Institut for Matematik og Datalogi
Syddansk Universitet

January 30, 2024
JBJ

DM553 (MM850) – Spring 2024 – Weekly Note 2

Approximate dates for the 3 sets of handin assignments. They may change a few days in either direction.

- Exam problems 1 (up to 3 people in a group): February 16 to March 11.
- Exam problems 2 (must be done alone): March 21 to April 11.
- Exam problems 3 (up to 3 people in a group): April 22 to May 8.

Material covered in Week 5:
We covered Sipser Chapter 1.1-1.4 on regular languages. The first 3 sections are known to almost all of you from DM565. Videos 1 and 2 cover the material.

Key points:

- Automata are mathematical models of computers.
- Finite automata (FA) use a constant amount of memory and determine whether a given string is in a certain language. The moves of a deterministic finite automaton (DFA) are completely determined by its initial state and the input string. A DFA M accepts a string w if the (**unique!**) walk that starts in the initial state and "spells" w ends in an accepting state of M.
- On the contrary, a nondeterministic finite automaton (NFA) can choose between several alternative moves, and it has the ability to guess the moves that lead to a favorable state. An NFA M accepts a string w if there exists a walk from the start state to one of the accepting (final states) which "spells" exactly w.
- However, NFA's do not have greater computational power than DFA's, as we can convert an arbitrary NFA M to an equivalent DFA M' (with $L(M') = L(M)$). Note that this conversion is **NOT a polynomial algorithm** since M' may have exponentially many states compared to M!
- A regular expression (RE) is a formal specification of a regular language.
- A language is accepted by some FA if and only if it is generated by some regular expresssion, since an equivalent FA can be constructed from a given RE and vice versa.
- The class of regular languages is closed under concatenation, complement, union, star, and intersection. The following holds:

  Let L, L' be regular languages, then each of the following are also regular:

  1. $L \cup L'$.
  2. The complement $\overline{L}$ of L.
  3. $L \cap L'$.
  4. $L \setminus L'$.
  5. LL'.
  6. L^*

To see that $L \cap L'$ is regular, it suffices to observe that $L \cap L' = \overline{\overline{L} \cup \overline{L'}}$.

- **Every finite language is regular**. To see this, just note that we can easily make an NFA $M(w)$ which accepts precisely the string w (how?) so if L consists of the strings $w_1, w_2, \ldots, w_k$ for some finite integer k, then make the NFAs $M(w_i)$, $i = 1, 2, \ldots, k$ and build (in $k-1$ union steps) an NFA whose language is precisely L. So every non-regular language contains arbitrarily long strings. In particular, it is infinite.

- Applying the pumping lemma is like a 2 person game between you and an adversary: In order to prove that an infinite language L is NOT regular you proceed as follows:

  – Assume (to reach a contradiction) that $L = L(M)$ for some DFA M and let p be the number of states of M. You may also think of getting p from **the adversary** (who claims that M exists).

  – **You** choose a suitable string $s \in L$ such that $|s| \geq p$.

  – Now **the adversary** must choose strings x, y, z over Σ such that

    1. $s = xyz$
    2. $xy^iz \in L$ for **all** $i \geq 0$
    3. $|y| > 0$ and $|xy| \leq p$.

    By the pumping Lemma (s)he can do so if L is indeed regular.

  – Then **you** choose a suitable $i \geq 0$ and show that $xy^iz \notin L$, contradiction.

  – Since the contradiction arose from the assumption that L was regular, it follows that L is not a regular language.

New material in Week 6

- Context free grammars. Section 2.1. Most of this is known from DM565 (Video 3).

- Pushdown automata. First part of Section 2.2 (Video 4).

- Non-context-free languages, Section 2.3 (Video 5)

- Equivalence between languages recognized by PDAs and context-free languages. Sipser pages 117-124. (Video 6)

Exercises in Week 6:

Please try to solve all of these **before** the exercises class with Magnus. Some of these are repetition of material you learned in DM565.
For those exercises with many subquestions, the instructor will select a subset of these to discuss with you.

- Solve the following problem: :

  A man is travelling with a wolf (w) and a goat (g). He also brings along a nice big cabbage (c). He encounters a small river which he must cross to continue his travel. Fortunately, there is a small boat at the shore which he can use. However, the boat is so small that the man cannot bring more than himself and exactly one more item along (from $\{w, g, c\}$). The man knows that if left alone with the goat, the wolf will surely eat it and the goat if left alone with the cabbage will also surely eat that. The man's task is hence to device a transportation scheme in which, at any time, at most one item from $\{w, g, c\}$ is in the boat and the result is that they all crossed the river and can continue unharmed.

(a) Describe a solution to the problem which satisfies the rules of the "game". You may use your answer to (b) to find a solution.

(b) Consider strings over the alphabet $\Sigma = \{m, w, g, c\}$ and interpret these as follows: The symbol m means that the man crosses the river alone, w means that he brings the wolf etc.

Design a finite automaton which accepts precisely those strings over Σ which correspond to a transportation sequence where everybody survives and is legal in the sense that the man can only bring an item (e.g. w) back across the river if it was actually on the shore where the boat just left from. For example $gmcg$ is a legal string (it is not a solution) whereas gc is not legal.

- Sipser 1.9, 1.11 and 1.12 page 85.
- Sipser 1.31
- 1.29(b) and 1.30 on page 88.
- 1.36 and 1.43 on page 89.
- 1.49 page 90.