SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

TME1. Introduction to frontend and backend aspects

BM. Bui-Xuan et A. Escriou

Exercise 1.

We plan to develop a search engine application on the text of books made available to the community by The Gutenberg Project. A basic wireframe schematic is given below.

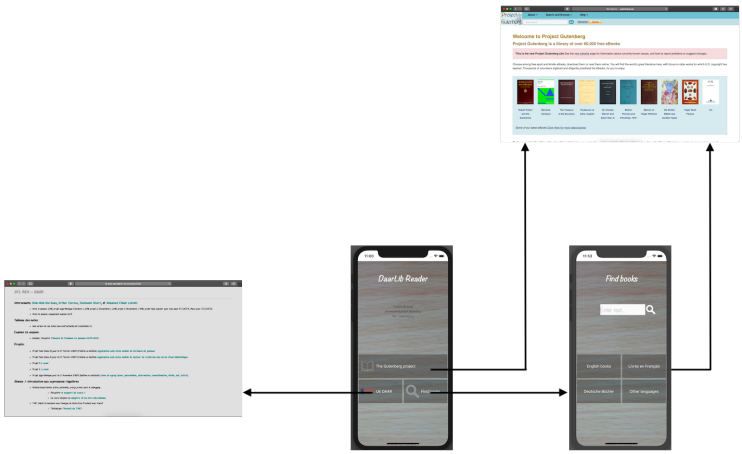

Figure 1: Two principal views of a search app for library books.

Implement the first view of this search app, by extending the VueJS code given at

- https://gutenberg-daar-2020.netlify.app.

Exercise 2.

We would like to prepare the backend of this app, starting with a search feature by RegEx matching. The algorithmic method is that of the book Aho-Ullman, chapter 10 pages 571 - 582, available at

- http://infolab.stanford.edu/~ullman/focs.html

For all following questions, your implementation must be confronted to a concrete example. One suggestion is

- RegEx: `"S(a|g|r)+on"` or `"S[a-z]+on"`.
- input text file: Book about Babylone.

- N.B.: we restrict ourselves to the following RegEx specifications: parenthesis, alternation, concatenation, star, dot, ASCII letter.

Question 2.1. Read *Extended Regular Expressions* specification available at

- http://pubs.opengroup.org/onlinepubs/7908799/xbd/re.html

Implement the parsing step in order to transform a RegEx string into a RegEx tree. An example of what is expected is given in Figure 10.29 of the Aho-Ullman book. We can be inspired by these lines of Java code.

Question 2.2. Implement the transformation of the RegEx tree into an automaton with epsilon transition. An example of what is expected is given in Figure 10.27 and 10.28 of the Aho-Ullman book.

Question 2.3. Implement the determinisation of the automaton previously obtained, using subset construction. A description of what is expected is given in Aho-Ullman book, pages 547 – 552.

Question 2.4. Implement a minimization of the previously obtained automaton. A naive method is given in Aho-Ullman book, page 555.