Espressioni e linguaggi regolari

Prof. A. Morzenti

aa 2008-2009

ESPRESSIONI E LINGUAGGI REGOLARI

I LINGUAGGI REGOLARI sono la più semplice famiglia di linguaggi formali. Può essere definita in molti modi diversi:
-In modo algebrico (noi cominceremo da qui)
-Con grammatiche generative
-Con algoritmi di riconoscimento

$$\boxed{\begin{array}{c} \Sigma = \{a_1, a_2, \ldots a_i\} \\ \cdot \quad \bigcup \quad * \\ \{a_1\}, \{a_2\}, \ldots \{a_i\} \ \varnothing \end{array}}$$

ESPRESSIONE REGOLARE (e.r.): stringa *r* costruita con i caratteri dell'alfabeto Σ e con i metasimboli ∅ ∪ · con le seguenti regole (dove s e t sono e.r.):

$$\boxed{\begin{array}{lll} 1.\ r = \varnothing & 3.\ r = (s \bigcup t) & 5.\ r = (s)^* \\ 2.\ r = a,\ a \in \Sigma & 4.\ r = (s.t) \text{ o } r = (st) & \end{array}}$$

NB: al posto di U spesso si usa |

PRECEDENZA OPERATORI: stella '*', concatenamento '.', unione 'U'

Si possono usare anche ε definito come $\varepsilon = \varnothing^*$
e^+ definito come $e.e^*$

IL SIGNIFICATO DI UNA e.r. r è un linguaggio L_r di alfabeto Σ secondo la seguente tabella

UN LINGUAGGIO E' DETTO REGOLARE se è denotato da una espressione regolare

espressione r	linguaggio L_r
1. ε	$\{\varepsilon\}$
2. $a \in \Sigma$	$\{a\}$
3. $s \bigcup t$ o $s \mid t$	$L_s \bigcup L_t$
4. $s.t$ o st	$L_s.L_t$
5. $s*$	L_s^*

ESEMPIO 1: L_e sia il linguaggio che contiene le sequenze di segnali multiple di tre

$$e = (111)*$$

$$L_e = \{\varepsilon, 111, 111111, ...\} = \{1^n \mid n \bmod 3 = 0\}$$

$$e_1 = 11(1)* \qquad L_e \neq L_{e_1}$$

$$L_{e_1} = \{11, 111, 1111, 11111, ...\} = \{111^n \mid n \geq 0\}$$

ESEMPIO 2: Sia Σ={+,-,*d*} con *d* che denota una cifra decimale 0,1,…,9.
Si definisce l'e.r. e che produce il linguaggio dei numeri interi con o senza segno.

$$\boxed{e = (+ \bigcup - \bigcup \varepsilon)dd*}$$

$$\boxed{L_e = \{+,-,\varepsilon\}\{d\}\{d\}^*}$$

ESEMPIO 3: Il linguaggio di alfabeto {a,b} è tale che il numero dei caratteri *a* è dispari e vi è almeno un *b*

$$A_p b A_d \mid A_d b A_p$$

$$A_p = b^*(ab^*ab^*)^* \quad A_d = b^*ab^*(ab^*ab^*)^*$$

LA FAMIGLIA DEI LINGUAGGI REGOLARI (REG)
È la collezione di tutti i linguaggi regolari

LA FAMIGLIA DEI LINGUAGGI FINITI (FIN)
E’ la collezione di tutti i linguaggi aventi cardinalità finita

OGNI LINGUAGGI FINITO è REGOLARE perché è l’unione
di un numero finito di stringhe e ciascuna stringa è il concatenamento
di un numero finito di caratteri

$$\boxed{(x_1 \cup x_2 \cup \ldots \cup x_k) = (a_{1_1} a_{1_2} \ldots a_{1_n} \cup \ldots \cup a_{k_1} a_{k_2} \ldots a_{k_m})}$$

La famiglia dei linguaggi regolari contiene anche linguaggi di cardinalità non finita,
quindi l’inclusione è stretta: FIN $\subset$ REG

SOTTOESPRESSIONE DI UNA E.R. (S.E.)

1. Consideriamo una e.r completamente parentesizzata
2. Produciamo una versione numerata della stessa
3. Produciame le sottoespressioni

$$e = (a \bigcup (bb))^* (c^+ \bigcup (a \bigcup (bb)))$$

$$e_N = (a_1 \bigcup (b_2 b_3))^* (c_4{}^+ \bigcup (a_5 \bigcup (b_6 b_7)))$$

$$(a_1 \bigcup (b_2 b_3))^* \qquad c_4{}^+ \bigcup (a_5 \bigcup (b_6 b_7))$$

$$a_1 \bigcup (b_2 b_3) \qquad c_4{}^+ \qquad a_5 \bigcup (b_6 b_7)$$

$$a_1 \quad b_2 b_3 \qquad c_4 \qquad a_5 \quad b_6 b_7$$

$$b_2 \quad b_3 \qquad\qquad b_6 \quad b_7$$

SCELTE: Gli operatori di unione e di ripetizione presenti in una e.r. corrispondono a possibili scelte. Fissando una scelta si ottiene una e.r. che Definisce un linguaggio più piccolo.

$$e_k, 1 \le k \le n, \ \grave{e}\ scelta\ dell'unione\ e_1 \cup ... \cup e_n$$

$$e^n, n \ge 1, \grave{e}\ scelta\ delle\ espressioni\ e^*, e^+$$

$$\varepsilon\ \grave{e}\ scelta\ dell'espressione\ e^*$$

Data una e.r. se ne può DERIVARE una seconda sostituendo al posto di una sottoespressione un'altra scelta da essa.

RELAZIONE DI DERIVAZIONE tra due e.r. *e'* ed *e''*

$$e' \Rightarrow e'' \quad \text{se le due e.r. si possono fattorizzare come}$$
$$e' = \alpha\beta\gamma \qquad e'' = \alpha\delta\gamma$$
$$\text{dove: } \beta \text{ è s.e. di } e',\ \delta \text{ è s.e. di } e'', \delta \text{ è una scelta di } \beta$$

LA RELAZIONE DI DERIVAZIONE è applicabile più volte di seguito (cfr. definizione di ***potenza*** di una relazione, ***chiusura transitiva e riflessiva***)

$$e_0 \overset{n}{\Rightarrow} e_n \quad \text{se} \quad e_0 \Rightarrow e_1, e_1 \Rightarrow e_2, \quad \ldots., \quad e_{n-1} \Rightarrow e_n$$
$$e_0 \overset{+}{\Rightarrow} e_n \qquad e_0 \text{ deriva } e_n \text{ in } n \geq 1 \text{ passi}$$
$$e_0 \overset{*}{\Rightarrow} e_n \qquad e_0 \text{ deriva } e_n \text{ in } n \geq 0 \text{ passi}$$

ESEMPI

Derivazioni immediate e derivazioni a più passi:

$$a^* \bigcup b^+ \Rightarrow a^*, \quad a^* \bigcup b^+ \Rightarrow b^+$$

$$a^* \bigcup b^+ \Rightarrow a^* \Rightarrow \varepsilon \text{ ossia } a^* \bigcup b^+ \overset{2}{\Rightarrow} \varepsilon \text{ o anche } a^* \bigcup b^+ \overset{+}{\Rightarrow} \varepsilon$$

$$a^* \bigcup b^+ \Rightarrow b^+ \Rightarrow bbb \text{ ossia } a^* \bigcup b^+ \overset{2}{\Rightarrow} bbb \text{ o anche } a^* \bigcup b^+ \overset{+}{\Rightarrow} bbb$$

Tra le e.r. ottenute per derivazione da una e.r. alcune contengono anche i metasimboli (operatori e parentesi), altre soltanto i simboli di Σ (detti anche ***terminali***) e la ε.
Queste ultime costituiscono il linguaggio definito dalla e.r.

Il LINGUAGGIO DEFINITO DA UNA E.R. è

$$L(r) = \left\{ x \in \Sigma^* \mid r \overset{*}{\Rightarrow} x \right\}$$

Due e.r sono dette EQUIVALENTI se definiscono lo stesso linguaggio

IL LINGUAGGIO DELLA E.R. DERIVATA E' INCLUSO IN QUELLO DELLA E.R. DERIVANTE

Ci possono essere molti ORDINI DISTINTI ma EQUIVALENTI per produrre una frase del linguaggio

ESEMPIO:

$1.(ab)^* \Rightarrow abab$	$5.a^*(b \cup c \cup d) f^+ \Rightarrow aaa(b \cup c \cup d) f^+$
$2.(ab \cup c) \Rightarrow ab$	$6.a^*(b \cup c \cup d) f^+ \Rightarrow a^* c f^+$
$3.a(ba \cup c)^* d \Rightarrow ad$	$7.a^*(b \cup c \cup d) f^+ \overset{+}{\Rightarrow} aaacf^+$ in 2 passi
$4.a(ba \cup c)^* d \Rightarrow a(ba \cup c)(ba \cup c)d$	$8.a^*(b \cup c \cup d) f^+ \overset{+}{\Rightarrow} aaacff$ in 3 passi

AMBIGUITA' DELLE ESPRESSIONI REGOLARI

Una frase può essere ottenuta con derivazioni che differiscono in modo più sostanziale del solo ordine di derivazione.

$$(a\bigcup b)^* a(a\bigcup b)^*$$

$$(a\bigcup b)^* a(a\bigcup b)^* \Rightarrow (a\bigcup b)a(a\bigcup b)^* \Rightarrow aa(a\bigcup b)^* \Rightarrow aa\varepsilon \Rightarrow aa$$

$$(a\bigcup b)^* a(a\bigcup b)^* \Rightarrow \varepsilon a(a\bigcup b)^* \Rightarrow \varepsilon a(a\bigcup b) \Rightarrow \varepsilon aa \Rightarrow aa$$

Una e.r. *f* E' AMBIGUA se e solo se, nel linguaggio definito dalla corrispondente e.r. marcata *f'* vi sono due diverse stringhe *x* e *y*, tali che, cancellando i numeri, esse vengono a coincidere.

$$f' = (a_1\bigcup b_2)^* a_3(a_4\bigcup b_5)^*$$

definisce un linguaggio regolare di alfabeto

$$\{a_1, b_2, a_3, a_4, b_5\}$$

a_1a_3 e a_3a_4 del ling. di f' mostrano l'ambiguità

ESEMPIO (ambiguità)

$(aa \mid ba)^* a \mid b(aa \mid ba)^*$ è ambigua

infatti, marcando

$(a_1 a_2 \mid b_3 a_4)^* a_5 \mid b_6 (a_7 a_8 \mid b_9 a_{10})^*$

si ottengono

$b_3 a_4 a_5 \quad b_6 a_7 a_8$

che si proiettano in modo ambiguo nella frase *baa*

APPLICAZIONE:

numeri reali in virgola (punto) mobile, con o senza segno ed esponente

$$\Sigma = \{+, -, \bullet, E, d\}$$

$$r = s.c.e$$

$$s = (+ \cup - \cup \varepsilon)\ \text{apporta il segno} \pm \text{opzionale}$$

$$c = (d^{+} \bullet d^{*} \cup d^{*} \bullet d^{+})\ \text{genera costanti intere o frazionarie senza segno}$$

$$e = (\varepsilon \cup E(+ \cup - \cup \varepsilon)d^{+})\ \text{genera l'esponente opzionale preceduto da E}$$

$$(+ \cup - \cup \varepsilon)(d^{+} \bullet d^{*} \cup d^{*} \bullet d^{+})(\varepsilon \cup E(+ \cup - \cup \varepsilon)d^{+})$$

$$+dd \bullet E - ddd \qquad +12 \bullet E - 341 \quad 12.10^{-341}$$

ALTRI OPERATORI

POTENZA: a^h = aa…a (h volte)

RIPETIZIONE: da k a n > k: $[a]_k^n = a^k \cup a^{k+1} \cup \ldots \cup a^n$

OPZIONALITA': $[a] = (\varepsilon \cup a)$

INTERVALLO ORDINATO (0...9) (a...z) (A...Z)

Operazioni insiemistiche INTERSEZIONE, DIFFERENZA, COMPLEMENTO

Si parla di ESPRESSIONI REGOLARI ESTESE con operatori insiemistici

Si dimostra (studiando relazione con automi finiti cfr. cap. 3 testo) che op. insiemistiche non aumentano potenza espressiva delle e.r.

(sono solo una ***comoda abbreviazione***)

INTERSEZIONE: comoda per definire linguaggi mediante congiunzione di condizioni

ESEMPIO: linguaggio $L \subset \{a,b\}^*$ delle stringhe di lunghezza pari e contenenti bb

Facile definire usando un'e.r. e con intersezione tale che $L = L_e$:

$$e = \underbrace{((a \mid b)^* bb (a \mid b)^*)}_{\text{frasi contenenti bb}} \cap \underbrace{((a \mid b)^2)^*}_{\text{fr. Lungh. pari}}$$

Senza l'intersezione:

bb **circondata da due stringhe pari o da due dispari**

$$\boxed{((a \mid b)^2)^* bb((a \mid b)^2)^* \mid (a \mid b)((a \mid b)^2)^* bb(a \mid b)((a \mid b)^2)^*}$$

ESEMPIO (uso di e.r. estesa con operatore di complemento)

Definiamo linguaggio $L \subset \{a,b\}^*$ delle stringhe che ***NON*** contengono sottostringa aa

Facile definire il linguaggio complemento $\neg L = \{x \in (a|b)^* \mid x \text{ contiene la sottostringa aa}\}$

$$\neg L = (\ (a|b)^*\ aa\ (a|b)^*\ )$$

Quindi L definibile con e.r. estesa

$$L = \neg(\ (a|b)^*\ aa\ (a|b)^*\ )$$

Definizione mediante e.r. ***NON*** estesa (forse meno leggibile)

$$L = (\ ab \mid b\ )^*\ (a \mid \varepsilon)$$

CHIUSURA DELLA FAMIGLIA REG RISPETTO ALLE OPERAZIONI / 1

Sia θ un operatore che applicato a un linguaggio o a una coppia di linguaggi
Ne produce un altro.

UNA FAMIGLIA DI LINGUAGGI SI DICE CHIUSA RISPETTO AD UN OPERATORE θ SE IL LINGUAGGIO RISULTANTE DALL'APPLICAZIONE DI θ AI LINGUAGGI DELLA FAMIGLIA APPARTIENE ANCORA ALLA FAMIGLIA

PROPRIETA'. La famiglia REG dei linguaggi regolari è CHIUSA RISPETTO AGLI OPERATORI di concatenamento, unione, stella (e quindi anche rispetto agli operatori derivati croce ed elevamento a potenza)
E' un'ovvia conseguenza della definizione stessa di espressione regolare)

Quindi i linguaggi regolari possono essere combinati tra loro con detti operatori senza pericolo di uscire dalla famiglia dei linguaggi definibili con e.r.

CHIUSURA DELLA FAMIGLIA REG RISPETTO ALLE OPERAZIONI / 2

PROPRIETA’ PIU’ FORTE. La famiglia REG dei linguaggi regolari è ***la più piccola*** famiglia di linguaggi che

(i) contiene tutti i linguaggi finiti ed

(ii) è chiusa rispetto a concatenamento, unione, stella.

Si puo` dare una facile dimostrazione per assurdo (cfr. testo §2.3.4)

REG è anche chiusa rispetto a INTERSEZIONE, COMPLEMENTO e RIFLESSIONE (useremo la teoria degli automi per dimostrarlo)

ASTRAZIONE LINGUISTICA

L'astrazione linguistica trasforma le frasi di un linguaggio reale, detto concreto, in una forma più semplice, detta rappresentazione astratta.
Essa trascura i simboli dell'alfabeto concreto e impiega al loro posto i caratteri di un altro alfabeto, quello astratto.

AL LIVELLO ASTRATTO, LE STRUTTURE DEI LINGUAGGI ARTIFICIALI SI POSSONO OTTENERE COME COMPOSIZIONE DI POCHI PARADIGMI ELEMENTARI, ATTRAVERSO LE OPERAZIONI DI CONCATENAMENTO, UNIONE E ITERAZIONE.

Dal linguaggio astratto a quello effettivo → scelta degli elementi lessicali (parole chiave, delimitatori, identificatori, ...)

LA COSTRUZIONE DEI COMPILATORI FA RIFERIMENTO ALLA STRUTTURA ASTRATTA

I linguaggi artificiali fanno uso di (poche) strutture astratte ricorrenti.
Tra queste, LE LISTE sono descrivibili con ESPRESSIONI REGOLARI.

LISTE ASTRATTE E CONCRETE

Una lista contiene un numero imprecisato di elementi *e* dello stesso tipo.
Essa è generata dalla e.r. e^+ o da e^*, se gli elementi possono mancare.

e può essere simbolo terminale o altro (stringa di un altro linguaggio formale, ...)

LISTE CON SEPARATORI E MARCHE DI APERTURA E CHIUSURA

$$ie(se)^* f \qquad\qquad i[e(se)^*]f$$

ESEMPI

$$\overbrace{begin}^{i}\,\overbrace{istr_1}^{e}\,\overbrace{;}^{s}\,\overbrace{istr_2}^{e}\,\overbrace{;}^{s}\ldots\overbrace{;}^{s}\,\overbrace{istr_n}^{e}\,\overbrace{end}^{f}$$

$$\overbrace{procedure\ STAMPA(}^{i}\,\overbrace{par_1}^{e}\,\overbrace{,}^{s}\,\overbrace{par_2}^{e}\,\overbrace{,}^{s}\ldots\overbrace{,}^{s}\,\overbrace{par_n}^{e}\,\overbrace{)}^{f}$$

$$\overbrace{array\ MATRICE\,'['}^{i}\,\overbrace{\mathrm{int}_1}^{e}\,\overbrace{,}^{s}\,\overbrace{\mathrm{int}_2}^{e}\,\overbrace{,}^{s}\ldots\overbrace{,}^{s}\,\overbrace{\mathrm{int}_n}^{e}\,\overbrace{']'}^{f}$$

LISTE A PRECEDENZE O LIVELLI

Un elemento di una lista può essere
Una lista di livello inferiore
NB: il numero di livelli è limitato,
altrimenti servono notazioni più
potenti (grammatiche)

$$lista_1 = i_1 lista_2 (s_1 lista_2)^* f_1$$
$$lista_2 = i_2 lista_3 (s_2 lista_3)^* f_2$$
$$\dots$$
$$lista_k = i_k e_k (s_k e_k)^* f_k$$

ESEMPI

$$\text{livello 1: } begin \quad istr_1; istr_2; \dots; istr_n \quad end$$
$$\text{livello 2: } STAMPA(\text{var}_1, \text{var}_2, \dots, \text{var}_n)$$

$$3 + \underbrace{5 \times 7 \times 4}_{\text{monomio1}} - \underbrace{8 \times 2 \div 5}_{\text{monomio2}} + 8 + 3$$

padre, madre, figlio e figlia
un padre forte, severo e giusto, una madre amorevole e fedele

un libro come lista di capitoli separati da pagine bianche,
chiusa tra due copertine
un capitolo come lista di sezioni

Si passa dalla forma astratta di un costrutto a quella concreta mediante SOSTITUZIONE, operazione semplice che rimpiazza un carattere terminale $b \in \Sigma$ di una stringa x di un primo linguaggio (detto sorgente) con le frasi di un secondo linguaggio $L_b \subseteq \Delta^*$ (detto pozzo)

La sostituzione del linguaggio L_b al posto di b nella stringa $x=a_1a_2\ldots a_n$ produce un linguaggio di alfabeto $(\Sigma \setminus \{b\}) \cup \Delta$ cosi' definito:

$$\boxed{\left\{ y \mid y = y_1 y_2 \ldots y_n \wedge (\text{if } a_i \neq b \text{ then } y_i = a_i \text{ else } y_i \in L_b \right\}}$$