Computer Science 220S2C (2009)

Automata and Pattern Matching: Recap

Main points

1. Strings, languages operations with strings (concatenation) and languages (complement, union, intersection, Kleene star, concatenation, power).

2. DFAs: definition by components and by diagram, examples, going from components to diagram and conversely, definition of a computation (trace), showing that a string is accepted by a DFA, showing that a string is rejected by a DFA, constructing a DFA accepting a given language, determining the language accepted by a DFA, proving that a language is not accepted by any DFA.

3. NFAs: definition by components and by diagram, examples, going from components to diagram and conversely, definition of a computation (trace), showing that a string is accepted by an NFA, showing that a string is rejected by an NFA, constructing an NFA accepting a given language, determining the language accepted by an NFA.

4. Constructing a DFA equivalent with a given NFA.

5. The following problems are algorithmically decidable:

    - a DFA M accepts the empty string
    - a DFA M accepts a string w
    - a DFA M accepts no string
    - a DFA M accepts only finitely many strings
    - a DFA M accepts infinitely many strings
    - an NFA N accepts infinitely many strings
    - two DFAs accept the same language
    - a DFA M accepts the complement of the language accepted by a DFA M'
    - a DFA M accepts the same language as an NFA N
    - a DFA M accepts only one string w.

6. The class of regular languages is closed under

    - complement
    - mirror (reverse)
    - union
    - intersection
    - Kleene star
    - concatenation
    - power

7. Regular expressions denote regular languages. Given a DFA M find a regular expression denoting the language accepted by M. Given a regular expression α find a DFA accepting the language denoted by α.

8. Finding the minimal DFA equivalent with a given DFA (NFA).

9. Pattern matching.

10. Revise all examples in lecture notes.

Sample exercises[1]

1. All examples in the textbook from pages 165–193.
2. All exercises in the textbook from pages 165–193.
3. Build DFAs for the following languages:
   - $\emptyset$,
   - $\{\varepsilon\}$,
   - $\{a^n b^m c^k \mid n \geq 0, m, k \geq 1\}$,
   - $\{1(01)^n \mid n \geq 0\}$,
   - $\{w \in \{a, b\}^* \mid w \neq \varepsilon\}$
4. Devise a general procedure that, given a DFA M, produces an equivalent DFA $M^{'}$ in which the start state, once left, cannot be re-entered.
5. Show that the language $A(w) = \{uwv \mid u, v \in \{a, b\}^*\}$ is regular for each string w.
6. Show that the language $\{a^n b^n c^n \mid n \geq 1\}$ is not accepted by any DFA.
7. Build NFAs for the following languages:
   - $\{w \in \{0, 1\}^* \mid w$ contains any of the substrings $010, 011$ or $1100\}$,
   - $\{w \in \{0, 1\}^* \mid w$ contains the substrings $010, 011$ and $1100\}$,
   - $\{w \in \{0, 1\}^* \mid w$ has a 0 in the third place$\}$,
   - $\{w \in \{0, 1\}^* \mid w$ has a 0 in the third place from the end$\}$,
   - $\{w \in \{a, b\}^* \mid |w| > 2\}$.
8. Given two DFAs M_1 and M_2, construct an NFA N such that $L(N) = L(M_1) \setminus L(M_2)$.
9. Given two DFAs M_1 and M_2, construct an NFA N such that $L(N) = L(M_1) \cup \overline{L(M_2)}$.
10. Using the equivalence between NFA and DFA, convert the following NFAs into equivalent DFAs:
    (a) Every NFA discussed in lecture notes.
    (b) $Q = \{q_1, q_2\}, \delta(q_1, a) = \{q_1, q_2\}, \delta(q_1, b) = \{q_2\}, \delta(q_2, a) = \emptyset, \delta(q_2, b) = \{q_1\}, S = F = \{q_1\}$.
    (c) $Q = \{q_1, q_2, q_3\}, \delta(q_1, a) = \{q_3\}, \delta(q_1, b) = \emptyset, \delta(q_2, a) = \emptyset, \delta(q_2, b) = \{q_1\}$, $\delta(q_3, a) = \emptyset, \delta(q_3, b) = \{q_3\}, S = F = \{q_1\}$.
11. Construct regular expressions denoting the languages accepted by each DFA/NFA discussed in lecture notes.
12. Minimise each DFA/NFA discussed in lecture notes. (For NFA, convert first to DFA, then minimise).
13. Design the Aho-Corasick automaton for a given simple pattern.
14. In regex the question mark $a?$ indicates there is zero or one of a. For example, $colou?r$ matches both "color" and "colour". Write a definition of $a?$ in terms of Kleene regular operations and apply it to $colou?r$.
15. What is the language denoted by the Kleene regular expression $(a|b)^*$?
16. What is the language denoted by the Kleene regular expression $ab^*(c|\varepsilon)$?
17. Write a Kleene regular expression (or NFA or DFA) for the set of all correct email addresses of the form `user@ec.auckland.ac.nz`, where `user` is a string on the alphabet of lower case letter and digits $\{a, b, \ldots, z\} \cup \{0, 1, \ldots, 9\}$ that starts with a letter, is followed by at least three letters and exactly three digits.
18. Write a Kleene regular expression (or NFA or DFA) for a simplified form of DNS (Domain Name Service) names: one or more "words", where each "word" is an arbitrary sequence of letters, digits, underscores, such as: `ec`
    `ec.auckland.ac.nz`
    `130.216.11.242`
    `Alfa_1.Beta_2` where the letters and digits are from the English ANSI alphabet $\{a, b, \ldots, z\} \cup \{0, 1, \ldots, 9\}$.

[1]Exercises in exam may be similar, but not identical.