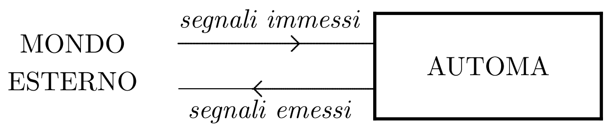
Capitolo D52:
Automi a stati finiti e linguaggi razionali

Contenuti delle sezioni

D52:a. Generalità sugli automi

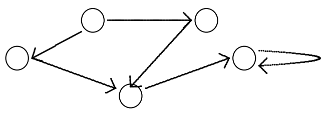
D52:a.01 Gli **automi** o **macchine formali** sono modelli matematici discreti che consentono di rappresentare in modo schematico e semplificato meccanismi e processi reali o realizzabili.
La gamma degli automi e delle loro applicazioni che risulta utile studiare sono assai vaste; in effetti non vi è un ampio accordo sulla precisa portata di questo termine. In questo capitolo, dopo una introduzione della nozione di automa, ma limitata all'ambito deterministico, presentiamo gli automi dei tipi più semplici, i riconoscitori di Rabin e Scott; successivamente studieremo modelli più dotati come i riconoscitori a pila (D54:) e le macchine di Turing (D58:).

D52:a.02 Prima caratteristica di ogni automa è quella di distinguersi chiaramente dal mondo che lo circonda e con il quale interagisce: il mondo esterno gli invia dei segnali i quali determinano una sua evoluzione che si sviluppa nel tempo. Durante una tale evoluzione l'automa a sua volta può emettere verso l'esterno segnali che caratterizzano il suo comportamento ovvero, da un punto di vista applicativo, le sue prestazioni; questi segnali si possono considerare emessi in risposta a quelli inviatigli dal mondo esterno.
La precedente caratteristica può raffigurarsi nel seguente modo:



D52:a.03 Altra caratteristica comune ad ogni macchina formale consiste nel fatto che nel corso della sua evoluzione essa si viene a trovare in successive situazioni, chiamate **stati interni**: l'insieme di questi possibili stati è finito.
Gli stati di un automa si possono rappresentare come i nodi di un digrafo dotato di archi arricchiti: l'arco che va da un nodo p ad un secondo nodo q è completato con le informazioni che individuano le

circostanze in grado di provocare il cambiamento dello stato. Questi cambiamenti si possono descrivere come spostamenti di una entità vagamente antropomorfa, chiamata **controllo**, che ad ogni passo evolutivo può portarsi da uno stato all'altro.
Riferendosi ai digrafi, si dice che il controllo si sposta da un nodo p ad un nodo q percorrendo un arco (q potrebbe coincidere con q e in questo caso l'arco sarebbe un cappio). A questo proposito si parla anche di **transizioni fra stati** dell'automa. Il termine transizione intuitivamente esprime l'azione del controllo, mentre formalmente consiste in una terna della forma $\langle p, C, q\rangle$, cioè in un arco munito delle indicazioni C sulle circostanze che possono comportare il passaggio dallo stato p al q. Il digrafo arricchito si dice **[di]grafo delle transizioni** dell'automa: esso si può infatti confondere con l'insieme delle suddette terne.



D52:a.04 Vari tipi di automi dispongono di **dispositivi di memoria** o **registri** nei quali si possono inserire informazioni discrete (elenchi, numeri, tabelle, grafi,...) costituite da simboli organizzati in stringhe o in complessi di dati che, in linea di principio, si possono ricondurre a stringhe. Ciascuna delle situazioni in cui si può trovare un automa è quindi caratterizzata da uno stato interno e dalle informazioni che sono registrate nei dispositivi di memoria; tutto questo costituisce la cosiddetta **configurazione attuale** dell'automa.
Alcuni tipi di automa si dicono dotati di **dispositivi di memoria illimitati**: questo significa che in ogni istante della loro evoluzione dispongono di un insieme finito di registri che, successivamente, se richiesto dalle circostanze, si può estendere quanto serve. Si parla anche di **dispositivi di memoria potenzialmente infiniti**.

D52:a.05 Il dispositivo di memoria basilare di molti automi è costituito da un **nastro**, supporto formato da una sequenza di caselle in ciascuna delle quali è registrabile un *simbolo* da scegliere in un *alfabeto* ben definito. In ogni istante solo un numero finito di caselle contiene registrazioni utilizzabili.
I più semplici tipi di automa hanno solo nastri finiti; un tale supporto corrisponde alla memoria fisica di un computer o ad un array disponibile in un programma scritto in un linguaggio procedurale.

a	b	c	d	a	b	a

Gli automi dotati di nastri illimitatamente estendibili talora vengono chiamati "macchine infinite", espressione che è opportuno considerare abbreviazione del termine "automi dotati di nastri potenzialmente infiniti".

D52:a.06 L'evoluzione di un automa è costituita da una sequenza di eventi che si verificano in una successione discreta di istanti; tra due di questi istanti consecutivi non si prende in considerazione alcun altro evento. Nella descrizione del comportamento di un automa 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 situazioni non perfettamente coincidenti con quelle rappresentate dagli stati e dai dispositivi di memoria.
Naturalmente altri studi si possono porre il problema dell'affidabilità e della adeguatezza dei dispositivi fisici o informatici che implementano i modelli costrituiti dagli automi formali.

L'evoluzione di un automa è dunque descritta interamente dalla sequenza dei **passi** o delle **mosse** che esso compie per portarsi sulle successive configurazioni. Nella teoria degli automi 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 biiezione con la successione delle mosse della macchina; in corrispondenza delle mosse e degli istanti della macchina si hanno le configurazioni nelle quali la macchina viene a trovarsi che 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 riguardano il suo comportamento nell'istante t_n.

D52:a.07 La transizione da una configurazione $\mathcal{C}_i$ di un automa alla successiva è determinata, oltre che dalla $\mathcal{C}_i$ stessa, dalle influenze esercitate sulla macchina dall'esterno. Anche queste sono trattate in modo discreto e possono essere rappresentate da stringhe di simboli appartenenti ad un alfabeto ben determinato chiamato **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 dall'automa mediante una **testina di lettura**.
I simboli di input possono essere pensati come comandi inviati all'automa da un operatore che costituisce il suo mondo esterno, oppure come informazioni che l'automa cattura dall'ambiente nel quale agisce.

D52:a.08 Nei casi più semplici gli automi sono influenzati dall'esterno solo attraverso la configurazione iniziale e hanno una evoluzione finita descritta da una sequenza finita di coppie di istanti e configurazioni:

$$\begin{array}{lccccc} \textit{istante} & 0 & 1 & 2 & \ldots & n \\ \textit{configurazione} & \mathcal{C}_0 & \mathcal{C}_1 & \mathcal{C}_2 & \ldots & \mathcal{C}_n \end{array}$$

Una di queste macchine dunque è in grado di fornire una funzione esplicita, quella che a un dato rappresentato in $\mathcal{C}_0$ associa un risultato reperibile in $\mathcal{C}_n$.

D52:a.09 Altri automi, invece, in certe circostanze hanno una evoluzione illimitata, che si può protrarre per un numero di passi grande a piacere, rappresentata da una coppia di successioni come le seguenti:

$$\begin{array}{lcccccc} \textit{istante} & 0 & 1 & 2 & \ldots & n & \ldots\ldots \\ \textit{configurazione} & \mathcal{C}_0 & \mathcal{C}_1 & \mathcal{C}_2 & \ldots & \mathcal{C}_n & \ldots\ldots \end{array}$$

Dalle evoluzioni più semplici di questo genere si possono ottenere insiemi numerabili come $\mathbb{N}$ e A^*, il monoide libero sull'alfabeto A.

Per un automa si pone quindi il problema di stabilire quali tipi di evoluzione subirà a partire dalle sue possibili configurazioni iniziali: come vedremo si tratta di un problema impegnativo che, tra l'altro, si rivela impossibile risolvere in generale (v. D57:, D58:).

D52:b. Riconoscitori di Rabin e Scott

D52:b.01 Introduciamo ora la più semplice tra le collezioni di automi in grado di individuare linguaggi, la classe dei cosiddetti **riconoscitori a stati finiti** o **riconoscitori di Rabin e Scott** o, in breve, **riconoscitori -RS**. La loro portata non è molto ampia, in quanto la varietà dei linguaggi da essi individuati non è molto ricca: si tratta comunque di linguaggi con interessanti applicazioni (come suggeriremo con esempi ed esercizi) e con il pregio di poter essere controllati con procedimenti piuttosto semplici e di poter essere trattati in modo sistematico con metodi algebrici.

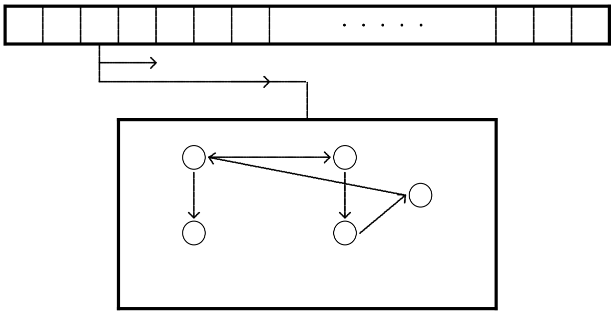
D52:b.02 Diciamo **riconoscitore di Rabin e Scott deterministico** o, in breve **riconoscitore -RSD**, una struttura della forma:

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

ove Q è un insieme finito detto **insieme degli stati** del riconoscitore -RSD, $\imath \in Q$ è detto **stato iniziale**, $F \subseteq Q$ è detto insieme degli **stati finali** di $\boldsymbol{R}$, T è un alfabeto detto **alfabeto di input** e $\delta \in \{Q \times \mathsf{T} \mapsto Q\}$ è detta **funzione di transizione** del riconoscitore -RSD.

Un tale automa $\boldsymbol{R}$ si può descrivere come costituito da una unità centrale formata da un insieme finito di stati collegati secondo le indicazioni di δ e 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.

Questo automa può presentarsi con una figura come la seguente:



D52: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 da sottoporre alla macchina.
Ad ogni mossa la testina legge un simbolo $\mathtt{a}$ nella casella sulla quale si trova ed 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. In pratica conviene usare per una configurazione una notazione del tipo $\langle q, w\rangle$, dove q è lo stato attuale e w è la stringa che la testina di lettura deve ancora leggere sul nastro di input (la testina si trova sulla sua prima casella). Chiaramente l'evoluzione di un riconoscitore *RRSD* 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 $\langle q_f, \mu\rangle$.

D52:b.04 Tra le stringhe che si possono sottoporre ad $\boldsymbol{R}$ si privilegiano quelle la cui lettura porta dallo stato iniziale ad uno degli stati finali. L'insieme di queste stringhe costituisce quello che si chiama il **linguaggio riconosciuto** o **accettato** dalla macchina e si indica con $\boldsymbol{R}^{\mathcal{A}}$.
Ogni evoluzione di un riconoscitore *RRSD* 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 e Scott si dice **linguaggio razionale** o anche **linguaggio regolare** o **linguaggio riconoscibile a stati finiti**.
Nel seguito indicheremo con **Rcn** la classe di tutti i riconoscitori -RS e con $\mathsf{Rcn}_{\mathsf{T}}$ la classe di quelli aventi T come alfabeto di input. L'insieme dei linguaggi razionali generici e sull'alfabeto T si indicano, risp., con $\mathsf{Rcn}^{\mathcal{A}}$ ed $\mathsf{Rcn}_{\mathsf{T}}{}^{\mathcal{A}}$.

D52: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 ed 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 il computer il loro comportamento.

Per presentare un riconoscitore -RSD ad una persona conviene servirsi di una raffigurazione del suo digrafo delle transizioni. Per precisare tale nozione ricordiamo una struttura che costituisce un arricchimento dei digrafi (D28:).

D52:b.06 Si dice **pluridigrafo** 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 $\mathcal{U} : \mathsf{T} \longmapsto \mathfrak{P}(Q \times Q)$. Ad ogni $\mathtt{a} \in \mathsf{T}$ corrisponde un digrafo $\langle Q, \mathcal{U}(\mathtt{a})\rangle$; le coppie costituenti $\mathcal{U}(\mathtt{a})$ si dicono **archi di $\boldsymbol{P}$ etichettati da** $\mathtt{a}$; le terne $\langle p, \mathtt{a}, q\rangle$ t.c. $\langle p, q\rangle \in \mathcal{U}(\mathtt{a})$ si dicono transizioni di $\boldsymbol{P}$; un arco etichettato da un simbolo $\mathtt{a}$ si dice anche **transizione indotta da** $\mathtt{a}$ o in breve $\mathtt{a}$-**transizione**.

Un pluridigrafo quindi si ottiene sovrapponendo alcuni digrafi aventi lo stesso insieme di nodi.
$\boldsymbol{P} = \langle Q, \mathsf{T}, \mathcal{U}\rangle$ si può rappresentare assommando le raffigurazioni dei digrafi che lo compongono etichettando con $\mathtt{a}$ ogni arco proveniente da un $\mathcal{U}(\mathtt{a})$; alternativamente si raffigura etichettando ogni arco del digrafo $\langle Q, \cup_{\mathtt{a} \in \mathsf{T}} \mathcal{U}(\mathtt{a})\rangle$ con i simboli dei sistemi di archi $\mathcal{U}(\mathtt{a})$ dai quali l'arco proviene.

D52:b.07 Si dice inoltre **pluridigrafo inizializzato-finalizzato** o **[di]grafo 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.

Per i pluridigrafi ed i digrafi di transizione, in quanto arricchimenti di digrafi, si possono utilizzare nozioni, notazioni e termini dei digrafi; in particolare si possono considerare passeggiate aperte e chiuse su queste strutture; su ogni loro arco si ha un'etichetta e su ogni passeggiata si può leggere una stringa.

D52:b.08 Definiamo ora il pluridigrafo ed il grafo di transizione di un riconoscitore -RSD $\boldsymbol{R}$. Il suo pluridigrafo ha come nodi gli stati di $\boldsymbol{R}$, ha sistemi di archi funzionali etichettati dai simboli dell'alfabeto di input: precisamente da ogni nodo $q \in Q$ escono gli archi etichettati $\langle q, \mathtt{a}, \delta(q, \mathtt{a})\rangle$ per $\mathtt{a} \in \mathsf{T}$. Il **digrafo di transizione** di $\boldsymbol{R}$ si ottiene arricchendo il suo pluridigrafo con lo stato iniziale e con l'insieme degli stati finali di $\boldsymbol{R}$. Nelle raffigurazioni dei digrafi di transizione contraddistingueremo lo stato iniziale con un segno "–" e gli stati finali con "+".

Il processo di analisi di una stringa $w \in \mathsf{T}^*$ può vantaggiosamente riferirsi alla passeggiata sul grafo 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 digrafo, passeggiate che vanno dallo stato iniziale ad uno stato finale, e leggendo su ogni passeggiata i simboli incontrati successivamente sugli archi.

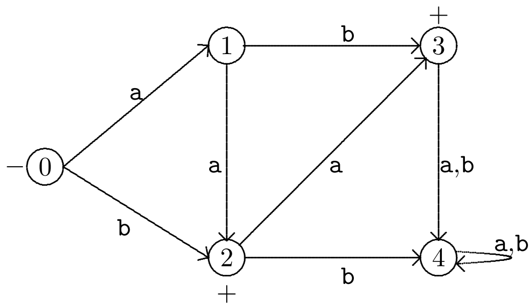
D52:b.09 Un primo esempio di riconoscitore -RSD è:

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

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 ad uno finale è la seguente:

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

Qui abbiamo usato il simbolo "$\vdash$" per indicare la relazione funzionale che associa ad una configurazione di un automa deterministico la successiva. Quando si considerano diversi riconoscitori come $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$, si possono rendere necessarie scritture più circostanziate come $\vdash_{\boldsymbol{R}_1}$ e $\vdash_{\boldsymbol{R}_2}$.
La suddetta evoluzione corrisponde al cammino $\langle[0, 1, 2, 3]\rangle$. 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 ${\boldsymbol{R}_a}^{\mathcal{A}} = \{\mathtt{b}, \mathtt{a}^2, \mathtt{ab}, \mathtt{ba}, \mathtt{a}^3\}$.

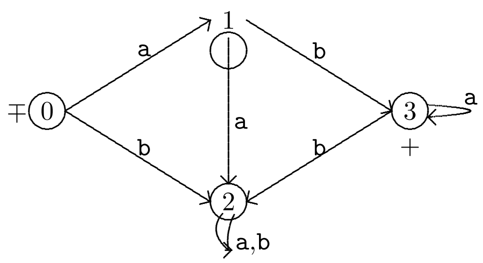
D52:b.10 Un secondo esempio di riconoscitore -RSD è:

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

ove per δ_b:

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

Esso ha come grafo di transizione:



D52:b.11 Si osservi che di ${\boldsymbol{R}_b}^{\mathcal{A}}$ fa parte la stringa muta μ. Questo è dovuto 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.: La stringa muta deve corrispondere ad un cammino di lunghezza 0 ▮

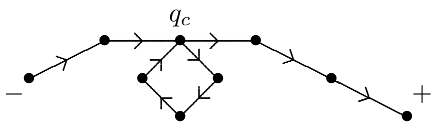
D52:b.12 Altre stringhe accettate da $\boldsymbol{R}_b$ sono: `ab, aba, abaa, abaaa`. In complesso si trova:

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

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

D52:b.13 Prop. Il linguaggio accettato da un riconoscitore $\boldsymbol{R}$ è infinito sse nel suo grafo di transizione si trova un circuito che tocca uno stato che appartiene ad 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. In tal caso la passeggiata relativa ad 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 ed una parte finale da q_c a un nodo finale.

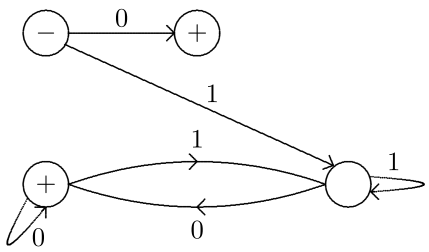
"Se" Se esiste una passeggiata utile con un nodo q_c appartenente ad un circuito, indichiamo con v la stringa che si legge sulla passeggiata tra il nodo iniziale e q_c, con w la stringa che si legge sul circuito e con y la stringa che si legge sulla passeggiata da q_c al nodo finale. È chiaro che tutte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

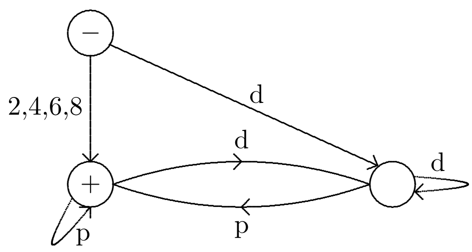


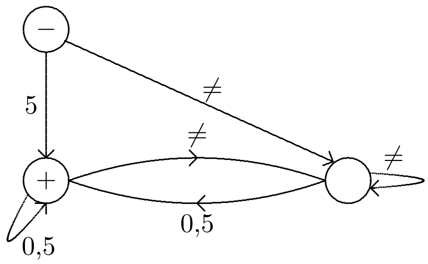
D52:b.14 Un linguaggio razionale può essere individuato da vari riconoscitori -RSD e la definizione consente che questi automi abbiano stati e transizioni ridondanti. Questa ridondanza degli strumenti che consentono di individuare un linguaggio è un fenomeno generale, tanto più marcato quanto più elaborati sono gli strumenti (come in particolare vedremo tra breve introducendo i riconoscitori non-deterministici). Si pone quindi il problema di trovare strumenti (come i riconoscitori) con pregi particolari; questi tipi di oggetti si dicono trovarsi in una qualche **forma canonica** o **forma normale**.

D52:b.15 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 $\boldsymbol{R}_b$. Il digrafo così semplificato è connesso ed il suo stato iniziale è una radice.
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 dati simboli. Per mantenere la definizione si può introdurre un cosiddetto **stato trappola**, stato non finale dal quale escono solo dei cappi. Si tratta allora di eliminare ogni stato non raggiungibile dallo stato iniziale e di fondere in un solo stato trappola tutti gli stati raggiungibili dall'iniziale ma dai quali non si può raggiungere alcun stato finale.
Nelle raffigurazioni, peraltro, lo stato trappola può essere trascurato.

D52:b.16 Presentiamo ora graficamente alcuni riconoscitori in grado di individuare scritture numeriche posizionali di interi particolari.





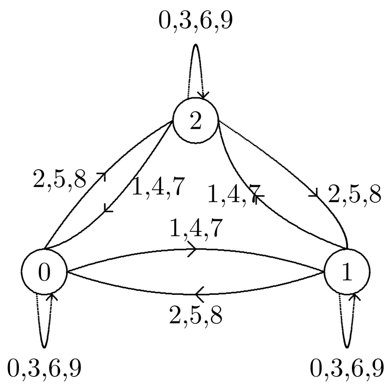


Il primo dei precedenti riconoscitori accetta le scritture binarie degli interi naturali pari; il linguaggio riconosciuto dal secondo riguarda le scritture decimali degli interi positivi pari; il terzo riconoscitore accetta le scritture decimali degli interi positivi divisibili per 5.

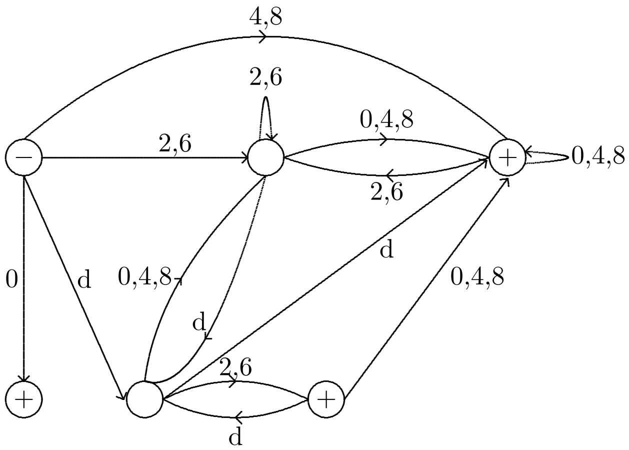
Un primo facile esercizio per prendere confidenza con queste strutture è dato da:

D52:b.17 Eserc. Tracciare un riconoscitore che accetta le stringhe costituite dalle lettere `a` e `b` e che non presentano due `a` ripetute, nè tre `b` ripetute.

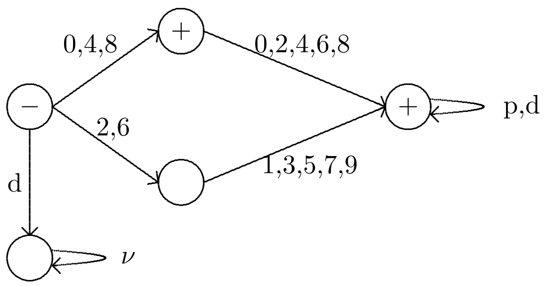
D52:b.18 Il seguente riconoscitore accetta le scritture decimali degli interi positivi divisibili per 3:



D52:b.19 Un riconoscitore che accetta le scritture decimali degli interi positivi divisibili per 4 è rappresentato da (in queste figure "d" rappresenta cifre dispari, "p" cifre pari e "ν" cifre arbitrarie):

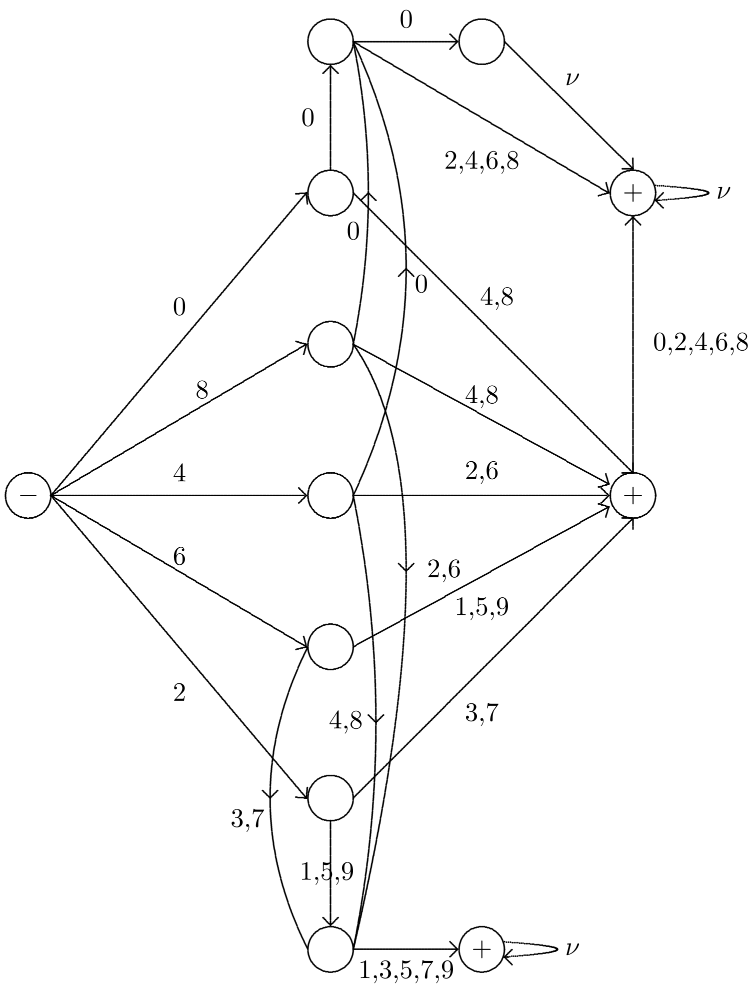


D52:b.20 Un riconoscitore che individua le riflesse delle scritture decimali degli interi positivi divisibili per 4 viene raffigurato da:



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.

D52:b.21 Il seguente riconoscitore accetta invece le stringhe riflesse delle scritture decimali dei numeri interi positivi divisibili per 8.



D52:b.22 Eserc. Precisare dei riconoscitori atti ad individuare le scritture decimali degli interi positivi divisibili per 7 e per 11.

D52:b.23 Eserc. Delineare riconoscitori che accettano le scritture che esprimono numeri reali in linguaggi di programmazione procedurale come BASIC, Fortran, C (e Java), ...

D52:b.24 Delineare un riconoscitore che accetti i numeri dei telefoni fissi italiani ed uno per gli internazionali.

D52:c. Riconoscitori non-deterministici

D52:c.01 Per capire 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 gli **automi** detti **deterministici**, 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 ad un altro unico stato; in altre parole ogni sua configurazione è determinata senza incertezze. Talora sono utili automi con comportamenti più "vaghi": qui tratteremo gli automi non-deterministici; di grande interesse sono anche gli automi probabilistici collegati alle catene di Markov.

Un riconoscitore di Rabin e Scott di tipo non-deterministico è associato ad un grafo di transizione generico, nel quale cioè si possono avere più stati iniziali e sistemi di archi non necessariamente funzionali; inoltre possono avere archi etichettati dalla stringa muta μ. Considerando su questo pluridigrafo le passeggiate utili e le stringhe che si leggono scorrendoli si individua ancora un linguaggio.

D52:c.02 Diciamo dunque **riconoscitore di Rabin e Scott non-deterministico**, 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 $\delta \in \{Q \times (\mathsf{T} \dot\cup \mu) \mapsto \mathfrak{P}(Q)\}$ 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 ad ogni stato di un automa corrisponda un indicatore luminoso. L'evoluzione di un automa deterministico si manifesta con l'accendersi in istanti successivi di singole luci; in un passo evolutivo di un automa non-deterministico, invece, si può assistere all'accendersi ed allo spegnersi di più luci.
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 $\langle p, \mathsf{a}, q \rangle \in Q \times (T \dot\cup \mu) \times Q$ t.c. $q \in \delta(p, \mathsf{a})$; anche un riconoscitore -RSNd è individuato dall'insieme delle sue transizioni e dagli insiemi dei suoi stati iniziali e finali.

D52:c.03 Dimostriamo ora l'equivalenza dei due tipi di riconoscitori introdotti.

Prop. Ogni riconoscitore -RSNd si può trasformare in un riconoscitore -RSD equivalente, cioè in grado di accettare lo stesso linguaggio.

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

Osserviamo preliminarmente che un riconoscitore -RSNd si può arricchire senza modificare il linguaggio accettato L aggiungendogli transizioni $\langle q_0, \mu, q_r \rangle$, se assenti, in corrispondenza di ogni passeggiata da q_0 a q_r con archi etichettati da μ e quindi aggiungendo ai suoi stati iniziali gli stati raggiungibili da stati iniziali con μ-transizioni ed ai suoi stati finali gli stati che li possono raggiungere con μ-transizioni. L viene fornito da quelle che chiameremo passeggiate μ-ridotte di questo nuovo riconoscitore, le passeggiate utili che non presentano μ-transizioni iniziali, finali e ripetute.
Si tratta ora di modificare il riconoscitore eliminando le sue μ-transizioni una alla volte. Vediamo come eliminare una $\langle p, \mu, q \rangle$; chiamiamo $\langle p_i, \mathsf{a}_i, p \rangle$ le transizioni con estremità finale p e $\langle q, \mathsf{b}_j, q_j \rangle$ le transizioni con estremità iniziale q, limitandoci a quelle etichettate da simboli di T.

Si può eliminare $\langle p, \mu, q\rangle$ pur di aggiungere le varie transizioni $\langle p_i, \mathtt{a}_i, q\rangle$ e $\langle p, \mathtt{b}_j, q_j\rangle$, qualora manchino: infatti le passeggiate μ-ridotte contenenti l'arco trascurato possono essere rimpiazzate con passeggiate di lunghezza inferiore di 1 contenenti uno dei nuovi archi.

Vediamo ora come si può trasformare un riconoscitore -RSNd $\boldsymbol{R}'$ privo di archi etichettati da μ in un riconoscitore -RSD $\boldsymbol{R}''$ equivalente. La trasformazione è formalmente assai semplice: si tratta di considerare come stati di $\boldsymbol{R}''$ alcune collezioni di stati di $\boldsymbol{R}$'. Come stato iniziale di $\boldsymbol{R}''$ si assume l'intero insieme I degli stati iniziali di $\boldsymbol{R}'$; come stato di $\boldsymbol{R}''$ ottenuto da I in seguito alla lettura di $a \in \mathsf{T}$ si assume $\cup_{q\in I}\delta(q, \mathtt{a})$. 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:

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

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

Infine, come collezione degli stati finali di $\boldsymbol{R}''$ si assume 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 grafo 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 ad una passeggiata utile su $\boldsymbol{R}'$ a cui si associa una passeggiata utile su $\boldsymbol{R}''$. Quindi ${\boldsymbol{R}'}^{\mathcal{A}} \subseteq {\boldsymbol{R}''}^{\mathcal{A}}$ ed in conclusione i due riconoscitori sono equivalenti ∎

D52: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 k a 2^k. Quindi si viene ad adottare uno strumento che presenta un tipo di struttura più semplice, ma contiene un maggior numero di componenti: questo è un aspetto tipico delle trasformazioni di uno strumento formale in una forma canonica.

L'elemento cruciale della costruzione precedente è il mantenimento della finitezza degli stati. A questo proposito occorre osservare che gli stati forniscono un dispositivo di memoria per il processo di riconoscimento delle stringhe. Quando nella analisi di una stringa 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 e Scott si basa su questo dispositivo di memoria e, come vedremo, trova nella sua finitezza i suoi limiti.

D52: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 stato trappola.

Prop. Un linguaggio razionale L è individuato da un riconoscitore deterministico privo di stato trappola sse è un linguaggio cometa $\mathsf{L} = \mathsf{T}^*\mathsf{N}$ con N linguaggio razionale.

Dim.: "Solo se" Sia $\boldsymbol{R}$ un riconoscitore deterministico privo di passeggiate palesememente inutili che riconosca L privo di stato trappola; sottoponendogli una qualsiasi stringa $x \in \mathsf{T}^*$ si fa passare il controllo dallo stato iniziale ad uno stato q t.c. esiste una passeggiata che porta da esso ad uno stato finale. Se y è la stringa leggibile su questa passeggiata, $xy \in \mathsf{L}$. Per l'arbitrarietà di x si ha $\mathsf{L} = \mathsf{T}^*\mathsf{N}$

con N accettato dal riconoscitore non-deterministico $\boldsymbol{R}$' ottenuto da $\boldsymbol{R}$ chiedendo che ogni suo stato sia iniziale.
"Se" Se $\mathsf{L} = \mathsf{T}^*\mathsf{N}$ con N razionale, una qualsiasi stringa $x \in \mathsf{T}^*$ deve essere prefisso di una stringa $xy \in \mathsf{L}$, quindi la sua lettura da parte di un riconoscitore che accetta L non può condurre ad uno stato trappola ▮

D52:c.06 Eserc. Ricavare riconoscitori -RSD equivalenti 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

D52:c.07 Eserc. Dimostrare che un linguaggio razionale è individuato da un riconoscitore deterministico con stati finali dai quali non esce alcun arco che vada in uno stato non finale sse è una anticometa $\mathsf{L} = \mathsf{N}\mathsf{T}^*$ con N razionale.

D52:c.08 Eserc. Dare una caratterizzazione di riconoscitori in grado di accettare linguaggi razionali che sono bicomete.

D52:d. Elaborazioni di linguaggi razionali

D52:d.01 In questo paragrafo vediamo come si possono costruire nuovi riconoscitori di Rabin e Scott componendo riconoscitori dati; dovremo considerare coppie di riconoscitori $\boldsymbol{R}_i := \langle Q_i, I_i, F_i, \mathsf{T}, \delta_i \rangle$ per $i = 1, 2$ ed i relativi linguaggi $\mathsf{L}_i := \boldsymbol{R}_i{}^{\mathcal{A}}$. Se servono riconoscitori deterministici, invece degli insiemi I_i si considerano i singoli stati $\imath_i$. Gli stati di queste macchine formali si può assumere che formino insiemi disgiunti.

D52:d.02 Prop. L'unione $\mathsf{L}_1 \cup \mathsf{L}_2$ di due linguaggi razionali è un linguaggio razionale.
Dim.: Dati due riconoscitori $\boldsymbol{R}_i$ atti ad accettare i due linguaggi, per avere un riconoscitore che accetti il linguaggio unione basta considerare l'unione disgiunta dei due corrispondenti digrafi di transizione. Formalmente si tratta del riconoscitore

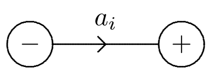
$$\langle Q_1 \dot\cup Q_2, I_1 \dot\cup I_2, F_1 \dot\cup F_2, \mathsf{T}, \delta_1 \dot\cup \delta_2 \rangle \text{ ▮}$$

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

D52: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 $\boldsymbol{R}_i$ t.c. $\boldsymbol{R}_i^{\mathcal{A}} = \mathsf{L}_i$, assumendo come insieme degli stati iniziali I_1, come insieme degli stati finali F_2 ed aggiungendo le transizioni $\langle f_h, \mu, i_k \rangle$ per ogni $f_h \in F_1$ ed 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 μ-transizioni e da una passeggiata utile su $\boldsymbol{R}_2$ ▮
Il precedente riconoscitore viene detto **composizione in serie** di $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$.

D52:d.04 Prop. Ogni stringa, comprese stringa muta e singoli simboli, 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 a_i è riconosciuto da:



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

D52:d.05 Prop. Tutti i linguaggi finiti sono razionali.

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

D52: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 $\boldsymbol{R} := \langle Q, \imath, F, \mathsf{T}, \delta\rangle$ atto ad accettare L e privo di stati palesemente inutili. Il riconoscitore $\langle Q, \imath, Q\setminus F, \mathsf{T}, \delta\rangle$, chiaramente, accetta le stringhe che il precedente rifiuta e rifiuta quelle che $\boldsymbol{R}$ accetta, cioè individua $\mathsf{T}^* \setminus \mathsf{L}$ ▮

D52: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 significativo riconoscitore per l'intersezione.
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
$\boldsymbol{R} = \big\langle Q_1 \times Q_2, \langle \imath_1, \imath_2\rangle, F_1 \times F_2, \mathsf{T}, \delta\big\rangle$, 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 rispettivamente in $\imath_1$ e $\imath_2$ e sono etichettati dalla stessa stringa. Questa corrispondenza è biunivoca e ad 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 ▮

D52:d.08 Prop. La 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 accettato L. Il riconoscitore -RSNd ottenuto aggiungendo al precedente le transizioni $\langle f_h, \mu, \imath\rangle$ 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 ▮

D52:d.09 Prop. La star-chiusura di un linguaggio razionale su T è razionale.

Dim.: Trascurando il fatto che questa proprietà si ottiene dalla precedente, dalla razionalità di $\{\mu\}$ e dalla razionalità della unione di due linguaggi razionali, consideriamo ancora il riconoscitore -RSD $R := \langle Q, \imath, F, \mathsf{T}, \delta\rangle$ ed il linguaggio individuato L. Il riconoscitore ottenuto aggiungendo al precedente un nuovo stato q_0 (assumendo che solo questo sia iniziale e finale), le transizioni $\langle f_h, \mu, q_0\rangle$ per ogni $f_h \in F$ e $\langle q_0, \mu, i_k\rangle$ per ogni $i_k \in I$ riconosce tutte e sole le stringhe di L^*. Infatti ogni passeggiata utile su questo riconoscitore 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 vuole ▮

D52:d.10 Prop. Il linguaggio riflesso di un linguaggio razionale è razionale.

Dim.: Si considerino il riconoscitore $R := \langle Q, I, F, \mathsf{T}, \delta\rangle$ ed il linguaggio individuato 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 $\boldsymbol{R}$▮

Ricordiamo le definizioni di funzione o e di derivata da sinistra rispetto alla stringa w del linguaggio L

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

D52:d.11 Prop. Per le derivate di un linguaggio accettato da un riconoscitore -RSD $\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta\rangle$ si ha:

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

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

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 è una conseguenza delle prima e della .07:3.1 ▮

D52:d.12 Prop. Le derivate da sinistra di un linguaggio razionale L sono linguaggi razionali e sono in numero finito ▮

D52:d.13 Sia L un linguaggio razionale su $\mathsf{T} = \{a_1, ..., a_n\}$ e per $i = 1, ..., n$ sia M_i un linguaggiio razionale su un alfabeto U. Il linguaggio N ottenuto sostituendo nelle stringhe di L ogni occorrenza di un simbolo a_i con il linguaggio M_i è razionale.

Dim.: Siano $\boldsymbol{R}$, $\boldsymbol{S}_1$, ... e $\boldsymbol{S}_n$ riconoscitori in grado di accettare risp. L, M_1, ... e M_n; consideriamo il riconoscitore $\boldsymbol{T}$ ottenuto sostituendo in $\boldsymbol{R}$ ogni arco etichettato $\langle p, \mathsf{a}_i, q\rangle$ con il pluridigrafo ottenuto dal digrafo di transizione di $\boldsymbol{S}_i$ aggiungendogli le transizioni $\langle p, \mu, p_h\rangle$ per tutti gli stati iniziali p_h di $\boldsymbol{S}_i$ e le transizioni $\langle q_k, \mu, q\rangle$ per tutti gli stati finali q_k di $\boldsymbol{S}_i$.
Il linguaggio N è 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 μ-transizioni dei tipi suddetti ▮

D52:d.14 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 ad una certa operazione unaria o binaria sui linguaggi. Osserviamo che tutte queste proprietà sono state dimostrate con procedimenti costruttivi.

D52:d.15 Eserc. Dimostrare che $\forall w \in \mathsf{T}^* \;:\; w \in L \Longleftrightarrow \mu \in w \setminus\!\!\setminus \mathsf{L}$.

D52:d.16 Eserc. Individuare un procedimento che consenta di decidere, per una qualsiasi coppia di linguaggi razionali L ed M, se $\mathsf{L} \subseteq \mathsf{M}$.

D52:d.17 Eserc. Individuare un riconoscitore -RSD in grado di accettare le scritture decimali degli interi divisibili per 6.

D52:e. Espressioni razionali e teorema di Kleene

D52:e.01 Quanto visto in precedenza mostra che la collezione dei linguaggi razionali si può considerare come un complesso di strumenti di elevata maneggevolezza.
Un altro pregio dei linguaggi razionali consiste nella possibilità di individuare ciascuno di essi con una espressione di un tipo particolare. Queste espressioni, come vedremo, consentono di effettuare agevolmente varie elaborazioni sui linguaggi razionali ed aumentano la loro maneggevolezza.

D52:e.02 Riferiamoci ad un alfabeto $\mathsf{T} = \{\mathtt{a}_1, ..., \mathtt{a}_n\}$ ed 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 $(\mathsf{E}_1 \cdot \mathsf{E}_2)$;
(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, giustapposizione e star-chiusura che possono essere assimilate alle ordinarie operazioni razionali di somma (in accordo con l'adozione del segno $+$), prodotto e divisione.

D52:e.03 Diciamo **linguaggio esprimibile razionalmente** ogni linguaggio ottenuto interpretando una espressione razionale come insieme di stringhe.

Esempi di espressioni razionali sono:

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

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 strumenti introdotti per scopi computazionali: partendo da una definizione semplice ed essenziale si ottiene uno strumento poco maneggevole. Nella pratica è opportuno servirsi di varianti semplificate nell'aspetto ma che devono riferirsi a definizioni più elaborate.
Nel caso delle espressioni razionali è naturale adottare semplificazioni come l'abolizione delle parentesi tonde che racchiudono l'intera espressione e di quelle semplificabili per implicita associatività di unione e giustapposizione, l'eliminazione delle parentesi graffe per i simboli, l'abolizione del segno di giustapposizione, l'uso delle potenze e l'uso di X^+ in luogo di XX^* o di $\mathsf{X}^*\mathsf{X}$. Inoltre si usa spesso il segno di somma invece del segno di unione binaria $\cup$.

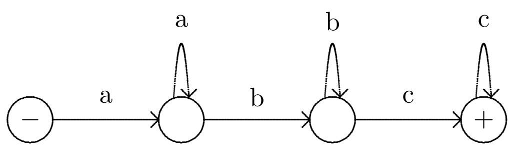
Le espressioni precedenti quindi si semplificano nelle seguenti:

$$\mathtt{a}^3 \qquad (\mathtt{ab})^* \qquad (\mathtt{a}+\mathtt{b})^* \qquad (\mathtt{a}^* + \mathtt{bc})^*,$$

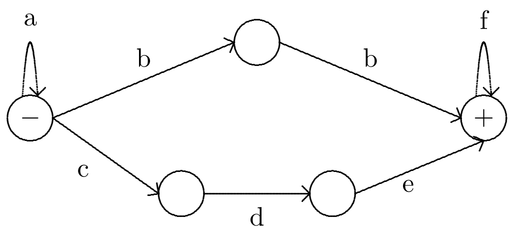
D52:e.04 Per le espressioni razionali più semplici non è difficile dare una descrizione del relativo linguaggio e conseguentemente individuare un riconoscitore capace di accettarlo.

Ad es. 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:

D52:e.05 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



D52:e.06 Il linguaggio dato dall'espressione: $\mathtt{a}^*(\mathtt{b}^2 + \mathtt{cde})\mathtt{f}^*$ è invece riconosciuto da:



D52:e.07 Ci proponiamo ora di dimostrare il classico teorema dovuto a Kleene della coincidenza dei linguaggi dati da espressioni razionali e dei linguaggi razionali, cioè dei linguaggi individuati da riconoscitori -RS.

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

D52:e.08 Prop. Ogni linguaggio esprimibile razionalmente è riconoscibile a stati finiti.

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

Abbiamo visto che esistono riconoscitori che individuano i linguaggi $\{\mu\}$ 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 i linguaggi 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 :d.02 .
Per avere un riconoscitore per il linguaggio espresso da $(\mathsf{E}_1 \cdot \mathsf{E}_2)$ basta considerare la composizione in serie di riconoscitori -RSNd vista in :d.03 .
Per avere un riconoscitore per il linguaggio espresso da $(\mathsf{E_1}^*)$ basta considerare l'arricchimento di $\boldsymbol{R}_1$ relativo alla star-chiusura visto in :d.09 ▮

D52:e.09 Poniamoci ora in grado di risolvere un tipo particolare di equazione per linguaggi.

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

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

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

D52:e.10 **Prop.** $\mathsf{U}^*\mathsf{V}$ è soluzione della $[*]$.

Dim.: Basta sostituire X con $\mathsf{U}^*\mathsf{V}$ nei due membri della $[*]$: $\mathsf{U}^*\mathsf{V} = \mathsf{UU}^*\mathsf{V} + \mathsf{V} = (\mathsf{U}^+ + \mu)\mathsf{V} = \mathsf{U}^*\mathsf{V}$ ▮

D52:e.11 **Prop.** $\mathsf{U}^*\mathsf{V}$ è contenuto in ogni soluzione Y della $[*]$.

Dim.: Dimostriamo per induzione che per ogni $n \in \mathbb{N}$ è $\mathsf{U}^n\mathsf{V} \subseteq \mathsf{Y}$.
La cosa vale per $n = 0$, in quanto $\mathsf{Y} = \mathsf{UY} + \mathsf{V}$ 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 $\mathsf{U}^{n+1}\mathsf{V} \subseteq \mathsf{Y}$.
$\mathsf{U}^{n+1}\mathsf{V} = \mathsf{U}(\mathsf{U}^n\mathsf{V}) \subseteq \mathsf{UY} \subseteq \mathsf{UY} + \mathsf{V} = \mathsf{Y}$ ▮

D52:e.12 **Prop.** Se $\mu \notin \mathsf{U}$, 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 e scriviamo $\ell := |z|$. Per essa $z \in \mathsf{Y} = \mathsf{UY} + \mathsf{V}$; dovendo essere $z \notin \mathsf{V} \subseteq \mathsf{U}^*\mathsf{V}$, si ricava $z \in \mathsf{UY} = \mathsf{U}^2\mathsf{Y} + \mathsf{UV}$; da questa, non potendo essere $z \in \mathsf{UV} \subseteq \mathsf{U}^*\mathsf{V}$, segue che $z \in \mathsf{U}^2\mathsf{Y}$. Procedendo in questo modo si ottiene $z \in \mathsf{U}^{\ell+1}\mathsf{Y}$; ma questa relazione è assurda, in quanto $\mu \notin \mathsf{U}$ implica che in $\mathsf{U}^{\ell+1}$ ci siano stringhe aventi lunghezza almeno pari ad $\ell + 1$ ▮

Osserviamo che l'equazione $[*]$ si può assimilare ad una equazione lineare nella X e che la soluzione si potrebbe pensare ottenuta con passaggi come i seguenti. $\mathsf{X} = \mathsf{UX} + \mathsf{V} \Longrightarrow (1 - \mathsf{U})\mathsf{X} = \mathsf{V} \Longrightarrow \mathsf{X} = (1 - \mathsf{U})^{-1}\mathsf{V} = (1 + \mathsf{U} + \mathsf{U}^2 + ...)\mathsf{V} = \mathsf{U}^*\mathsf{V}$.
Qui peraltro non possiamo approfondire l'analogia fra operazioni su linguaggi ed operazioni numeriche e la precedente "deduzione" può essere considerata solo come un aiuto mnemonico.

D52:e.13 Occorre ora una uguaglianza riguardante le derivate di un linguaggio.

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

$$\mathsf{L} = o(\mathsf{L}) + \mathtt{a}_1(\mathtt{a}_1 \mathbin{\backslash\!\backslash} \mathsf{L}) + ... + \mathtt{a}_n(\mathtt{a}_n \mathbin{\backslash\!\backslash} \mathsf{L}).$$

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 diversi simboli iniziali ▮

D52:e.14 Osserviamo che il secondo membro della precedente relazione è lineare a destra nelle derivate. La precedente relazione viene talora chiamata **formula di Taylor per i linguaggi**. Questo termine si giustifica sulla base della analogia tra i cosidetti anello delle funzioni di variabili reali e semianello dei linguaggi. In questa ottica i simboli dell'alfabeto sono visti come variabili, $o(L)$ si può pensare come una costante (che per il semianello dei linguaggi può essere solo lo zero $\emptyset$ o l'unità μ) e le derivate portano all'abbassamento del grado delle potenze senza comportare moltiplicazioni per fattori numerici.

D52:e.15 A questo punto siamo in grado di associare ad 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 $q \in Q$; L_q corrisponde al linguaggio accettato da $\boldsymbol{R}_q := \langle Q, q, F, \mathsf{T}, \delta\rangle$; per le incognite si ha $\mathsf{L}_q = \boldsymbol{R}_q{}^{\mathcal{A}}$.
L'equazione associata allo stato q corrisponde alla formula di Taylor per L_q:

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

In questa formula $o(\mathsf{L}_q)$ 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 a_i lo si ricava facilmente dalla osservazione di q e degli archi uscenti da tale stato.

A questo proposito serve la seguente relazione trovata in :D.11 per un riconoscitore -RSD $\boldsymbol{R} = \langle Q, \imath, F, \mathsf{T}, \delta\rangle$:

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

Dalle formule precedenti segue anche che le derivate rispetto a stringhe di un linguaggio razionale costituiscono un numero finito di linguaggi razionali, fatto già enunciato in :D.12 .

D52:e.16 Rimane ora da dimostrare l'ultimo fatto per il collegamento tra espressioni razionali e riconoscitori -RS.

Prop. Il sistema di equazioni associabili ad un riconoscitore -RSD è risolubile univocamente e conduce ad 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 successivi $|Q|$ passi in ciascuno dei quali si trasforma una equazione in una espressione della incognita a primo membro nelle rimanenti e quindi si può eliminare dal sistema l'incognita e la relativa espressione, cioè si può abbassare il suo grado (come avviene nel noto procedimento risolutivo dei sistemi delle ordinarie equazioni lineari).

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 dati da espressioni razionali e privi della stringa muta. Diciamo che le 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 può essere immediatamente 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 è del tipo [$*$]; 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**.

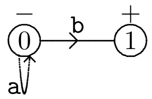
Quindi il procedimento risolutivo può essere portato fino alla fine, cioè fino ad avere una espressione razionale per $\mathsf{L}_\imath$ ed una catena di espressioni razionali per le altre incognite esplicitabili progressivamente.

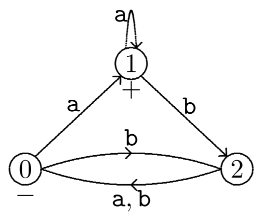
Risulta quindi che un linguaggio accettato a stati finiti è fornito da una espressione razionale e con esso anche tutte le sue derivate ▮

D52:e.17 Possiamo quindi enunciare:

Teorema (Kleene 1956) Un linguaggio è razionale, cioè è riconoscibile a stati finiti sse è esprimibile razionalmente ▮

D52:e.18 Illustriamo il procedimento risolutivo precedente con gli esempi relativi ai seguenti semplici riconoscitori:





Il primo porta alle equazioni:

$$\mathtt{L}_0 = \mathtt{a}\mathtt{L}_0 + \mathtt{b}\mathtt{L}_1$$
$$\mathtt{L}_1 = \mu;$$

dalle quali seguono evidentemente $\mathtt{L}_0 = \mathtt{a}\mathtt{L}_0 + \mathtt{b}$ e $\mathtt{L}_0 = \mathtt{a}^*\mathtt{b}$.
Si noti che abbiamo risolto la più semplice delle equazioni di tipo [∗].

D52:e.19 Al secondo dei precedenti riconoscitori è associato il seguente sistema di equazioni:

$$\mathtt{L}_0 = \mathtt{a}\mathtt{L}_1 + \mathtt{b}\mathtt{L}_2$$
$$\mathtt{L}_1 = \mathtt{a}\mathtt{L}_1 + \mathtt{b}\mathtt{L}_2 + \mu$$
$$\mathtt{L}_2 = \mathtt{a}\mathtt{L}_0 + \mathtt{b}\mathtt{L}_0.$$

Eliminando dalle prime due equazioni $\mathtt{L}_2$ grazie alla terza si ha:

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

Risolvendo la seconda equazione si ottiene

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

a questo punto si può riscrivere la prima come

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

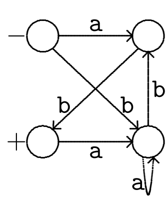
e risolvendola si ricava l'espressione razionale richiesta

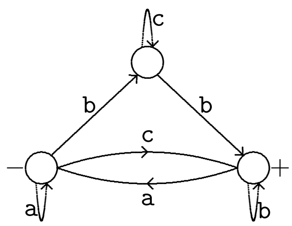
$$\mathtt{L}_0 = (\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\mathtt{a}^+.$$

Le uguaglianze che si erano ottenute consentono di avere anche le espressioni per le derivate di $\mathtt{L}_0$:

$$\mathtt{L}_1 = \mathtt{a} \mathbin{\backslash\!\backslash} \mathtt{L}_0 = a^*(\mathtt{b}(\mathtt{a}+\mathtt{b})(\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\mathtt{a}^+ + \mu),$$
$$\mathtt{L}_2 = \mathtt{b} \mathbin{\backslash\!\backslash} \mathtt{L}_0 = (\mathtt{a}+\mathtt{b})(\mathtt{a}^+\mathtt{b}(\mathtt{a}+\mathtt{b}))^*\mathtt{a}^+.$$

D52:e.20 Eserc. Trovare una espressione razionale per i linguaggi accettati dai riconoscitori:

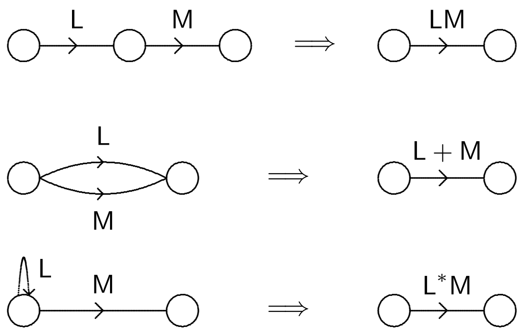




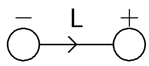
D52:e.21 Eserc. Esaminare alcuni riconoscitori -RSD con digrafo delle transizioni aciclico e mostrare come il procedimento risolutivo delle relative equazioni lineari porti ad espressioni razionali polinomiali, cioè ad espressioni nelle quali non compare la star-chiusura, e quindi a linguaggi finiti corrispondenti agli insiemi finiti delle passeggiate utili sui grafi.

D52:e.22 Eserc. Data un'espressione razionale come la $(\mathtt{a} + \mathtt{b}^*)(\mathtt{a} + \mathtt{b})(\mathtt{a}^* + \mathtt{b})$, individuare un riconoscitore -RSNd ad essa equivalente, trasformarlo in un riconoscitore -RSD e risolvere le relative equazioni; confrontare quindi l'espressione razionale trovata con quella di partenza.

D52:e.23 Eserc. Generalizzare la definizione di digrafo 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 digrafo delle transizioni come progressiva riduzione di tali digrafi ottenuti mediante fusione di archi in serie, fusione di archi in parallelo ed eliminazione di cappi, cioè mediante trasformazioni del tipo:



fino ad ottenere un digrafo del tipo a transizione unica:



D52:f. Riconoscitore minimo di un linguaggio razionale

D52:f.01 Un importante problema riguarda l'equivalenza di due espressioni razionali, ovvero l'equivalenza di due riconoscitori di Rabin e Scott $\boldsymbol{R}_1$ ed $\boldsymbol{R}_2$. Un modo per risolverlo consiste nel provare che $\boldsymbol{R}_1^{\mathcal{A}} \subseteq \boldsymbol{R}_2^{\mathcal{A}}$ e $\boldsymbol{R}_1^{\mathcal{A}} \supseteq \boldsymbol{R}_2^{\mathcal{A}}$, ovvero che $\boldsymbol{R}_1^{\mathcal{A}} \setminus \boldsymbol{R}_2^{\mathcal{A}} = \boldsymbol{R}_2^{\mathcal{A}} \setminus \boldsymbol{R}_1^{\mathcal{A}} = \emptyset$, ovvero che $\boldsymbol{R}_1^{\mathcal{A}} \cap (\mathsf{T}^* \setminus \boldsymbol{R}_2^{\mathcal{A}}) = \boldsymbol{R}_2^{\mathcal{A}} \cap (\mathsf{T}^* \setminus \boldsymbol{R}_1^{\mathcal{A}}) = \emptyset$.
Questo procedimento però può risultare pesante; un metodo più efficiente e significativo riguarda la individuazione di un tipo particolarmente pregevole di riconoscitore per un linguaggio razionale.

D52:f.02 Consideriamo per il momento il linguaggio generico $L \subseteq \mathsf{T}^*$ ed indichiamo con $\mathcal{D} := \{w \in \mathsf{T}^* \text{ t.c. } w \mathbin{\backslash\!\backslash} \mathsf{L}\}$ l'insieme delle sue derivate da sinistra. Questo insieme, non necessariamente finito, si può considerare come insieme degli stati di una sorta di automa detto **riconoscitore delle derivate** di L:

$$\Big\langle \mathcal{D} \; , \; \mathsf{L} \; , \; \{z \in \mathsf{L} \mid z \mathbin{\backslash\!\backslash} \mathsf{L}\} \; , \; \mathsf{T} \; , \; \ulcorner \langle w \mathbin{\backslash\!\backslash} \mathsf{L}, \mathsf{a}\rangle \in \mathcal{D} \times \mathsf{T} \mapsto (w\mathsf{a}) \mathbin{\backslash\!\backslash} \mathsf{L} \urcorner \Big\rangle .$$

La precedente scrittura individua un vero e proprio riconoscitore -RSD sse $\mathcal{D}$ è finito.

D52:f.03 Prop. Il riconoscitore delle derivate di un linguaggio L avente un insieme finito di derivate 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 $w \in \mathsf{L}$ ▮

D52:f.04 A questo punto possiamo anche affermare:

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 (v. :d.12). D'altro lato qui sopra si è mostrato come la finitezza delle derivate da sinistra garantisce l'esistenza di un riconoscitore -RS che lo accetta ▮

D52:f.05 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. Un riconoscitore in effetti può presentare stati ridondanti anche se privo di stati palesemente inutili.
Per poter disporre effettivamente dei riconoscitori delle derivate occorre individuare un procedimento che permette di stabilire tutte e sole le coincidenze fra derivate 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.

D52:f.06 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 $\mathsf{L} := \boldsymbol{R}^{\mathcal{A}}$.

Prop. Consideriamo due stati q_j e q_h del riconoscitore $\boldsymbol{R}$, due stringhe u_j e u_h t.c. $q_j = \delta(\imath, u_j)$ 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 $\mathsf{L}_{q_h} := \langle Q, q_h, F, \mathsf{T}, \delta\rangle^{\mathcal{A}}$. $\mathsf{L}_{q_j} = \mathsf{L}_{q_h} \iff u_j \setminus\!\!\setminus \mathsf{L} = u_h \setminus\!\!\setminus \mathsf{L}$ ▮

D52:f.07 Prop. Se in $\boldsymbol{R}$ vi sono due stati come i precedenti q_j e q_h che forniscono la stessa derivata di L si può ottenere un riconoscitore che accetta lo stesso L mediante la fusione di q_j e q_h ed eventuali 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 il suo 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 a $\boldsymbol{R}$▮

D52:f.08 La precedente dimostrazione non indica chiaramente un procedimento costruttivo per ridurre il numero degli stati di un riconoscitore -RSD, in quanto non dice come stabilire l'uguaglianza $\mathsf{L}_{q_j} = \mathsf{L}_{q_h}$, ovvero l'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 fra stringhe.

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

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

Evidentemente $u \,\mathrm{M}_{\mathsf{L}}\, v$ sse $u \setminus\!\!\setminus \mathsf{L} = \{z \in \mathsf{T}^* \text{ t.c. } uz \in \mathsf{L}\} = \{z \in \mathsf{T}^* \text{ t.c. } vz \in \mathsf{L}\} = v \setminus\!\!\setminus \mathsf{L}$.

Si tratta quindi di trovare un procedimento per individuare l'equivalenza M_{L} per un generico linguaggio razionale L.

D52:f.09 Consideriamo la seguente successione di equivalenze entro l'insieme degli stati del riconoscitore -RSD $\boldsymbol{R}$:

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

Le suddette equivalenze si possono individuare facilmente operando sul digrafo delle transizioni di $\boldsymbol{R}$.
Evidentemente E_0 corrisponde alla bipartizione $Q = (Q \setminus F) \dot\cup F$.
Per trovare E_1 si considerano i diversi simboli $\mathsf{a} \in \mathsf{T}$ e per ciascuno di essi si distinguono i nodi finali che la lettura di a porta in un altro finale, quelli che a porta in uno non finale, i nodi di $Q \setminus F$ che a lascia in $Q \setminus F$ e i nodi non finali che la lettura di a sposta in F. Tutte queste distinzioni si ottengono con un semplice esame degli archi etichettati da a uscenti dai vari nodi.
Osserviamo poi che al crescere di h le equivalenze E_h sono sempre più fini, cioè che $\forall h \in \mathbb{N} : \; \mathrm{E}_h \supseteq \mathrm{E}_{h+1}$.
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.
Essendo Q finito, il processo di raffinamento della equivalenze E_h non può continuare illimitatamente, ma per un certo t si ha $\mathrm{E}_t = \mathrm{E}_{t+1}$. A questo punto non si può più avere nessun altro raffinamento: se si trovasse un raffinamento per il quale $\mathrm{E}_{t+1} \supset \mathrm{E}_{t+2}$, si sarebbe potuto trovare un raffinamento anche per E_t.

D52:f.10 L'equivalenza trovata si verifica facilmente coincidere con la congruenza di Myhill.
Inoltre $q \, \mathrm{M}_L \, q'$ 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 di M_{L}:

$$\delta/\mathrm{M}_{\mathsf{L}} \; := \; \left\lceil \, \langle q ⌑ \mathrm{M}_{\mathsf{L}}, \mathsf{a} \rangle \in Q/\mathrm{M}_{\mathsf{L}} \times \mathsf{T} \;\; \not\mapsto \;\; \delta(q, \mathsf{a}) ⌑ \mathrm{M}_{\mathsf{L}} \, \right\rceil \; .$$

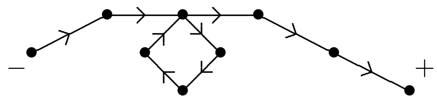
È dunque lecito considerare il riconoscitore i cui stati sono ottenuti "fondendo" gli elementi delle varie classi dell'equivalenza M_{L}:

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

Gli stati di questo riconoscitore corrispondono biunivocamente alle derivate di L. Quindi se si applicasse la stessa costruzione ad un altro riconoscitore -RSD che accetta L, si giungerebbe allo stesso automa, a meno di inessenziali differenze nel modo di individuare gli stati, cioè a meno di isomorfismi.
È quindi lecito chiamare il precedente automa **riconoscitore minimo** del linguaggio razionale L o **riconoscitore delle derivate** di L.

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

D52:f.11 Eserc. Costruire il riconoscitore minimo equivalente al seguente:



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