Make-up Credit Assignment

1 Languages

▶ 1.1 Use set-builder notation and the set operations $\cup$, $\cap$, $\cdot$ (language concatenation) and $*$ (Kleene star) to construct the following languages (the first is given for you):

1. The language of zero-or-more `a`'s *or* one-or-more `b`'s.

$$\{\mathtt{a}^m \mid m \geq 0\} \cup \{\mathtt{b}^n \mid n \geq 1\}$$

2. The language of an even number of `a`'s where each string ends with a single `b`. E.g., `b`, `aab`, `aaaab`, ...

3. The language of zero-or-more `a`'s *followed by* twice as many `b`'s.

4. The language of strings of any number of `a`'s or `b`'s, or some mixture, in any order, such that the length of every string is *even* (including the zero-length string).

▶ 1.2 Which of the above languages are regular?

▶ 1.3 Consider all possible languages over $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ where the length of the longest string is 2. How many such languages are there? That is, how many L satisfy

$$\forall L \in \Sigma^*, s \in L\colon |s| = 2$$

▶ 1.4 Give examples of languages in the following classes, using set notation:

1. A finite language with $|L| = 2$.
2. A null language.
3. A nullable language.
4. An empty language.
5. A regular language.
6. A non-regular language.

▶ 1.5 Consider the recursive language definition:

$$L = \{\varepsilon\} \cup (\{\mathsf{a}, \mathsf{b}\} \cdot L)$$

What language does this define? Is it regular? If so, give a regular expression that generates the same language.

2 Regular languages

▶ 2.1 For each of the following strings and regular expressions, write Yes or No to indicate whether the string is or is not in the language generated by the RE.

1. $\mathsf{aabb} \in L(\mathsf{aa^*b^*})$
2. $\mathsf{aaa} \in L(\mathsf{a^*a^*})$
3. $\mathsf{aaa} \in L((\mathsf{aa})^*)$
4. $\mathsf{abab} \in L((\mathsf{a} + \mathsf{b})^*)$
5. $\mathsf{baba} \in L(\mathsf{a^*} + \mathsf{b^*})$

▶ 2.2 True or false: The number of *distinct* regular expressions that can generate the empty language is infinite.

▶ 2.3 Consider the following non-regular grammar, which generates a regular language. Translate this grammar into a regular one which generates the same language.

$$\begin{aligned} S &\to S\mathtt{a} \mid A \mid B\mathtt{b} \\ A &\to \mathtt{a}A\mathtt{a} \mid \varepsilon \\ B &\to B\mathtt{b} \mid \mathtt{b} \end{aligned}$$

▶ 2.4 Prove that your grammar generates the same language as the one given in 2.3. This will require you to show that every derivation possible in the original grammar has a corresponding, and equivalent, derivation in your grammar.

3 Machines

▶ 3.1 Construct the DFA machine which accepts strings over $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ which have an odd number of a's and an odd number of b's, in any order or mixture, but which end with a single c.

▶ 3.2 Use the node deletion algorithm on the machine from the previous problem to determine the regular expression corresponding to it.

▶ 3.3 Use ε transitions to build the NDA-ε corresponding to the regular expression from the previous problem.

▶ 3.4 Use the ε-removal transformation to build an ε-free NDA corresponding to the machine in the previous question.

▶ 3.5 Use the method shown in lecture and described in the notes ("parallelizing" the machine by running multiple instances) to convert the machine from the previous problem into DFA (fully deterministic). How does this compare to the machine you started with in problem 2.1?

4 Proofs

▶ 4.1 Prove by induction on $s \in \Sigma^*$ that

$$s \cdot \varepsilon = s$$

▶ 4.2 Prove by induction on $s \in \Sigma^*$ that

$$ss^R = (ss^R)^R$$

(That is, that all even-length palindromes are the same backwards and forwards.)

To do this, you will first need to prove three lemmas:

- Lemma 1: $(cs)^R = s^R c$ where $c \in \Sigma$, by induction on s.
- Lemma 2: $(uv)^R = v^R u$ (by induction on u). This is an extension of the previous lemma to strings instead of single characters.
- Lemma 3: $(u^R)^R = u$ (by induction on u).

▶ 4.3 Use the pumping lemma for regular languages to prove that the language

$$L = \{\mathtt{a}^n\mathtt{b}\mathtt{a}^n \mid n \geq 0\}$$

is not regular.

▶ 4.4 Is the language

$$L = \{\mathtt{a}^{2^n} \mid n \geq 0\}$$

(consisting of a power-of-2 number of a's) regular? Why or why not?