1

Capitolo C12
macchine sequenziali e linguaggi razionali

Contenuti delle sezioni

a. macchine sequenziali p. 3
b. riconoscitori di Rabin-Scott p. 8
c. riconoscitori nondeterministici p. 14
d. composizioni di linguaggi razionali p. 18
e. espressioni razionali e teorema di Kleene p. 22
f. riconoscitore minimo di un linguaggio razionale p. 28

30 pagine

C12 0.01 Ai termini **macchina sequenziale** e **macchina formale** che qui consideriamo equivalenti, assegnamo il significato di modello matematico discreto che ha lo scopo di rappresentare in modo schematico e semplificato un meccanismo, tangibile o condivisibilmente giudicato realizzabile, che sia in grado di operare con buona autonomia in modo da effettuare processi che producono effetti osservabili.

In altre parole possiamo definire macchina sequenziale un meccanismo sibolico, ossia in grado di trasformare stringhe, che consenta di progettare (simulare) processi tangibili che producono trasformazioni precisamente valutabili.

La gamma delle macchine sequenziali e delle loro applicazioni che risulta utile studiare è assai vasta e si è andata ampliando soprattutto negli ultimi decenni che hanno visto la crescita delle tecnologie elettroniche e telematiche che consentono di implementarle, testarle, valutarle e migliorarle.

Va anche segnalato che non vi è un ampio accordo sopra una precisa portata del termine "macchina sequenziale" e dei termini collegati: qui introdurremo vari loro tipi preoccupandoci solo di mantenere una coerenza interna per i termini e le caratterizzazioni e di segnalare le scelte alternative più rilevanti.

Occorre aggiungere che in genere si usa anche il termine automa come equivakente di macchina sequenziale, ma che qui riserviamo questo nome a un tipo specifico di tali meccanismi.
Inoltre spesso in seguito abbreviamo "macchina sequenziale" e "macchina formale" con il semplice "macchina".
Su questo ultimo termine dobbiamo aggiungere che ci allontaniamo dalla accezione classica (archimedea) di meccanismo in grado di svolgere lavori da valutare in temini energetici, accezione da adottare invece nella fisica matematica e in particolare nella termodinamica.

C12 0.02 Le prime pagine che seguono riguardano una introduzione delle macchine sequenziali intese in modo generico e caratterizzate da comportamenti deterministici, introduzione assimilabile a quelle riguardanti le macchine sequenziali multinastro o programmabili.

Sono poi presentate macchine piuttosto semplici, corrispondenti a dispositivi con pochi tipi di componenti e a prestazioni piuttosto rigide, i riconoscitori di Rabin-Scott, ma già in grado di prestazioni significative.

Successivamente saranno studiati vari tipi di macchine internamente più elaborate e più versatili, a partire dai riconoscitori a pila [C14] e dalle macchine di Turing [C21].

I riconoscitori di Rabib e Scott individuano linguaggi che possono essere caratterizzati bene in termini algebrici (sono strettamente collegati con i semigruppi finiti) e conseguentemente attraverso espressioni molto maneggevoli (che chiamiamo espressioni razionali) e mediante equazioni, fino ad arrivare a forme canoniche ben definite in termini algoritmici.

In questa direzione si ottengono risultati di carattere basilare che sono di notevole utilità sia per le teorie dei linguaggi formali e della computabilità, sia per tecniche di elevato interesse per la programmazione dei computers.

C12 a. macchine sequenziali

C12 a.01 Prima caratteristica di ogni macchina sequenziale è quella di essere chiaramente distinguibile dall'ambiente che la circonda e di avere interazioni con una parte ben delimitata dell'ambiente che la circonda collegate e funzionali alle trasformazioni che ha il compito di eseguire.
Queste azioni in genere sono definite in modo ben preciso, ma occorre segnalare che si studiano anche macchine che presentano comportamenti probabilistici.

L'ambiente esterno invia a una macchina sequenze di segnali, ciascuna delle quali induce una evoluzione della macchina che vediamo svilupparsi nel tempo.

Durante una sua evoluzione la macchina a sua volta può emettere verso l'esterno segnali che vengono percepiti come prodotti di una sua attività e quindi, dal punto di vista delle applicazioni, vengono giudicati come sue prestazioni.
Questi segnali si possono considerare come emessi in risposta a quelli inviati alla machina dal mondo esterno, a loro volta giudicabili come stimoli.

La precedente descrizione si può raffigurare nel seguente modo:

Si può osservare che le macchine sequenziali che stiamo introducendo si possono avvicinare agli esecutori artificiali di elaborazioni [B01e, B17a] e che la loro collezione comprende le macchine sequenziali multinastro [B01e-B01g] e le macchine sequenziali programmabili [B17d].

C12 a.02 Altra caratteristica comune a ogni macchina sequenziale consiste nel fatto che nel corso della sua evoluzione essa si viene a trovare in successive situazioni, chiamate **configurazioni della macchina sequenziale**.
Ogni configurazione la caratterizziamo primariamente con un cosiddetto **stato interno** che chiediamo sia ben definito e potenzialmente riconoscibile.
In accordo con i requisiti di tangibilità e di realizzabilità delle macchine l'insieme degli stati nei quali la macchina si può trovare deve essere un insieme finito.
L'importanza primaria che si assegna agli stati induce a chiamare taluni loro tipi "macchine a stati finiti".

Gli stati di una macchina si possono rappresentare come i nodi di un digrafo dotato di archi arricchiti [D28a]: l'arco che va da un nodo p a un secondo nodo q è munito delle informazioni che, proveniendo dall'esterno, inducono il cambiamento dello stato.
I cambiamenti di stato si possono descrivere come spostamenti di una entità spesso descritta in termini antropomorfi, chiamata **controllo della macchina**, che in ciascuno dei **passi evolutivi** che riconosciamo come componenti di una evoluzione si sposta da uno stato all'altro.

Riferendosi ai digrafi, si usa dire che il controllo si sposta da un nodo p a un nodo q percorrendo l'arco che li collega; può accadere che q coincida con q ed in un tale caso l'arco costituisce un cappio.

A questo proposito si parla anche di **transizioni tra stati della macchina**. Il termine transizione intuitivamente richiama la effettuazione di un passo evolutivo, mentre formalmente si definisce come una terna della forma $\langle p, C, q\rangle$, cioè in un arco $\langle p, q\rangle$ munito delle informazioni C sulle condizioni che possono comportare il passaggio dallo stato p allo stato q.

Questo digrafo arricchito lo chiamiamo **pluridigrafo delle transizioni** della macchina: esso infatti porta le stesse informazioni fornite dall'insieme delle suddette terne.

Va anche segnalato che spesso "pluridigrafo delle transizioni" viene semplificato con "digrafo delle transizioni" o con "grafo delle transizioni".

C12 a.03 Vari tipi di macchine sequenziali dispongono di **dispositivi di memoria**.

I più semplice tra questi dispositivi sono chiamati **registri** e in ciascuno di essi si possono inserire informazioni semplici, come caratteri, stringhe e numeri interi.

Dispositivi più articolati possono contenere e rendere disponibili complessi di informazioni che chiamiamo **strutture di dati**.
Questi complessi, evidentemente finiti, in linea di principio, si possono codificare mediante stringhe più o meno articolate, ma in discorsi interessati alla pratica, conviene formularli mediante informazioni più precisamente organizzate: tabelle, digrafi, espressioni che forniscono regole, ordini o condizioni.

Una macchina compie evoluzioni che nel caso più semplice sono rappresentate da sequenze di situazioni nelle quali essa si viene progressivamente a trovare e che abbiamo già chiamate "configurazioni".

Ogni configurazione è determinata dalle informazioni che sono registrate attualmente nei suoi dispositivi di memoria e, primariamente, dalla informazione che abbiamo chiamata "stato [interno]".
Lo stato di una macchina può pensarsi come dato memorizzato in un registro privilegiato o come nodo del corrispondente pluridigrafo delle transizioni.

In alcune descrizioni delle macchine si dice che possono essere dotate di "dispositivi di memoria illimitati": sarebbe opportuno che questa espressione fosse sostituita dall'affermazione che in ogni istante della sua evoluzione una macchina dispone di un insieme finito di dispositivi di memoria che in una fase successiva, se richiesto dall'andamento dell'elaborazione tive, possa essere esteso.

Questa caratteristica è preferibile chiamarla dotazione di **dispositivi di memoria illimitatamente ampliabili**.

La effettiva possibilità di ampliare la memoria per soddisfare le varie esigenze che si possono manifestare nelle situazioni reali, evidentemente, deve essere esaminata dai responsabili delle elaborazioni sulla base delle complessive risorse disponibili o acquisibili nelle singole situazioni concrete.

C12 a.04 Un dispositivo di memoria considerato basilare per molti tipi di macchine è costituito da un **nastro**, supporto formato da una sequenza di caselle in ciascuna delle quali è registrabile un segno da scegliere in un alfabeto ben definito.
In ogni istante solo un numero finito di caselle contiene registrazioni utilizzabili.

I più semplici tipi di macchine dispongono solo di nastri finiti; un tale supporto si può assimilare alla memoria fisica di un computer o a un array disponibile in un programma scritto in un linguaggio procedurale.

Le macchine dotate di nastri illimitatamente estendibili talora vengono chiamati "macchine infinite", espressione alla quale pensiamo preferibile anche una locuzione come "macchine dotate di nastri potenzialmente infiniti".

C12 a.05 L'evoluzione di una macchina, quindi, si considera costituita da una sequenza di eventi che si riscontrano in una successione discreta di istanti; tra due di questi istanti consecutivi non si prende in considerazione alcun accadimento ipotizzando che non si incontrino ostacoli al realizzarsi della transizione dettata dalle regole formali che caratterizzano la macchina.

Nella descrizione del comportamento di una macchina sequenziale che fa da modello per un'apparecchiatura o per un processo materiale non si tiene conto delle situazioni intermedie e neppure della possibilità che il sistema reale si trovi in una situazione non perfettamente coincidente con una rappresentata da uno stato attuale e dai contenuti degli altri eventuali dispositivi di memoria.

Naturalmente altri studi sui processi reali che si schematizzano con le evoluzioni di una macchina sequenziale $\boldsymbol{M}$ si possono porre i problemi dell'adeguatezza della macchina a rappresentare i processi. Altri studi possono riguardare l'adeguatezza e l'affidabilità dei dispositivi meccanici o informatici utilizzati per implementano la macchina $\boldsymbol{M}$ e ottenere sistematici risultati sulle sue evoluzioni.

Ogni evoluzione di una macchina $\boldsymbol{M}$ è dunque descritta interamente dalla sequenza dei **passi**, ovvero delle **mosse** che essa compie per portarsi sulle successive configurazioni, sequenza a sua volta dettata dalle regole caratterizzanti $\boldsymbol{M}$.

Nella teoria delle macchine formali anche il tempo viene trattato in modo discreto: si considera una successione di istanti $t_0, t_1, t_2, ..., t_n, ...$ (oppure 0, 1, ..., n, ...) che sono in stretta corrispondenza con la successione delle mosse della macchina. In corrispondenza biunivoca con gli istanti concernenti una evoluzione della macchina vi sono le configurazioni nelle quali la macchina si viene a trovare e che tipicamente scriviamo $\mathcal{C}_0,\ \mathcal{C}_1,\ \mathcal{C}_2, ..., \mathcal{C}_n, ...$.

Si chiede che la configurazione $\mathcal{C}_n$ sia in grado di fornire tutte le caratteristiche della macchina che influiscono sul suo comportamento nel passo $n+1$-esimo.

C12 a.06 La transizione di una macchina da una configurazione $\mathcal{C}_i$ alla successiva è determinata dalla $\mathcal{C}_i$ stessa e in taluni casi dalle influenze che sono esercitate sulla macchina dall'esterno.

Anche queste sono trattate in modo discreto e nei casi basilari sono rappresentate da stringhe di simboli appartenenti a un alfabeto ben determinato che diciamo **alfabeto di input**; le relative stringhe sono dette **stringhe di input**.

Una stringa di input si può pensare registrata su un nastro, chiamato **nastro di input**, il quale si dice che viene letto dalla macchina mediante un suo dispositivo chiamato **testina di lettura**.

I caratteri di input possono essere descritti come comandi inviati alla macchina da un operatore che riassume il suo esterno influente, oppure come informazioni che la macchina cattura dalla parte di 'ambiente estern sulla quale può effettuare osservazioni.

C12 a.07 Nei casi più semplici l'evoluzione di una macchina è determinata dalla sua configurazione iniziale e dalla sequenza finita dei caratteri di input che procede a ricevere (o a catturare) nei passi successivi.

Tale evoluzione, come si è detto, può essere letta esaurientemente dalla sequenza finita delle configurazioni che va assumendo e che scriviamo

$$\langle \mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{n-1}, \mathcal{C}_n \rangle \ .$$

Le configurazioni successive vengono riferite a successivi istanti individuati da t_0 (l'istante iniziale della evoluzione), da t_1, da t_2, ..., da t_{n-1} e da un t_n che, nei casi più semplici, costituisce l'istante finale della evoluzione.

Il passaggio da una configurazione $\mathcal{C}_{i-1}$ alla successiva $\mathcal{C}_i$ viene detta **passo dell'evoluzione** e va associato al carattere di input $\mathtt{a}_{j_i}$ che viene letto nell'istante t_{i-1} e che contribuisce a provocare il passaggio stesso.
In questo passaggio consiste la **transizione** dalla configurzione $\mathcal{C}_{i-1}$ lla $\mathcal{C}_i$.

Questo processo si può descrivere con il seguente schema

istante	0	1	2	. . .	$n-1$	n
carattere di input		$\mathtt{a}_1$	$\mathtt{a}_2$	. . .	$\mathtt{a}_n$	
configurazione	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	. . .	$\mathcal{C}_{n-1}$	$\mathcal{C}_n$

Consideriamo una evoluzione di una macchina che si arresta nella n-esima configurazione; essa è in grado di fornire una funzione esplicita, quella che a $\mathcal{C}_0$ e alla sequenza dei caratteri di input associa la configurazione $\mathcal{C}_n$.

Una tale macchina, in particolare consente di prendere una decisione (in particolare una accettazione o un rifiuto) sopra la sequenza dei caratteri di input, ricavabile dal processo volutivo.

C12 a.08 Un genere di evoluzione finita con maggiori prestazioni del precedente riguarda processi nei quali la macchina è in grado di emettere ad ogni passo una stringa di caratteri di output da registrare nelle successive caselle di un suo nastro adibito alla emissione.

Una tale evoluzione si descrive adeguatamente con il seguente schema più ricco, ovvero più completo, del precedente:

istante	0	1	2	. . .	$n-1$	n
carattere di input		$\mathtt{a}_1$	$\mathtt{a}_2$	. . .	$\mathtt{a}_n$	
configurazione	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	. . .	$\mathcal{C}_{n-1}$	$\mathcal{C}_n$
stringhe di output		$\mathtt{u}_1$	$\mathtt{u}_2$	. . .	$\mathtt{u}_n$	

Una macchina con questo genere di evoluzione è quindi in grado di fornire una funzione che (considerando $\mathcal{C}_0$ fisso e quindi ininfluente) ad ogni stringa immessa $\mathtt{a}_1\,\mathtt{a}_2 \ldots \mathtt{a}_n$ fa corrispondere la stringa emessa $\mathtt{u}_1\,\mathtt{u}_2 \ldots \mathtt{u}_n$.

C12 a.09 Tutte le macchine accennate finora hanno un cosiddetto **comportamento deterministico**, cioè seguono regole che garantiscono che stando in una definita configurazione si può avere una sola configurazione successiva o l'arresto.
Inoltre finora abbiamo prese in considerazione solo macchine per le quali si deve avere un arresto o dopo un numero finito di passi.

Si possono invece prospettare macchine per le quali si garantisce o non si esclude che possono effettuare quelle che diciamo **evoluzioni illimitate**, ossia evoluzioni deterministiche costituite da una sequenza di passi per la quale non è facilmente prevedibile una conclusione dopo un numero finito di passi.

Si possono dunque prospettare evoluzioni che si svolgono attraverso una sequenza di passi grande a piacere.

Si prospettno anche, in generale, evoluzioni per le quali è lasciata aperta la possibilità di giungere a configurazioni di arresto, cioè prive della possibilità di proseguire; per ciascuna evoluzione per assicurare o escludere il verificarsi di un arresto si rende necessario analizzare l'insieme dei possibili comportamenti della macchina, e va segnalato che questo insieme potrebbe essere difficilmente delimitabile e problematico.

Siamo quindi di fronte a diversi generi di evoluzioni: si può avere una evoluzione che giunge a una conclusione (che a sua volta potrebbe avere caratteristiche auspicate da chi è interessato alla evoluzione o all'opposto caratteristiche deprecate); si potrebbe invece avere una evoluzione in grado di proseguire senza mai giungere a un'arresto conclusivo; ma inoltre si potrebbe avere una evoluzione che a un certo punto mostra di poter proseguire ma non fa capire a un suo controllore se potrà proseguire fino a giungere a un arresto oppure avrà garantita la possibilità di proseguire illimitatamente, condizioni ambientali permettendolo.

Le più semplici di queste evoluzioni illimitate possono essere illustrate da schemi generici come il seguente nel quale non compaiono caratteri letti ma solo stringhe emesse nei passi successivi:

istante	0	1	2	$\ldots$	$n-1$	n	$\ldots\ldots$
configurazione	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\ldots$	$\mathcal{C}_{n-1}$	$\mathcal{C}_n$	$\ldots\ldots$
stringhe di output		$\mathtt{u}_1$	$\mathtt{u}_2$	$\ldots$		$\mathtt{u}_n$	$\ldots\ldots$

Un genere di evoluzioni illimitate, molto utile al livello della organizzazione di conoscenze generali, riguarda la possibilità (in linea di principio, risorse disponibili permettendolo) di proseguire quanto si vuole con la emissione di informazioni $\mathtt{u}_1\,\mathtt{u}_2\,...\,\mathtt{u}_n\,....$
Le successioni di questo tipo potrebbero riguardare gli elementi di insiemi numerabili di grande importanza come $\mathbb{N}$, come A^* per qualche alfabeto rilevante A, come l'insieme dei numeri primi, come l'insieme degli alberi [D26d12], o come l'insieme delle arborescenze di derivazione nell'ambito di una grammatica acontestuale [C14a02].

A questo punto è evidente che si pone il problema generale di stabilire per le varie macchine utilizzabili proficuamente quali tipi di evoluzione subirà a partire da sue possibili configurazioni iniziali: come vedremo si tratta di un problema impegnativo che si rivela impossibile risolvere nella sua generalità [C20, C21].

C12 b. riconoscitori di Rabin-Scott

C12 b.01 La più semplice tra le collezioni di macchine in grado di individuare linguaggi formali è la classe dei cosiddetti **riconoscitori di Rabin-Scott** o, in breve, **riconoscitori-RS**; queste macchine spesso sono chiamate anche riconoscitori a stati finiti.

La loro portata non è molto ampia, in quanto la varietà dei linguaggi da essi individuati è alquanto circoscritta.
Tuttavia questi linguaggi, che qui chiamiamo **linguaggi razionali**, possono essere controllati con procedimenti relativamente semplici ed efficaci che si servono di digrafi e di equazioni riguardanti linguaggi con buone proprietà algebriche.

I linguaggi razionali hanno molte notevoli applicazioni sia nell'ambito della organizzazione della teoria dei linguaggi formali [C30, C32], che all'interno delle attività per i linguaggi di programmazione e per l'elaborazione dei dati, come vedremo in C13.

Di essi inoltre si può ottenere una classificazione costruttiva e funzionale mediante la cosiddetta teoria algebrica della decomposizione dei semigruppi finiti dovuta a Kenneth Krohn e John Rhodes.

C12 b.02 Diciamo **riconoscitore di Rabin-Scott deterministico** o, in breve **riconoscitore-RSD**, una struttura della forma:

$$\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta \rangle \ ,$$

ove Q è un insieme finito detto **insieme degli stati**, $\imath \in Q$ è detto **stato iniziale**, $F \subseteq Q$ è detto insieme degli **stati finali**, T è un alfabeto detto **alfabeto di input** e $\delta \in \lceil Q \times \mathsf{T} \longmapsto Q \rceil$ è detta **funzione di transizione**.

Una macchina come $\boldsymbol{R}$ si può interpretare come modello formale di una macchina tangibile costituita da una testina di lettura che può scorrere in una sola direzione un nastro di input le cui caselle possono contenere solo simboli di T e da una unità centrale formata da un insieme finito di stati collegati secondo le indicazioni fornite dalla δ in modo da costituire un digrafo con archi etichettati da caratteri di T e con lo stato $\imath$ in evidenza.
Un riconoscitore-RSD può essere presentato completamente dal solo suddetto digrafo arricchito.

Un esempio come è il seguente:

C12 b.03 Ogni evoluzione di $\boldsymbol{R}$ inizia con il controllo nello stato iniziale e con la testina di lettura sulla casella più a sinistra del nastro di input su cui è registrata una stringa w che viene sottoposta al processo di riconoscimento a carico della macchina.

Ad ogni mossa la testina legge un simbolo $\mathtt{a}$ nella casella sulla quale si trova e il controllo passa dallo stato corrente q allo stato $\delta(q, \mathtt{a})$ fornito dalla funzione di transizione.

Le successive configurazioni toccate sono determinate, oltre che dai successivi stati, dall'avanzamento della testina di lettura sul nastro.

Se per la generica configurazione toccata usiamo una notazione del tipo $\langle q, v \rangle$, dove $q \in Q$ è lo stato attuale e v è il suffisso della stringa da riconoscere w, intendiamo segnalare che si fotografa l'istante nel quale la testina di lettura è collocata sulla prima casella di input contenente il suffisso v dopo aver letto il prefisso $u := w /\!\!/ v$.

Chiaramente l'evoluzione di un riconoscitore-RSD si conclude sempre dopo un numero di mosse pari alla lunghezza della stringa da analizzare con la testina sulla casella alla destra di quella contenente l'ultimo simbolo letto, ovvero con una configurazione $\langle q_f, \sqcup \rangle$.

C12 b.04 Tra le stringhe che si possono sottoporre ad $\boldsymbol{R}$ si privilegiano quelle la cui lettura porta dallo stato iniziale a uno degli stati finali. L'insieme di queste stringhe costituisce quello che si chiama il **linguaggio riconosciuto** o **linguaggio accettato** dalla macchina e che si denota con $\boldsymbol{R}^{\mathcal{A}}$.

Ogni evoluzione di un riconoscitore-RSD in linea di principio è facilmente controllabile e quindi si può decidere agevolmente se una data stringa w appartiene o meno ad $\boldsymbol{R}^{\mathcal{A}}$.

Un linguaggio accettabile da un riconoscitore di Rabin-Scott, che come si è preannunciato qui chiamiamo **linguaggio razionale**, più spesso viene detto anche linguaggio regolare o linguaggio riconoscibile a stati finiti.
Qui preferiamo "linguaggio razionale" in quanto a "razionale" può essere attribuito un significato più preciso di "regolare", mentre la terza dizione può essere attribuita ad ogni macchina con insieme di stati non ampliabile e finalizzata a riconoscere un linguaggio.

Nel seguito denoteremo con **Rcn** l'insieme di tutti i riconoscitori-RSD e con $\mathbf{Rcn}_{\mathsf{T}}$ l'insieme di quelli aventi T come alfabeto di input.

L'insieme dei linguaggi razionali sull'alfabeto T si denota con $\mathbf{Rcn}_{\mathsf{T}}{}^{\mathcal{A}}$, mentre l'insieme dei linguaggi razionali su un alfabeto qualsiasi si denota con $\mathbf{Rcn}^{\mathcal{A}}$.

C12 b.05 La funzione di transizione δ si presenta comodamente come **matrice di transizione**, matrice con le righe associate agli stati, le colonne ai simboli di input e avente nella casella che appartiene alla riga relativa allo stato q ed alla colonna relativa al simbolo a lo stato $\delta(q, \mathtt{a})$.

Osserviamo che questa matrice consente di implementare agevolmente i riconoscitori-RSD e di simulare con un semplice programma i loro comportamenti.

Per presentare un riconoscitore-RSD a una persona conviene servirsi di una raffigurazione del suo digrafo delle transizioni, cioè di un digrafo arricchito opportunamente.

C12 b.06 Si dice **pluridigrafo** [v.a. D28d] una struttura della forma $\boldsymbol{P} = \langle Q, \mathsf{T}, \mathcal{U} \rangle$, dove Q è un insieme finito detto **insieme dei nodi** o **insieme degli stati**, T un alfabeto detto **alfabeto delle etichette**, ed $\mathcal{U} \in \left\lceil \mathsf{T} \longmapsto \mathfrak{P}(Q \times Q) \right\rceil$, funzione chiamata **etichettatura dei nodi**.

Ad ogni $\mathtt{a} \in \mathsf{T}$ risulta associato il digrafo $\langle Q, \mathcal{U}(\mathtt{a}) \rangle$; le coppie che costituiscono $\mathcal{U}(\mathtt{a})$ si dicono **archi del plurigrafo etichettati dalla lettera** a.

Le terne $\langle p, \mathtt{a}, q \rangle$ tali che $\langle p, q \rangle \in \mathcal{U}(\mathtt{a})$ si dicono **transizioni del plurigrafo**; **transizione indotta da** a, o in breve transizione-a.

Il plurigrafo $\boldsymbol{P} = \langle Q, \mathsf{T}, \mathcal{U} \rangle$ quindi si può ottenere sovrapponendo le raffigurazioni dei digrafi che lo compongono, digrafi che presentano lo stesso insieme di nodi, dopo aver etichettati con $\mathtt{a}$ tutti gli archi provenienti da ciascuno degli $\mathcal{U}(\mathtt{a})$.

C12 b.07 Si dice **pluridigrafo inizializzato-finalizzato** o plurigrafo di transizione ogni $\boldsymbol{P} = \langle Q, I, F, \mathsf{T}, \mathcal{U} \rangle$, dove $\langle Q, \mathsf{T}, \mathcal{U} \rangle$ è un pluridigrafo, mentre I ed F sono due sottoinsiemi di Q detti, risp., insieme degli stati iniziali e insieme degli stati finali.

I pluridigrafi e i pluridigrafi di transizione, in quanto arricchimenti di digrafi, "ereditano", ossia possono utilizzare, le nozioni, le notazioni e i termini attinenti ai digrafi [B16b, B16c, B53c]; in particolare su queste strutture si possono prendere in considerazione passeggiate aperte e chiuse.

Dato che su ogni loro arco si ha un'etichetta, su ogni passeggiata si può leggere una stringa sull'alfabeto di output.

C12 b.08 Definiamo ora il pluridigrafo e il pluridigrafo di transizione di un riconoscitore-RSD $\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta \rangle$.
Questo pluridigrafo ha come nodi gli stati di $\boldsymbol{R}$ e ha sistemi di archi funzionali etichettati dai simboli dell'alfabeto di input T: precisamente per $\mathtt{a} \in \mathsf{T}$ da ogni nodo $q \in Q$ esce l'arco etichettato $\big\langle q, \mathtt{a}, \delta(q, \mathtt{a}) \big\rangle$.

Il **pluridigrafo di transizione** di $\boldsymbol{R}$ si ottiene arricchendo il suo pluridigrafo con lo stato iniziale $\imath$ e con l'insieme F degli stati finali di $\boldsymbol{R}$.

Nelle raffigurazioni dei pluridigrafi di transizione contrassegneremo lo stato iniziale con un segno "–" e gli stati finali con un "+".

Il processo di analisi di una stringa $w \in \mathsf{T}^*$ può vantaggiosamente riferirsi alla passeggiata sul pluridigrafo di transizione che inizia nel nodo iniziale $\imath$ ed è formato dagli archi relativi alle transizioni indotte dalla lettura dei successivi simboli letti dal nastro di input.

Il linguaggio accettato da $\boldsymbol{R}$ si ottiene considerando la totalità delle cosiddette **passeggiate utili** sul pluridigrafo, passeggiate che vanno dallo stato iniziale a uno stato finale, e leggendo su ogni passeggiata i simboli incontrati successivamente sugli archi che la compongono.

C12 b.09 Un primo esempio di riconoscitore-RSD è:

$$\boldsymbol{R}_a := \big\langle \{0,1,2,3,4\}, \{0\}, \{2,3\}, \{\mathtt{a},\mathtt{b}\}, \delta_a \big\rangle$$

ove δ_a è data dalla seguente matrice:

	a	b
0	1	2
1	2	3
2	3	4
3	4	4
4	4	4

Il corrispondente grafo di transizione è:

Una evoluzione che porta dallo stato iniziale a uno finale è la seguente:

$$\langle 0, \mathtt{aaa} \rangle \vdash \langle 1, \mathtt{aa} \rangle \vdash \langle 2, \mathtt{a} \rangle \vdash \langle 3, \sqcup \rangle.$$

Qui abbiamo usato il simbolo "$\vdash$" per denotare la relazione funzionale che associa a una configurazione di una macchina deterministica la successiva.

Quando si considerano diversi riconoscitori come $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$, è opportuno usare scritture più complete e circostanziate della forma $\vdash_{\boldsymbol{R}_i}$.

La suddetta evoluzione corrisponde al cammino $\big\langle [0,1,2,3] \big\rangle$. Tutte le altre passeggiate che danno stringhe accettate sono $\langle [0,2] \rangle$, $\langle [0,1,2] \rangle$, $\langle [0,1,3] \rangle$ e $\langle [0,2,3] \rangle$. Il linguaggio accettato da questo riconoscitore è quindi ${\boldsymbol{R}_a}^{\mathcal{A}} = \{\mathtt{b}, \mathtt{a}^2, \mathtt{ab}, \mathtt{ba}, \mathtt{a}^3\}$.

C12 b.10 Un secondo esempio di riconoscitore-RSD è:

$$\boldsymbol{R}_b := \big\langle \{0,1,2,3\}, \{0\}, \{0,3\}, \{\mathtt{a},\mathtt{b}\}, \delta_b \big\rangle,$$

ove per δ_b:

	a	b
0	1	2
1	2	3
2	2	2
3	3	2

Esso ha come grafo di transizione:

C12 b.11 Si osservi che di $\boldsymbol{R}_b{}^{\mathcal{A}}$ fa parte la stringa muta $\sqcup$; questo corrisponde al fatto che lo stato iniziale è anche finale. In effetti in generale vale quanto segue:

Prop. La stringa muta è accettata da un riconoscitore-RSD sse il suo stato iniziale è anche finale.

Dim.: Per un riconoscitore-RSD la stringa muta viene accettata sse sul pluridigrafo di transizione si trova un cammino utile di lunghezza 0 ∎

Altre stringhe accettate da $\boldsymbol{R}_b$ sono: `ab, aba, abaa, abaaa`. In complesso si trova:

$$\boldsymbol{R}_b{}^{\mathcal{A}} = \{\sqcup\} \dot\cup \{n \in \mathbb{N} :| \texttt{aba}^n\} \ ,$$

Osserviamo che $\boldsymbol{R}_b{}^{\mathcal{A}}$, contrariamente a $\boldsymbol{R}_a{}^{\mathcal{A}}$, è un linguaggio infinito.

C12 b.12 Prop. Il linguaggio accettato da un riconoscitore $\boldsymbol{R}$ è infinito sse nel suo pluridigrafo di transizione si trova un circuito che tocca uno stato che appartiene a una passeggiata utile.

Dim.: "Solo se" Se $\boldsymbol{R}^{\mathcal{A}}$ è infinito deve contenere una stringa con un numero di simboli grande quanto si vuole e in particolare non inferiore al numero degli stati di $\boldsymbol{R}$. Ad una tale stringa corrisponde una passeggiata utile sul digrafo che deve presentare almeno un nodo ripetuto q_c (e che potrebbe ridursi ad un cappio etichettato. In tal caso la passeggiata relativa a una tale stringa si può suddividere in tre parti: una parte iniziale dal nodo iniziale al nodo q_c, un circuito da q_c a q_c e una parte finale da q_c a un nodo finale.

"Se" Se esiste una passeggiata utile con un nodo q_c appartenente ad un circuito, denotiamo con v la stringa che si legge sulla passeggiata tra il nodo iniziale e q_c, con w la stringa che si legge sul circuito e con y la stringa che si legge sulla passeggiata da q_c al nodo finale. È chiaro che tutte le stringhe della forma vw^ny con $n \in \mathbb{N}$ appartengono ad $\boldsymbol{R}^{\mathcal{A}}$, che quindi è un linguaggio infinito ∎

C12 b.13 Si constata facilmente che un linguaggio razionale può essere individuato da vari riconoscitori-RSD e si osserva che la definizione di queste macchine non esclude che esse abbiano stati e transizioni ridondanti, la cui eliminazione non porta a riduzioni del linguaggio accettato.

Queste situazioni di ridondanza si riscontrano in generale per tutti i vari tipi di strumenti che hanno lo scopo di individuare linguaggi formali, siano essi macchine formali, insiemi di espressioni sistemi di equazionie o tipi di sistemi di regole.
In effetti, come vedremo, questo fenomeno è tanto più marcato quanto più elaborati sono i tipi di strumenti; in particolare vedremo tra breve le ridondanze che si riscontrano per i riconoscitori nondeterministici.

Si pone quindi il problema di trovare strumenti (in particolare riconoscitori) con pregi particolari; per taluni di questi oggetti si dice che si trovano in una qualche **forma canonica** o **forma normale**.

C12 b.14 Per un riconoscitore-RSD tutti gli stati che non si trovano su una passeggiata utile, che chiameremo **stati palesemente inutili**, possono essere eliminati, insieme ai relativi archi, senza ridurre l'insieme delle passeggiate che forniscono le stringhe del linguaggio riconosciuto.

Gli stati palesemente inutili sono facilmente individuabili e la suddetta semplificazione si può attuare agevolmente.

Due esempi di questi stati sono lo stato 4 in $\boldsymbol{R}_a$ e lo stato 2 in $\boldsymbol{R}_b$.
In questo caso e in generale il digrafo privato di questi stati è un digrafo connesso e il suo stato iniziale è una sua radice, cioè un nodo dal quale sono raggiungubili tutti gli altri.

Eliminando tutti gli stati palesemente inutili, però, può cadere il carattere deterministico in senso stretto, in quanto si possono avere nodi privi di archi uscenti etichettati da una parte dei caratteri di T.

Si può mantenere il carattere deterministico di un riconoscitore-RSD introducendo un cosiddetto **stato trappola**, stato nonfinale dal quale escono solo archi che sono dei cappi; più precisamente da uno stato trappola escono $|\mathsf{T}|$ cappi, ciascuno etichettato da un diverso carattere di T.

Da un qualsiasi riconoscitore di Rabin-Scott deterministico si può ottenere il riconoscitore deterministico equivalente attraverso l'eliminazione di ogni stato non raggiungibile dallo stato iniziale e la fusione in un solo stato trappola tutti gli stati raggiungibili dall'iniziale ma dai quali non si può raggiungere alcuno stato finale.
Questo accade ad esempio per lo stato 2 di $\boldsymbol{R}_b$.

Occorre segnalare che in genere nelle raffigurazioni dei pluridigrafi di transizione lo stato trappola viene trascurato.

C12 b.15 Presentiamo ora graficamente alcuni riconoscitori in grado di individuare scritture numeriche posizionali di interi particolari.

!pinput pC12b15

!pinput pC12b15B

Il primo dei precedenti riconoscitori accetta le scritture binarie degli interi naturali pari; il linguaggio riconosciuto dal secondo riguarda le scritture decimali degli interi positivi pari; il terzo riconoscitore accetta le scritture decimali degli interi positivi divisibili per 5.

C12 b.16 Proponiamo alcuni facili esercizi per prendere confidenza con queste macchine.

(1) Eserc. Tracciare un riconoscitore-RSD che accetta le stringhe costituite dalle lettere `a` e `b` e che non presentano due `a` ripetute, né tre `b` ripetute.

(2) Eserc. Tracciare un riconoscitore=RSD che accetta le stringhe che iniziano con alcune occorrenze della lettera `a`, proseguoo con occorrenze della lettera `b` e che si concludono con caratteri `c` oppure con caratteri `d`.

(3) Eserc. - Tracciare il digrafo delle transizioni che riconosce le stringhe costituite da infissi della forma $[\mathtt{a}^+]$ oppure della forma $(\mathtt{b}^2\mathtt{c}^3)$.

C12 b.17 Il seguente riconoscitore accetta le scritture decimali degli interi positivi divisibili per 3:

Un riconoscitore che accetta le scritture decimali degli interi positivi divisibili per 4 è rappresentato come segue (in queste figure "d" rappresenta le cifre dispari, "p" le cifre pari e "ν" le cifre arbitrarie):

C12 b.18 Un riconoscitore che individua le riflesse delle scritture decimali degli interi positivi divisibili per 4 viene raffigurato da:

Si noti quanto esso sia più semplice del precedente: questo è dovuto al fatto che le due cifre analizzate hanno un ruolo chiaramente determinato, mentre nel caso dell'analisi da sinistra a destra il ruolo delle cifre dipende da quanto segue e quindi il giudizio su di esse deve rimanere sospeso fino alla conclusione della lettura.

Il seguente riconoscitore accetta invece le stringhe riflesse delle scritture decimali dei numeri interi positivi divisibili per 8.

C12 b.19 (1) Eserc. Precisare dei riconoscitori atti a individuare le scritture decimali degli interi positivi divisibili per 7 e per 11.

Eserc. 2 Delineare riconoscitori che accettano le scritture che esprimono numeri reali in linguaggi di programmazione procedurale come BASIC, Fortran, C e Java.

Eserc. 3 Delineare un riconoscitore che accetti i numeri dei telefoni fissi italiani e uno per gli internazionali.

C12 c. riconoscitori nondeterministici

C12 c.01 Per capire e maneggiare meglio i linguaggi razionali occorre introdurre una generalizzazione dei riconoscitori-RSD che a prima vista sembrerebbe in grado di individuare una classe di linguaggi più ampia, ma che in realtà fornisce solo strumenti più flessibili per trattare gli stessi linguaggi.

Finora abbiamo esaminato le cosiddette **macchine deterministiche**, in quanto in ogni istante dell'evoluzione di uno di essi la lettura del simbolo sul quale è posizionata la testina comporta la transizione dallo stato attuale a un unico stato successivo; in altre parole il passaggio da una sua configurazione ad una successiva in conseguenza della lettura di un carattere è determinata senza incertezze.

In molte circostanze si rivelano vantaggiose macchine con comportamenti più "elastici": qui tratteremo le macchine nondeterministiche; di grande interesse sono anche le macchine probabilistiche collegati alle catene di Markov [C65e].

Un riconoscitore di Rabin-Scott di tipo nondeterministico è associato a un pluridigrafo di transizione generale, nel quale cioè si possono avere più stati iniziali, sistemi di archi non necessariamente funzionali e archi etichettati dalla stringa muta $\sqcup$.
Un riconoscitore di questo tipo consente di individuare il linguaggio costituito dalle stringhe che si leggono scorrendo le sue passeggiate utili, cioè ancora le passeggiate che vanno da uno degli stati iniziali a uno degli stati finali.

C12 c.02 Diciamo dunque **riconoscitore di Rabin-Scott nondeterministico**, o in breve **riconoscitore-RSND**, una struttura della forma:

$$\boldsymbol{R} = \langle Q, I, F, \mathsf{T}, \delta \rangle ,$$

ove Q è un insieme finito detto **insieme degli stati del riconoscitore**, $I \subseteq Q$ è detto **insieme degli stati iniziali** , $F \subseteq Q$ è detto insieme degli **stati finali** , T è un alfabeto detto **alfabeto di input** e
$\delta \in \lceil Q \times (\mathsf{T} \dot\cup \sqcup) \longmapsto \mathfrak{P}(Q) \rceil$ è detta **relazione di transizione**.

L'interpretazione meccanicistica dell'evoluzione di uno di questi riconoscitori è solo poco meno intuitiva di quella di un riconoscitore-RSD.

Una configurazione interna del riconoscitore-RSND è determinata da più stati contemporaneamente attivi.

Per immaginare tale situazione può essere utile pensare che a ogni stato di una macchina sequenziale qualsiasi corrisponda una spia luminosa. L'evoluzione di una macchina deterministica si manifesta con l'accendersi di singole luci negli istanti successivi di una evoluzione; in un passo evolutivo di una macchina nondeterministica, invece, si può assistere a un succedersi di accensioni e spegnimenti di più spie luninose.

Osserviamo anche che un riconoscitore-RSND è caratterizzato da una matrice di transizione con una colonna corrispondente alla stringa muta e con caselle che possono contenere nessuno, uno o più stati.

Per **transizione di un riconoscitore-RSND** $\boldsymbol{R} = \langle Q, I, F, \mathsf{T}, \delta \rangle$ si intende una terna

$$\langle p, \mathtt{a}, q \rangle \in Q \times (T \dot\cup \sqcup) \times Q \quad \text{tale che } q \in \delta(p, \mathtt{a}) .$$

Anche un riconoscitore-RSND è individuato dall'insieme delle sue transizioni e dagli insiemi dei suoi stati iniziali e finali.

C12 c.03 Evidentemente l'insieme dei riconoscitori-RSD si può considerare un sottoinsieme proprio piuttosto circoscritto dell'insieme dei riconoscitori-RSND.

Diciamo anche, in generale, che due riconoscitori-RSND sono **riconoscitori equivalenti-L** sse individuano lo stesso linguaggio.
Più in generale diciamo che due strumenti in grado di individuare linguaggi formali sono equivalenti-L sse individuano lo stesso linguaggio.

Ci proponiamo quindi di dimostrare l'equivalenza-L dei due tipi di riconoscitori di Rabin-Scott sopra introdotti; per questo basta dimostrare che ogni riconoscitore-RSND può essere sostituito da un riconoscitore-RSD equivalente-L.

Prop. Ogni riconoscitore-RSND si può trasformare in un riconoscitore-RSD equivalente-L.

Dim.: Descriveremo in termini di pluridigrafi di transizione un procedimento che permette di trasformare ogni riconoscitore-RSND $\boldsymbol{R}$ in uno equivalente-L deterministico.
In una prima parte mostriamo come si può eliminare da $\boldsymbol{R}$ una generica transizione etichettata dalla $\sqcup$; nella seconda mostreremo come si può trasformare un riconoscitore-RSND privo dei suddetti archi in un riconoscitore-RSD equivalente-L.

Osserviamo preliminarmente che un riconoscitore-RSND si può arricchire con nuove particolari transizioni senza modificare il linguaggio accettato L: si tratta di aggiungergli transizioni $\langle q_0, \sqcup, q_r\rangle$ prima assenti, in corrispondenza di ogni passeggiata da q_0 a q_r con archi etichettati da $\sqcup$ e successivamente aggiungere ai suoi stati iniziali gli stati raggiungibili da precedenti stati iniziali con transizioni-$\sqcup$ e aggiungere ai suoi stati finali gli stati dai quali questi si possono raggiungere attraverso transizioni-$\sqcup$.

L viene fornito dalle passeggiate utili di questo nuovo riconoscitore, che non presentano transizioni-$\sqcup$ iniziali, ripetute e finali; queste passeggiate le diciamo passeggiate ridotte-$\sqcup$.
In una fase successiva si procede a modificare il riconoscitore eliminando una per una le sue transizioni-$\sqcup$

Descriviamo la sottofase nella quale si elimina la transizione $\langle p, \sqcup, q\rangle$; chiamiamo $\langle p_i, \mathtt{a}_i, p\rangle$ le transizioni con estremità finale p e $\langle q, \mathtt{b}_j, q_j\rangle$ le transizioni con estremità iniziale q, dove le $\mathtt{a}_i$ e le $\mathtt{b}_j$ sono caratteri qualsiasi di T.

Si può eliminare $\langle p, \sqcup, q\rangle$ pur di aggiungere le varie transizioni $\langle p_i, \mathtt{a}_i, q\rangle$ e le $\langle p, \mathtt{b}_j, q_j\rangle$, qualora manchino: infatti ogni passeggiata contenente l'arco-$\sqcup$ trascurato può essere rimpiazzata con qualche passeggiata con lunghezza diminuita di 1 contenenti uno dei nuovi archi.

Il riconoscitore privo di transizioni-$\sqcup$ così ottenuto lo denotiamo con $\boldsymbol{R}'$.

Con una nuova fase si trasforma $\boldsymbol{R}'$ nel riconoscitore-RSD equivalente-L che denotiamo con $\boldsymbol{R}''$.
Si tratta di definire come stati di $\boldsymbol{R}''$ alcune collezioni di stati di $\boldsymbol{R}'$.

Come stato iniziale di $\boldsymbol{R}''$ si assume l'intero insieme degli stati iniziali di $\boldsymbol{R}'$ e lo si chiama I; come stato di $\boldsymbol{R}''$ ottenuto da I in seguito alla lettura di $\mathtt{a} \in \mathsf{T}$ si assume l'insieme $\cup_{q\in I}\delta(q, \mathtt{a})$.

Per quanto riguarda gli stati ottenibili con la lettura di stringhe di più simboli di T conviene ampliare la definizione della relazione di transizione definendo ricorsivamente:

$$\forall w \in \mathsf{T}^+,\ \mathtt{a} \in \mathsf{T},\ q \in Q \quad : \quad \delta(q, w\,\mathtt{a}) := \bigcup_{r\in\delta(q,w)} \delta(r, \mathtt{a})\ .$$

Procedendo a considerare sempre nuove stringhe accade di ritrovare stati di $\boldsymbol{R}''$ già incontrati e l'ampliamento dell'insieme degli stati di $\boldsymbol{R}''$ ha sicuramente termine, in quanto, ovviamente, il numero dei sottoinsiemi dell'insieme finito Q è finito.
Osserviamo anche che il procedimento descritto fornisce direttamente la funzione di transizione di $\boldsymbol{R}''$.

Conclusa questa fase, si assume come insieme degli stati finali di $\boldsymbol{R}''$ la collezione di tutti i nuovi stati contenenti almeno uno degli stati finali di $\boldsymbol{R}'$.

Ogni stringa w del linguaggio accettato da $\boldsymbol{R}''$ si legge su una passeggiata che corrisponde a passeggiate etichettate dalla stessa w sul pluridigrafo di $\boldsymbol{R}'$; tutte queste passeggiate partono da un nodo iniziale di $\boldsymbol{R}'$ ed almeno uno raggiunge uno stato finale di $\boldsymbol{R}'$: quindi

$$\boldsymbol{R}''^{\mathcal{A}} \subseteq \boldsymbol{R}'^{\mathcal{A}} \ .$$

Viceversa ogni stringa accettata da $\boldsymbol{R}'$ corrisponde a una passeggiata utile su $\boldsymbol{R}'$ a cui si associa una passeggiata utile su $\boldsymbol{R}''$. Quindi

$$\boldsymbol{R}'^{\mathcal{A}} \subseteq \boldsymbol{R}''^{\mathcal{A}} \ .$$

Si può quindi assicurare che i due riconoscitori sono equivalenti-L ∎

C12 c.04 Osserviamo che con le precedenti trasformazioni si ottengono riconoscitori con il pregio di essere deterministici, ma che possono essere formati da un numero molto maggiore di stati e di archi del riconoscitore di partenza: il numero degli stati potrebbe passare da un intero k a un intero che si avvicina a 2^k.

Quindi si viene ad adottare uno strumento che presenta una struttura che gode di una proprietà semplice da enunciare, ma che richiede un numero di nodi che può essere sensibilmente maggiore.

Questo è una caratteristica comune alle trasformazioni che da uno strumento formale che presenta requisiti poco stringenti ne ricavano uno soggetto a regole espresse in forme più semplici, ma più stringenti.
Queste ultime caratteristiche sono tipiche delle cosiddette forme canoniche o delle cosiddette forme normali; queste forme tendenzialmente hanno il vantaggio di poter essere elaborate con procedimenti di concezione più semplice e più incisivi.

L'elemento cruciale della costruzione precedente è il mantenimento della finitezza degli stati del riconoscitore.

A questo proposito occorre osservare che gli stati di un riconoscitore si possono considerare un dispositivo di memoria per il processo di riconoscimento delle stringhe che gli vengono proposte.

Quando nel corso dell'analisi di una di queste stringhe candidate da parte di un riconoscitore-RSD si è letto un suo prefisso x ed il controllo è giunto in un certo stato q, è solo questo stato che tiene traccia dei simboli letti.

La capacità di discriminare le stringhe sottoposte a lettura dei riconoscitori di Rabin-Scott si basa sugli stati cha fungono da dispositivi di memoria e, come vedremo, trova nella loro finitezza i suoi limiti.

C12 c.05 Tra i linguaggi razionali individuati da riconoscitori privi di stati palesemente inutili vogliamo ora trovare elementi distintivi riguardanti la presenza o meno di uno stato trappola.

Prop. Consideriamo un linguaggio razionale L. L è accettato da un riconoscitore deterministico privo di stato trappola $\iff$ L è un linguaggio cometa della forma $\mathsf{L} = \mathsf{T}^*\mathsf{N}$ con N linguaggio razionale.

Dim.: "$\Longrightarrow$" Sia $\boldsymbol{R}$ un riconoscitore deterministico privo di passeggiate palesememente inutili e privo di stato trappola che riconosce L. Sottoponendogli una qualsiasi stringa $x \in \mathsf{T}^*$ si fa passare il controllo

dallo stato iniziale a uno stato q tale che esiste una passeggiata κ che porta da esso a uno stato finale. Se y è la stringa leggibile su κ, $x\,y \in \mathsf{L}$. Per l'arbitrarietà di x si ha $\mathsf{L} = \mathsf{T}^*\mathsf{N}$ con N accettato dal riconoscitore nondeterministico ottenuto modificando $\boldsymbol{R}$ solo imponendo che ogni suo stato sia iniziale.

"$\Longleftarrow$" Se $\mathsf{L} = \mathsf{T}^*\mathsf{N}$ con N razionale, una qualsiasi stringa $x \in \mathsf{T}^*$ deve essere prefisso di una stringa $x\,y \in \mathsf{L}$; quindi la sua lettura da parte di un riconoscitore che accetta L non può condurre a uno stato trappola ∎

C12 c.06 Eserc. Ricavare riconoscitori-RSD equivalenti-L a quelli forniti dalle seguenti matrici di transizione

	a	b
$1\,-$	1	2
2	3	2
$3\,+$	$\{1,3\}$	$\emptyset$

	a	b
$1\,-$	2	$\emptyset$
$2\,\pm$	$\{2,4\}$	3
3	$\emptyset$	4
$4\,+$	3	2

	a	b	c
$1\,-$	$\emptyset$	3	$\{1,3\}$
$2\,-$	2	$\emptyset$	$\{1,3\}$
$3\,+$	$\{1,2\}$	$\emptyset$	3

C12 c.07 (1) Eserc. Dimostrare che un linguaggio razionale è accettato da un riconoscitore deterministico con stati finali dai quali non esce alcun arco che vada in uno stato nonfinale sse è una anticometa della forma $\mathsf{L} = \mathsf{N}\,\mathsf{T}^*$ con N razionale.

(2) Eserc. Dare una caratterizzazione di riconoscitori in grado di accettare linguaggi razionali che sono bicomete.

C12 d. composizioni di linguaggi razionali

C12 d.01 In questa sezione vediamo come a partire da riconoscitori di Rabin-Scott si possano costruire nuovi riconoscitori in grado di accettare semplici composizioni dei linguaggi accettati dai riconoscitori di partenza.

A questo scopo dovremo considerare coppie di riconoscitori $\boldsymbol{R}_i := \langle Q_i, I_i, F_i, \mathsf{T}, \delta_i \rangle$ per $i = 1, 2$ e i linguaggi da essi accettati $\mathsf{L}_i := \boldsymbol{R}_i{}^{\mathcal{A}}$.

Se occorre partire da qualche riconoscitore deterministico, qualche insieme $I_i = \{\imath_i\}$ può essere sostituito da $\imath_i$.

Nel seguito, per semplicità, assumiamo che i due insiemi Q_1 e Q_2 degli stati delle nostre macchine formali siano disgiunti; questa assunzione è del tutto lecita a causa della natura puramente strumentale degli stati dei riconoscitori.

C12 d.02 Prop. L'unione $\mathsf{L}_1 \cup \mathsf{L}_2$ di due linguaggi razionali è un linguaggio razionale.

Dim.: Dati due riconoscitori $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$ atti ad accettare, risp., i due linguaggi, per avere un riconoscitore (nondeterministico) che accetti il linguaggio loro unione basta considerare l'unione disgiunta dei due corrispondenti digrafi di transizione.

Il riconoscitore così ottenuto è individuato dalla espressione

$$\langle Q_1 \dot\cup Q_2 \ , \ I_1 \dot\cup I_2 \ , \ F_1 \dot\cup F_2 \ , \ \mathsf{T} \ , \ \delta_1 \dot\cup \delta_2 \rangle \ \blacksquare$$

Il precedente riconoscitore viene detto **composizione in parallelo** di $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$.

Si tratta chiaramente di una composizione commutativa.

C12 d.03 Prop. La giustapposizione $\mathsf{L}_1 \cdot \mathsf{L}_2$ di due linguaggi razionali è un linguaggio razionale.

Dim.: Consideriamo il riconoscitore-RSND ottenuto mettendo assieme le transizioni di due riconoscitori $\boldsymbol{R}_i$ tali che $\boldsymbol{R}_i^{\mathcal{A}} = \mathsf{L}_i$ per $i = 1, 2$, assumendo come insieme degli stati iniziali I_1, come insieme degli stati finali F_2 e aggiungendo le transizioni $\langle f_h, \sqcup, i_k \rangle$ per ogni $f_h \in F_1$ e ogni $i_k \in I_2$.

Questo riconoscitore individua $\mathsf{L}_1 \cdot \mathsf{L}_2$, in quanto tutte le sue passeggiate utili sono giustapposizioni di una passeggiata utile su $\boldsymbol{R}_1$, di una delle precedenti transizioni-$\sqcup$ e da una passeggiata utile su $\boldsymbol{R}_2$ $\blacksquare$

Il precedente riconoscitore viene detto **composizione in serie** di $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$.

Questa composizione è commutativa sse i due linguaggi coincidono.

C12 d.04 Prop. Ogni stringa, comprese stringa muta e singoli caratteri in T, costituisce un linguaggio razionale.

Dim.: Un riconoscitore per la stringa muta è dato da un solo stato che è contemporaneamente iniziale e finale e da nessun arco.

Il simbolo $\mathtt{a}_i$ è riconosciuto da:

Un riconoscitore per una stringa $w = \mathtt{a}_{j_1}\mathtt{a}_{j_2}...\mathtt{a}_{j_r}$ si ottiene con un pluridigrafo di transizione costituito da una sola catena formata dalle transizioni $\langle q_{i-1}, \mathtt{a}_{j_i}, q_i \rangle$ per $i = 1, ..., r$ e avente come unico stato iniziale q_0 e come solo stato finale q_r $\blacksquare$

In altre parole il riconoscitore di una stringa con due o più caratteri si ottiene dalla composizione in serie dei riconoscitori dei successivi caratteri componenti.

C12 d.05 Prop. Tutti i linguaggi finiti sono razionali.

Dim.: Ogni linguaggio finito si può considerare unione finita delle sue stringhe ∎

C12 d.06 Prop. Il complemento $\mathsf{T}^* \setminus \mathsf{L}$ di un linguaggio razionale L su T è razionale.

Dim.: Si consideri il riconoscitore-RSD $\boldsymbol{R} := \langle Q, \imath, F, \mathsf{T}, \delta \rangle$ in grado di accettare L e privo di stati palesemente inutili.
Il riconoscitore $\langle Q, \imath, Q \setminus F, \mathsf{T}, \delta \rangle$ nel quale si scambiano gli stati finali e i non inali, chiaramente, accetta le stringhe che il precedente rifiuta e rifiuta quelle che $\boldsymbol{R}$ accetta, cioè individua $\mathsf{T}^* \setminus \mathsf{L}$ ∎

C12 d.07 Prop. La intersezione di due linguaggi razionali L_1 ed L_2 è un linguaggio razionale.

Dim.: Trascuriamo il fatto che questa proposizione si può ricavare dalle due precedenti in virtù della

$$\mathsf{L}_1 \cap \mathsf{L}_2 \;=\; \mathsf{T}^* \setminus ((\mathsf{T}^* \setminus \mathsf{L}_1) \cup (\mathsf{T}^* \setminus \mathsf{L}_2))$$

e costruiamo un riconoscitore significativo per $\mathsf{L}_1 \cap \mathsf{L}_2$ a partire dai riconoscitori dei linguaggi intersecandi $\boldsymbol{R}_1$ e $\boldsymbol{R}_2$.

Se per $i = 1, 2$ $\boldsymbol{R}_i := \langle Q_i, \imath_i, F_i, \mathsf{T}, \delta_i \rangle$ è in grado di accettare L_i, consideriamo

$$\boldsymbol{R} := \langle Q_1 \times Q_2, \langle \imath_1, \imath_2 \rangle, F_1 \times F_2, \mathsf{T}, \delta \rangle \ ,$$

dove $\delta(a, \langle q_1, q_2 \rangle) := \langle \delta(a, q_1), \delta(a, q_2) \rangle$.

Ad ogni passeggiata sul nuovo riconoscitore $\boldsymbol{R}$ che inizia in $\langle \imath_1, \imath_2 \rangle$ corrisponde una coppia di passeggiate su $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$ che iniziano, risp., in $\imath_1$ e $\imath_2$ e sono etichettate dalla stessa stringa. Questa corrispondenza è biunivoca e a una passeggiata utile su $\boldsymbol{R}$ fa corrispondere una coppia di passeggiate utili su $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$.
Quindi una stringa è accettata da $\boldsymbol{R}$ sse appartiene sia ad L_1 che ad L_2 ∎

C12 d.08 Prop. La chiusura-cross o chiusura di giustapposizione di un linguaggio razionale su T è razionale.

Dim.: Si considerino il riconoscitore-RSD $R := \langle Q, \imath, F, \mathsf{T}, \delta \rangle$ ed il linguaggio che esso accetta L.
Il riconoscitore-RSND ottenuto aggiungendo al precedente le transizioni $\langle f_h, \sqcup, \imath \rangle$ per ogni $f_h \in F$ riconosce tutte e sole le stringhe della chiusura per giustapposizione L^+.
Infatti ogni passeggiata utile su questo riconoscitore si ottiene considerando una qualsiasi sequenza di passeggiate utili su $\boldsymbol{R}$ "saldandone" due successive con una delle transizioni aggiunte ∎

C12 d.09 Prop. La chiusura-star di un linguaggio razionale su T è razionale.

Dim.: Trascurando il fatto che questa proprietà si ottiene dalla precedente, dalla razionalità di $\{\sqcup\}$ e dalla razionalità della unione di due linguaggi razionali, consideriamo ancora il riconoscitore-RSD $\boldsymbol{R} := \langle Q, \imath, F, \mathsf{T}, \delta \rangle$ ed il linguaggio individuato L.

Consideriamo il riconoscitore ottenuto aggiungendo al precedente un nuovo stato q_0 (assumendo che solo questo sia iniziale e finale), le transizioni $\langle f_h, \sqcup, q_0 \rangle$ per ogni $f_h \in F$ e $\langle q_0, \sqcup, i_k \rangle$ per ogni $i_k \in I$. Le sue passeggiate utili, oltre a quella ridotta allo stato q_0, sono ottenute giustapponendo in tutti i modi possibili le passeggiate utili di $\boldsymbol{R}$ precedute da una transizione-$\sqcup$ $\langle f_h, \sqcup, q_0 \rangle$ e da una transizione $\langle q_0, \sqcup, i_k \rangle$; quindi $\boldsymbol{R}$ riconosce tutte e sole le stringhe di L^*.

In altre parole, ogni passeggiata utile su $\boldsymbol{R}$ inizia e finisce in q_0 e, qualora non abbia lunghezza nulla (e quindi individui $\{\mu\}$), deve raggiungere uno stato iniziale di $\boldsymbol{R}$, percorrere una passeggiata utile su $\boldsymbol{R}$, tornare in q_0 e proseguire con spostamenti analoghi quante altre volte si vogliono ∎

C12 d.10 Prop. Il linguaggio riflesso di un linguaggio razionale è razionale.

Dim.: Si considerino il riconoscitore $\boldsymbol{R} := \langle Q, I, F, \mathsf{T}, \delta \rangle$ ed il corrispondente linguaggio accettato L.
Il riconoscitore ottenuto da $\boldsymbol{R}$ cambiando la direzione degli archi e scambiando il ruolo degli stati iniziali e finali individua esattamente il linguaggio costituito dalle stringhe riflesse di quelle di L.
Infatti le passeggiate utili sul nuovo riconoscitore sono tutte e sole le riflesse delle passeggiate utili su $\boldsymbol{R}$ ▮

C12 d.11 Ricordiamo le definizioni di funzione $\mathbf{o}$ e di derivata da sinistra rispetto alla stringa w del linguaggio L [C10d11]

$$\mathbf{o}(\mathsf{L}) \;:=\; \begin{cases} \mu & \text{sse } \mu \in \mathsf{L}, \\ \emptyset & \text{altrimenti;} \end{cases} \qquad w \setminus\!\!\setminus \mathsf{L} \;:=\; \{z \in \mathsf{T}^* \,\P\, w \cdot z \in \mathsf{L}\}.$$

Prop. Per le derivate da sinistra di un linguaggio accettato da un riconoscitore-RSND $\boldsymbol{R} := \langle Q, I, F, \mathsf{T}, \delta \rangle$ si ha:

$$\forall \mathsf{a}_i \in \mathsf{T} \;:\; \mathsf{a}_i \setminus\!\!\setminus \langle Q, I, F, \mathsf{T}, \delta \rangle^{\mathcal{A}} \;=\; \langle Q, \delta(I, \mathsf{a}_i), F, \mathsf{T}, \delta \rangle^{\mathcal{A}} \;.$$

$$\forall w \in \mathsf{T}^* \;:\; w \setminus\!\!\setminus \langle Q, I, F, \mathsf{T}, \delta \rangle^{\mathcal{A}} \;=\; \langle Q, \delta(I, w), F, \mathsf{T}, \delta \rangle^{\mathcal{A}} \;.$$

Dim.: La prima uguaglianza esprime il fatto che la derivata porta alle stringhe ottenute da quelle in $\boldsymbol{R}^{\mathcal{A}}$ aventi come iniziale a_i per eliminazione della iniziale stessa.
La seconda uguaglianza si ottiene come reiterazione della prima ▮

C12 d.12 Prop. Le derivate da sinistra di un linguaggio razionale L su $\mathsf{T} := \mathsf{L}^{minalf}$ rispetto a linguaggi qualsiasi M sono linguaggi razionali e sono in numero finito.

Dim.: Le derivate da sinistra rispetto a stringhe su T si ottengono modificando un riconoscitore-RSND di L solo attraverso cambiamenti dell'insieme degli stati finali e portano a linguaggi razionali.
Questo vale anche per le derivate da sinistra di L rispetto a linguaggi arbitrari sullo stesso alfabeto T.

Si osserva poi che la presenza in M di stringhe con caratteri estranei all'alfabeto TSs non contribuiscono al linguaggio derivato
Le possibili varianti dell'insieme degli stati iniziali sono solo in numero finito e questo implica che l'insieme delle derivate da sinistra è una collezione finita. ▮

C12 d.13 Per le derivate da destra dei linguaggi razionali valgono proprietà, espressioni e procedimenti simili a quelle trovate per le derivate da sinistra.
Esse sono ottenibili attraverso la dualità-LR e in particolare attraverso la relazione [C10d14]

$$\mathsf{L} \mathbin{/\!\!/} \mathsf{M} = ((\mathsf{M}^{\leftarrow}) \setminus\!\!\setminus (\mathsf{L}^{\leftarrow}))^{\leftarrow} \;.$$

(1) Prop.: Per le derivate da destra di un linguaggio accettato da un riconoscitore-RSND $\boldsymbol{R} := \langle Q, I, F, \mathsf{T}, \delta \rangle$ si ottengono dalle espressioni:

$$\forall \mathsf{a}_i \in \mathsf{T} \;:\; \langle Q, I, F, \mathsf{T}, \delta \rangle^{\mathcal{A}} \mathbin{/\!\!/} \mathsf{a}_i \;=\; \langle Q, I, (\delta^{\mathsf{T}})_{\mathsf{R}}(F, \mathsf{a}_i)), \mathsf{T}, \delta \rangle^{\mathcal{A}} \;.$$

$$\forall w \in \mathsf{T}^* \;:\; (\langle Q, I, F, \mathsf{T}, \delta \rangle^{\mathcal{A}}) \mathbin{/\!\!/} w \;=\; \langle Q, I, (\delta^{\mathsf{T}})_{\mathsf{R}}(F, w), \mathsf{T}, \delta \rangle^{\mathcal{A}} \;▮$$

(2) Prop.: Le derivate da destra di un linguaggio razionale L su $\mathsf{T} := \mathsf{L}^{minalf}$ rispetto a linguaggi qualsiasi M sono linguaggi razionali e sono in numero finito.

Dim.: Ottenibile modificando per dualità-LR la dimostrazione di d12 ▮

C12 d.14 Sia L un linguaggio razionale su $\mathsf{T} = \{\mathtt{a}_1, ..., \mathtt{a}_n\}$ e per $i = 1, ..., n$ sia M_i un linguaggio razionale su un alfabeto U.
Il linguaggio N ottenuto sostituendo nelle stringhe di L ogni occorrenza di ciascuno dei carattere $\mathtt{a}_i$ con il linguaggio M_i è razionale.

Dim.: Siano $\boldsymbol{R}$, $\boldsymbol{S}_1$, ... e $\boldsymbol{S}_n$ riconoscitori-RSND in grado di accettare, risp., L, M_1, ... e M_n.

Consideriamo il riconoscitore $\boldsymbol{T}$ ottenuto sostituendo in $\boldsymbol{R}$ ogni arco etichettato $\langle p, \mathtt{a}_i, q\rangle$ con il pluridigrafo ottenuto dal pluridigrafo di transizione di $\boldsymbol{S}_i$ aggiungendogli le transizioni $\langle p, \sqcup, p_h\rangle$ per tutti gli stati iniziali p_h di $\boldsymbol{S}_i$ e le transizioni $\langle q_k, \sqcup, q\rangle$ per tutti gli stati finali q_k di $\boldsymbol{S}_i$.

Il linguaggio N è il linguaggio accettato da $\boldsymbol{T}$: infatti le sue passeggiate utili si ottengono giustapponendo, come indicato dalle stringhe di L (leggibili sulle passeggiate utili di $\boldsymbol{R}$), passeggiate utili sui digrafi di transizione degli $\boldsymbol{S}_i$ e "saldandoli" con transizioni-$\sqcup$ dei tipi suddetti ∎

C12 d.15 Le proprietà dimostrate in precedenza si dicono **proprietà di chiusura dei linguaggi razionali**, in quanto ciascuna di esse afferma che la collezione dei linguaggi razionali generici o quella dei linguaggi razionali su un certo alfabeto è chiusa rispetto a certe determinate elaborazioni sui linguaggi razionali e in particolare su certe composizioni unarie o binarie su tali linguaggi.

Conviene osservare che tutte queste proprietà sono state dimostrate con procedimenti costruttivi.

Nel seguito denotiamo con $\mathbf{R}_{\mathsf{A}}$ la collezione dei linguaggi razionali sull'alfabeto A e con $\mathbf{R}$ la collezione dei linguaggi razionali su qualsiasi alfabeto.

Le proprietà di chiusura dimostrate in precedenza si possono esprimere con le seguenti formule.

$$\mathbf{R} \in \left[\ \cup, \text{ı}, \cap, ^+, ^*, \setminus, \ominus\ \right] .$$

$$\forall TSs, \mathsf{U} \in \boxplus \quad : \quad \mathbf{R}\mathsf{Sost}_{\mathsf{T},\mathbf{R}[\mathsf{U}]} \ = \ \mathbf{R}_{\mathsf{U}} .$$

C12 d.16 (1) Eserc. A partire da due pluridigrafi che individuano, risp., i linguaggi razionali L_1 ed L_2 individure il pluridigrafo che individua $\mathsf{L}_1 \setminus \mathsf{L}_2$ e individuare il plurigrafo che individua $\mathsf{L}_1 \ominus \mathsf{L}_2$.

Eserc. 2 Individuare un procedimento che consenta di decidere, per una qualsiasi coppia di linguaggi razionali L ed M, se $\mathsf{L} \subseteq \mathsf{M}$.

(3) Eserc. Individuare i riconoscitori-RSD in grado di accettare le scritture decimali degli interi divisibili per 6, per 18, per 72 e per 81.

C12 e. espressioni razionali e teorema di Kleene

C12 e.01 Quanto visto in precedenza mostra che la collezione dei riconoscitori-RS e la collezione dei linguaggi razionali $\mathbf{R}$ si possono considerare come un complesso di strumenti molto maneggevoli.

Un altro pregio dei linguaggi razionali consiste nella possibilità di individuare ciascuno di essi con una espressione di un tipo ben definito
Come vedremo queste espressioni, come i riconoscitori-RS, consentono di effettuare agevolmente varie elaborazioni sui linguaggi razionali ed aumentano la loro maneggevolezza.

C12 e.02 Riferiamoci a un alfabeto $\mathsf{T} = \{\mathtt{a}_1, ..., \mathtt{a}_n\}$ e introduciamo l'insieme delle **espressioni razionali** su T con le seguenti richieste ricorsive:

(1) $\{\mu\}$ è espressione razionale;
(2) per ogni $\mathtt{a}_i \in \mathsf{T}$, $\{\mathtt{a}_i\}$ è espressione razionale;
(3) se E_1 ed E_2 sono espressioni razionali, è tale anche $(\mathsf{E}_1 \cup \mathsf{E}_2)$;
(4) se E_1 ed E_2 sono espressioni razionali, è tale anche $(\mathsf{E}_1 \, \mathsf{E}_2)$;
(5) se E è espressione razionale, lo è anche (E^*).

Il termine espressione razionale dipende dal fatto che in esse entrano con ruoli essenziali le operazioni di unione e giustapposizione che possono essere assimilate alle usuali operazioni numeriche razionali somma e prodotto; inoltre, come vedremo, la chiusura-star si può accostare alla divisione tra serie numeriche.
La suddetta assimilazione contribuisce ad utilizzare in alcuni contesti il segno "+" per l'unione e il segno "·" o la semplice spaziatura per la giustapposizione.

C12 e.03 Diciamo **linguaggio esprimibile razionalmente** ogni linguaggio ottenuto dalla interpretazione di una espressione razionale come insieme di stringhe.

Esempi di espressioni razionali sono:

$$(\{\mathtt{a}\}\,\{\mathtt{a}\})\,\{\mathtt{a}\}) \qquad (\{\mathtt{a}\}\,\{\mathtt{b}\})^*) \qquad (\{\mathtt{a}\} \cup \{\mathtt{b}\})^*) \qquad ((\{\mathtt{a}\}^*) \cup (\{\mathtt{b}\}\,\{\mathtt{c}\}))^*).$$

Queste espressioni sono state ottenute applicando alla lettera le precedenti richieste, ma sono decisamente pesanti alla lettura.
Questa è una situazione che si verifica per molti linguaggi e automatismi introdotti per scopi computazionali generali: partendo da una definizione semplice ed essenziale spesso si ottiene uno strumento poco maneggevole.

Nella pratica è opportuno servirsi di varianti che presentano un aspetto più leggibile, ma che sono rette da regole più elaborate di quelle usate per le definizioni introduttive.
Per le espressioni razionali e per espressioni analoghe le accennate regole più elaborate si possono precisare soddisfacentemente servendosi delle cosiddette grammatiche acontestuali [C16].

Nel caso delle espressioni razionali risulta opportuno adottare le seguenti semplificazioni.
abolizione delle parentesi tonde che racchiudono l'intera espressione e di quelle semplificabili per implicita associatività delle operazioni unione e giustapposizione;
eliminazione delle parentesi graffe per i caratteri e le loro stringhe;
abolizione del segno di giustapposizione,
adozione delle potenze e della notazione X^+ in luogo di $\mathsf{X}\mathsf{X}^*$ o di $\mathsf{X}^*\mathsf{X}$.

Inoltre, come già detto spesso si usa il segno di somma invece del segno di unione binaria $\cup$.

Le espressioni precedenti quindi si semplificano nelle seguenti:

$$\mathtt{a}^3 \qquad (\mathtt{a}\,\mathtt{b})^* \qquad (\mathtt{a}+\mathtt{b})^* \qquad (\mathtt{a}^*+\mathtt{b}\,\mathtt{c})^*.$$

C12 e.04 Per le espressioni razionali più semplici non è difficile dare una descrizione del relativo linguaggio e conseguentemente individuare un riconoscitore capace di accettarlo.

Per esempio il linguaggio dato dall'espressione $\mathtt{a}^*\mathtt{b}^*\mathtt{c}^*$ è formato dalle stringhe contenenti alcune repliche della lettera `a` (eventualmente nessuna), seguite da alcune `b` (eventualmente nessuna), seguite da alcune `c` (eventualmente nessuna). Esso è riconosciuto da:

Il linguaggio espresso da: $\mathtt{a}^+\mathtt{b}^+\mathtt{c}^+$ è costituito da una o più lettere `a` seguite da una o più lettere `b` seguite a loro volta da una da una o più lettere `c`. Esso corrisponde al digrafo di transizione

Il linguaggio dato dall'espressione: $\mathtt{a}^*(\mathtt{b}^2+\mathtt{c}\,\mathtt{d}\,\mathtt{e})\mathtt{f}^*$ è invece riconosciuto da:

C12 e.05 Ci proponiamo ora di dimostrare il classico teorema dovuto a Stephen Kleene sulla coincidenza dei linguaggi dati da espressioni razionali e dei linguaggi razionali, cioè dei linguaggi individuati da riconoscitori-RSND.

Dimostreremo dapprima, servendoci di risultati del paragrafo precedente, che a ogni espressione razionale si riesce ad associare un riconoscitore-RSND che accetta lo stesso linguaggio.
Successivamente vedremo come ad un riconoscitore-RSD $\boldsymbol{R}$ si associa un sistema di equazioni avente per incognite dei linguaggi che può essere risolto da un algoritmo in modo completo fino a ottenere una espressione razionale che individua il linguaggio riconosciuto da $\boldsymbol{R}$.

C12 e.06 Prop. Ogni linguaggio esprimibile razionalmente è riconoscibile-RS.

Dim.: La dimostrazione procede induttivamente su espressioni razionali via via più complesse.

Abbiamo visto che esistono riconoscitori che individuano i linguaggi $\{\sqcup\}$ e $\{\mathtt{a}_i\}$ richiesti dai punti (1) e (2) della definizione di espressione razionale.
Supponiamo allora che per due espressioni razionali E_1 ed E_2 esistano due riconoscitori $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$ in grado di accettare, risp., i linguaggi L_1 e L_2 da esse espressi.

Per avere un riconoscitore per il linguaggio espresso da $(\mathsf{E}_1+\mathsf{E}_2)$ basta considerare la composizione in parallelo di riconoscitori-RSND vista in d02 .

Per avere un riconoscitore per il linguaggio espresso da $(\mathsf{E}_1\,\mathsf{E}_2)$ basta considerare la composizione in serie di riconoscitori-RSND vista in d03 .

Per avere un riconoscitore per il linguaggio espresso da $(\mathsf{E}_1{}^*)$ basta considerare l'arricchimento di $\boldsymbol{R}_1$ relativo alla chiusura-star visto in d09 ∎

C12 e.07 Poniamoci ora in grado di risolvere una equazione avente linguaggi come incognite di forma particolare.

Dati due linguaggi U e V sull'alfabeto T, si consideri l'equazione

$$\mathsf{X} = \mathsf{U}\,\mathsf{X} + \mathsf{V}\ , \qquad [*]$$

nella quale X è la incognita che può assumere come valori linguaggi su T.

(1) Prop.: $\mathsf{U}^*\mathsf{V}$ è soluzione della $[*]$.

Dim.: Basta sostituire X con $\mathsf{U}^*\mathsf{V}$ nei due membri della $[*]$:

$$\mathsf{U}^*\mathsf{V} \;=\; \mathsf{U}\,\mathsf{U}^*\mathsf{V} + \mathsf{V} \;=\; (\mathsf{U}^+\mathsf{V} + \sqcup)\mathsf{V} \;=\; \mathsf{U}^*\mathsf{V}$$ ∎

(2) Prop.: $\mathsf{U}^*\mathsf{V}$ è contenuto in ogni soluzione Y della $[*]$.

Dim.: Dimostriamo per induzione che per ogni $n \in \mathbb{N}$ si ha $\mathsf{U}^n\mathsf{V} \subseteq \mathsf{Y}$.
La cosa vale per $n = 0$, in quanto $\mathsf{Y} = \mathsf{U}\mathsf{Y} + \mathsf{V}$ implica $\mathsf{V} \subseteq \mathsf{Y}$.
Supponiamo che per un certo $n \in \mathbb{N}$ sia $\mathsf{U}^n\mathsf{V} \subseteq \mathsf{Y}$ e deduciamo che $\mathsf{U}^{n+1}\mathsf{V} \subseteq \mathsf{Y}$:

$$\mathsf{U}^{n+1}\mathsf{V} = \mathsf{U}(\mathsf{U}^n\mathsf{V}) \subseteq \mathsf{U}\mathsf{Y} \subseteq \mathsf{U}\,\mathsf{Y} + \mathsf{V} = \mathsf{Y}$$ ∎

C12 e.08 Prop. Se $\sqcup \notin \mathsf{U}$, allora $\mathsf{U}^*\,\mathsf{V}$ è la sola soluzione della $[*]$.

Dim.: Se così non fosse si potrebbe scegliere in $\mathsf{Y} \setminus \mathsf{U}^*\mathsf{V}$ una stringa z avente lunghezza minima, misura che scriviamo $\ell := |z|$.
Per essa $z \in \mathsf{Y} = \mathsf{U}\,\mathsf{Y} + \mathsf{V}$, in quanto deve essere $z \notin \mathsf{V} \subseteq \mathsf{U}^*\mathsf{V}$ e si ricava $z \in \mathsf{U}\mathsf{Y} = \mathsf{U}^2\mathsf{Y} + \mathsf{U}\mathsf{V}$; da questa, non potendo essere $z \in \mathsf{U}\,\mathsf{V} \subseteq \mathsf{U}^*\mathsf{V}$, segue che $z \in \mathsf{U}^2\,\mathsf{Y} = \mathsf{U}^3\,\mathsf{Y} + \mathsf{U}^2\,\mathsf{V}$.
Procedendo in questo modo si ottiene $z \in \mathsf{U}^{\ell+1}\mathsf{Y}$; ma questa relazione è assurda, in quanto $\sqcup \notin \mathsf{U}$ implica che in $\mathsf{U}^{\ell+1}$ si possono trovare solo stringhe aventi lunghezza maggiore o uguale di $\ell + 1$ ∎

Osserviamo che l'equazione $[*]$ si può assimilare a una equazione numerica lineare nella X e che la soluzione si potrebbe pensare ottenuta con passaggi come i seguenti:

$$\mathsf{X} = \mathsf{U}\mathsf{X} + \mathsf{V} \implies (1 - \mathsf{U})\mathsf{X} = \mathsf{V} \implies \mathsf{X} = (1 - \mathsf{U})^{-1}\mathsf{V} = (1 + \mathsf{U} + \mathsf{U}^2 + ...)\mathsf{V} = \mathsf{U}^*\mathsf{V} \;.$$

Tuttavia a questo punto dell'*esposizione* non possiamo ancora rendere chiara la proposta analogia tra operazioni su linguaggi e operazioni numeriche e la precedente Catena deduttiva può essere considerata solo come un aiuto per la memoria.

C12 e.09 Una espressione riguardante le derivate di un linguaggio prevedibile, ma utile, è la seguente.

Prop. Per ogni linguaggio L sull'alfabeto $\mathsf{T} = \{\mathsf{a}_1, ..., \mathsf{a}_n\}$ si ha

$$\mathsf{L} \;=\; \mathbf{o}(\mathsf{L}) + \mathsf{a}_1\,(\mathsf{a}_1 \setminus\setminus \mathsf{L}) + \cdots + \mathsf{a}_n\,(\mathsf{a}_n \setminus\setminus \mathsf{L}) \;.$$

Dim.: Questa relazione non dice altro che in L ci può essere la stringa muta e che le stringhe rimanenti si possono ripartire secondo i loro diversi caratteri iniziali ∎

Osserviamo che il secondo membro della precedente relazione è lineare a destra nelle derivate da sinistra.

La precedente relazione viene talora chiamata formula di Taylor per i linguaggi.

Questo termine si giustifica sulla base della analogia tra il cosiddetto anello delle funzioni di variabili reali e il semianello dei linguaggi.
In questa ottica i simboli dell'alfabeto sono visti come variabili, $\mathbf{o}(L)$ si può pensare come una costante (che per il semianello dei linguaggi può essere solo lo zero $\emptyset$ o l'unità $\sqcup$) e le derivate portano all'abbassamento del grado delle potenze senza comportare moltiplicazioni per fattori numerici.

Presentiamo anche la duale-LR della formula precedente riguardante le derivate da destra

$$\mathsf{L} \;=\; \mathbf{o}(\mathsf{L}) + (\mathsf{L} \mathbin{/\!\!/} \mathsf{a}_1)\,\mathsf{a}_1 + \cdots + (\mathsf{L} \mathbin{/\!\!/} \mathsf{a}_n)\,\mathsf{a}_n \;.$$

C12 e.10 A questo punto siamo in grado di associare a ogni riconoscitore-RSD $\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta\rangle$ un sistema di equazioni per linguaggi.

Le incognite L_q e le equazioni di questo sistema sono associate ai vari stati $q \in Q$; ciascuna delle incognite L_q esprime il linguaggio accettato da $\boldsymbol{R}_q := \langle Q, q, F, \mathsf{T}, \delta\rangle$, ovvero abbiamo $\mathsf{L}_q = \boldsymbol{R}_q{}^{\mathcal{A}}$.

L'equazione associata allo stato q corrisponde alla formula di Taylor per L_q:

$$\mathsf{L}_q = o(\mathsf{L}_q) + \mathtt{a}_1 \cdot \mathsf{L}_{\delta(q,\mathtt{a}_1)} + \cdots + \mathtt{a}_n \cdot \mathsf{L}_{\delta(q,\mathtt{a}_n)} \ .$$

In questa formula $\mathbf{o}(\mathsf{L}_q)$ si può esplicitare facilmente, in quanto vale μ sse q è stato finale, vale $\emptyset$ in caso contrario; quali L_p porre a destra dei simboli $\mathtt{a}_i$ lo si ricava facilmente dalla osservazione degli archi uscenti dallo stato q.

Ricordiamo poi la seguente proprietà trovata in d11 per un generico riconoscitore-RSD:

$$\forall \mathtt{a}_i \in \mathsf{T} \ : \ \mathtt{a}_i \setminus\!\setminus (\langle Q, \imath, F, \mathsf{T}, \delta\rangle^{\mathcal{A}}) \ = \langle Q, \delta(\imath, \mathtt{a}_i), F, \mathsf{T}\rangle\delta^{\mathcal{A}} \ .$$

Si ricorda anche il fatto conseguente [d12]: le derivate rispetto a stringhe di un linguaggio razionale costituiscono una collezione finita di linguaggi razionali.

C12 e.11 Per stabilire l'equivalenza tra espressioni razionali e riconoscitori-RSND rimane da dimostrare un ultimo fatto.

Prop. Il sistema di equazioni associabili a un riconoscitore-RSD è risolubile univocamente e conduce a espressioni razionali per il linguaggio accettato e le sue derivate.

Dim.: Si tratta di mostrare che la soluzione del sistema di $|Q|$ equazioni in $|Q|$ incognite si ottiene con successive $|Q|$ manovre in ciascuna dei quali si trasforma una equazione che presenta a primo membro una incognita L_q e a secondo membro una combinazione lineare di alcune incognite in una espressione esplicita della L_q nelle incognite rimanenti.

Una tale manovra quindi consente di eliminare dal sistema l'incognita che stava a primo membro e quindi conduce a un sistema della stessa forma con un numero di incognite diminuito di 1 (come avviene nei noti procedimenti risolutivi dei sistemi delle equazioni numeriche lineari [B32j]).

La caratteristica essenziale delle equazioni del sistema e delle equazioni ottenute con i passi preannunciati è quella di avere a primo membro una incognita ed al secondo membro una espressione lineare a destra in alcune incognite, i coefficienti delle quali sono linguaggi forniti da espressioni razionali che sono privi della stringa muta.
Diciamo che queste espressioni a secondo membro sono di tipo **d** e che le equazioni con le caratteristiche descritte sono di tipo **e**.

Una equazione nella quale l'incognita a primo membro non compare nel secondo, ovviamente, può essere direttamente utilizzata per l'abbassamento del grado del sistema.

Una equazione nella quale l'incognita a primo membro compare anche nel secondo ha la forma
$\mathsf{L}_q = \mathsf{U}\mathsf{L}_q + \mathsf{V}$, ovvero presenta la forma dell'equazione [$*$]; quindi se il secondo membro è di tipo **d**, ha come unica soluzione $\mathsf{U}^*\mathsf{V}$.

Come la V, anche $\mathsf{U}^*\mathsf{V}$ è un'espressione di tipo **d**. Quando si sostituisce una incognita L_q con un'espressione di tipo **d** nell'espressione di tipo **d** che è secondo membro di una equazione di tipo **e**, si ottiene ancora un'espressione di tipo **d** e quindi l'equazione rimane di tipo **e**.

Quindi il procedimento risolutivo può essere portato fino alla fine, cioè fino ad avere una espressione razionale per $\mathsf{L}_\imath$ ed una catena di espressioni razionali per le altre incognite che sono state successivamente esplicitate.

Risulta dunque che un linguaggio accettato-RS, ossia accettato da un riconoscitore-RS è fornito da una espressione razionale e da espressioni razionali sono rappresentate anche tutte le sue derivate ∎

C12 e.12 Possiamo quindi enunciare un risultato classico.

Teorema (**teorema di Kleene 1956**)
Un linguaggio è razionale, cioè accettabile da un riconoscitore-RS sse è esprimibile razionalmente ∎

C12 e.13 Illustriamo il procedimento risolutivo dei sistemi di equazioni lineari per i linguaggi con gli esempi relativi ai seguenti semplici riconoscitori:

Il primo porta alle equazioni:

$$\begin{aligned} \mathsf{L}_0 &= \mathtt{a}\,\mathsf{L}_0 + \mathtt{b}\,\mathsf{L}_1 \\ \mathsf{L}_1 &= \mu\,; \end{aligned}$$

dalle quali seguono evidentemente $\mathsf{L}_0 = \mathtt{a}\mathsf{L}_0 + \mathtt{b}$ e $\mathsf{L}_0 = \mathtt{a}^*\mathtt{b}$.
Si noti che abbiamo ottenuta la soluzione della più semplice delle equazioni della forma $[*]$.

Al secondo dei precedenti riconoscitori è associato il seguente sistema di equazioni:

$$\begin{aligned} \mathsf{L}_0 &= \mathtt{a}\,\mathsf{L}_1 + \mathtt{b}\,\mathsf{L}_2 \\ \mathsf{L}_1 &= \mathtt{a}\,\mathsf{L}_1 + \mathtt{b}\,\mathsf{L}_2 + \mu \\ \mathsf{L}_2 &= \mathtt{a}\,\mathsf{L}_0 + \mathtt{b}\,\mathsf{L}_0 \end{aligned} \quad .$$

Eliminando dalle prime due equazioni L_2 grazie alla terza si ha:

$$\begin{aligned} \mathsf{L}_0 &= \mathtt{a}\,\mathsf{L}_1 + \mathtt{b}(\mathtt{a}+\mathtt{b})\mathsf{L}_0 \\ \mathsf{L}_1 &= \mathtt{a}\,\mathsf{L}_1 + \mathtt{b}(\mathtt{a}+\mathtt{b})\mathsf{L}_0 + \mu \end{aligned} \quad .$$

Risolvendo la seconda equazione si ottiene

$$\mathsf{L}_1 = \mathtt{a}^*(\mathtt{b}(\mathtt{a}+\mathtt{b})\mathsf{L}_0 + \mu) \ .$$

a questo punto si può riscrivere la prima equazione come

$$\mathsf{L}_0 = a^+\mathtt{b}(\mathtt{a}+\mathtt{b})\mathsf{L}_0 + \mathtt{a}^+$$

e risolvendola si ricava l'espressione razionale richiesta

$$\mathsf{L}_0 = (\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\mathtt{a}^+ \ .$$

Le uguaglianze che si erano ottenute consentono di avere anche le espressioni per le derivate di L_0:

$$\begin{aligned} \mathsf{L}_1 &= \mathtt{a} \setminus\!\setminus \mathsf{L}_0 = a^*(\mathtt{b}(\mathtt{a}+\mathtt{b})(\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\mathtt{a}^+ + \mu) \ , \\ \mathsf{L}_2 &= \mathtt{b} \setminus\!\setminus \mathsf{L}_0 = (\mathtt{a}+\mathtt{b})(\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\,\mathtt{a}^+ \ . \end{aligned}$$

C12 e.14 Eserc. Trovare una espressione razionale per i linguaggi accettati dai riconoscitori:

C12 e.15 Eserc. Esaminare alcuni riconoscitori-RSD con pluridigrafo delle transizioni aciclico e mostrare come il procedimento risolutivo delle relative equazioni lineari porti a espressioni razionali polinomiali, cioè a espressioni nelle quali non compare la chiusura-star, e quindi a linguaggi finiti corrispondenti agli insiemi finiti delle passeggiate utili sui pluridigrafi.

C12 e.16 Eserc. Data un'espressione razionale come la $(\mathtt{a} + \mathtt{b}^*)(\mathtt{a} + \mathtt{b})(\mathtt{a}^* + \mathtt{b})$, individuare un riconoscitore-RSND a essa equivalente-L, trasformarlo in un riconoscitore-RSD e risolvere le relative equazioni; confrontare quindi l'espressione razionale trovata con quella di partenza.

C12 e.17 Eserc. Generalizzare la definizione di pluridigrafo delle transizioni ammettendo che su ogni arco si possa avere una espressione razionale (in particolare priva della stringa muta).

Reinterpretare quindi il procedimento risolutivo del sistema di equazioni di un pluridigrafo delle transizioni come progressiva riduzione di tale pluridigrafo ottenuta attraverso varianti che provengono da fusioni di archi in serie, da fusioni di archi in parallelo e da eliminazioni di cappi, cioè che provengono da trasformazioni dei tipi:

fino a ottenere un pluridigrafo del tipo a transizione unica:

C12 f. riconoscitore minimo di un linguaggio razionale

C12 f.01 Un problema di rilevante importanza riguarda l'equivalenza-L di due espressioni razionali, e quindi l'equivalenza-L di due riconoscitori-RS $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$.

Un procedimento per risolvere un tale problema consiste nel dimostrare la validità delle due relazioni $\boldsymbol{R}_1^{\mathcal{A}} \subseteq \boldsymbol{R}_2^{\mathcal{A}}$ e $\boldsymbol{R}_1^{\mathcal{A}} \supseteq \boldsymbol{R}_2^{\mathcal{A}}$.

Un altro modo di procedere consiste nel dimostrare che $\boldsymbol{R}_1^{\mathcal{A}} \setminus \boldsymbol{R}_2^{\mathcal{A}} = \boldsymbol{R}_2^{\mathcal{A}} \setminus \boldsymbol{R}_1^{\mathcal{A}} = \emptyset$; un altro ancora consiste nel provare che $\boldsymbol{R}_1^{\mathcal{A}} \cap (\mathsf{T}^* \setminus \boldsymbol{R}_2^{\mathcal{A}}) = \boldsymbol{R}_2^{\mathcal{A}} \cap (\mathsf{T}^* \setminus \boldsymbol{R}_1^{\mathcal{A}}) = \emptyset$.

Questi modi di procedere richiedono l'esame dei corrispondenti riconoscitori composti e in genere risultano alquanto pesanti.
Un atteggiamento che di solito risulta più efficiente e che fornisce informazioni molto significative consiste nella individuazione di un tipo di riconoscitore-RSD univocamente associabile a un linguaggio razionale e nel ricondurre ad esso ciascuno dei riconoscitori dei quali si cerca di stabilire l'equivalenza.

In effetti si dimostra la possibilità effettiva di individuare un tipo di riconoscitore-RS univocamente ad una qualsiasi presentazione di un linguaggio razionale il quale si rivela particolarmente vantaggioso e al quale risulta opportuno assegnare il ruolo di riconoscitore-RSD canonico.

C12 f.02 Consideriamo il linguaggio generico $L \subseteq \mathsf{T}^*$ e denotiamo con $\mathcal{D}_{\mathsf{L}} := \{w \in \mathsf{T}^* :| \; w \setminus\setminus \mathsf{L}\}$ la collezione delle sue derivate da sinistra.
Questo insieme, non necessariamente finito, si può considerare come insieme degli stati di una sorta di macchina detta **riconoscitore delle derivate** di L:

$$\Big\langle \mathcal{D} \;,\; \mathsf{L} \;,\; \{z \in \mathsf{L} :| \; z \setminus\setminus \mathsf{L}\} \;,\; \mathsf{T} \;,\; \lceil \langle w \setminus\setminus \mathsf{L}, \mathtt{a}\rangle \in \mathcal{D} \times \mathsf{T} \mapsto (w\mathtt{a}) \setminus\setminus \mathsf{L}rF \Big\rangle.$$

Va rilevato che la precedente espressione individua un vero e proprio riconoscitore-RSD [b02] sse $\mathcal{D}$ è finito.

(1) Prop.: Il riconoscitore delle derivate di un linguaggio L avente un insieme finito di derivate da sinistra accetta L stesso.

Dim.: La lettura di una stringa w fa passare dallo stato iniziale L allo stato $w \setminus\setminus \mathsf{L}$ e questo è stato finale sse $w \in \mathsf{L}$ ∎

A questo punto possiamo anche affermare:

(2) Prop.: Un linguaggio è razionale sse possiede un numero finito di derivate da sinistra.

Dim.: Abbiamo già visto che un linguaggio razionale possiede un numero finito di derivate da sinistra [d12]. Per il viceversa qui sopra si è mostrato come la finitezza delle derivate da sinistra garantisca l'esistenza di un riconoscitore-RSD che lo accetta ∎

C12 f.03 Le precedenti considerazioni non forniscono direttamente indicazioni costruttive: infatti non è chiaro come si possono individuare precisamente le diverse derivate di un linguaggio razionale L.
Le derivate associate ai diversi stati di un riconoscitore-RSD che accetta L potrebbero presentare delle coincidenze non evidenti. In effetti un riconoscitore-RSD, anche se privo di stati palesemente inutili, può presentare stati ridondanti, ossia stati che possono essere fusi.

Per poter disporre effettivamente dei riconoscitori delle derivate occorre individuare un procedimento che permette di stabilire tutte e sole le coincidenze tra derivate individuate da varianti del riconoscitore corrispondenti all'assunzione di stati iniziali diversi.

Un tale procedimento consente di avere un riconoscitore privo di nodi inutili, cioè uno strumento con buone caratteristiche di essenzialità.

C12 f.04 L'enunciato che segue serve a chiarire le circostanze nelle quali un riconoscitore-RSD $\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta \rangle$ presenta stati diversi associati alla stessa derivata da sinistra di $\mathsf{L} := \boldsymbol{R}^{\mathcal{A}}$.

(1) Prop.: Consideriamo due stati q_j e q_h del riconoscitore $\boldsymbol{R}$, due stringhe u_j e u_h tali che $q_j = \delta(\imath, u_j)$ e $q_h = \delta(\imath, u_h)$ e i due linguaggi razionali $\mathsf{L}_{q_j} := \langle Q, q_j, F, \mathsf{T}, \delta \rangle^{\mathcal{A}}$ ed $\mathsf{L}_{q_h} := \langle Q, q_h, F, \mathsf{T}, \delta \rangle^{\mathcal{A}}$.

$$\mathsf{L}_{q_j} \;=\; \mathsf{L}_{q_h} \iff u_j \setminus\!\setminus \mathsf{L} = u_h \setminus\!\setminus \mathsf{L}$$

(2) Prop.: Se in $\boldsymbol{R}$ vi sono due stati come i precedenti q_j e q_h che forniscono la stessa derivata da sinistra di L si può ottenere un riconoscitore che accetta lo stesso L mediante la fusione di q_j e q_h ed eventuali altre conseguenti fusioni di stati.

Dim.: Osserviamo che si può stabilire una corrispondenza biunivoca tra le passeggiate che iniziano in q_j e quelle che iniziano in q_h indotta dalla lettura delle varie stringhe $v \in \mathsf{T}$.
Per l'uguaglianza $\mathsf{L}_{q_j} = \mathsf{L}_{q_h}$, una passeggiata che inizia in q_j termina in uno stato finale sse anche la sua corrispondente che inizia in q_h termina in uno stato finale.
Fondendo q_j e q_h e conseguentemente $\delta(q_j, v)$ con $\delta(q_h, v)$ per ogni $v \in \mathsf{T}^+$, se già non coincidono, si ottiene un riconoscitore equivalente-L a $\boldsymbol{R}$.

C12 f.05 La precedente dimostrazione non individua un chiaro procedimento costruttivo per ridurre il numero degli stati di un riconoscitore-RSD, in quanto non precisa in termini operativi dice come stabilire l'uguaglianza $\mathsf{L}_{q_j} = \mathsf{L}_{q_h}$, ovvero la sua equivalente $u_j \setminus\!\setminus \mathsf{L} = u_h \setminus\!\setminus \mathsf{L}$.

Per individuare un procedimento che consenta questa riduzione relativamente a tutte le coincidenze delle derivate disponibili è opportuno esaminare una classica equivalenza tra stringhe.

Si dice **congruenza di Myhill** relativa al linguaggio L la relazione M_{L} definita da:

$$u \,\mathrm{M}_{\mathsf{L}}\, v \quad \text{sse} \quad \forall w \in \mathsf{T}^* \;:\; u\,w \in \mathsf{L} \iff v\,w \in \mathsf{L} \;.$$

Evidentemente $u \,\mathrm{M}_{\mathsf{L}}\, v$ sse $u \setminus\!\setminus \mathsf{L} = \{z \in \mathsf{T}^* \,\S\, u\,z \in \mathsf{L}\} = \{z \in \mathsf{T}^* \,\S\, v\,z \in \mathsf{L}\} = v \setminus\!\setminus \mathsf{L}$.
Si tratta quindi di trovare un procedimento per individuare effettivamente l'equivalenza M_{L} per un generico linguaggio razionale L.

C12 f.06 Consideriamo la seguente successione di equivalenze entro l'insieme degli stati del riconoscitore-RSD $\boldsymbol{R}$:

$$\forall h \in \mathbb{N},\quad p, q \in Q \;:\; p \,\mathsf{E}_h\, q \quad \text{sse} \quad \forall w \in \mathsf{T}^{\leq h} \;:\; \delta(p, w) = \delta(q, w).$$

Le suddette equivalenze si possono individuare facilmente esaminando il pluridigrafo delle transizioni di $\boldsymbol{R}$.

Evidentemente E_0 corrisponde alla bipartizione dell'insieme degli stati $Q = (Q \setminus F) \,\dot\cup\, F$.

Per trovare E_1 si considerano i diversi caratteri $\mathtt{a} \in \mathsf{T}$ e per ciascuno di essi si distinguono i nodi finali che la lettura di $\mathtt{a}$ porta in un altro nodo finale, i nodi finali che $\mathtt{a}$ porta in un nodo nonfinale, i nodi di $Q \setminus F$ che $\mathtt{a}$ lascia in $Q \setminus F$ e i nodi nonfinali che la lettura di $\mathtt{a}$ sposta in F.
Tutte queste distinzioni si ottengono con un semplice esame degli archi etichettati da $\mathtt{a}$ uscenti dai vari nodi.

Per raffinare una generica equivalenza E_h si procede in modo del tutto simile. Si tratta di vedere in quali classi di E_h vengono mandati i nodi di ciascuna classe di E_h in conseguenza della lettura di ciascuno dei simboli di T.

Osserviamo poi che al crescere di h le equivalenze E_h sono sempre più fini, cioè che $\forall h \in \mathbb{N} \ : \ \mathsf{E}_h \supseteq \mathsf{E}_{h+1}$.

Essendo Q finito, il processo di raffinamento della equivalenze E_h non può continuare illimitatamente, ma si incontra necessariamente un intero positivo t tale che $\mathsf{E}_t = \mathsf{E}_{t+1}$.

A questo punto non si può più avere nessun altro raffinamento: se si trovasse un raffinamento per il quale $\mathsf{E}_{t+1} \supset \mathsf{E}_{t+2}$, si sarebbe potuto trovare un raffinamento anche per E_t.

C12 f.07 $\mathsf{RcnDmin}_{\mathsf{L}}$ L'equivalenza trovata E_t si verifica facilmente coincidere con la congruenza di Myhill.
Inoltre $q \, \mathrm{M}_{\mathsf{L}} \, q'$ implica che $\forall \mathsf{a} \in \mathsf{T} \ : \ \delta(q, \mathsf{a}) \mathrm{M}_{\mathsf{L}} \delta(q', \mathsf{a})$.
Questo rende lecito introdurre la seguente applicazione riguardante classi di equivalenza della rlazione M_{L}:

$$\delta/\mathrm{M}_{\mathsf{L}} \ := \ \left[\ \langle q \, \mathrm{M}_{\mathsf{L}}, \mathsf{a} \rangle \in Q/\mathrm{M}_{\mathsf{L}} \times \mathsf{T} \quad \mapsto \quad \delta(q, \mathsf{a}) \, \mathrm{M}_{\mathsf{L}} \ \right] \ .$$

È dunque lecito considerare il riconoscitore i cui stati sono ottenuti "fondendo" gli elementi delle varie classi dell'equivalenza M_{L}:

$$\left\langle \ Q/\mathrm{M}_{\mathsf{L}} \ , \ \imath \, \mathrm{M}_{\mathsf{L}} \ , \ F/\mathrm{M}_{\mathsf{L}} \ , \ \mathsf{T} \ , \ \delta/\mathrm{M}_{\mathsf{L}} \ \right\rangle \ .$$

Gli stati di questo riconoscitore corrispondono biunivocamente alle derivate di L.

Quindi se si applicasse la stessa costruzione a un altro riconoscitore-RSD che accetta L, si giungerebbe alla stessa macchina, a meno di inessenziali differenze nel modo di individuare gli stati, cioè a meno di isomorfismi tra pluridigrafi etichettati dello stesso alfabeto T.

È quindi lecito chiamare la precedente macchina **riconoscitore minimo del linguaggio razionale** L o riconoscitore delle derivate di L.
Per questo riconoscitore proponiamo la notazione $\mathsf{RcnDmin}_{\mathsf{L}}$.

Evidentemente tra i riconoscitori-RSD che accettano L questo è quello che permette di operare meglio su L stesso.

Occorre tuttavia segnalare che per certi linguaggi razionali si trovano riconoscitori-RSND con meno stati di quelli del riconoscitore delle derivate.

C12 f.08 Eserc. Costruire il riconoscitore minimo equivalente-L al seguente:

Si osservare come si fondono stati palesemente inutili, cioè stati dai quali non è raggiungibile alcuno stato finale.

Testo fruibile in https://www.mi.imati.cnr.it/alberto/ e https://arm.mi.imati.cnr.it/Matexp/matexp_main.php