German University in Cairo
Department of Computer Science
Assoc. Prof. Haythem O. Ismail

CSEN1002 Compilers Lab, Spring Term 2023
Task 3: Fallback Deterministic Finite Automata

Due: Week starting 11.03.2023

1 Objective

For this task you need to implement a fallback deterministic finite automaton with actions (FDFA) abstract data type. Recall that an FDFA is a sextuple $(Q, \Sigma, \delta, q_0, F, \mathcal{A})$: Q is a non-empty, finite set of states; Σ is non-empty, finite set of symbols (an alphabet); $\delta : Q \times \Sigma \longrightarrow Q$ is the transition function; $q_0 \in Q$ is the start state; $F \subseteq Q$ is the set of accept states; and $\mathcal{A}$ is function that maps every state in Q to an action. Refer to the slides of Lecture 2 of CSEN1003 for more details about the operation of FDFA.

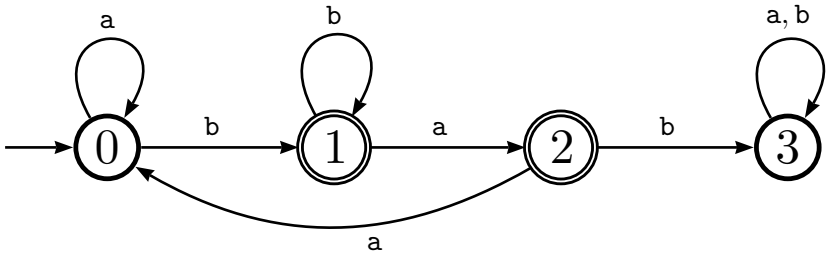
2 Requirements

- We make the following assumptions about FDFA for simplicity.
  a) The set of states Q is always of the form $\{0, \ldots, n\}$, for some $n \in \mathbb{N}$.
  b) The alphabet Σ is always a subset of the Latin alphabet, not including `e`.
  c) $q_0 \notin F$.
  d) $\mathcal{A}(q)$ is the action which appends the token "lex`,`q" to a list, where lex is as indicated in Lecture 2 of CSEN1003, and q is the state name.
- You should implement a class constructor `FallbackDfa` and a method `run`.
- `FallbackDfa`, a class constructor, takes one parameter which is a string description of an FDFA and constructs an FDFA instance as per the description. A string describing an FDFA is of the form $Q\#A\#T\#I\#F$.
  - Q is a string representation of the set of states; a semicolon-separated sequence of sorted integer literals.
  - A is a string representation of the input alphabet; a semicolon-separated sequence of alphabetically sorted symbols
  - T is a string representation of the transition function. T is a semicolon-separated sequence of triples. Each triple is a string representing a single transition; a comma-separated sequence i, a, j where i is a state of Q, a a symbol of A, and j a state of Q representing a transition from i to j on input a. These triples are sorted by the source state i and then by the input a.
  - I is an integer literal representing the initial state.
  - F is a string representation of the set of accept states; a semicolon-separated sequence of sorted integer literals.

– Note that the function $\mathcal{A}$ is not encoded in the string representation since it is fixed for all FDFA as indicated in the simplifying assumptions above.

– For example, the following string represents the FDFA whose state diagram appears in the figure below.

`0;1;2;3#a,b#0,a,0;0,b,1;1,a,2;1,b,1;2,a,0;2,b,3;3,a,3;3,b,3#0#1;2`



- `run` simulates the operation of the constructed FDFA on a given binary string, and returns a semicolon-separated sequence of tokens. For example, running the above FDFA on the string `baababb` produces the output `baaba`, `2`; `bb`, `1`.
- Important Details:
  - Your implementation should be done within the template file "`FallbackDfa.java`" (uploaded to the CMS).
  - You are not allowed to change package, file, constructor, or method names/signatures.
  - You are allowed to implement as many helper classes/methods within the same file (if needed).
  - Public test cases have been provided on the CMS for you to test your implementation.
  - Please ensure that the public test cases run correctly without modification before coming to the lab to maintain a smooth evaluation process.
  - Private test cases will be uploaded before your session and will have the same structure as the public test cases.

3 Evaluation

- Your implementation will be tested by constructing two FDFAs and running each on five strings.
- You get one point for each correct output of `run`; hence, a maximum of ten points.
- The evaluation will take place during your lab session of the week starting Saturday March 11.

4 Online Submission

- You should submit your code at the following link.

`https://forms.gle/8ZGyx4pBdz4GLfhTA`

- Submit one Java file (`FallbackDfa.java`) containing executable code.
- **Online submission is due by the end of your lab session.**