Capitolo 5

Grammatiche e linguaggi liberi dal contesto

Volgiamo ora l'attenzione a una classe più ampia di linguaggi rispetto a quelli regolari: i linguaggi "liberi dal contesto". Si tratta di linguaggi che hanno una notazione naturale ricorsiva, chiamata "grammatiche libere dal contesto". Le grammatiche libere dal contesto sono state cruciali nei compilatori sin dagli anni '60: grazie a loro la realizzazione di un *parser* (una funzione che estrae la struttura di un programma) è passata da un'attività di implementazione *ad hoc* che richiedeva molto tempo a un lavoro di routine che può essere svolto in un solo pomeriggio. Di recente le grammatiche libere dal contesto sono state usate per descrivere formati di documenti attraverso le cosiddette DTD (*Document Type Definition*, definizione di tipo di documento), utilizzate dagli utenti di XML (*eXtensible Markup Language*) per lo scambio di informazioni nel Web.

In questo capitolo introduciamo la notazione delle grammatiche libere dal contesto e mostriamo come definiscono i linguaggi. Presentiamo inoltre l'"albero sintattico" (*parse tree*), una rappresentazione grafica della struttura che una grammatica impone alle stringhe del suo linguaggio. L'albero sintattico è prodotto dal *parser* di un linguaggio di programmazione e costituisce il modo tipico di rappresentare la struttura dei programmi.

I linguaggi liberi dal contesto sono descritti compiutamente anche da una notazione in forma di automa, l'automa "a pila" (*pushdown automaton*). L'automa a pila sarà materia del Capitolo 6. Sebbene meno importanti degli automi a stati finiti, gli automi a pila, soprattutto perché sono equivalenti alle grammatiche libere dal contesto come modo di definire linguaggi, si rivelano molto utili nell'esplorazione delle proprietà di chiusura e di decisione dei linguaggi liberi dal contesto (vedi Capitolo 7).

5.1 Grammatiche libere dal contesto

Cominciamo col presentare informalmente la notazione delle grammatiche libere dal contesto. Passeremo alle definizioni formali dopo averne esaminato alcune importanti caratteristiche. Mostreremo come si definisce una grammatica in termini formali e presenteremo il processo di "derivazione", che determina quali stringhe appartengono al linguaggio di una grammatica.

5.1.1 Un esempio informale

Consideriamo il linguaggio delle palindrome. Una stringa è *palindroma* se si può leggere indifferentemente da sinistra a destra e da destra a sinistra, come per esempio `otto` o `madamimadam` ("Madam, I'm Adam", la prima frase che si suppone Eva abbia udito nel Giardino dell'Eden). In altri termini una stringa w è palindroma se e solo se $w = w^R$. Per semplificare le cose descriveremo solo le stringhe palindrome sull'alfabeto $\{0, 1\}$; questo linguaggio comprende stringhe come 0110, 11011 e ϵ, ma non 011 o 0101.

È facile verificare che il linguaggio L_{pal} delle palindrome di 0 e 1 non è un linguaggio regolare. Per farlo ci serviamo del *pumping lemma*. Se L_{pal} è regolare, sia n la costante associata, e consideriamo la stringa palindroma $w = 0^n 1 0^n$. Per il lemma possiamo scomporre w in $w = xyz$, in modo tale che y consista di uno o più 0 dal primo gruppo. Di conseguenza xz, che dovrebbe trovarsi in L_{pal} se L_{pal} fosse regolare, avrebbe meno 0 a sinistra dell'unico 1 rispetto a quelli a destra. Dunque xz non può essere palindroma. Abbiamo così confutato l'ipotesi che L_{pal} sia un linguaggio regolare.

Per stabilire in quali casi una stringa di 0 e 1 si trova in L_{pal}, possiamo avvalerci di una semplice definizione ricorsiva. La base della definizione stabilisce che alcune stringhe semplici si trovano in L_{pal}; si sfrutta poi il fatto che se una stringa è palindroma deve cominciare e finire con lo stesso simbolo. Inoltre, quando il primo e l'ultimo simbolo vengono rimossi, la stringa risultante dev'essere palindroma.

BASE ϵ, 0 e 1 sono palindrome.

INDUZIONE Se w è palindroma, lo sono anche $0w0$ e $1w1$. Nessuna stringa di 0 e 1 è palindroma, salvo che non risulti dalla base e dalla regola di induzione appena esposte.

Una grammatica libera dal contesto è una notazione formale per esprimere simili definizioni ricorsive di linguaggi. Una grammatica consiste di una o più variabili che rappresentano classi di stringhe, ossia linguaggi. Nell'esempio ci occorre una sola variabile, P, che rappresenta l'insieme delle palindrome, cioè la classe delle stringhe che formano il linguaggio L_{pal}. Opportune regole stabiliscono come costruire le stringhe in ogni classe. La costruzione può impiegare simboli dell'alfabeto, stringhe di cui si conosce già l'appartenenza a una delle classi, oppure entrambi gli elementi.

Esempio 5.1 Le regole che definiscono le palindrome, espresse nella notazione delle grammatiche libere dal contesto, sono riportate nella Figura 5.1. Il loro significato verrà chiarito nel Paragrafo 5.1.2.

1. $P \rightarrow \epsilon$
2. $P \rightarrow 0$
3. $P \rightarrow 1$
4. $P \rightarrow 0P0$
5. $P \rightarrow 1P1$

Figura 5.1 Una grammatica libera dal contesto per le stringhe palindrome.

La prime tre regole costituiscono la base e indicano che la classe delle palindrome include le stringhe ϵ, 0 e 1. Nessuno dei membri destri delle regole (le porzioni che seguono le frecce) contiene una variabile; questo è il motivo per cui formano la base della definizione.

Le ultime due regole costituiscono la parte induttiva. Per esempio la regola 4 dice che, se si prende una qualunque stringa w dalla classe P, anche $0w0$ si trova nella classe P. Analogamente la regola 5 indica che anche $1w1$ è in P. □

5.1.2 Definizione delle grammatiche libere dal contesto

La descrizione grammaticale di un linguaggio consiste di quattro componenti importanti.

1. Un insieme finito di simboli che formano le stringhe del linguaggio da definire. Nell'esempio delle palindrome l'insieme è $\{0, 1\}$. Chiameremo quest'alfabeto i *terminali* o *simboli terminali*.

2. Un insieme finito di *variabili*, talvolta dette anche *non terminali* oppure *categorie sintattiche*. Ogni variabile rappresenta un linguaggio, ossia un insieme di stringhe. Nell'esempio precedente c'è una sola variabile, P, usata per rappresentare la classe delle stringhe palindrome sull'alfabeto $\{0, 1\}$.

3. Una variabile, detta *simbolo iniziale*, che rappresenta il linguaggio da definire. Le altre variabili rappresentano classi ausiliarie di stringhe, che contribuiscono a definire il linguaggio del simbolo iniziale. Nell'esempio, P, l'unica variabile, è il simbolo iniziale.

4. Un insieme finito di *produzioni*, o *regole*, che rappresentano la definizione ricorsiva di un linguaggio. Ogni produzione consiste di tre parti.

   (a) Una variabile che viene definita (parzialmente) dalla produzione ed è spesso detta la *testa* della produzione.

(b) Il simbolo di produzione $\rightarrow$.

(c) Una stringa di zero o più terminali e variabili, detta il *corpo* della produzione, che rappresenta un modo di formare stringhe nel linguaggio della variabile di testa. Le stringhe si formano lasciando immutati i terminali e sostituendo ogni variabile del corpo con una stringa appartenente al linguaggio della variabile stessa.

Esempi di produzioni sono illustrati nella Figura 5.1.

I quattro componenti appena descritti formano una *grammatica libera dal contesto*, o semplicemente *grammatica*, o *CFG*, (*Context-Free Grammar*). Rappresenteremo una CFG per mezzo dei suoi quattro componenti, ossia $G = (V, T, P, S)$, dove V è l'insieme delle variabili, T i terminali, P l'insieme delle produzioni ed S il simbolo iniziale.

Esempio 5.2 La grammatica G_{pal} per le palindrome è rappresentata da

$$G_{pal} = (\{P\}.\{0,1\}, A, P)$$

dove A rappresenta l'insieme delle cinque produzioni nella Figura 5.1. □

Esempio 5.3 Esaminiamo una CFG più complessa, che rappresenta, semplificandole, le espressioni in un tipico linguaggio di programmazione. Dapprima ci limitiamo agli operatori + e $*$, corrispondenti a somma e moltiplicazione. Ammettiamo che gli operandi siano identificatori, ma al posto dell'insieme completo di identificatori tipici (una lettera seguita da zero o più lettere e cifre) accettiamo solo le lettere a e b e le cifre 0 e 1. Ogni identificatore deve iniziare per a o b e può continuare con una qualunque stringa in $\{a, b, 0, 1\}^*$.

In questa grammatica sono necessarie due variabili. La prima, che chiameremo E, rappresenta le espressioni. È il simbolo iniziale e rappresenta il linguaggio delle espressioni che stiamo definendo. L'altra variabile, I, rappresenta gli identificatori. Il suo linguaggio è regolare; è il linguaggio dell'espressione regolare

$$\mathbf{(a+b)(a+b+0+1)^*}$$

Eviteremo di usare direttamente le espressioni regolari nelle grammatiche, preferendo piuttosto un insieme di produzioni che abbiano lo stesso significato dell'espressione in esame.

La grammatica per le espressioni è definita formalmente da $G = (\{E, I\}, T, P, E)$, dove T è l'insieme di simboli $\{+, *, (,), a, b, 0, 1\}$ e P è l'insieme di produzioni nella Figura 5.2. L'interpretazione delle produzioni è la seguente.

La regola (1) è la regola di base per le espressioni, e afferma che un'espressione può essere un singolo identificatore. Le regole dalla (2) alla (4) descrivono il caso induttivo

1.	E	$\rightarrow$	I
2.	E	$\rightarrow$	$E + E$
3.	E	$\rightarrow$	$E * E$
4.	E	$\rightarrow$	(E)
5.	I	$\rightarrow$	a
6.	I	$\rightarrow$	b
7.	I	$\rightarrow$	Ia
8.	I	$\rightarrow$	Ib
9.	I	$\rightarrow$	$I0$
10.	I	$\rightarrow$	$I1$

Figura 5.2 Una grammatica libera dal contesto per espressioni semplici.

per le espressioni. La regola (2) afferma che un'espressione può essere formata da due espressioni connesse dal segno +; la regola (3) dice la stessa cosa per il segno di moltiplicazione. La regola (4) dichiara che, se si prende una qualunque espressione e la si racchiude fra parentesi, il risultato è ancora un'espressione.

Le regole dalla (5) alla (10) descrivono gli identificatori. La base è data dalle regole (5) e (6), secondo le quali a e b sono identificatori. Le rimanenti quattro regole sono il caso induttivo, e affermano che se abbiamo un identificatore possiamo farlo seguire da a, b, 0 oppure 1, e il risultato è comunque un altro identificatore. □

5.1.3 Derivazioni per mezzo di una grammatica

Le produzioni di una CFG si applicano per dedurre che determinate stringhe appartengono al linguaggio di una certa variabile. La deduzione può seguire due strade. Quella più convenzionale si serve delle regole utilizzando il corpo per passare alla testa. In altre parole prendiamo stringhe di cui conosciamo l'appartenenza al linguaggio di ognuna delle variabili del corpo, le concateniamo nell'ordine adeguato, con i terminali che compaiono nel corpo, e deduciamo che la stringa risultante è nel linguaggio della variabile che compare in testa. Chiameremo questa procedura *inferenza ricorsiva*.

Il secondo modo per definire il linguaggio di una grammatica applica le produzioni dalla testa al corpo. Espandiamo il simbolo iniziale usando una delle sue produzioni (cioè usando una produzione la cui testa sia il simbolo iniziale). Espandiamo ulteriormente la stringa risultante sostituendo una delle variabili con il corpo di una delle sue produzioni, e così via, fino a derivarne una stringa fatta interamente di terminali. Il linguaggio della grammatica è l'insieme di tutte le stringhe di terminali ottenute con questa procedura. Questa tecnica è detta *derivazione*.

Notazione compatta per le produzioni

È opportuno pensare a una produzione come "appartenente" alla variabile della sua testa. Useremo spesso termini come "le produzioni per A" oppure "le A-produzioni" per riferirci alle produzioni la cui testa è la variabile A. Possiamo scrivere le produzioni di una grammatica elencando ogni variabile una volta, e facendola seguire dai corpi delle sue produzioni, separati da barre verticali. In altre parole le produzioni $A \to \alpha_1$, $A \to \alpha_2, \ldots, A \to \alpha_n$ possono essere sostituite dalla notazione $A \to \alpha_1 | \alpha_2 | \cdots | \alpha_n$. Per esempio la grammatica per le palindrome della Figura 5.1 può essere scritta come $P \to \epsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$.

Cominceremo da un esempio di inferenza ricorsiva. Poiché spesso è più naturale considerare una grammatica secondo lo schema della derivazione, come passo successivo svilupperemo la notazione per descrivere le derivazioni.

Esempio 5.4 Consideriamo alcune inferenze possibili nella grammatica per le espressioni della Figura 5.2. La Figura 5.3 riassume queste inferenze. Per esempio la riga (i) dice che possiamo dedurre che la stringa a è nel linguaggio per I usando la produzione 5. Le righe dalla (ii) alla (iv) affermano che possiamo dedurre che $b00$ è un identificatore usando una volta la produzione 6 (per ottenere la b) e poi applicando due volte la produzione 9 (per accodare i due 0).

	Stringa ricavata	Per il linguaggio di	Produzione impiegata	Stringhe impiegate
(i)	a	I	5	—
(ii)	b	I	6	—
(iii)	$b0$	I	9	(ii)
(iv)	$b00$	I	9	(iii)
(v)	a	E	1	(i)
(vi)	$b00$	E	1	(iv)
(vii)	$a + b00$	E	2	$(v), (vi)$
$(viii)$	$(a + b00)$	E	4	(vii)
(ix)	$a * (a + b00)$	E	3	$(v), (viii)$

Figura 5.3 Inferenza di stringhe dalla grammatica della Figura 5.2.

Ogni identificatore è un'espressione; le righe (v) e (vi) sfruttano quindi la produzione 1 per dedurre che anche le stringhe a e $b00$, che risultano essere identificatori per inferenza nelle righe (i) e (iv), sono nel linguaggio della variabile E. La riga (vii) usa la produzione 2 per inferire che la somma di questi identificatori è un'espressione; la riga $(viii)$ usa la produzione 4 per inferire che la stessa stringa, posta tra parentesi, è un'espressione, e la riga (ix) usa la produzione 3 per moltiplicare l'identificatore a per l'espressione che abbiamo trovato nella riga $(viii)$. □

Il processo di derivazione di stringhe per applicazione di produzioni dalla testa al corpo richiede la definizione di un nuovo simbolo di relazione, $\Rightarrow$. Supponiamo che $G = (V, T, P, S)$ sia una CFG. Sia $\alpha A \beta$ una stringa di terminali e variabili, dove A è una variabile. In altre parole α e β sono stringhe in $(V \cup T)^*$, e A è in V. Sia $A \to \gamma$ una produzione di G. Allora scriviamo $\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$. Se G risulta chiara dal contesto, scriviamo semplicemente $\alpha A \beta \Rightarrow \alpha \gamma \beta$. Si noti che un passo di derivazione sostituisce una variabile in un punto qualsiasi della stringa con il corpo di una delle sue produzioni.

Possiamo estendere la relazione $\Rightarrow$ fino a farle rappresentare zero, uno o più passi di derivazione, analogamente a come la funzione di transizione δ di un automa a stati finiti si estende a $\hat{\delta}$. Per le derivazioni usiamo il simbolo * per denotare "zero o più passi", come segue.

BASE Per qualsiasi stringa α di terminali e variabili, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$. In altri termini: qualunque stringa deriva se stessa.

INDUZIONE Se $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ e $\beta \underset{G}{\Rightarrow} \gamma$, allora $\alpha \overset{*}{\underset{G}{\Rightarrow}} \gamma$. Ossia, se α può diventare β in zero o più passi, e un passo ulteriore trasforma β in γ, allora α può diventare γ. Detto in altri termini, $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$ significa che esiste una sequenza di stringhe $\gamma_1, \gamma_2, \ldots, \gamma_n$, con $n \geq 1$, tale che

1. $\alpha = \gamma_1$
2. $\beta = \gamma_n$
3. per $i = 1, 2, \ldots, n-1$, abbiamo $\gamma_i \Rightarrow \gamma_{i+1}$.

Se la grammatica G è chiara dal contesto, allora scriviamo $\overset{*}{\Rightarrow}$ anziché $\overset{*}{\underset{G}{\Rightarrow}}$.

Esempio 5.5 L'inferenza che $a * (a + b00)$ appartiene al linguaggio della variabile E si riflette in una derivazione della stringa, a partire dalla stringa E. Eccone un esempio.

$$E \Rightarrow E * E \Rightarrow I * E \Rightarrow a * E \Rightarrow$$

$$a * (E) \Rightarrow a * (E + E) \Rightarrow a * (I + E) \Rightarrow a * (a + E) \Rightarrow$$

$$a * (a + I) \Rightarrow a * (a + I0) \Rightarrow a * (a + I00) \Rightarrow a * (a + b00)$$

Al primo passo E viene sostituita dal corpo della produzione 3 (dalla Figura 5.2). Al secondo passo si usa la produzione 1 per rimpiazzare la prima E con I, e così via. Si noti che abbiamo sistematicamente sostituito la variabile più a sinistra nella stringa. A ogni passo è comunque possibile scegliere quale variabile sostituire, e al posto di questa usare qualunque produzione. Per esempio al secondo passo avremmo potuto sostituire la seconda E con (E) servendoci della produzione 4. In tal caso avremmo avuto $E * E \Rightarrow E * (E)$. Potevamo anche scegliere una sostituzione incapace di condurre alla stessa stringa di terminali, per esempio la produzione 2 al primo passo: $E \Rightarrow E + E$. Infatti nessuna sostituzione delle due E può trasformare $E + E$ in $a * (a + b00)$.

Possiamo usare la relazione $\overset{*}{\Rightarrow}$ per abbreviare la derivazione. Dalla base sappiamo che $E \overset{*}{\Rightarrow} E$. L'uso reiterato della parte induttiva fornisce $E \overset{*}{\Rightarrow} E * E$, $E \overset{*}{\Rightarrow} I * E$, e così via, finché si ottiene $E \overset{*}{\Rightarrow} a * (a + b00)$.

I due punti di vista, inferenza ricorsiva e derivazione, sono equivalenti. In altre parole si deduce che una stringa di terminali w si trova nel linguaggio di una certa variabile A se e solo se $A \overset{*}{\Rightarrow} w$. La dimostrazione è però laboriosa, e la rimandiamo al Paragrafo 5.2. □

5.1.4 Derivazioni a sinistra e a destra

Per ridurre il numero di scelte possibili nella derivazione di una stringa, spesso è comodo imporre che a ogni passo si sostituisca la variabile all'estrema sinistra con il corpo di una delle sue produzioni. Tale derivazione viene detta *derivazione a sinistra* (*leftmost derivation*), e si indica tramite le relazioni $\underset{lm}{\Rightarrow}$ e $\overset{*}{\underset{lm}{\Rightarrow}}$, rispettivamente per uno o molti passi. Se la grammatica G in esame non è chiara dal contesto, possiamo porre il nome G sotto la freccia.

Analogamente è possibile imporre che a ogni passo venga sostituita la variabile più a destra con uno dei suoi corpi. In tal caso chiamiamo la derivazione *a destra* (*rightmost*), e usiamo i simboli $\underset{rm}{\Rightarrow}$ e $\overset{*}{\underset{rm}{\Rightarrow}}$ per indicare rispettivamente uno o più passi di derivazione a destra. Qualora non fosse evidente, il nome della grammatica può comparire anche in questo caso sotto i simboli.

Esempio 5.6 La derivazione dell'Esempio 5.5 è una derivazione a sinistra. Possiamo dunque descriverla come segue.

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E \underset{lm}{\Rightarrow}$$

$$a * (E) \underset{lm}{\Rightarrow} a * (E + E) \underset{lm}{\Rightarrow} a * (I + E) \underset{lm}{\Rightarrow} a * (a + E) \underset{lm}{\Rightarrow}$$

$$a * (a + I) \underset{lm}{\Rightarrow} a * (a + I0) \underset{lm}{\Rightarrow} a * (a + I00) \underset{lm}{\Rightarrow} a * (a + b00)$$

Notazione per le derivazioni delle CFG

Per aiutarci a ricordare il ruolo dei simboli usati nel trattare le CFG, si usano comunemente alcune convenzioni, che elenchiamo.

1. Le lettere minuscole in prossimità dell'inizio dell'alfabeto, a, b, e così via, sono simboli terminali. Assumiamo inoltre che cifre o altri caratteri, come + o le parentesi, sono terminali.
2. Le lettere maiuscole in prossimità dell'inizio dell'alfabeto, A, B, e così via, sono variabili.
3. Le lettere minuscole in prossimità della fine dell'alfabeto, come w o z, sono stringhe di terminali. Questa convenzione ci ricorda che i terminali sono analoghi ai simboli di input di un automa.
4. Le lettere maiuscole in prossimità della fine dell'alfabeto, come X o Y, sono terminali oppure variabili.
5. Le lettere minuscole greche, come α e β, sono stringhe che consistono di terminali e variabili.

Dato che non si tratta di un fattore importante, non esiste alcuna notazione specifica per le stringhe che consistono solo di variabili. Può accadere che una stringa denominata α, o con un'altra lettera greca, contenga solo variabili.

Possiamo inoltre riassumere la derivazione a sinistra scrivendo $E \overset{*}{\underset{lm}{\Rightarrow}} a*(a+b00)$, oppure esprimerne alcuni passi tramite espressioni come $E * E \overset{*}{\underset{lm}{\Rightarrow}} a * (E)$.

Esiste una derivazione a destra che applica le stesse sostituzioni per ogni variabile, sebbene in ordine diverso. Eccola.

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E) \underset{rm}{\Rightarrow}$$

$$E * (E + I) \underset{rm}{\Rightarrow} E * (E + I0) \underset{rm}{\Rightarrow} E * (E + I00) \underset{rm}{\Rightarrow} E * (E + b00) \underset{rm}{\Rightarrow}$$

$$E * (I + b00) \underset{rm}{\Rightarrow} E * (a + b00) \underset{rm}{\Rightarrow} I * (a + b00) \underset{rm}{\Rightarrow} a * (a + b00)$$

Questa derivazione permette di concludere $E \overset{*}{\underset{rm}{\Rightarrow}} a * (a + b00)$. □

Qualunque derivazione ha una derivazione a sinistra e una a destra equivalenti. In altre parole, se w è una stringa terminale e A una variabile, allora $A \stackrel{*}{\Rightarrow} w$ se e solo se $A \underset{lm}{\stackrel{*}{\Rightarrow}} w$, e $A \stackrel{*}{\Rightarrow} w$ se e solo se $A \underset{rm}{\stackrel{*}{\Rightarrow}} w$. Dimostreremo anche queste affermazioni nel Paragrafo 5.2.

5.1.5 Il linguaggio di una grammatica

Se $G = (V, T, P, S)$ è una CFG, il *linguaggio* di G, denotato con $L(G)$, è l'insieme delle stringhe terminali che hanno una derivazione dal simbolo iniziale. In altri termini

$$L(G) = \{w \text{ in } T^* \mid S \underset{G}{\stackrel{*}{\Rightarrow}} w\}$$

Se un linguaggio L è il linguaggio di una grammatica libera dal contesto, allora L è detto *linguaggio libero dal contesto*, o CFL (*Context-Free Language*). Per esempio abbiamo affermato che la grammatica della Figura 5.1 definisce il linguaggio delle palindrome sull'alfabeto $\{0, 1\}$. Di conseguenza l'insieme delle palindrome è un linguaggio libero dal contesto. Dimostriamo l'enunciato.

Teorema 5.7 $L(G_{pal})$, dove G_{pal} è la grammatica dell'Esempio 5.1, è l'insieme delle palindrome su $\{0, 1\}$.

DIMOSTRAZIONE Dimostreremo che una stringa w in $\{0, 1\}^*$ è in $L(G_{pal})$ se e solo se è palindroma, ossia $w = w^R$.

(Se) Supponiamo che w sia palindroma. Mostriamo per induzione su $|w|$ che w è in $L(G_{pal})$.

BASE Usiamo le lunghezze 0 e 1 come base. Se $|w| = 0$ o $|w| = 1$, allora w è ϵ, 0, o 1. Dato che esistono le produzioni $P \to \epsilon$, $P \to 0$ e $P \to 1$, concludiamo che $P \stackrel{*}{\Rightarrow} w$ in tutti i casi di base.

INDUZIONE Supponiamo che $|w| \geq 2$. Poiché $w = w^R$, w deve iniziare e finire con lo stesso simbolo. In altri termini $w = 0x0$ oppure $w = 1x1$. Oltre a ciò x deve essere palindroma, ossia $x = x^R$. Si noti che abbiamo bisogno che $|w| \geq 2$ per concludere che esistono due simboli distinti agli estremi di w.

Se $w = 0x0$ invochiamo l'ipotesi induttiva per sostenere che $P \stackrel{*}{\Rightarrow} x$. Allora esiste una derivazione di w da P, cioè $P \Rightarrow 0P0 \stackrel{*}{\Rightarrow} 0x0 = w$. Se $w = 1x1$ il ragionamento è lo stesso, ma usiamo la produzione $P \to 1P1$ al primo passo. In entrambi i casi concludiamo che w è in $L(G_{pal})$. La dimostrazione è completa.

(Solo se) Ora assumiamo che w sia in $L(G_{pal})$; cioè $P \stackrel{*}{\Rightarrow} w$. Dobbiamo concludere che w è palindroma. La dimostrazione è un'induzione sul numero dei passi in una derivazione di w da P.

BASE Se la derivazione è un solo passo, allora dobbiamo usare una delle tre produzioni che non abbiano P nel loro corpo. In altre parole la derivazione è $P \Rightarrow \epsilon$, $P \Rightarrow 0$ o $P \Rightarrow 1$. Dato che ϵ, 0 e 1 sono tutti palindromi, la base è dimostrata.

INDUZIONE Supponiamo ora che la derivazione compia $n + 1$ passi, dove $n \geq 1$, e l'enunciato sia vero per tutte le derivazioni di n passi. Ossia, se $P \stackrel{*}{\Rightarrow} x$ in n passi, allora x è un palindromo.

Consideriamo una derivazione di $(n + 1)$ passi, che deve essere della forma

$$P \Rightarrow 0P0 \stackrel{*}{\Rightarrow} 0x0 = w$$

oppure $P \Rightarrow 1P1 \stackrel{*}{\Rightarrow} 1x1 = w$, dato che $n + 1$ passi significa almeno due passi e le produzioni $P \to 0P0$ e $P \to 1P1$ sono le uniche che permettono passi aggiuntivi. Si osservi che in entrambi i casi $P \stackrel{*}{\Rightarrow} x$ in n passi.

Per l'ipotesi induttiva sappiamo che x è palindromo, cioè $x = x^R$. Ma se è così, allora anche $0x0$ e $1x1$ sono palindromi. Per esempio $(0x0)^R = 0x^R0 = 0x0$. Concludiamo che w è palindromo e completiamo così la dimostrazione. □

5.1.6 Forme sentenziali

Le derivazioni dal simbolo iniziale producono stringhe che hanno un ruolo speciale e che chiameremo "forme sentenziali". Ossia, se $G = (V, T, P, S)$ è una CFG, allora qualunque stringa α in $(V \cup T)^*$ tale che $S \stackrel{*}{\Rightarrow} \alpha$ è una *forma sentenziale*. Se $S \underset{lm}{\stackrel{*}{\Rightarrow}} \alpha$, allora α è una *forma sentenziale sinistra*; se $S \underset{rm}{\stackrel{*}{\Rightarrow}} \alpha$, allora α è una *forma sentenziale destra*. Si noti che il linguaggio $L(G)$ è formato dalle forme sentenziali che sono in T^*, cioè che consistono unicamente di terminali.

Esempio 5.8 Consideriamo la grammatica per le espressioni della Figura 5.2. La stringa $E * (I + E)$ è un esempio di forma sentenziale perché esiste una derivazione

$$E \Rightarrow E * E \Rightarrow E * (E) \Rightarrow E * (E + E) \Rightarrow E * (I + E)$$

Dato che all'ultimo passo viene sostituita la E centrale, questa derivazione non è né a sinistra né a destra.

Come esempio di forma sentenziale sinistra consideriamo $a * E$, con la derivazione a sinistra

$$E \underset{lm}{\Rightarrow} E * E \underset{lm}{\Rightarrow} I * E \underset{lm}{\Rightarrow} a * E$$

La derivazione

$$E \underset{rm}{\Rightarrow} E * E \underset{rm}{\Rightarrow} E * (E) \underset{rm}{\Rightarrow} E * (E + E)$$

mostra che $E * (E + E)$ è una forma sentenziale destra. □

La forma delle dimostrazioni sulle grammatiche

Il Teorema 5.7 è un tipico caso di dimostrazione che una grammatica definisce un particolare linguaggio, descritto informalmente. Si parte da un'ipotesi induttiva che enuncia le proprietà di cui sono dotate le stringhe derivate da ciascuna variabile. Nell'esempio in questione c'è una sola variabile, P. Dunque abbiamo dovuto soltanto affermare che le sue stringhe sono palindrome.

Si dimostra la parte "se": se una stringa w soddisfa l'enunciato informale sulle stringhe di una delle variabili A, allora $A \stackrel{*}{\Rightarrow} w$. Nell'esempio, dato che P è il simbolo iniziale, abbiamo dichiarato "$P \stackrel{*}{\Rightarrow} w$", dicendo che w appartiene al linguaggio della grammatica. Di solito la parte "se" si dimostra per induzione sulla lunghezza di w. Se ci sono k variabili l'enunciato induttivo da dimostrare ha k parti, che devono essere dimostrate per induzione mutua.

Si deve inoltre dimostrare la parte "solo-se": se $A \stackrel{*}{\Rightarrow} w$, allora w soddisfa l'enunciato informale sulle stringhe derivate dalla variabile A. Nell'esempio, dovendo trattare solo il simbolo iniziale P, abbiamo supposto che w fosse nel linguaggio di G_{pal} come equivalente di $P \stackrel{*}{\Rightarrow} w$. La dimostrazione di questa parte si compie di solito per induzione sul numero dei passi nella derivazione. Se la grammatica contiene produzioni in cui due o più variabili compaiono nelle stringhe derivate, allora una derivazione di n passi va scomposta in più parti, con una derivazione per ognuna delle variabili. Queste derivazioni possono avere meno di n passi, e si deve dunque compiere un'induzione supponendo l'enunciato valido per tutti i valori minori o uguali a n, come discusso nel Paragrafo 1.4.2.

5.1.7 Esercizi

Esercizio 5.1.1 Ideate una grammatica libera dal contesto per ognuno dei seguenti linguaggi.

* a) L'insieme $\{0^n1^n \mid n \geq 1\}$, ossia l'insieme di tutte le stringhe di uno o più 0 seguiti da un uguale numero di 1.

*! b) L'insieme $\{a^ib^jc^k \mid i \neq j \text{ o } j \neq k\}$, ossia l'insieme delle stringhe di a seguite da un certo numero di b seguite da un certo numero di c, tali che il numero di a sia diverso dal numero di b o il numero di b sia diverso dal numero di c, o entrambi.

! c) L'insieme di tutte le stringhe di a e b che *non* sono della forma ww, cioè non sono uguali a una stringa ripetuta.

!! d) L'insieme di tutte le stringhe con un numero di 0 doppio rispetto al numero di 1.

Esercizio 5.1.2 La seguente grammatica genera il linguaggio dell'espressione regolare $\mathbf{0^*1(0+1)^*}$:

$$\begin{aligned} S &\rightarrow A1B \\ A &\rightarrow 0A \mid \epsilon \\ B &\rightarrow 0B \mid 1B \mid \epsilon \end{aligned}$$

Scrivete le derivazioni a sinistra e a destra delle seguenti stringhe:

* a) 00101

b) 1001

c) 00011.

! **Esercizio 5.1.3** Dimostrate che ogni linguaggio regolare è un linguaggio libero dal contesto. *Suggerimento*: costruite una CFG per induzione sul numero di operatori nell'espressione regolare.

! **Esercizio 5.1.4** Una CFG è detta *lineare a destra* se il corpo di ogni produzione ha al massimo una variabile, e la variabile si trova all'estremità destra. In altre parole tutte le produzioni di una grammatica lineare a destra sono della forma $A \rightarrow wB$ o $A \rightarrow w$, dove A e B sono variabili e w una stringa di zero o più terminali.

a) Dimostrate che ogni grammatica lineare a destra genera un linguaggio regolare. *Suggerimento*: costruite un ϵ-NFA che simula derivazioni a sinistra, usando il suo stato per rappresentare l'unica variabile nella forma sentenziale sinistra corrente.

b) Mostrate che ogni linguaggio regolare ha una grammatica lineare a destra. *Suggerimento*: cominciate da un DFA e fate in modo che le variabili della grammatica rappresentino gli stati.

*! **Esercizio 5.1.5** Sia $T = \{0, 1, (,), +, *, \emptyset, e\}$. Possiamo considerare T come l'insieme dei simboli usati nelle espressioni regolari sull'alfabeto $\{0, 1\}$; l'unica differenza è che si usa e al posto del simbolo ϵ per evitare confusione in ciò che segue. Il vostro compito è definire una CFG, con T come insieme di terminali, che generi esattamente le espressioni regolari con alfabeto $\{0, 1\}$.

Esercizio 5.1.6 Abbiamo definito la relazione $\stackrel{*}{\Rightarrow}$ con una base "$\alpha \Rightarrow \alpha$" e un'induzione che dice "$\alpha \stackrel{*}{\Rightarrow} \beta$ e $\beta \Rightarrow \gamma$ implicano $\alpha \stackrel{*}{\Rightarrow} \gamma$". La relazione $\stackrel{*}{\Rightarrow}$ può essere definita in altri modi che a loro volta equivalgono a dire "$\stackrel{*}{\Rightarrow}$ significa zero o più $\Rightarrow$-passi". Dimostrate i seguenti enunciati.

a) $\alpha \stackrel{*}{\Rightarrow} \beta$ se e solo se esiste una sequenza di una o più stringhe

$$\gamma_1, \gamma_2, \ldots, \gamma_n$$

tali che $\alpha = \gamma_1$, $\beta = \gamma_n$, e per $i = 1, 2, \ldots, n-1$ abbiamo $\gamma_i \Rightarrow \gamma_{i+1}$.

b) Se $\alpha \stackrel{*}{\Rightarrow} \beta$, e $\beta \stackrel{*}{\Rightarrow} \gamma$, allora $\alpha \stackrel{*}{\Rightarrow} \gamma$. *Suggerimento*: procedete per induzione sul numero dei passi nella derivazione $\beta \stackrel{*}{\Rightarrow} \gamma$.

! **Esercizio 5.1.7** Consideriamo la CFG G definita dalle produzioni:

$$S \to aS \mid Sb \mid a \mid b$$

a) Dimostrate per induzione sulla lunghezza della stringa che nessuna stringa in $L(G)$ ha ba come sottostringa.

b) Descrivete $L(G)$ in termini informali e giustificate la vostra risposta servendovi della parte (a).

!! **Esercizio 5.1.8** Considerate la CFG G definita dalle produzioni

$$S \to aSbS \mid bSaS \mid \epsilon$$

Dimostrate che $L(G)$ è l'insieme di tutte le stringhe con lo stesso numero di a e di b.

5.2 Alberi sintattici

La rappresentazione ad albero delle derivazioni si è rivelata particolarmente utile. L'albero mostra in modo chiaro come i simboli di una stringa terminale sono raccolti in sottostringhe, ciascuna appartenente al linguaggio di una delle variabili della grammatica. Forse è ancora più importante che un albero di questo tipo, detto "albero sintattico" (*parse tree*), sia la struttura dati ideale per rappresentare il programma sorgente in un compilatore. La struttura ad albero del programma sorgente facilita la traduzione del programma stesso in codice eseguibile, delegando in modo naturale il processo di traduzione a funzioni ricorsive.

In questo paragrafo presentiamo gli alberi sintattici e dimostriamo che la loro esistenza è strettamente legata alle derivazioni e alle inferenze ricorsive. Studieremo poi la questione dell'ambiguità nelle grammatiche e nei linguaggi, che costituisce un'applicazione importante degli alberi sintattici. In alcune grammatiche una stringa terminale può avere più di un albero sintattico; ciò rende una grammatica inadatta a un linguaggio di programmazione perché il compilatore non sarebbe in grado di stabilire la struttura di certi programmi, e quindi non potrebbe dedurre con sicurezza il codice eseguibile appropriato.

Terminologia relativa agli alberi

Supponiamo che il lettore conosca già la nozione di albero e che le definizioni più comuni in quest'ambito gli siano familiari. Quanto segue sarà tuttavia un utile ripasso terminologico.

- Gli alberi sono collezioni di *nodi*, con una relazione *genitore-figlio*. Un nodo ha al massimo un genitore, disegnato sopra il nodo, e zero o più figli, disegnati al di sotto. Una linea collega un genitore a ogni figlio. Le Figure 5.4, 5.5 e 5.6 sono esempi di alberi.
- Esiste un unico nodo senza genitori, posto alla sommità dell'albero: la *radice*. I nodi privi di figli sono detti *foglie*. I nodi che non sono foglie sono *nodi interni*.
- Il figlio di un figlio di un $\cdots$ di un nodo è un *discendente* del nodo. Il genitore di un genitore di un $\cdots$ è un *antenato*. Ogni nodo è discendente e antenato di se stesso.
- I figli di un nodo vengono ordinati da sinistra e disegnati di conseguenza. Se il nodo N è a sinistra del nodo M, allora si considera che tutti i discendenti di N siano alla sinistra di tutti i discendenti di M.

5.2.1 Costruzione di alberi sintattici

Fissiamo una grammatica $G = (V, T, P, S)$. Gli *alberi sintattici* di G sono alberi che soddisfano le seguenti condizioni.

1. Ciascun nodo interno è etichettato da una variabile in V.
2. Ciascuna foglia è etichettata da una variabile, da un terminale, o da ϵ. Se una foglia è etichettata ϵ, deve essere l'unico figlio del suo genitore.
3. Se un nodo interno è etichettato A e i suoi figli sono etichettati, a partire da sinistra,

$$X_1, X_2, \ldots, X_k$$

allora $A \to X_1 X_2 \cdots X_k$ è una produzione in P. Si noti che un X può essere ϵ solo nel caso in cui è l'etichetta di un figlio unico, e quindi $A \to \epsilon$ è una produzione di G.

Esempio 5.9 La Figura 5.4 presenta un albero sintattico per la grammatica delle espressioni della Figura 5.2. La radice è etichettata dalla variabile E. La produzione applicata alla radice è $E \to E + E$: i tre figli della radice hanno, a partire da sinistra, le etichette E, $+$, ed E. Per il figlio più a sinistra della radice si applica la produzione $E \to I$: il nodo ha un solo figlio, etichettato I. □

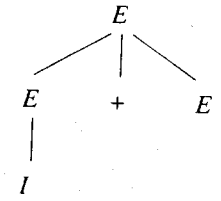

Figura 5.4 Un albero sintattico che illustra la derivazione di $I + E$ da E.

Esempio 5.10 La Figura 5.5 presenta un albero sintattico per la grammatica delle palindrome della Figura 5.1. Alla radice si applica la produzione $P \to 0P0$ e al figlio di mezzo della radice $P \to 1P1$. Al livello più basso è stata applicata la produzione $P \to \epsilon$. Questo caso, in cui il nodo etichettato dalla testa di una produzione ha un unico figlio, etichettato ϵ, è il solo in cui un nodo etichettato ϵ può fare la sua comparsa in un albero sintattico. □

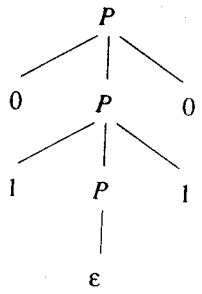

Figura 5.5 Un albero sintattico che illustra la derivazione $P \stackrel{*}{\Rightarrow} 0110$.

FINE

5.2.2 Il prodotto di un albero sintattico

Se concateniamo le foglie di un albero sintattico a partire da sinistra otteniamo una stringa, detta il *prodotto* dell'albero, che è sempre una stringa derivata dalla variabile della radice. Dimostreremo questa asserzione fra poco. Di particolare importanza sono gli alberi sintattici che soddisfano queste due condizioni.

1. Il prodotto è una stringa terminale. In questo caso tutte le foglie sono etichettate da un terminale o da ϵ.
2. La radice è etichettata dal simbolo iniziale.

Questi sono gli alberi sintattici i cui prodotti sono stringhe nel linguaggio della grammatica associata. Fra breve dimostreremo che si può descrivere il linguaggio di una grammatica anche come insieme dei prodotti degli alberi sintattici che hanno il simbolo iniziale alla radice e una stringa terminale come prodotto.

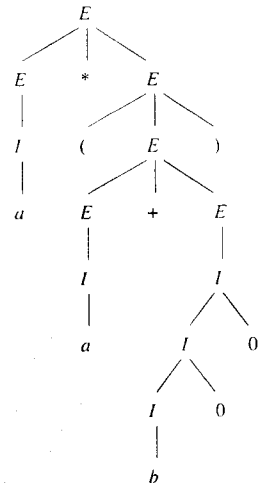

Figura 5.6 Un albero sintattico che illustra come $a * (a + b00)$ sia nel linguaggio della grammatica delle espressioni.

Esempio 5.11 La Figura 5.6 è un esempio di albero con una stringa terminale come prodotto e un simbolo iniziale alla radice; esso è basato sulla grammatica delle espressioni

presentata nella Figura 5.2. Il prodotto di quest'albero è la stringa $a * (a + b00)$, derivata nell'Esempio 5.5. Vedremo che questo particolare albero sintattico è una rappresentazione di quella derivazione. □

5.2.3 Inferenza, derivazioni e alberi sintattici

Ognuna delle nozioni presentate finora per descrivere il funzionamento di una grammatica dà essenzialmente gli stessi risultati sulle stringhe. In altre parole, data una grammatica $G = (V, T, P, S)$, dimostriamo che i seguenti enunciati si equivalgono:

1. la procedura di inferenza ricorsiva stabilisce che la stringa terminale w è nel linguaggio della variabile A
2. $A \stackrel{*}{\Rightarrow} w$
3. $A \underset{lm}{\stackrel{*}{\Rightarrow}} w$
4. $A \underset{rm}{\stackrel{*}{\Rightarrow}} w$
5. esiste un albero sintattico con radice A e prodotto w.

Se escludiamo l'inferenza ricorsiva, definita solo per stringhe terminali, le altre condizioni – l'esistenza di derivazioni generiche, a sinistra o a destra, e di alberi sintattici – sono equivalenti anche se w contiene variabili.

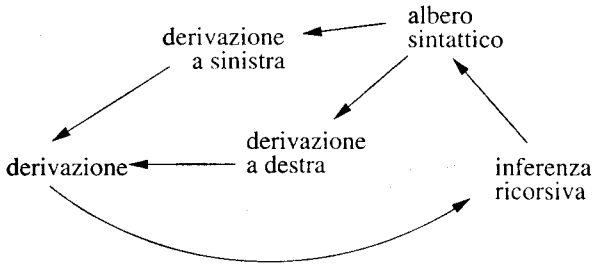

Figura 5.7 Dimostrazione dell'equivalenza di alcuni enunciati sulle grammatiche.

Dimostriamo queste equivalenze secondo lo schema della Figura 5.7. Ogni arco nel diagramma denota un teorema secondo il quale, se w soddisfa la condizione nella coda dell'arco, allora soddisfa anche la condizione alla testa. Per esempio nel Teorema 5.12 dimostreremo che, se si deduce che w è nel linguaggio di A per inferenza ricorsiva, allora esiste un albero sintattico con radice A e prodotto w.

Due degli archi sono molto semplici e se ne tralascerà dunque la dimostrazione formale. Se w ha una derivazione a sinistra da A, allora ha di certo una derivazione da A, dato che una derivazione a sinistra *è* una derivazione. Analogamente, se w ha una derivazione a destra, ha di certo una derivazione. Passiamo ora alla dimostrazione dei passi più ardui dell'equivalenza.

5.2.4 Dalle inferenze agli alberi

Teorema 5.12 Sia $G = (V, T, P, S)$ una CFG. Se la procedura di inferenza ricorsiva indica che la stringa terminale w è nel linguaggio della variabile A, allora esiste un albero sintattico con radice A e prodotto w.

DIMOSTRAZIONE La dimostrazione è un'induzione sul numero dei passi usati per dedurre che w è nel linguaggio di A.

BASE Un solo passo. In questo caso deve essere stata usata soltanto la base della procedura di inferenza. Di conseguenza deve esistere una produzione $A \to w$. L'albero della Figura 5.8, in cui esiste una sola foglia per ogni posizione di w, soddisfa le condizioni degli alberi sintattici per la grammatica G, e ha evidentemente prodotto w e radice A. Nel caso speciale che $w = \epsilon$, l'albero ha una foglia singola etichettata ϵ, ed è quindi un albero sintattico lecito, con radice A e prodotto w.

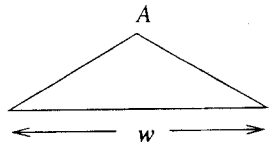

Figura 5.8 Albero costruito nel caso di base del Teorema 5.12.

INDUZIONE Supponiamo di aver dedotto che w è nel linguaggio di A dopo $n + 1$ passi di inferenza, e che l'enunciato del teorema sia valido per tutte le stringhe x e variabili B tali che l'appartenenza di x al linguaggio B si deduca in n, o meno, passi di inferenza. Consideriamo l'ultimo passo dell'inferenza che w è nel linguaggio di A. Questa inferenza impiega una certa produzione per A, poniamo $A \to X_1 X_2 \cdots X_k$, dove ogni X_i è una variabile oppure un terminale.

Possiamo scomporre w in $w_1 w_2 \cdots w_k$ soddisfacendo le seguenti clausole.

1. Se X_i è un terminale, allora $w_i = X_i$; cioè w_i consiste solamente di questo terminale.