Automata Theory :: Regular Languages

Jörg Endrullis

Vrije Universiteit Amsterdam

Alternative Descriptions of Regular Languages

Recall that:

The following statements are equivalent:

- The language L is regular.
- There is a DFA M with $L(M) = L$.
- There is an NFA M with $L(M) = L$.
- There is a right linear grammar G with $L(G) = L$.
- There is a left linear grammar G with $L(G) = L$.
- There is a regular expression r with $L(r) = L$.

Elementary Properties of Regular Languages

Elementary Properties of Regular Languages

Theorem

If L_1, L_2, L are regular languages, then also

$$L_1 \cup L_2 \qquad L_1 \cap L_2 \qquad L_1 L_2 \qquad \overline{L} \qquad L_1 \backslash L_2 \qquad L^* \qquad L^R$$

Proof.

Let r_1, r_2, r be regular expr. with $L(r_1) = L_1$, $L(r_2) = L_2$, $L(r) = L$.

- $L_1 \cup L_2 = L(r_1 + r_2)$ is regular.
- $L_1 L_2 = L(r_1 \cdot r_2)$ is regular.
- $L^* = L(r^*)$ is regular.
- L is accepted by some DFA $(Q, \Sigma, \delta, q_0, F)$.
  $\overline{L} = L((Q, \Sigma, \delta, q_0, Q \backslash F))$ is regular.
- $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ is regular. □
- $L_1 \backslash L_2 = L_1 \cap \overline{L_2}$ is regular.
- For L^R, take an NFA that accepts L. Reverse all arrows. Swap final and starting states. The result accepts L^R.

Decidable Properties of Regular Languages

Decidability of Emptyness

Roughly speaking, a property is **decidable** if there is an **algorithm/program** that can tell whether the property holds.

Convention

If we say that a **property of regular languages is decidable**, we implicitly assume that the language is **given as a DFA** (or a description that can be translated into a DFA by an algorithm).

Theorem

It is decidable whether a regular language L is empty.

Proof.

- Construct a DFA (or NFA) M with $L(M) = L$.
- Check if M has a path from starting state to a final state.
- If **yes**, then $L \neq \varnothing$. If **no**, then $L = \varnothing$. □

Decidability of Membership

Theorem

It is decidable if a word u is member of a regular language L.

Proof.

- Represent L in the form of a DFA M.
- Check if u is accepted by M. □

Practical difficulty: state-space explosion

The conversion to DFA might require an exponential number of states. (E.g. when L is given as NFA or regular expression.)

Solution

On-the-fly generation of DFA prevents state-space explosion. We only generate those states visited when reading u.

Decidability of Subsets

Theorem

It is decidable for regular languages L_1 and L_2 if $L_1 \subseteq L_2$.

Proof.

We have

$$L_1 \subseteq L_2 \quad \Longleftrightarrow \quad L_1 \setminus L_2 = \emptyset$$

The language $L_1 \setminus L_2$ is regular.

Finally, emptyness is decidable. □

Decidability of Equivalence

Theorem

It is decidable if two regular languages L_1 and L_2 are equal.

Proof.

We have

$$L_1 = L_2 \iff (L_1 \subseteq L_2) \wedge (L_2 \subseteq L_1)$$

Both problems on the right are decidable. □