Gli automata

Lezione 03

10 marzo 2015

Ing. Chiara Foglietta

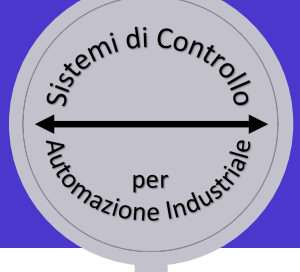
foglietta.chiara@gmail.com

Sistemi di Controllo per l'Automazione Industriale
Ingegneria Gestionale
A.A. 2014 - 2015
Università degli Studi di Cassino e del Lazio Meridionale

Agenda

Lezione 03

Chiara Foglietta



I modelli di linguaggi

Chiara Foglietta

Un metodo formale per studiare il comportamento logico dei SED è basato sulle teorie dei linguaggi e degli automata. Il punto di partenza è che qualsiasi SED ha un insieme di evento sottostante associato con esso. L'insieme *E* è pensato come un alfabeto del linguaggio e la sequenza di eventi sono pensati come parole nel linguaggio.

I modelli di linguaggi: esempio

Lezione 03

Chiara Foglietta

Si consideri una macchina che di solito accendiamo una o due volte al giorno (come una macchina o un computer), e si voglia progettare un sistema per eseguire compiti semplici: quando la macchina è accesa, per prima cosa deve mandare un segnale per dirci che è accesa, poi inviarci un report sul suo stato (come, nel caso della macchina, 'tutto a posto', 'controlla olio' o 'manca la benzina'), e quindi concludere con un altro segnale per informarci che 'report sullo stato finito'. Ognuno di questi segnali definisce un evento, e quindi tutti i possibili segnali della macchina possono definire un alfabeto (insieme di eventi).

I modelli di linguaggi: esempio

Lezione 03

Chiara Foglietta

Il nostro sistema deve essere un SED guidato dagli eventi. Questo SED è responsabile per riconoscere eventi e dare una propria interpretazione della particolare sequenza ricevuta.

1. 'Accesa', 'tutto a posto', 'report sullo stato finito' $\rightarrow$ completa con successo il compito
2. 'Accesa', 'report sullo stato finito' $\rightarrow$ riporta un'anomalia

Linguaggio

Linguaggio

Un linguaggio definito sopra un insieme di eventi E è un insieme di stringhe di lunghezza finita formata dagli eventi in E

Come esempio, sia $E = \{a, b, g\}$ l'insieme di eventi. Si può definire il linguaggio:

$$L_1 = \{\epsilon, a, abb\}$$

che contiene 3 stringhe solamente; oppure il linguaggio

$$L_1 = \{\text{tutte le stringhe di lunghezza 3 che cominciano con } a\}$$

che contiene 9 stringhe

Concatenazione

Chiara Foglietta

L'operazione coinvolta nella costruzione di stringhe da un insieme di eventi E è la concatenazione.
La stringa abb in L_1 è la concatenazione della stringa ab con l'evento b; ab è essa stessa la concatenazione di a e di b.
La concatenazione uv di due stringhe è la nuova stringa che consiste negli eventi in u immediatamente seguita dagli eventi in v.
La stringa vuota ϵ è l'elemento identità della concatenazione

$$u\epsilon = \epsilon u = u, \qquad \forall u$$

Chiusura di Kleene

Lezione 03

Chiara Foglietta

Si identifica con E^* l'insieme di *tutte* le stringhe finite di elementi di E, inclusa la stringa vuota ϵ. L'operatore * è detto chiusura di Kleene.
Si osservi che l'insieme E^* è un infinito numerabile perché contiene le stringhe di lunghezza arbitraria.

Per esempio, se $E = \{a, b, c\}$ allora

$$E^* = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \dots\}$$

Un linguaggio su un insieme di eventi E è quindi un

sottoinsieme di E^*. In particolare sono linguaggi $\emptyset, E, E^*$.

Altre definizioni

Se $tuv = s$ con $t, u, v, \in E^*$, allora

- t è chiamato *prefisso* di s
- u è chiamata *sottostringa* di s
- v è chiamato suffisso di s

A volte si usa la notazione s/t (s dopo t) per indicare il suffisso di s dopo il suo prefisso t. Se t non è un prefisso di s, allora s/t non è definito.

Si osservi che sia ϵ e s sono prefissi, sottostringhe, suffissi di s.

Operazioni sui linguaggi

Lezione 03

Chiara Foglietta

Le operazioni sugli insiemi, come unione, intersezione, differenza e complemento rispetto a E^*, sono applicabili ai linguaggio poiché i linguaggi sono insiemi.

Concatenazione

Concatenazione

Siano $L_a, L_b \subseteq E^*$, allora

$$L_a L_b := \{s \in E^* : (s = s_a s_b) \text{ and } (s_a \in L_a) \text{ and } (s_b \in L_b)\}$$

Una stringa è in $L_a L_b$ se può essere scritta come la concatenazione della stringa in L_a con una stringa in L_b

Chiusura del prefisso

Chiusura del prefisso

Sia $L \subseteq E^*$, allora

$$\bar{L} := \{s \in E^* : (\exists t \in E^*)\,[st \in L]\}$$

La chiusura del prefisso di L è il linguaggio identificato da $\bar{L}$ e consiste di tutti i prefissi di tutte le stringhe in L. In generale, $L \subseteq \bar{L}$.

L si dice chiuso rispetto al prefisso se $L = \bar{L}$.

Chiusura in Kleene

Sia $L \subseteq E^*$, allora

$$L^* := \{\epsilon\} \cup L \cup LL \cup LLL \cup \dots$$

Questa è la stessa operazione che si è definita per l'insieme E, ad eccezione che ora è applicato all'insieme L i cui elementi sono stringhe di lunghezza maggiore di 1.
Un elemento di L^* è costituito dalla concatenazione di un numero finito (ma possibilmente grande) di elementi di L; questo include la concatenazione di elementi "zeri" che è la stringa vuota ϵ.
Si noti che l'operazione * è idempotente: $(L^*)^* = L^*$

Post-linguaggio

Post-linguaggio

Sia $L \subseteq E^*$ e $s \in L$. Allora il post-linguaggio di L dopo s, indicato con L/s, è il linguaggio

$$L/s := \{t \in E^* : st \in L\}$$

Per definizione, $L/s = \emptyset$ se $s \notin \bar{L}$

Esempio per le operazioni sui linguaggi

Chiara Foglietta

Sia $E = \{a, b, g\}$ e si consideri i due linguaggi $L_1 = \{\epsilon, a, abb\}$ e $L_4 = \{g\}$.

Né L_1 né L_4 sono chiuse rispetto al prefisso poiché $ab \notin L_1$ e $\epsilon \notin L_4$.

$$
\begin{aligned}
L_1 L_4 &= \{g, ag, abbg\} \\
\bar{L}_1 &= \{\epsilon, a, ab, abb\} \\
\bar{L}_4 &= \{\epsilon, g\} \\
L_1 \bar{L}_4 &= \{\epsilon, a, abb, g, ag, abbg\} \\
L_4^* &= \{\epsilon, g, gg, ggg, \dots\} \\
L_1^* &= \{\epsilon, a, abb, aa, aabb, abba, abbabb, \dots\}
\end{aligned}
$$

Proiezione di stringhe

Lezione 03

Chiara Foglietta

Un'altra operazione è chiamata proiezione naturale, o semplicemente proiezione, da un insieme di eventi E_l, in un insieme più piccolo di eventi E_s, dove $E_s \subset E_l$.
Le proiezioni naturali sono indicati dalla lettera P; un indice è tipicamente aggiunto per specificare l'insieme destinazione.
Nel nostro caso si assume che i due insieme sono fissati e quindi si evita l'indice.

Proiezione di stringhe

Si definisce la *proiezione P* come

$$P : E_l^* \rightarrow E_s^*$$

dove

$$P(\epsilon) := \epsilon$$

$$P(e) := \begin{cases} e & \text{if } e \in E_s \\ \epsilon & \text{if } e \in E_l \backslash E_s \end{cases}$$

$$P(se) := P(s)P(e) \text{ for } s \in E_l^*, e \in E_l$$

Inversa della proiezione

Lezione 03

Chiara Foglietta

Si può anche definire la mappa inversa

$$P^{-1} : E_s^* \rightarrow 2^{E_l^*}$$

$$P^{-1}(t) := \{s \in E_l^* : P(s) = t\}$$

La notazione 2^A significa il power set di A, che è l'insieme di tutti i sottoinsiemi di A.

Proiezione di stringhe: esempi

Sia $E_l = \{a, b, c\}$ e si consideri due sottoinsiemi propri $E_1 = \{a, b\}, E_2 = \{b, c\}$. Si consideri

$$L = \{c, ccb, abc, cacb, cabcbbca\} \subset E_l^*$$

Si considerino le due proiezioni $P_i : E_l^* \to E_i^*, i = 1, 2$:

$$P_1(L) = \{\epsilon, b, ab, abbba\}$$
$$P_2(L) = \{c, ccb, bc, cbcbbc\}$$

Automata

Lezione 03

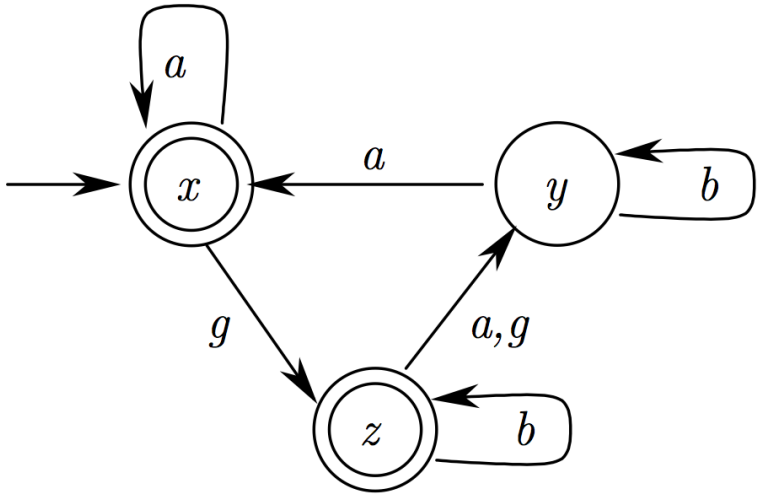
Chiara Foglietta

Un automata è un approccio che è in grado di rappresentare un linguaggio in accordo a regole ben definite.
Il modo più semplice per presentare la nozione di automata è quella di considerare la rappresentazione tramite grafo diretto, o un diagramma di transizione dello stato.

Automata: esempio

Lezione 03

Chiara Foglietta



Sia un insieme di eventi $E = \{a, b, g\}$. I nodi rappresentano gli stati e gli archi rappresentano le transizioni tra gli stati. Il grafo diretto presenta una descrizione delle dinamiche di un automata. Gli archi nel grafo forniscono una rappresentazione grafica della funzione di transizione dell'automata, che si denota come $f : X \times E \to X$:

$$f(x, a) = x, f(y, a) = x, f(z, b) = z,$$
$$f(x, g) = z, f(y, b) = y, f(z, a) = f(z, g) = y$$

Automata: esempio

Chiara Foglietta

Un evento può accadere senza alcun cambiamento dello stato, come in $f(x, a) = x$.

Due distinti eventi possono accadere ad un certo stato causando esattamente la stessa transizione, come in $f(z, a) = f(z, g) = y$.

La funzione f è una *funzione parziale* sul dominio $X \times E$, for ogni evento in E ad ogni stato X.

Lezione 03

Chiara Foglietta

Automata deterministico

Un automata deterministico, denotato da G, come una tupla di sei elementi

$$G = (X, E, f, \Gamma, x_0, X_m)$$

- X è l'insieme degli stati
- E è l'insieme finito degli eventi associati con G
- $f : X \times E \to X$ è la funzione di transizione: $f(x, e) = y$ indica che esiste una transizione etichettata da un evento e dallo stato x allo stato y; in generale, f è una funzione parziale sul suo dominio
- $\Gamma : X \to 2^E$ è la funzione degli eventi attivo o la funzione degli eventi possibili; $\Gamma(x)$ è l'insieme di tutti gli eventi e per cui $f(x, e)$ è definita ed è chiamata insieme degli eventi attivi di G a x
- x_0 è lo stato iniziale
- $X_m \subseteq X$ è l'insieme degli stati marcati

Linguaggi generati e marcati

Il linguaggio generati da $G = (X, E, f, \Gamma, x_0, X_m)$ è

$$\mathcal{L}(G) := \{s \in E^* : f(x_0, s) \textit{is defined}\} \tag{1}$$

Il linguaggio marcato da G è

$$\mathcal{L}_m(G) := \{s \in \mathcal{L}(G) : f(x_0, s) \in X_m\} \tag{2}$$

Le definizioni assumono che si sta lavorando con la funzione di transizione estesa $f : X \times E^* \to X$.

Linguaggio marcato: esempio

Lezione 03

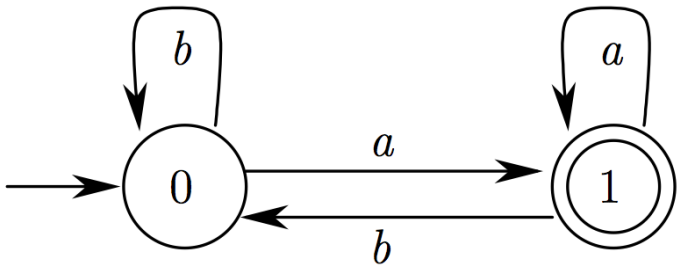
Chiara Foglietta

Dato un insieme di eventi $E = \{a, b\}$. Si consideri il linguaggio

$$L = \{a, aa, ba, aaa, aba, baa, bba, \dots\}$$

che consiste di tutte le stringhe seguite sempre da un evento a. Questo linguaggio è marcato dall'automata a stato finito $G = (E, X, f, \Gamma, x_0 X_m)$ dove $X = \{0, 1\}$, $x_0 = 0$, $X_m = \{1\}$ ed f è definito da $f(0, a) = 1$, $f(0, b) = 0$, $f(1, a) = 1$, $f(1, b) = 0$.

$$\mathcal{L}_m(G) = L \tag{3}$$



Linguaggio generato e marcato: esempio

Chiara Foglietta

Se si modifica l'automata nel caso precedente togliendo l'auto-ciclo a causa dell'evento b allo stato 0, che si ottiene considerando $f(0, b)$ indefinita e quindi $\mathcal{L}(G)$ consiste di ϵ insieme con le stringhe in E^* che comincia con l'evento a e dove non ci sono occorrenze consecutive dell'evento b. Qualsiasi b nella stringa è o l'ultimo evento della stringa o è immediatamente seguito da un a. $\mathcal{L}_m(G)$ è il sotto-insieme di $\mathcal{L}(G)$ che contiene tutte le stringhe che terminano con un evento a.

Linguaggio equivalenti ad un automata

Due automata sono detti *linguaggio equivalenti* se generato e marcano il medesimo linguaggio.

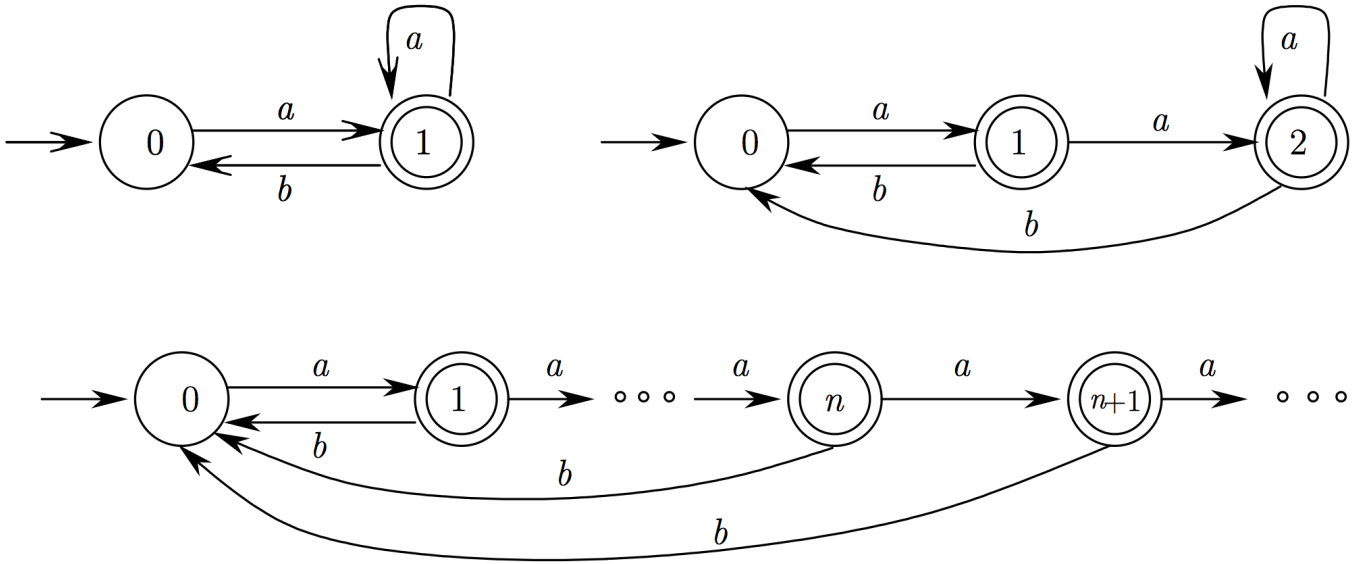
Automata a linguaggio equivalenti

Gli automata G_1 e G_2 sono detti linguaggio equivalenti se

$$\mathcal{L}(G_1) = \mathcal{L}(G_2), \qquad \text{and} \qquad \mathcal{L}_m(G_1) = \mathcal{L}_m(G_2)$$

Linguaggio equivalenti ad un automata: esempio

Si noti come il terzo esempio ha uno insieme di stati infinito.



Blocco

In generale, vale la seguente osservazione:

$$\mathcal{L}_m(G) \subseteq \bar{\mathcal{L}}_m(G) \subseteq \mathcal{L}(G)$$

La prima inclusione è poiché X_m può essere un sotto-insieme proprio di X, mentre la seconda inclusione è una conseguenza della definizione di $\mathcal{L}_m(G)$ e del fatto che $\mathcal{L}(G)$ è chiuso rispetto al prefisso per definizione.

Blocco

Un automata G può raggiungere uno stato x dove $\Gamma(x) = \emptyset$ ma con $x \notin X_m$. Questo è chiamato *punto morto* (deadlock) perchè nessun altro evento può essere eseguito. Data la definizione di marcatura, si dice che il sistema si blocca perchè entra in uno stato a punto morto senza avere terminato il task. Se ci si trova in tale situazione, allora $\bar{\mathcal{L}}_m(G)$ sarà un sottoinsieme proprio di $\mathcal{L}(G)$, poichè ogni stringa in $\mathcal{L}(G)$ che termina con stato x non può essere un prefisso di una stringa in $\mathcal{L}_m(G)$.

Blocco

Blocco

Un automata G è in stato di blocco se

$$\bar{\mathcal{L}}_m(G) \subset \mathcal{L}(G)$$

e in stato di non-blocco quando

$$\bar{\mathcal{L}}_m(G) = \mathcal{L}(G)$$

Se un automata è bloccato, questo significa che può accadere un punto morto o un livelock.

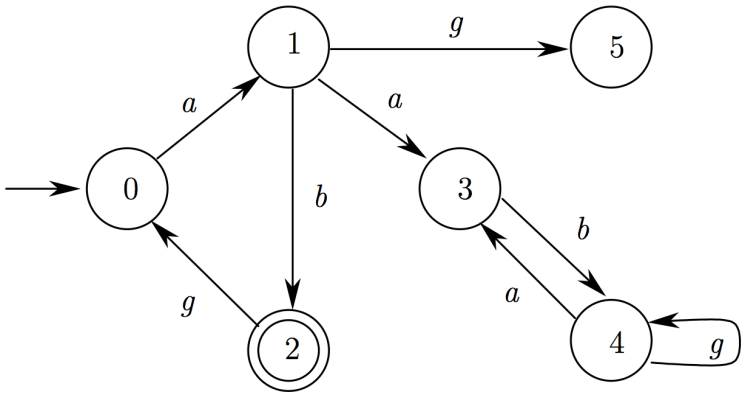
Livestock

Chiara Foglietta

Un altro problema da considerare è quando esiste un insieme di stati non marcati in G che forma una componente fortemente connessa (ossia gli stati sono raggiungibili tra di loro), ma con nessuna transizione che esce fuori dall'insieme. Se il sistema entra in questo insieme di stati, allora si ottiene un *livelock*. Quando il sistema è "vivo" nel senso che può sempre eseguire un evento, non può mai completare il compito cominciato poichè nessuno stato nell'insieme è marcato e il sistema non può lasciare questo insieme di stati. Se un livelock è possibile, allora $\bar{\mathcal{L}}_m(G)$ è un sottoinsieme proprio di $\mathcal{L}(G)$. Qualsiasi stringa in $\mathcal{L}(G)$ che raggiunge l'insieme di stati non-marcati, non possono essere un prefisso di una stringa in $\mathcal{L}_m(G)$, poichè si assume che non esista modo per uscire da tale insieme. Si può dire che il sistema è "bloccato" nel livestock.

Blocco: esempio

Lo stato 5 è un punto morto.
Gli stati 3 e 4, con le associate transizioni a, b, g, formano una componente fortemente connessa ; poichè nè 3 nè 4 sono marcati, qualsiasi stringa che raggiunge lo stato 3 genera un livestock.
La stringa $ag \in \mathcal{L}(G)$ ma $ag \notin \mathcal{L}_m(G)$; questo vale per qualsiasi stringa in $\mathcal{L}(G)$ che comincia con aa. Quindi l'automata G può andare in blocco poichè $\bar{\mathcal{L}}_m(G)$ è un sottoinsieme proprio di $\mathcal{L}(G)$.



Esempio: check dello stato della macchina

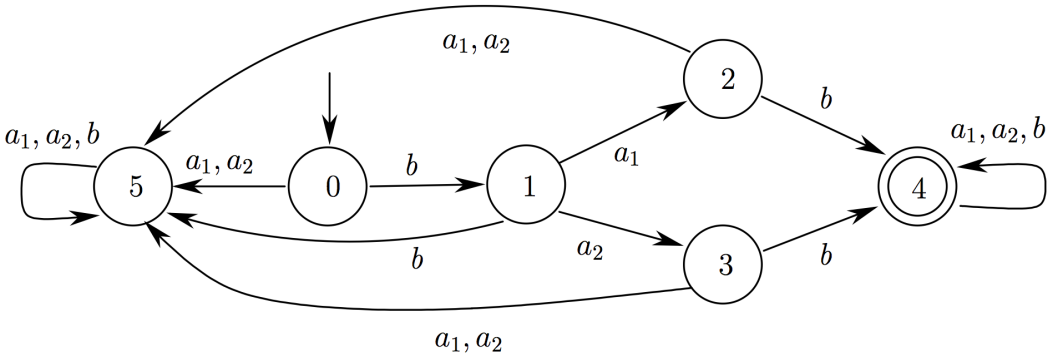
Chiara Foglietta

Sia $E = \{a_1, a_2, b\}$ l'insieme degli eventi. Un task è definito come una sequenza di tre eventi che comincia con l'evento b, seguito dall'evento a_1 o da a_2, e quindi l'evento b, seguito da qualsiasi sequenza di eventi arbitraria. Quindi, sivuole progettare un apparecchio che legge qualsiasi stringa formata dall'insieme di eventi, ma riconosce solamente, ossia marca, le stringhe di lunghezza 3 o più che soddisfa la definizione precedente di compito.Ognuna di queste stringhe deve cominciare con b e include a_1 o a_2 nella seconda posizione, seguita da b nella terza posizione.

Esempio: check dello stato della macchina

Chiara Foglietta

L'insieme degli stati $X = \{0, 1, 2, 3, 4, 5\}$ consiste di stati etichettati arbitrariamente con $x_0 = 0$ e $X_m = 4$. Si noti come qualsiasi stringa con meno di 3 eventi finisce sempre negli stati $0, 1, 2, 3$ o 5 e quindi non è marcato. La sola stringa con 3 o più eventi che sono marcati sono quelli che cominciano con b, seguiti da a_1 o a_2, e quindi raggiunge lo stato 4 attraverso l'evento b.



Esempio: check dello stato della macchina

Chiara Foglietta

Quando una macchina è accesa, si consideri ”l’inizializzazione” (stato 0), e si aspetta l’evento *b* che indica l’accensione. Questo porta allo stato 1, che significa ”sono accesa”. A questo punto la macchina si suppone che esegua un test di diagnostica che genera uno dei due possibili eventi a_1 o a_2. L’evento a_1 indica ”il mio stato è ok”, mentre l’evento a_2 indica ” ho un problema”. Infine la macchina ci deve informare che la procedura iniziale è completa tramite l’evento *b*, andando nello stato 4 che significa ”report sulla stato finito”. Qualsiasi altra combinazione porta allo stato 5, che deve essere interpretata come un ”errore”. Qualsiasi altro evento che accade dopo lo stato 4, è semplicemente ignorato per lo scopo del compito, il cui stato marcato è già stato raggiunto.

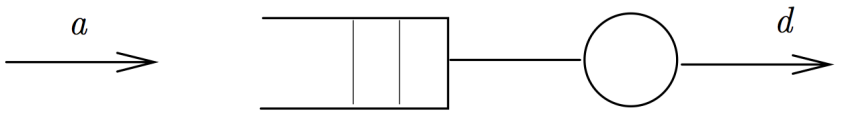
Esempio: check dello stato della macchina

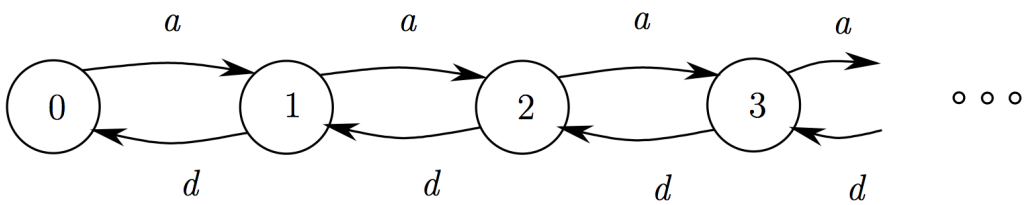
Chiara Foglietta

Poichè l'automata è progettato per accettare eventi di ingresso generati dal sistema, tutti gli ingressi non attesi, ossia quelli non nell'insieme $\bar{\mathcal{L}}_m(G)$ portano l'automata nello stato 5. Questo significa che f è una funzione totale; questo assicura che l'automata è in grado di processare qualsiasi tipo di ingresso.
Lo stato 5 è lo stato in cui accade un livelock, in quanto è impossibile raggiungere lo stato 4 dallo stato 5. Quindi, tale automata si può bloccare.
Lo stato 2 e 3 possono essere uniti senza avere conseguenze sul linguaggio generato e marcato dall'automata.

Sistema a coda

Gli utenti arrivano e richiedono l'accesso ad un servente. Se il servente è già occupato, allora aspetta in coda. Quando un utente completa il servizio, lascia il sistema e il prossimo utente nella coda (se esiste) immediatamente riceve il servizio.





Sistema a coda

Gli eventi del sistema sono:

- a arrivo dell'utente
- d partenza dell'utente

Si definisce un automata nel seguente modo

$$
\begin{aligned}
E &= \{a, d\} \\
X &= \{0, 1, 2, \dots\} \\
\Gamma(x) &= \{a, d\}, \qquad \forall x > 0, \Gamma(0) = \{a\} \\
f(x, a) &= x + 1, \qquad \forall x \geq 0 \\
f(x, d) &= x - 1, \qquad x > 0
\end{aligned}
$$

La variabile di stato x rappresenta il numero di utenti nel sistema.

Sistema a coda

Lezione 03

Chiara Foglietta

Lo stato iniziale x_0 è scelto come il numero iniziale di utenti nel sistema.
L'insieme di eventi possibili $\Gamma(0)$ è limitato agli eventi di arrivo a, poichè nessuna partenza è possibili quando il sistema a coda è vuoto. Quindi $f(x, d)$ non è definito per $x = 0$.
Lo spazio di stato è infinito, ma numerabile; X_m non è specificato.

Sistema a coda

Lezione 03

Chiara Foglietta

Ci si concentra ora sullo stato del servente piuttosto che dell'intero sistema.
Il servente può essere o in modalità Idle (I) o in modalità Busy (B). Il servente occasionalmente si può rompere (D). Quando il servente si rompe, l'utente viene perso; e dopo la riparazione il servente è in idle.

Sistema a coda

Gli eventi definiti dall'ingresso di questo sistema sono assunti come:

- α servizio iniziato
- β servizio completato
- λ servente rotto
- μ servente riparato

Il modello dell'automata del servente è dato da:

$$E = \{\alpha, \beta, \lambda, \mu\} \qquad X = \{I, B, D\}$$

$$\Gamma(I) = \{\alpha\} \qquad f(I, \alpha) = B$$

$$\Gamma(B) = \{\beta, \lambda\} \qquad f(B, \beta) = I$$

$$f(B, \lambda) = D$$

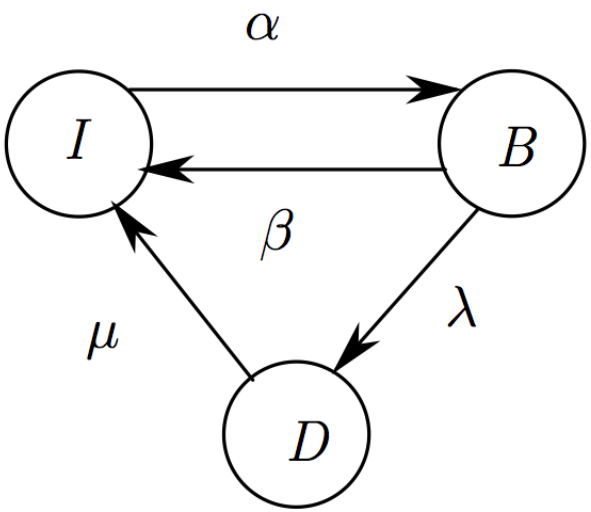
$$\Gamma(D) = \{\mu\} \qquad f(D, \mu) = I$$

Sistema a coda

Lezione 03

Chiara Foglietta

Intuitivamente, assumendo che non si vuole che il nostro servente rimanga in idle, l'evento "servizio iniziato" accade immediatamente appena entrato nello stato *I*. Questo non è possibile quando la coda è vuota, ma in questo modello non si ha la conoscenza della lunghezza della coda.



Le operazioni sugli automata

Lezione 03

Chiara Foglietta

Per analizzare i modelli a eventi discreti con gli automata, sono necessarie un insieme di operazioni su un singolo automata per modificare appropriatamente il diagramma di transizione dello stato in accordo ad alcune operazioni di linguaggio.

Si deve anche definire le operazioni per consentire la combinazione o la composizione di due o più automata , in modo che i modelli del sistema completa possano essere costruiti a partire dai modelli dei componenti individuali.

Operazioni unarie

Lezione 03

Chiara Foglietta

In questa sezione si considerano operazioni che alterano il diagramma di transizione di un automata. L'insieme degli eventi E rimane inalterato.

Si può cancellare da G tutti gli stati che non sono accessibili o raggiungibili da x_0 con stringhe in $\mathcal{L}(G)$ senza conseguenze di linguaggi generati e marcati da G. Quando si cancella uno stato, si cancellano anche tutte le transizioni che sono collegate allo stato. Tale operazione è denotata da $Ac(G)$. Senza nessuna perdita di generalità si può assumere che l'automata è accessibile, per cui $G = Ac(G)$.

Operazioni unarie

Lezione 03

Chiara Foglietta

Parte co-accessibile

Uno stato x di G è detto co-accesibile da X_m se esiste un cammino nel diagramma di transizione di G dallo stato x allo stato marcato. Si identifica l'operazione di cancellare tutti gli stati di G che non sono co-accessibili con $CoAc(G)$. L'operazione di $CoAc$ può ridurre $\mathcal{L}(G)$, perchè cancella stati che sono accessibili da x_0; tuttavia, l'operazione $CoAc$ non impatta su $\mathcal{L}_m(G)$ poichè uno stato cancellato non può essere su qualsiasi cammino da x_0 a X_m.

Operazioni unarie

Lezione 03

Chiara Foglietta

Operazione trim

Un automata che è sia accessibile che co-accessibile è detto trim.

$$Trim(G) := CoAc[Ac(G)] := Ac[CoAc(G)]$$

dove vale la proprietà commutativa

Operazioni unarie

Complemento

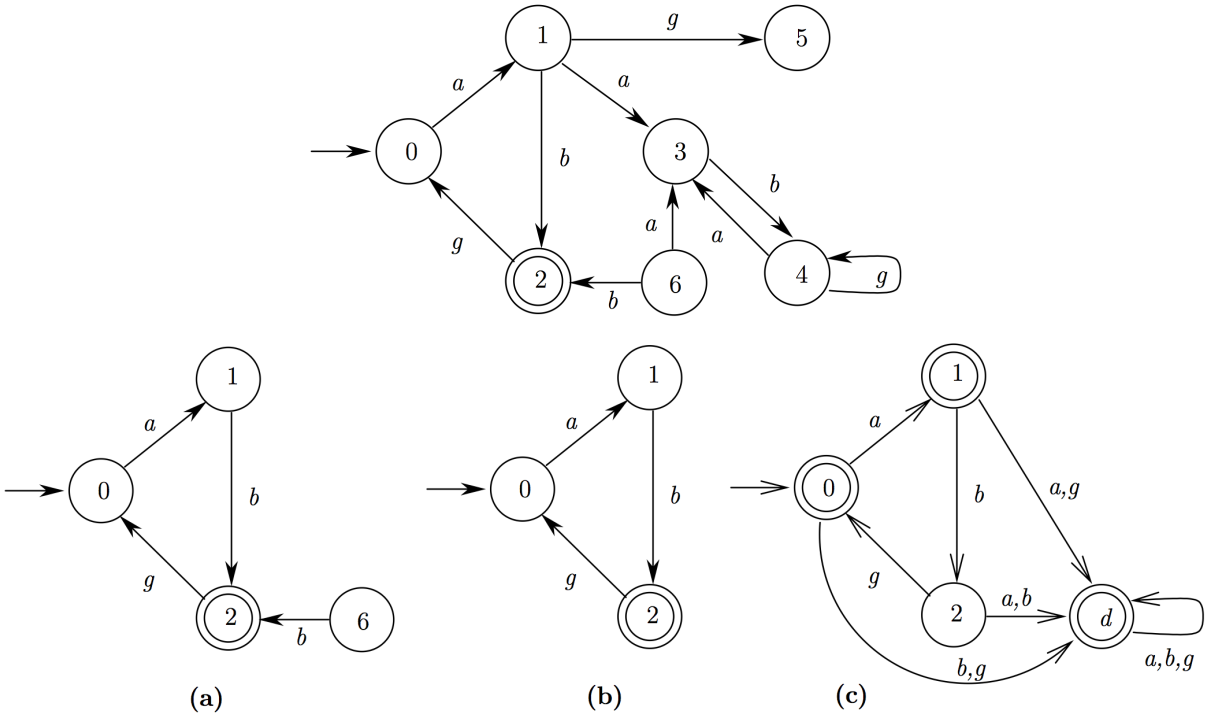
Si supponga di avere un automata trim $G = (X, E, f, \Gamma, x_0, X_m)$ che marca il linguaggio $L \subseteq E^*$. Quindi G genera il linguaggio $\bar{L}$. Si vuole creare un nuovo automata, denotato con G^{comp} che marca il inguaggio $E^* \setminus L$.

Il primo step di questa operazione è di completare la funzione di transizione f di G; la nuova funzione di transizione è detto f_{tot}. Questo è fatto aggiungendo un nuovo stato x_d a X, chiamato stato ”morto” o ”finto”. Tutte gli $f(x, e)$ indefiniti in G sono assegnati a x_d.
Il secondo step è di cambiare lo stato della marcatura di tutti gli stati in G_{tot} marcando tutti gli stati non marcati, e non-marcando tutti gli stati marcati.

Operazioni unarie: esempio

Lezione 03

Chiara Foglietta



(a) *CoAc*(*G*) cancellano tutti gli stati che non possono raggiungere lo stato marcato 2.

(b) *Trim*(*G*) identifica l'automata che è sia *CoAc*(*G*) che *Ac*(*G*), ossia cancello lo stato 6.

(c) *Comp*[*Trim*(*G*)], in cui si aggiunge lo stato *d*, si completa la funzione di transizione e infine si inverte la marcatura.

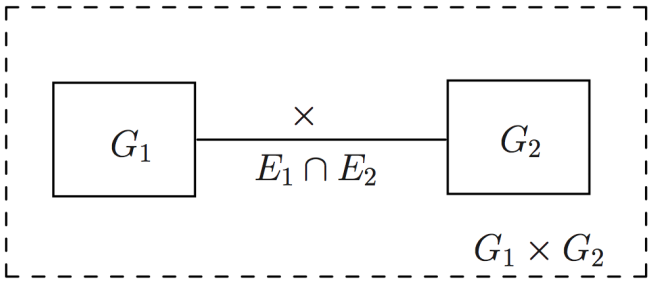
Le operazioni di composizione

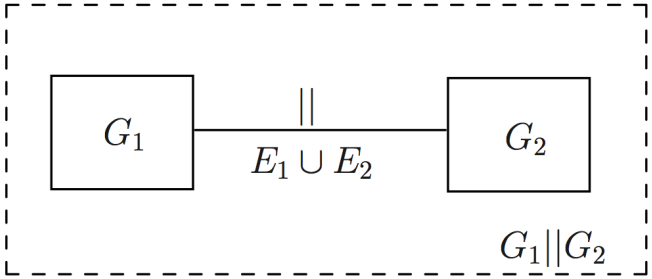
Lezione 03

Chiara Foglietta

Si definiscono due operazioni sugli automata: prodotto, denotato da $\times$ e la composizione parallela, denotata da $||$.
Si considerano due automata:

$$G_1 = (X_1, E_1, f_1, \Gamma_1, x_{01}, X_{m1})$$
$$G_2 = (X_2, E_2, f_2, \Gamma_2, x_{02}, X_{m2})$$





Le operazioni di composizione

Prodotto

Il prodotto di G_1 e G_2 è l'automata

$$G_1 \times G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1\times 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2})$$

dove

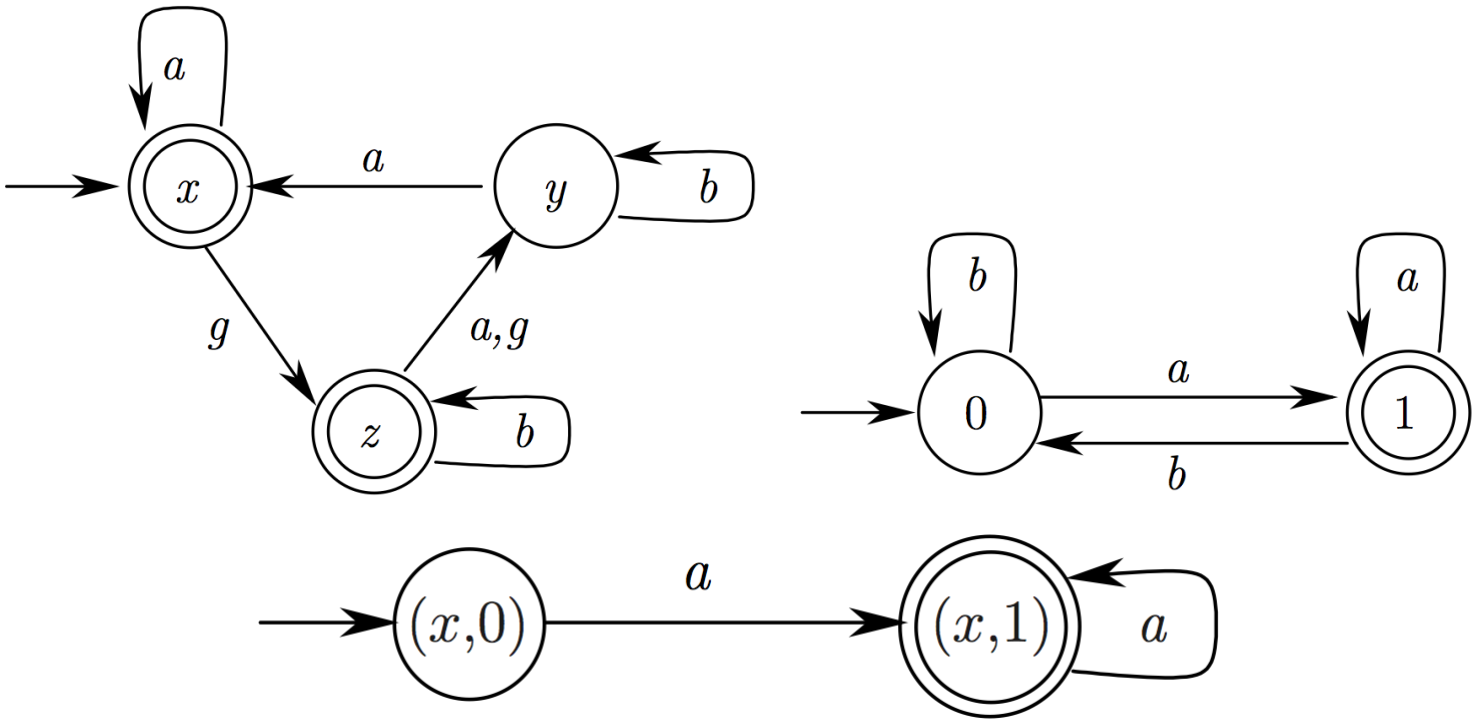
$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

Il prodotto è commutativo ed associativo

Prodotto: esempio

Lezione 03

Chiara Foglietta



gli eventi in comune sono $\{a, b\}$, mentre l'unico evento che fa modificare l'automata è a

Le operazioni di composizione

Lezione 03

Chiara Foglietta

Parallelo

La composizione parallela di G_1 e di G_2 è l'automata

$$G_1||G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1||2}, (x_{01}, x_{02}), X_{m1} \times X_{m2})$$

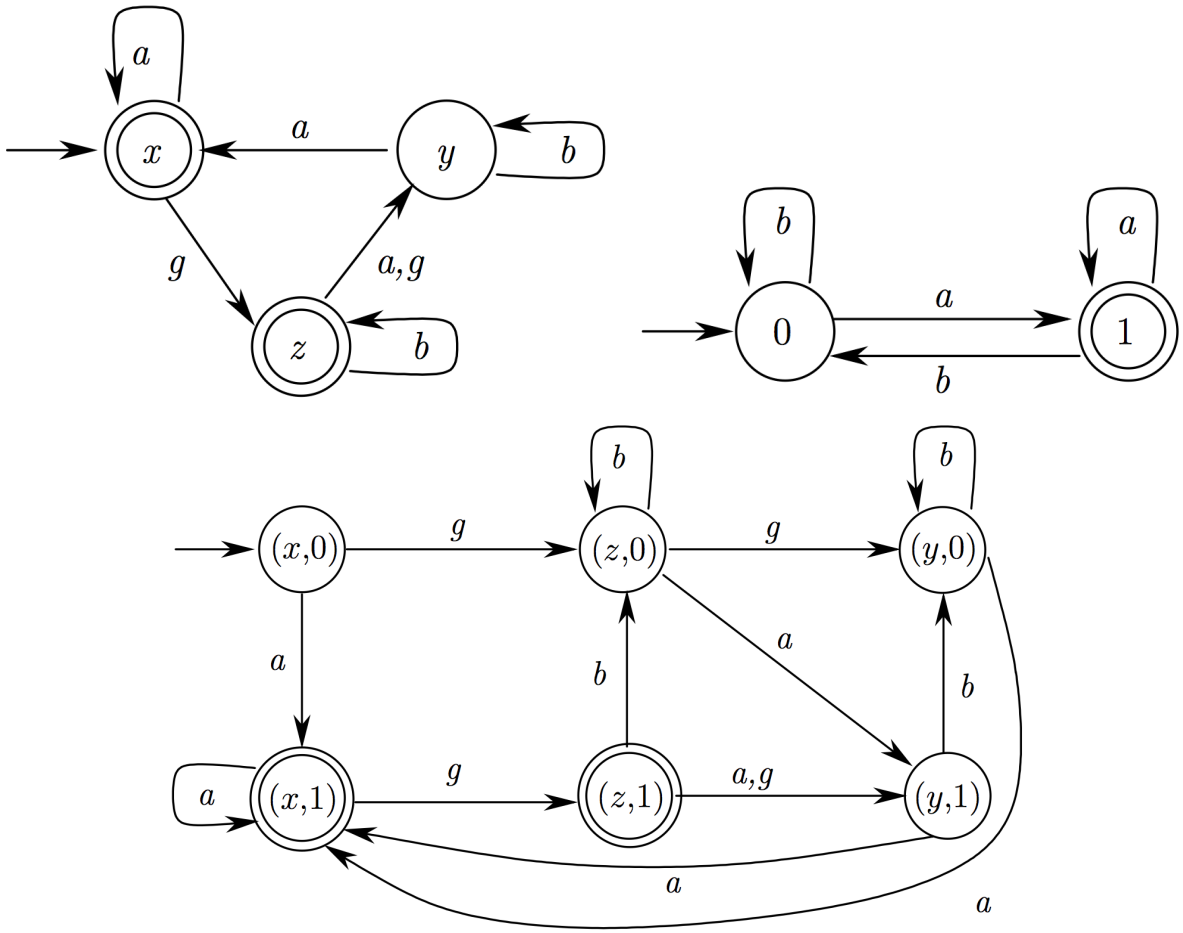
dove

$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1, e), x_2) & e \in \Gamma_1(x_1) E_2 \\ (x_1, f_2(x_2, e)) & e \in \Gamma_2(x_2) E_1 \\ \textit{undefined} & \textit{otherwise} \end{cases}$$

La composizione parallela è commutativa e associativa

Parallelo: esempio

Chiara Foglietta



Proprietà di sicurezza

Lezione 03

Chiara Foglietta

Sono quelle proprietà che riguardano la raggiungibilità di alcuni stati non-desiderabili nell'automata, la presenza di certe stringhe o sotto-stringhe non-desiderabili nel linguaggio generato dall'automata, o più generalmente l'inclusione del linguaggio generato dall'automata in un dato linguaggio.

Proprietà di sicurezza

Lezione 03

Chiara Foglietta

Le questioni di sicurezza sono:

1. Per determinare se un dato stato y è raggiungibile da un altro stato x, si considera l'operazione di accessibile sull'automata, con x dichiarato come stato iniziale, e cercando per lo stato y nei risultati;
2. Per determinare se una data sottostringa è possibile nell'automata, si prova ad eseguire la sottostringa da tutti gli stati accessibili nell'automata;
3. Per testare l'inclusione $A \subseteq B$ è equivalente a testare $A \cap B^c = \emptyset$. L'intersezione è implementata come il prodotto del corrispondente automata.

Le proprietà di blocco

Lezione 03

Chiara Foglietta

Sono quelle proprietà che sono relative alla co-accessibilità degli stati rispetto all'insieme degli stati marcati. La proprietà di blocco più comune è determinata se

$$\bar{\mathcal{L}}_m(G) = \mathcal{L}(G)$$
$$\bar{\mathcal{L}}_m(G) \subset \mathcal{L}(G)$$

Nell'ultimo caso G è bloccato. Per determinare se un dato G è bloccato o no, si usa l'operazione di co-accessibilità. Se qualsiasi stato viene cancellato, allora G è bloccato, altrimenti no.