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

CSEN1002 Compilers Lab, Spring Term 2020
Task 3: Fallback DFA

Due: Week starting 22.02.2020

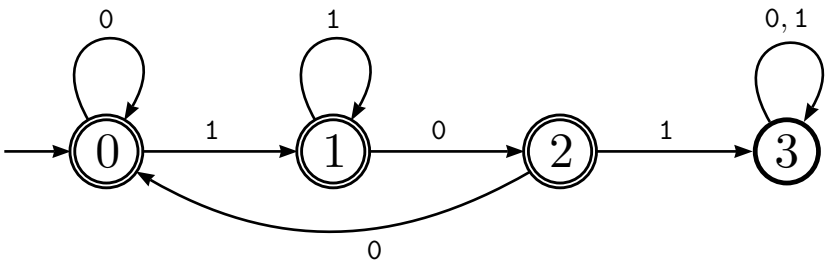
1 Objective

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

- You may use the programming language of your choice.
- We make the following assumptions for simplicity.
  a) The alphabet Σ is always the binary alphabet $\{\texttt{0}, \texttt{1}\}$.
  b) The set of states Q is always of the form $\{0, \ldots, n\}$, for some $n \in \mathbb{N}$.
  c) The start state is always state 0.
  d) $\mathcal{A}$ maps each state to a binary string; the action is to print this string.
- You should implement two functions: `fdfa` and `run`.
- `fdfa` (which could be a class constructor) takes one parameter which is a string description of an FDFA and returns (or constructs) an FDFA instance as per the description.
- A string describing an FDFA is of the form $P\#S$, where P is a prefix representing both the transition function δ and the action function $\mathcal{A}$ and S is a suffix representing the set F of accept state.
- P is a semicolon-separated sequence of quadruples. Each quadruple is a comma-separated sequence of items; the first three items are states and the fourth is a binary string. A quadruple i, j, k, s means that $\delta(i, \texttt{0}) = j$, $\delta(i, \texttt{1}) = k$, and $\mathcal{A}(i) = s$.
- S is a comma-separated sequence of states.
- For example, consider the FDFA for which the state diagram appears below. Suppose that, for state i, $\mathcal{A}(i)$ is the two-bit binary representation of i. Thus, such an FDFA may have the following string representation.

```
0,0,1,00;1,2,1,01;2,0,3,10;3,3,3,11#0,1,2
```



- `run` simulates the operation of the constructed FDFA on a given binary string. For example, running the above FDFA on the string `1011100` produces the output `1000`.

3 Evaluation

- Your implementation will be tested by constructing two FDFA and running each on five strings.
- You get one point for each correct output of `run`; hence, a maximum of ten points.