Automi

a.a. 2020-2021

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma "Tor Vergata"

Prof. Giorgio Gambosi

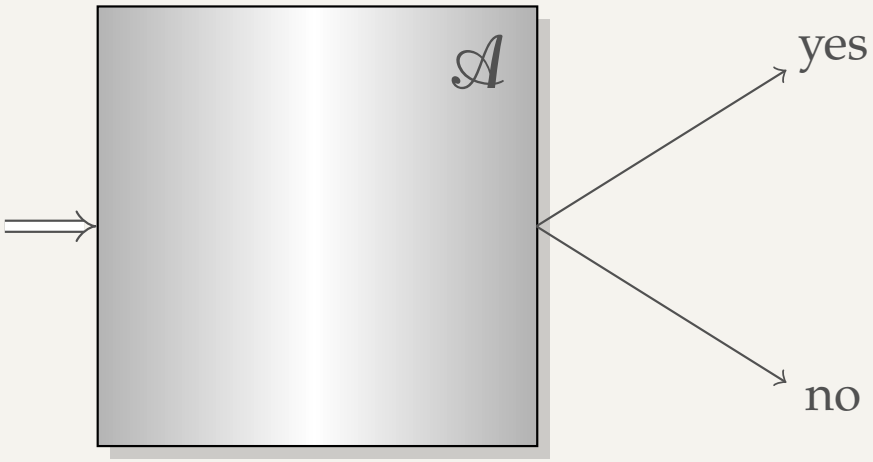
Automa

Dispositivo astratto che, data un stringa x fornitagli in input, esegue una computazione

Se la computazione termina, restituisce, secondo una qualche modalità, un valore

Nel caso del problema del riconoscimento, il valore restituito è booleano

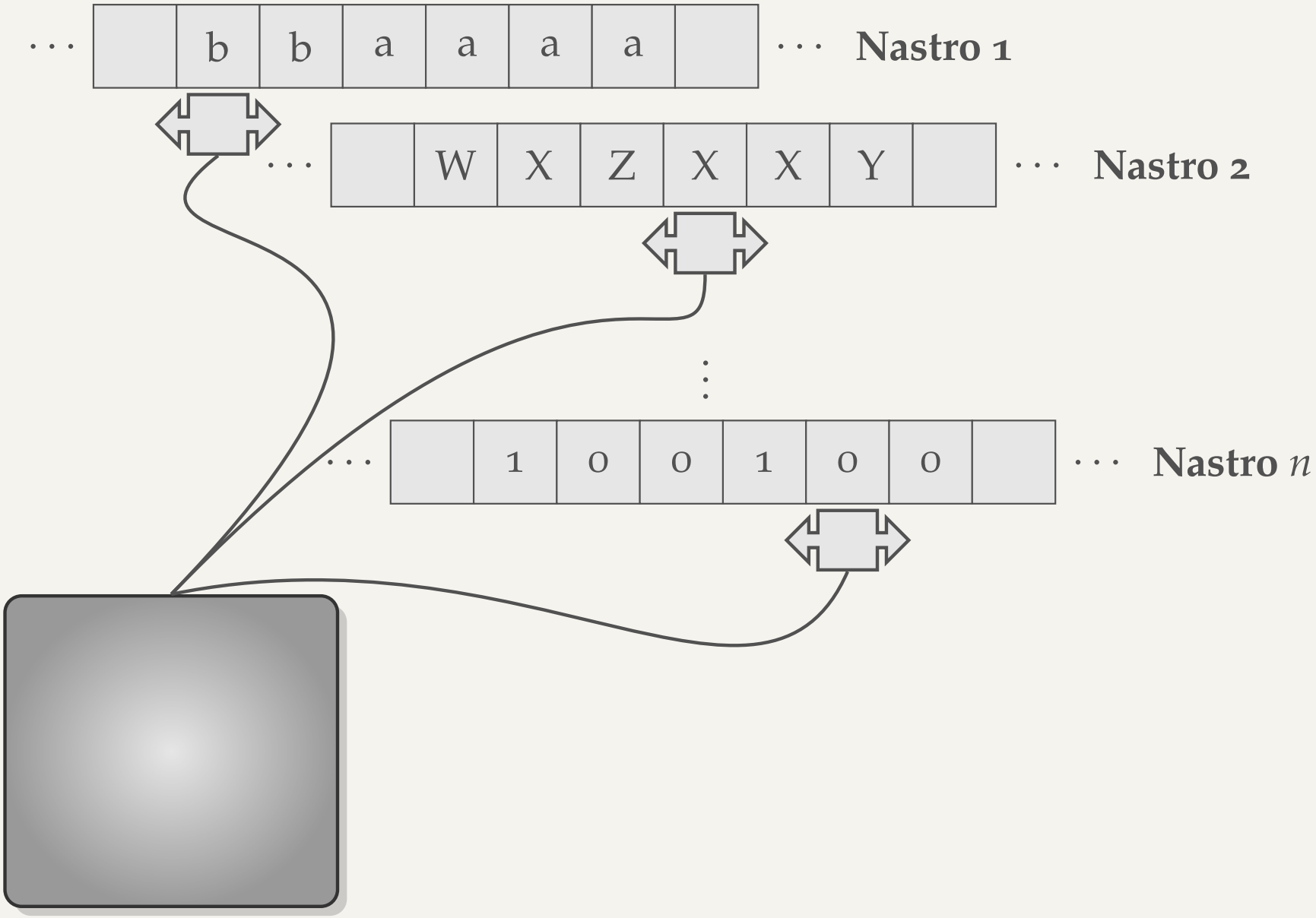
Automa



Automa

- dispositivo interno, che ad ogni istante assume uno stato in un possibile insieme finito predefinito
- uno o più dispositivi di memoria (nastri), sui quali è possibile memorizzare delle informazioni, sotto forma di stringhe di caratteri da alfabeti predefiniti
- nastri costituiti da celle; ogni cella può contenere un carattere
- caratteri letti o scritti per mezzo di testine che possono muoversi lungo i nastri, posizionandosi sulle diverse celle

Automa



Configurazione di un automa

Insieme delle informazioni necessarie e sufficienti per determinare, in un certo istante, il comportamento futuro dell'automa.

1. stato interno dell'automa;
2. contenuto di tutti i nastri di memoria;
3. posizione di tutte le testine sui nastri.

Configurazione iniziale

Configurazione in cui si assume si trovi inizialmente un automa, in presenza di una stringa in input.

Ad esempio:

1. stato predefinito come iniziale;
2. nastri di memoria vuoti eccetto il nastro di input, contenente la stringa;
3. testina del nastro di input sulla cella contenente il primo carattere della stringa

Funzione di transizione

Induce una relazione di transizione tra configurazioni, che associa ad una configurazione un'altra (o più di una) configurazione successiva.

Definita non sull'insieme delle possibili configurazioni, ma su domini e codomini che rappresentano parti di configurazioni, quelle che effettivamente determinano e sono determinate dalla transizione.

L'applicazione della funzione di transizione ad una configurazione si dice transizione o mossa o passo computazionale dell'automa.

Transizioni

Dato un automa $\mathscr{A}$ e due sue configurazioni c_i, c_j, la notazione

$$c_i \underset{\mathscr{A}}{\vdash\!\!\!-\!\!\!-} c_j$$

indica che c_j deriva da c_i per effetto dell'applicazione della funzione di transizione di $\mathscr{A}$.

Configurazioni di accettazione

Sottoinsieme delle possibili configurazioni: determinano, se raggiunte, l'accettazione della stringa in input da parte dell'automa.

Tutte le altre configurazioni sono definite come configurazioni di non accettazione, o di rifiuto.

Computazione

Un automa esegue una computazione applicando iterativamente, ad ogni istante, la propria funzione di transizione alla configurazione attuale, a partire dalla configurazione iniziale.

Sequenza di configurazioni attraversate

$$c_0, c_1, c_2, \ldots$$

tale che $c_i \underset{\mathcal{A}}{\vdash\!\!\!-\!\!\!-} c_{i+1}$ per $i = 0, 1, \ldots$

Computazione

$\overset{*}{\underset{\mathscr{A}}{\vdash\!\!-\!\!-}}$: chiusura transitiva e riflessiva della relazione $\underset{\mathscr{A}}{\vdash\!\!-\!\!-}$.

Date due configurazioni c_i, c_j di $\mathscr{A}$,

$$c_i \overset{*}{\underset{\mathscr{A}}{\vdash\!\!-\!\!-}} c_j$$

se e solo se esiste una computazione che porta $\mathscr{A}$ da c_i a c_j.

Computazione massimale

Se

1. $c_0, c_1, c_2, \ldots, c_n$ ha lunghezza finita
2. non esiste nessuna configurazione c tale che $c_n \vdash_{\mathcal{A}} c$

la computazione termina.

Esiti di una computazione

- c_n esiste ed è una configurazione di accettazione, l'automa accetta la stringa in input
- c_n esiste ed non è una configurazione di accettazione, l'automa rifiuta la stringa in input
- c_n non esiste, la computazione non termina

Automi deterministici

Ad ogni stringa di input associa una sola computazione, e quindi una singola sequenza di configurazioni.

Un automa deterministico, data una stringa di input, può eseguire una sola computazione: se la computazione termina in una configurazione di accettazione, allora la stringa viene accettata.

Stringhe e linguaggio accettato

1. Automa deterministico $\mathcal{A}$
2. Stringa x in input ad $\mathcal{A}$
3. $c_0(x)$ configurazione iniziale di $\mathcal{A}$ corrispondente alla stringa x

$\mathcal{A}$ accetta x se e solo se esiste una configurazione di accettazione c di $\mathcal{A}$ per la quale

$$c_0(x) \overset{*}{\underset{\mathcal{A}}{\vdash\!\!-\!\!-}} c$$

Il linguaggio accettato da $\mathcal{A}$ è l'insieme $L(\mathcal{A})$ di tutte le stringhe x accettate da $\mathcal{A}$.

Linguaggio riconosciuto

Se la computazione eseguita dall'automa $\mathscr{A}$ termina per ogni stringa, e quindi se ogni stringa viene o accettata o rifiutata, allora diciamo che il linguaggio $L(\mathscr{A})$ è riconosciuto (o deciso) da $\mathscr{A}$

Automa non deterministico

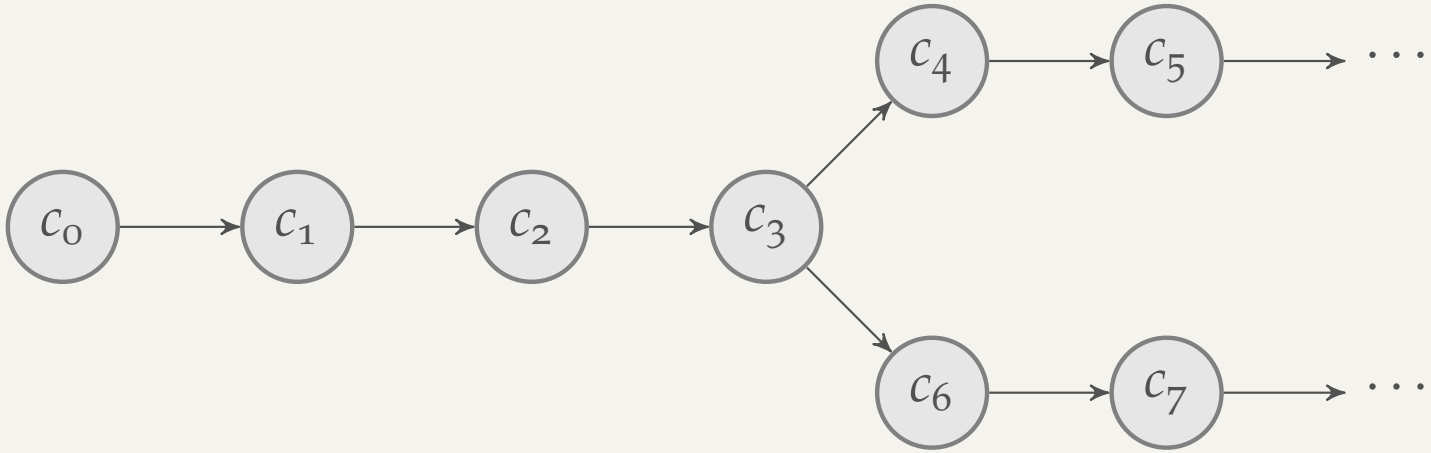
Generalizzazione dell'automa deterministico.

Associa ad ogni stringa di input un numero qualunque, in generale maggiore di uno, di computazioni.

La funzione di transizione associa ad almeno una configurazione c più di una configurazione successiva.

Per ogni computazione che conduce a c sono definite continuazioni diverse, che definiscono diverse computazioni.

Automa non deterministico



Il grado di non determinismo di un automa è il massimo numero di configurazioni che la funzione di transizione associa ad una configurazione.

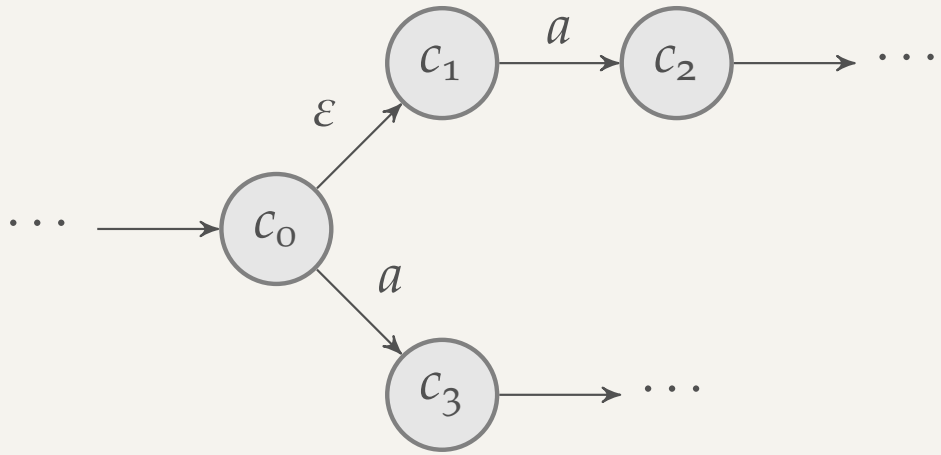
ε-transizioni

Transizioni che un automa può eseguire senza leggere alcun carattere in input.

La presenza di ε-transizioni introduce non determinismo.

Se da una configurazione c una automa può passare o meno in un'altra configurazione c' senza leggere caratteri in input, sono possibili due diverse continuazioni della computazione attuale, l'una da c e l'altra da c'.

ε-transizioni



- Computazione 1: $\cdots c_0 c_1 c_2 \cdots$
- Computazione 2: $\cdots c_0 c_3 \cdots$

Computazioni non deterministiche

Automa deterministico: associa alla stringa di input una sequenza di configurazioni (computazione lineare), con la prima uguale alla configurazione iniziale della stringa

Automa non deterministico: associa alla stringa di input un albero di configurazioni (albero di computazione), con la radice uguale alla configurazione iniziale della stringa. Ogni computazione corrisponde ad un cammino avente origine dalla radice stessa.

Computazioni non deterministiche: interpretazione 1

L'automa non deterministico può eseguire una qualunque computazione, associata ad una sequenza di "scelte" su quale transizione applicare in corrispondenza ad ogni applicazione della funzione di transizione.

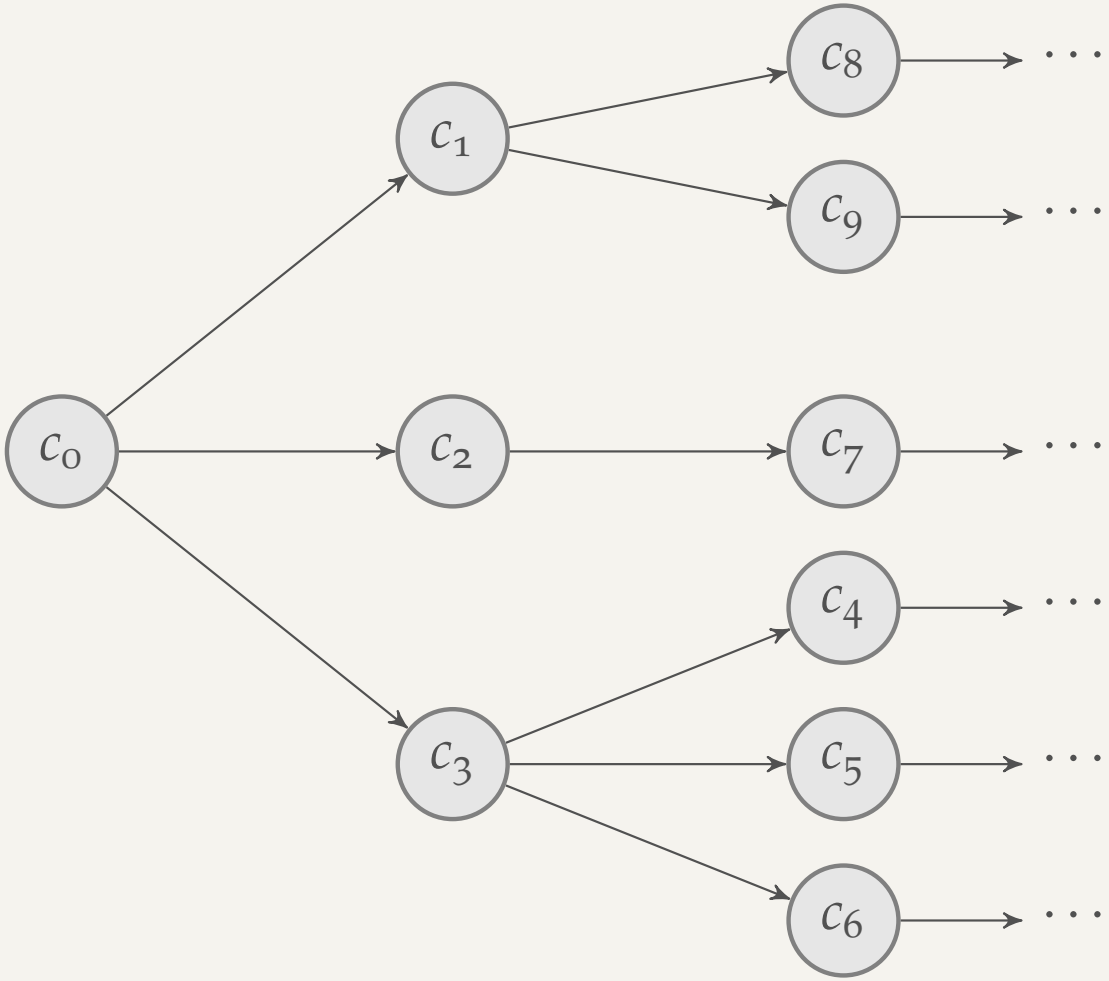
L'automa accetta la stringa di input se, tra tutte le computazioni possibili, ne esiste almeno una di accettazione (se esiste, nell'albero di computazione, un cammino dalla radice ad un nodo rappresentante una configurazione di accettazione).

Computazioni non deterministiche: interpretazione 2

L'automa esegue una sola computazione non deterministica, per la quale, ad ogni passo, assume non una sola, ma un insieme di configurazioni, transitando, ad ogni passo, non da configurazione a configurazione ma da un insieme di configurazioni ad un insieme di configurazioni.

Un insieme di configurazioni è di accettazione se include almeno una consfigurazione si accettazione.

Albero di computazione



Accettazione

Asimmetria tra accettazione e rifiuto di una stringa, introdotta dal non determinismo.

- Nel caso deterministico la stringa viene accettata se la singola computazione definita è di accettazione, e rifiutata se essa termina in una configurazione non di accettazione
- Nel caso non deterministico la stringa viene accettata se una qualunque delle computazioni definite è di accettazione, mentre non lo viene se tutte le possibili computazioni che terminano non sono di accettazione.

Decidibilità

Dato un modello di calcolo, un problema decisionale (o un linguaggio) è detto decidibile nel modello di calcolo se esiste un algoritmo (in particolare, un automa) che per ogni istanza del problema risponde correttamente VERO oppure FALSO. In caso contrario il problema è detto indecidibile.

Semidecidibilità

Dato un modello di calcolo, un problema decisionale (o un linguaggio) è detto semidecidibile nel modello di calcolo se esiste un algoritmo (in particolare, un automa) che per tutte e sole le istanze positive del problema risponde correttamente VERO.

Decidibilità e classi di linguaggi

I linguaggi di tipo 1, 2, 3 sono decidibili all'interno di modelli di calcolo opportuni.

Non esistono modelli di calcolo al cui interno i linguaggi di tipo 0 siano decidibili.

Esiste un modello di calcolo all'interno del quale i linguaggi di tipo 0 sono semidecidibili.

Decidibilità e classi di linguaggi

Linguaggi di tipo 3: decidibili per automi con memoria di dimensione costante, dipendente dal linguaggio, ma non dalla particolare stringa (Automi a stati finiti)

Linguaggi di tipo 2: decidibili per automi non deterministici con memoria di dimensione lineare e modalità di accesso limitate, di tipo *last-in-first-out* (Automi a pila non deterministici)

Linguaggi di tipo 1: decidibili per automi non deterministici con memoria di dimensione lineare e accesso casuale (Automi lineari non deterministici)

Linguaggi di tipo 0: semidecidibili per automi con memoria di dimensione potenzialmente illimitata e accesso casuale (Macchine di Turing)