

Esercizio 1. Laboratorio di Ingegneria del Software AA 2023/2024

Definizione di Word Automata

Un **word automata** è un tipo specifico di automa a stati finiti definito come segue:

$$A = (Q, q_0, F, \Sigma, \delta)$$

dove:

- Q rappresenta l'insieme degli stati,
- $q_0 \in Q$ è lo stato iniziale,
- $F \subseteq Q$ indica l'insieme degli stati finali o accettanti,
- Σ è l'alfabeto di input,
- $\delta : Q \times \Sigma^+ \rightarrow Q$ è una funzione che mappa parole distinte su Σ in uno stato di A .

Nota: la funzione di transizione δ considera sequenze non vuote di simboli dall'alfabeto di input (Σ^+), mappando queste sequenze in uno stato dell'automata. Un word automata accetta una parola $w = \sigma_1 \dots \sigma_n$, dove $\sigma_i \in \Sigma$ per ogni $1 \leq i \leq n$, se e solo se esistono $k \geq 1$ parole w_1, \dots, w_k su Σ^+ tali che:

$$w = w_1 \dots w_k$$

e

$$\delta(\dots \delta(\delta(q_0, w_1), w_2) \dots, w_k) \in F$$

Nella figura è presentato un esempio di automa insieme a tre esempi di computazione. Nel primo caso, la parola è accettata perché viene consumata interamente e termina in uno stato finale. Nel secondo caso, benché la parola sia consumata interamente, essa termina in uno stato non finale, quindi viene rifiutata. Nel terzo caso, la parola viene processata fino a raggiungere lo stato q_3 ; tuttavia, il suo suffisso ab non permette nessuna transizione uscente da q_3 , pertanto la parola viene rifiutata.

Consegna

Obiettivo: Sviluppare un'applicazione con interfaccia grafica che permetta agli utenti di disegnare e testare automi a stati finiti (word automata) per alfabeti arbitrari.

Descrizione:

L'applicazione dovrà offrire un'interfaccia utente grafica (GUI) che consenta la creazione di un word automa. Gli utenti devono poter definire stati, transizioni e segnare gli stati iniziali e finali. Una volta che l'automata è stato definito, se questo rispetta i criteri di correttezza, l'utente dovrà avere la possibilità di testare il funzionamento dell'automata inserendo una parola tramite un campo di testo. Premendo un pulsante dedicato, l'applicazione mostrerà il processo di accettazione o rifiuto della parola, evidenziando il percorso degli stati attraversati.

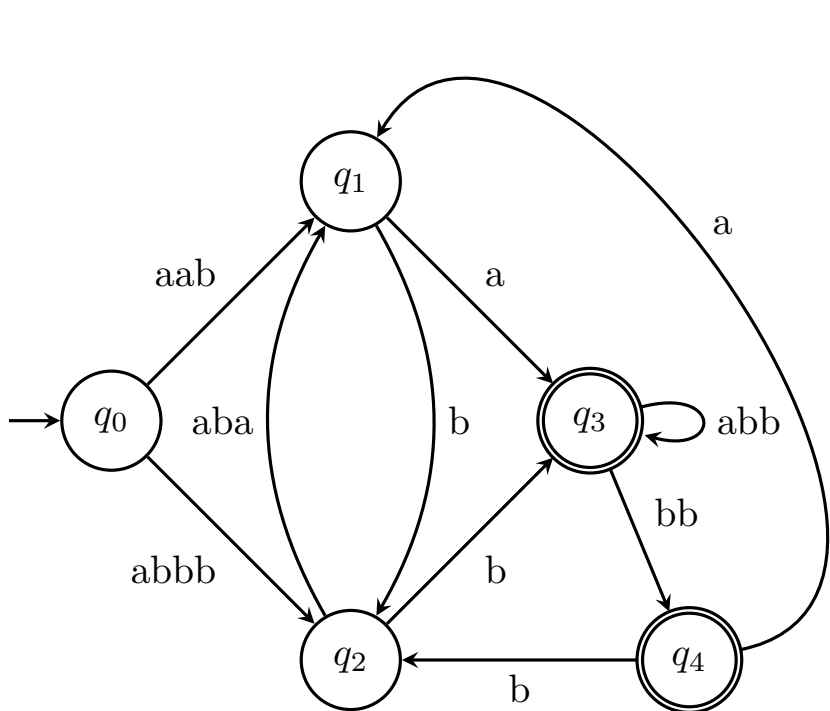
Requisiti di Correttezza dell'Automa:

L'applicazione deve implementare un controllo di validità sull'automata disegnato prima di permettere il test delle parole. Gli automi validi devono rispettare i seguenti criteri:

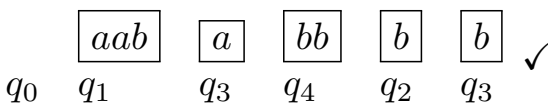
1. **Stato Iniziale Unico:** L'automata deve avere uno e un solo stato iniziale.
2. **Transizioni Non Vuote:** Ogni transizione che esce da uno stato deve essere etichettata con una parola non vuota.
3. **Determinismo:** Per ogni stato, le parole etichettate su coppie di transizioni uscenti devono essere diverse. In altre parole, da uno stato non possono partire due transizioni con la stessa etichetta. **Importante:** se da uno stato escono due transizioni con associate due parole w e w' tali che w è un prefisso proprio di w' (i.e., $w = aa$ e $w' = aab$) e nella computazione corrente è possibile eseguire sia la transizione di w che quella di w' , allora si prende sempre la transizione "attivabile" con la parola più lunga associata, w' nel nostro caso. Tale condizione è fondamentale per garantire che la computazione sia deterministica.

Funzionalità:

- **Disegno dell'automata:** l'utente può aggiungere stati, transizioni e marcare stati iniziali e finali.
- **Inserimento e test di parole:** tramite un campo di testo, l'utente può inserire una parola per testare se viene accettata dall'automata.
- **Visualizzazione del processo:** su inserimento di una parola valida, l'applicazione mostrerà il processo di accettazione o rifiuto, evidenziando il cammino seguito all'interno dell'automata in maniera analoga a quanto mostrato in figura.

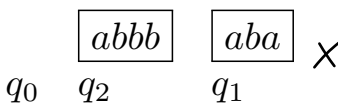


aababbbb



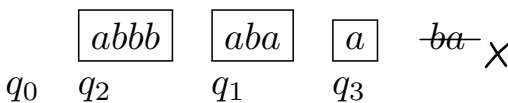
Caso 1

abbbaba



Caso 2

abbbabaaba



Caso 3