya$, we can simplify the left-hand side:

$$\begin{aligned} \hat{\delta}((q_1, q_2), xa) &= \delta(\hat{\delta}((q_1, q_2), x), a) && \text{by definition of } \hat{\delta} \\ &= \delta((\hat{\delta_1}(q_1, x), \hat{\delta_2}(q_2, x)), a) && \text{by Inductive Hypothesis} \\ &= (\delta_1(\hat{\delta_1}(q_1, x), a), \delta_2(\hat{\delta_2}(q_2, x), a)) && \text{by definition of } \delta \\ &= (\hat{\delta_1}(q_1, xa), \hat{\delta_2}(q_2, xa)) && \text{by definition of } \hat{\delta_1} \text{ and } \hat{\delta_2} \end{aligned}$$

Which is equal to the right hand side. This concludes the inductive proof of the claim.

Using this fact and the definition of $L(M)$ we can prove that $L(M) = L(M_1) \cup L(M_2)$. We have

$$\begin{aligned}
L(M) &= \{x \in \Sigma^* \mid \hat{\delta}((q_{01}, q_{02}), x) \in F\} && \text{by definition} \\
&= \{x \in \Sigma^* \mid \hat{\delta_1}(q_{01}, x) \in F_1 \text{ or } \hat{\delta_2}(q_{02}, x) \in F_2\} && \text{by defintion of } F \text{ and the claim} \\
&= \{x \in \Sigma^* \mid \hat{\delta_1}(q_{01}, x) \in F_1\} \cup \{x \in \Sigma^* \mid \hat{\delta_2}(q_{02}, x) \in F_2\} \\
&= L(M_1) \cup L(M_2)
\end{aligned}$$

as claimed.

2 Regular expressions

We discussed another way of describing sets of strings, regular expressions. Regular expressions are a compact way of representing certain sets of strings.

The set of regular expressions is defined inductively:

- $\emptyset$ is a regular expression.
- ϵ is a regular expression.
- for all $a \in \Sigma$, a is a regular expression.
- if r_1 and r_2 are regular expressions, then $r_1 r_2$ is a regular expression (called the concatenation of r_1 and r_2).
- if r_1 and r_2 are regular expressions, then $r_1 | r_2$ is a regular expression (called the alternation of r_1 and r_2).
- if r is a regular expression, then so is r^*, called the Kleene closure of r.

Each regular expression has an associated *language* of strings that it *matches*, written as $L(r)$. The function L is defined inductively:

- $L(\emptyset) = \emptyset$
- $L(\epsilon) = \{\epsilon\}$
- $L(a) = \{a\}$
- $L(r_1 r_2) = \{xy \mid x \in L(r_1) \text{ and } y \in L(r_2)\}$
- $L(r_1 | r_2) = L(r_1) \cup L(r_2)$
- $L(r^*) = \{x_1 x_2 x_3 \cdots x_n \mid x_i \in L(r)\}$