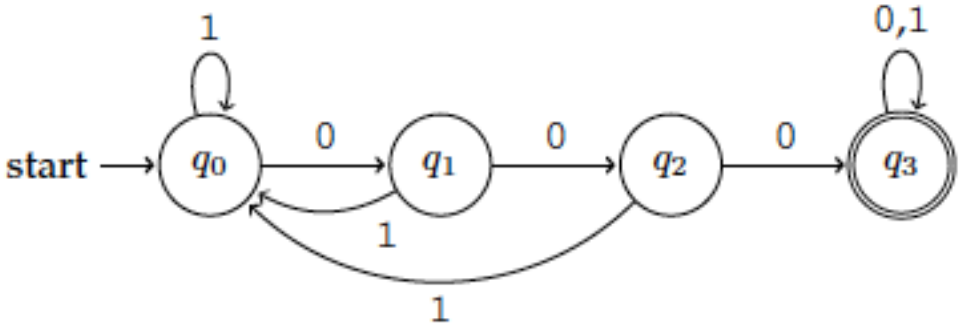
1. Implementazione di un DFA in Java



Esercizio 1.7

- Progettare e implementare un DFA con alfabeto {a, b} che riconosca il linguaggio delle stringhe tali che a occorre almeno una volta in una delle **prime tre** posizioni della stringa.
- Il DFA deve accettare anche stringhe che contengono meno di tre simboli (ma almeno uno dei simboli deve essere a).
- Ad esempio, il DFA deve accettare le stringhe:
  - "abb", "abbbbbb", "bbaba", "baaaaaaa", "aaaaaaa", "a", "ba", "bba", "aa" e "bbabbbbbbbb"

    ma non
  - "bbbababab" oppure "b".

Esercizio 1.8

- Progettare e implementare un DFA con alfabeto $\{a, b\}$ che riconosca il linguaggio delle stringhe tali che a occorre almeno una volta in una delle **ultime tre** posizioni della stringa.
- Come nell'esercizio 1.7, il DFA deve accettare anche stringhe che contengono meno di tre simboli (ma almeno uno dei simboli deve essere a).
- Ad esempio, il DFA
  - Deve accettare le stringhe: "abb", "bbaba", "baaaaaaa", "aaaaaaa", "a", "ba", "bba", "aa" e "bbbababab"

  - Non deve accettare le stringhe "abbbbbb", "bbabbbbbbbb" oppure "b".

Esercizio 1.9

- Progettare e implementare un DFA che riconosca il linguaggio di stringhe che contengono il tuo nome e tutte le stringhe ottenute dopo la sostituzione di un carattere del nome con un altro qualsiasi.
- Ad esempio, nel caso di uno studente che si chiama Paolo, il DFA
  - deve accettare la stringa “Paolo” (cioe` il nome scritto correttamente), ma anche le stringhe “Pjolo”, “caolo”, “Pa%lo”, “Paola” e “Parlo”
    (il nome dopo la sostituzione di un carattere)

  - Non deve accettare
    “Eva”, “Perro”, “Pietro” oppure “P*o*o”.

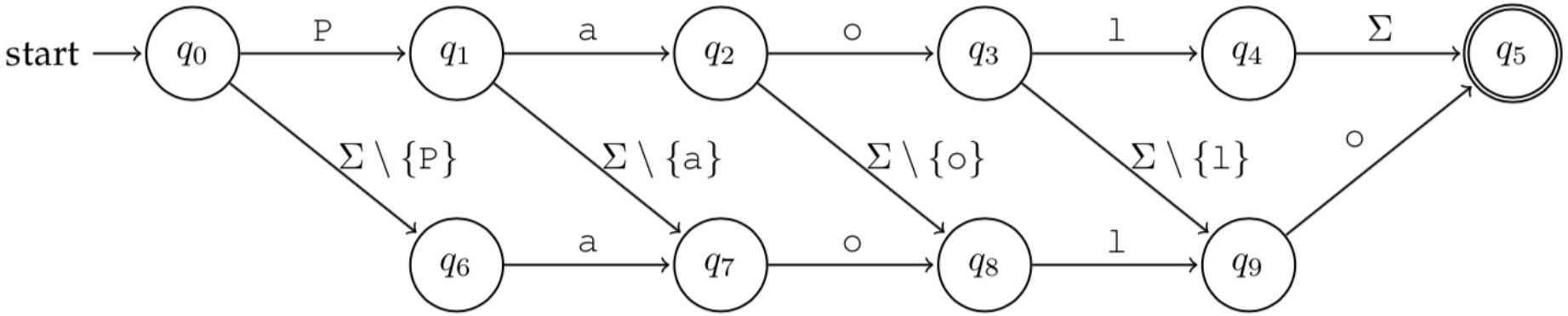


Figura 8: Esercizio 1.9.

Esercizio 1.10

- Progettare e implementare un DFA definito sull'alfabeto {/ ,*,a} che riconosca il linguaggio di "commenti" delimitati da /* (all'inizio) e */ (alla fine)
- L'automa deve accettare stringhe sull'alfabeto che
  - contengono almeno 4 caratteri
  - che iniziano con /*, che finiscono con */
  - e che contengono una sola occorrenza della sequenza */, quella finale (dove l'asterisco della sequenza */ non deve essere in comune con quello della sequenza /* all'inizio,)
- Esempio: l'automa
  - deve accettare le stringhe:
    "/****/", "/*a*a*/", "/*a/**/", "/**a///a/a**/", "/**/" e "/*/*/"

  - non deve accettare le stringhe:
    "/*/", oppure "/**/***/".

Esercizio 1.11

- Modificare l'automa dell'esercizio precedente in modo che riconosca il linguaggio di stringhe (sull'alfabeto {/, *, a}) che contengono "commenti" delimitati da /* e */ , ma con la possibilità di avere stringhe prima e dopo come specificato qui di seguito.
- L'idea è che sia possibile avere eventualmente commenti (anche multipli) in una sequenza di simboli dell'alfabeto. Quindi l'unico vincolo è che l'automa deve accettare le stringhe in cui un'occorrenza della sequenza /* deve essere seguita (anche non immediatamente) da un'occorrenza della sequenza */ .
- Le stringhe del linguaggio possono non avere nessuna occorrenza della sequenza /* (caso della sequenza di simboli senza commenti).
- Ad esempio, il DFA
  - deve accettare le stringhe:
    "aaa/****/aa ", "aa/*a*a*/ ",
    "aaaa ", "///"
    "/****/ ", "/*aa*/ ", "*/a ", "a/**/***a ", "a/**/***/a " e "a/**/aa/***/a "

  - ma non deve accettare
    "aaa/*/aa " oppure "aa/*aa ". Implementare l'automa seguendo la costruzione vista in Figura 2.