1

Capitolo C13
applicazioni di riconoscitori-RS e trasduttori-R

Contenuti delle sezioni

a. applicazioni dei linguaggi razionali p. 2
b. accettatori-trasduttori-R p. 5

8 pagine

C13 0.01 Questo capitolo è dedicato quasi esclusivamente alle applicazioni dei riconoscitori di Rabin-Scott e delle macchine sequenziali che consistono in loro arricchimenti mediante l'aggiunta di etichette o di altre semplici informazioni ai loro nodi e ai loro archi.

Un solo paragrafo è dedicato a cenni delle macchine e dei linguaggi di portata più generale.

C13 a. applicazioni dei linguaggi razionali

C13 a.01 Come abbiamo avuto modo di vedere, i linguaggi razionali costituiscono una classe di linguaggi particolarmente maneggevoli. Sono note procedure che consentono di risolvere molti problemi che li riguardano ed essi si possono definire e trattare con strumenti comodi e significativi come i riconoscitori minimi, le espressioni razionali e con strumenti algebrici collocabili nella teoria dei semigruppi finiti.

D'altra parte la classe dei linguaggi razionale ha dei limiti piuttosto precisi. Vediamo a questo proposito alcuni semplici linguaggio che non sono razionali.

C13 a.02 Prop. Il linguaggio $\mathsf{L} := \sum_{n\in\mathbb{N}} \mathtt{a}^n\,\mathtt{b}^n$ non è razionale.

Dim.: L'enunciato equivale ad affermare che non è finito l'insieme delle sue derivate a sinistra.
Infatti questa collezione è costituita dai seguenti linguaggi:

$\{\emptyset\} = \mathtt{b}^m \setminus\!\setminus \mathsf{L} = b^m\{\mathtt{a},\mathtt{b}\}^*$, per ogni $m \in \mathbb{P}$;
$\mathtt{a}^n \setminus\!\setminus \mathsf{L} = \mathsf{L}\mathtt{b}^n$, per ogni $n \in \mathbb{N}$;
$\mathtt{b}^* = a^m \setminus\!\setminus \mathsf{L}$ per ogni $m \in \mathbb{P}$ ∎

In modo analogo, cioè riconoscendo insiemi non finiti di linguaggi derivati si può stabilire che non siano razionali ampliamenti del linguaggio precedente come il linguaggio di Dyck e gli ulteriori ampliamenti relativi alle usuali espressioni algebriche e alle espressioni disponibili nei linguaggi di programmazione procedurali (Fortran, BASIC, C, C++, Java, PHP, Python, ...).

In effetti vedremo che per controllare questi linguaggi artificiali è necessario mettere a punto strumenti assai più elaborati dei riconoscitori-RSND: un esempio è fornito dalle grammatiche e dagli algoritmi costituenti la tecnologia del parsing.

Altri linguaggi non razionali che che incontreremo più avanti ma conviene segnalare subito sono $\sum_{n\in\mathbb{N}} a^n\,b^n\,c^n$, $\sum_{niN} a^n\,b^n\,a^n$, $\sum_{niN} a^n\,b^{2n}\,c^{3n}$ e il linguaggio su un solo carattere formato dalle stringhe $|^p$ con p numero primo.

C13 a.03 Peraltro i linguaggi razionali, in virtù della loro maneggevolezza, hanno molteplici applicazioni nella pratica della programmazione, in molti capitoli della teoria dei linguaggi formali, nella teoria dei codici [C65, C66], nel trattamento dei testi, nello studio dei circuiti elettrici, nell'analisi di processi produttivi, nella chimica, nella biologia, ...

Le espressioni razionali consentono di esprimere comodamente ed efficacemente schemi di stringhe e questo fa di loro uno strumento molto utile per definire in modo semplice la sintassi di frammenti dei linguaggi artificiali e di semplici frasi dei linguaggi naturali.

Inoltre va detto che per implementare le analisi degli accennati frammenti e per realizzare ricerche e le manipolazioni dei testi vengono utilizzati sistematicamente i riconoscitori di Rabin-Scott deterministici, le espressioni razionali e parecchie loro varianti

Accade quindi che entrambe le entità collegate dal teorema di Kleene vengono sfruttate in applicazioni molto concrete.

C13 a.04 Tra le attività di programmazione nelle quali si utilizzano macchine a stati finiti del livello-R, cioè riconducibili a quelle presentati in C12 e a quelle che vedremo nelle pagine che seguono vi sono le manovre per la lettura e la validazione di dati strutturati che sono richiesti nelle fasi iniziali di una grande varietà di elaborazioni delle informazioni.

Per esempio il seguente riconoscitore

consente di accettare presentazioni di digrafi nelle quali ν rappresenta, genericamente, le possibili etichette dei nodi.

C13 a.05 Nella pratica si utilizzano digrafi delle transizioni con molte decine di stati e archi; questi digrafi vengono utilizzati con numerose varianti che consentono di semplificare la redazione dei dati di ingresso delle elaborazioni in modo da poter trattare dati esprimibili con raffigurazioni di pluridigrafi arricchiti più concise, ma che per essere controllate algoritmicamente devono soddisfare regole che si aggiungono a quelle gestibili dai puri riconoscitori-RSD.

Nelle presentazioni di molti riconoscitori-RSD utilizzati come schemi operativi per le accennate applicazioni lo stato trappola viene trascurato.
Va anche detto che le presentazioni per utenti di questi riconoscitori e di altri simili meccanismi in genere omettono le etichette degli stati ritenendo che essi possono essere individuati su schemi grafici o su descrizioni che mettono in primo piano le prestazioni che possono essere richieste.

Ad esempio consideriamo il seguente pluridigrafo di transizione:

Questo può essere vantaggiosamente semplificato nel seguente:

Ed ancora più concisamente:

Come illustrato nell'esempio, talora non è necessario esplicitare le direzioni degli archi: nel caso di riconoscitori di linguaggi finiti, e quindi aciclici, bastano le indicazioni degli stati iniziale e finali.

C13 a.06 Un altro esempio chiarificatore è il riconoscitore che accetta il linguaggio delle 16 orientazioni costituenti la cosiddetta "rosa dei venti":

$$\{\ \texttt{E,ENE,NE,NNE,N,NNW,NW,WNW,W,WSW,SW,SSW,S,SSE,SE,ESE}\ \}\ .$$

La sua semplificazione è raffigurabile da:

C13 a.07 Per descrivere linguaggi nei quali intervengono sottostringhe fisse, come per le frasi dichiarative dei linguaggi procedurali nelle quali si incontrano parole riservate come `int` e `char`, è naturale ridurre passeggiate senza deviazioni a semplici archi etichettati da intere sottostringhe.

In molti casi si devono accettare sottostringhe specifiche che devono appartenere a liste ben definite costituite da molti items: si pensi ad esempio alle denominazioni dei comuni italiani, ai loro codici di 4

caratteri, alle componenti che si incontrano in numerose applicazioni della Pubblica Amministrazione o dei circuiti bancari.
La gestione di questi elementi conviene che sia gestita da routines che fanno riferimento a lunghi elenchi di origine empirica che in molti casi sono variabili nel tempo e quindi richiedono che siano curati con precione i loro aggiornamenti.

Molte varianti dei riconoscitori-RSD sono utilizzate per gestire insiemi di stringhe che devono appartenere ad elenchi di milioni di items e che consistono in una successione di porzioni di determinate lunghezze e che sono costituite da stringhe che devono appartenere ad elenchi specifici di centinaia o migliaia di items.

Si pensi ad esempio ai codici fiscali italiani e ai codici bancari IBAN.

C13 a.08 Questo accade, in particolare, quando le strutture di dati che individuano strumenti governati come gli stessi riconoscitori-RSND, le espressioni razionali e le grammatiche e le macchine che vedremo in C14, C16, C21, C22 e C23, oltre ai sistemi di Lindenmayer [C26] vengono presentate a programmi per il computer incaricati di simulare le loro evoluzioni o a procedure con il compito di codificarle per poterle trattare "nella loro totalità.

Questo accade anche per procedure incaricate di gestire le informazioni che devono essere digitate nelle caselle da compilare interattivamente per ottenere la registrazione a qualche servizio da fruire in linea.

In questi casi può essere utile fare riferimento a un riconoscitore arricchito che gestisce le possibili sequenze di manovre ciascuna corrispondente con la digitazione in una casella. I movimenti da una casella all'altra vengono gestiti da un riconoscitore, mentre ciascuna manovra di digitazione viene controllata richiamando una routine che si serve di una lista specifica, spesso massiccia: lista dei CAP, dei codici fiscali, dei codici bancari, dei codici ISBN, dei codici di una grande varietà di prodotti che entrano nei circuiti commerciali di grandi aziende, soprattutto se multinazionali, ...

C13 a.09 (1) Eserc. Tracciare pluridigrafi di transizione riguardanti le presentazioni formali di un generico riconoscitore di Rabin-Scott deterministico e di un trasduttore (v. paragrafo successivo).

(2) Eserc. Tracciare pluridigrafi di transizione riguardanti le presentazioni formali di una generica grammatica acontestuale e di una più generale grammatica a struttura di frase ((v. capitoli successivi).

C13 b. accettatori-trasduttori-R

C13 b.01 Diciamo **accettatore-trasduttore** o **macchina di Mealy** una macchina formale ottenuta arricchendo un riconoscitore di Rabin-Scott deterministico con un **alfabeto di output** (che può essere diverso da quello di input) e con una **funzione di output**, funzione che in corrispondenza a ogni stato interno e a ogni simbolo di input fornisce una stringa sull'alfabeto di output destinata a contribuire ai risultati delle evoluzioni.

L'evoluzione di un accettatore-trasduttore è un arricchimento del processo evolutivo di un riconoscitore-RSD: ora la lettura di una stringa di input porta non solo ai successivi cambiamenti di stato, ma anche alla emissione su un dispositivo sequenziale di uscita di successive stringhe.
Queste si possono descrivere come emesse una dopo l'altra nei successivi passi dell'evoluzione su un nastro di sola scrittura che chiamiamo **nastro di output**.

Anche questo nastro può essere percorso solo da sinistra a destra da una **testina di scrittura** che all'inizio dell'evoluzione si trova sulla casella iniziale e che a ogni mossa emette una stringa e si posiziona sulla casella successiva all'ultima scritta.

Le singole azioni emissive possono consistere nell'emissione della stringa muta e nel non avanzamento della testina di scrittura; questo può accadere spesso negli accettatori-trasduttori di interesse pratico.

Alla fine di una evoluzione sul nastro di uscita, nelle posizioni dalla iniziale a quella che precede la casella su cui si trova alla fine la testina di scrittura, si trova una stringa che viene detta **stringa trasdotta** di quella che è stata letta.

È possibile, e in genere opportuno, trattare anche un accettatore trasduttore attraverso un pluridigrafo di transizione ed emissione che costituisce un arricchimento del pluridigrafo di transizione del riconoscitore-RSD sottostante.

Ora ogni arco viene etichettato non solo dal carattere che induce la transizione e che può comportare il passaggio dallo stato corrente a un nuovo stato, ma anche dalla stringa che viene conseguentemente emessa sul nastro di output.

C13 b.02 Se non ci si preoccupa dello stato finale raggiunto alla fine dell'evoluzione (ovvero se si assume che tutti gli stati della macchina sono stati finali), non si hanno distinzioni tra le stringhe lette e tra le stringhe ottenute per trasduzione. In questo caso la macchina sequenziale in esame svolge soltanto il **ruolo del trasduttore**.

Se invece si distingue tra tra stati finali e nonfinali, cioè ci si preoccupa anche di accettare o di rifiutare le stringhe che vengono presentate alla macchina, oltre che di trasformarle nelle stringhe da emettere. In questo caso si ha un accettatore-trasduttore a pieno titolo.

Con gli accettatori-trasduttori si ha il più semplice modello formale tra le macchine governate (ossia dai meccanismi per i quali si distingue una parte hardware ed una software) che compiono trasformazioni di stringhe in stringhe.

Queste trasformazioni di stringhe possono fornire prestazioni di codifica e di decodifica e possono eseguire elementari traduzioni e semplici calcoli, come vedremo più estesamente in seguito.

C13 b.03 In generale le trasduzioni effettuate da un trasduttore si possono considerare come attività volte a risolvere un problema $\mathcal{P}$ secondo Lo schema che segue.

Una stringa di input costituisce la rappresentazione di una istanza di $\mathcal{P}$.

La stringa di output corrispondente costituisce la rappresentazione della soluzione di tale istanza trovata dalla macchina trasduttrice.

Le attività di una macchina accettatrice-trasduttrice hanno una interpretazione un po' più elaborata: ora le stringhe di input rifiutate si possono considerare come formulazioni non accettabili di istanze del problema $\mathcal{P}$; queste formulazioni rifiutate sono spesso chiamate "formulazioni illegali".

C13 b.04 Definiamo **accettatore-trasduttore alla Rabin-Scott deterministico**, in breve **accettatore-trasduttore-RS** un sistema

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

dove $\langle Q, \imath, F, \mathsf{T}, \delta \rangle$ è un riconoscitore-RSD, U un alfabeto che chiamiamo alfabeto di output ed η una funzione del genere $\lceil\!\lceil\, Q \times \mathsf{T} \longmapsto \mathsf{U}^* \,\rceil\!\rceil$.

Se si trascura l'insieme degli stati finali F si ha una struttura che chiamiamo **trasduttore-RD**.

C13 b.05 Torniamo sul fatto che gli accettatori-trasduttori-RD e i trasduttori-RD sono le più semplici macchine formali in grado di fare da modelli per elaborazioni di effettivo interesse pratico.

Si può considerare una tale struttura $\boldsymbol{T}$ come un meccanismo nel quale vengono immesse stringhe $w \in \mathsf{T}^*$ le quali, in seguito alla elaborazione da parte di $\boldsymbol{T}$, possono essere rifiutate e quindi portano a enunciati della forma $(w \notin \boldsymbol{T}^{\mathcal{A}})$, oppure possono essere accettate e trasformate in nuove stringhe in U^*.
Si può considerare che le stringhe w siano i dati immessi (accettabili o meno) e le stringhe trasformate i risultati delle elaborazioni effettuate dalla T sopra dati correttamente espressi.

Più avanti presenteremo una gamma di esempi semplici ma significativi di queste macchine sequenziali. In effetti esse consentono di risolvere con facilità un certo numero di problemi riconducibili a trasformazioni di stringhe.

La facilità dipende dal fatto che, quando si è capito che la soluzione di un problema si può ricondurre all'azione di un trasduttore si devono effettuare due azioni in genere agevoli: precisare i dettagli del pluridigrafo di transizione ed emissione e implementare quest'ultimo.

C13 b.06 L'implementazione è un lavoro sostanzialmente routinario, mentre la precisazione del pluridigrafo è una attività meno prevedibile, soprattutto quando non sono completamente chiari l'insieme delle stringhe accttabili che devono essere trasdotte e le modalità delle trasduzioni.

Serve spesso una attività di precisazione del problema stesso e per questa può essere molto vantaggioso un atteggiamento empirico di chiarimento progressivo.
Si tratta di procedere a una sequenza di fasi di esame di successive versioni dell'accettatore-trasduttore con il sostegno delle rispettive raffigurazioni.
A ciascuna versione vengono proposte possibili stringhe da accettare e trasformare; quando si incontrano stringhe di input rifiutate si possono avere indicazioni per il confine tra stringhe legali e stringhe illegali; quando si ottengono stringhe emesse insoddisfacenti si possono ricavare suggerimenti sopra il confine suddetto e sopra le manovre di trasduzione.
Le successive fasi dovrebbero consentire la messa a punto graduale dell'accettatore-trasduttore fino a giungere ad una versione giudicata soddisfacente.

Lo sviluppo della applicazione spesso conviene effettuarlo alternando la precisazione del pluridigrafo e la sua rapida implementazione seguita da una verifica al computer sopra stringhe di input che negli stadi iniziali possono essere semplici e in seguito possono essere scelte sempre più impegnative e pertinenti nei confronti dei risultati di interesse concreto.

In molti casi si riesce a procedere attraverso aggiustamenti effettuati per gradi, senza dover ricorrere a rifacimenti profondi, interventi che possono essere interpretati come cambiamenti di strategia.

C13 b.07 Purtroppo gli accettatori-trasduttori-R hanno una gamma piuttosto circoscritta di applicazioni e si individuano facilmente problemi di evidente importanza applicativa riconducibili ad accettazioni e trasformazioni di stringhe che gli strumenti riconducibili a Atrdn non consentono di affrontare.

Per risolvere questi problemi chiaramente al di sopra di quello che abbiamo chiamato livello-R, è necessario servirsi di macchine formali dotate di "dispositivi" decisamente più versatili di quelli visti finora.

Un primo arricchimento di grande interesse per versatilità e possibile efficienza lo presentano le macchine dotate di deposito a pila che introdurremo nel cap. C14 e che attribuiano ad un cosiddetto livello-F.
Queste consentono di affrontare problemi di grande interesse come quelli collegati alla gestione di ampie porzioni dei linguaggi di programmazione

Anche i limiti di questo livello di problemi sono individuabili abbastanza agevolmente e per affrontare sistematicamente molti altri problemi si rendono necessari strumenti computazionali via via più prestanti.

Dando peso alle considerazioni generali sono stati introdotti due successivi livelli di automatismi e di linguaggi ai quali nelle presenti pagine associamo, risp., le specificazioni -S e -T.

Entrambi i tipi di automatismi e linguaggi presentano caratteristiche esprimibili piuttosto semplicemente ed hanno ampia portata. In particolare quelli del livello -T hanno una portata che buone ragioni consentono di ipotizzare che sia la più ampia concepibile.

Dando invece peso alle moltissime applicazioni che sono state effettivamente affrontate con gli strumenti informatici, è stata introdotta un grande varietà di automatismi e di linguaggi per i quali in una prima fase di sviluppo ci si è preoccupati quasi esclusivamente di adeguarsi alla esigenze applicative specifiche. Solo successivamente è stata esaminata con maggiore astrazione la tipologia delle loro caratteristiche al fine di collocarle in uno scenario più generale e lungimirante delle attività sistematiche di computazione.

C13 b.08 Tornando alle applicazioni degli strumenti del livello-R, una casistica ben comprensibile riguarda la normalizzazione dei testi tendenzialmente discorsivi che spesso vengono digitati senza curare meticolosamente la separazione delle parole, la punteggiatura, la composizione delle linee di stampa e neppure la ortografia.

Per migliorare questi testi possono essere utili vari accettatori-trasduttori di diversi livelli di complessità.

Un trasduttore per l'eliminazione dei blanks superflui da un tale testo che potrebbe essere stato digitato frettolosamente, cioè dei blanks iniziali, ripetuti e finali è il seguente.

Un trasduttore-accettatore che elimina da periodi discorsivi digitati affrettatamnte i blanks superflui e che sistema i blanks che precedono o seguono i segni ",", ";" e "." è:

C13 b.09 Passando ad applicazioni diverse è interessante esaminare il seguente trasduttore che consente di effettuare la somma di due interi naturali dati in forma binaria: si suppone che sul nastro di input

sono registrate le coppie costituite da cifre binarie e blanks che si incontrano scorrendo da destra a sinistra le scritture degli addendi.

C13 b.10 Un altro esempio interessante è costituito dalla schematizzazione di un set di una partita a tennis nella quale si incontrano due giocatori che chiamiamo A e B.
Sul pluridigrafo compaiono i caratteri di input a e b che rappresentano i punti ottenuti, risp., da A e B.

C13 b.11 **(1) Eserc.** Tracciare il pluridigrafo di un trasduttore per la moltiplicazione di due interi positivi in scrittura binaria.

(2) Eserc. Tracciare il pluridigrafo di una intera partita a tennis che possa riguardare fino a 3 sets o fino a 5 sets.

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