Linguaggi ed espressioni regolari

Da Wikiversità, l'apprendimento libero.

In questa lezione analizzeremo la famiglia delle **espressioni regolari** (in inglese *regular expression* o, in forma abbreviata, *regexp*, *regex* o *RE*) di cui si invita a leggere come introduzione la relativa pagina di Wikipedia.

Indice

Definizione

Formalmente definiamo **espressione regolare** una stringa r costruita su un alfabeto $\Sigma = \{a_1, a_2, \ldots, a_k\}$ e in unione ai seguenti metasimboli:

- $\varnothing$: insieme vuoto
- $\cup$: unione (notazione alternativa: |)
- $\cdot$: concatenazione
- $*$: star
- $()$: parentesi

Una RE è detta **ben formata** se si presenta in una delle seguenti forme:

- $r = \varnothing$
- $r = a,\ a \in \Sigma$
- $r = (s \cup t)$ o $r = (s|t)$
- $r = (s \cdot t)$ o $r = (st)$ (notazione alternativa)
- $r = (s)$

dove s e t sono a loro volta espressioni regolari. Si noti che la precedenza degli operatori è:

- $*$
- $\cdot$
- $\cup$

Definiamo inoltre altri operatori non essenziali ma frequentemente usati, utilizzando solo le proprietà sopra descritte:

- $\varepsilon = \varnothing^*$

- $r^+ = r \cdot r^*$
- $r^h = \underbrace{rr\ldots r}_{m}$ (potenza)
- $[r]_k^n = r^k \cup r^{k+1} \cup \ldots \cup r^n$ con $n \geq k$ (ripetizione)
- $[r] = \varepsilon \cup r$
- $(0...9) = 0123456789, (a...m) = abcdefghijklm$ (intervalli ordinati)

Altri operatori possono essere quelli insiemistici teorici: intersezione, differenza e complemento. Una espressione regolare che contiene questi operatori è detta **espressione regolare estesa**. Nota: Il potere espressivo di una RE estesa **non** è maggiore di quello di una RE standard.

Definizione di linguaggio regolare

Diciamo che un linguaggio è un **linguaggio regolare** se è denotato da una RE. Formalmente, un linguaggio regolare L_r è un linguaggio su un alfabeto Σ che ha una corrispondente RE in accordo con la seguente tabella:

Espressione	Linguaggio
$r = \varepsilon$	$L_r = \{\varepsilon\}$
$r = a \in \Sigma$	$L_r = \{a\}$
$r = s \cup t$ $r = s\|t$	$L_r = L_s \cup L_t$
$r = s \cdot t$ $r = st$	$L_r = L_s \cdot L_t$
$r = s^*$	$L_r = L_s^*$

Denotiamo con $\mathbf{REG}$ la famiglia di tutti linguaggi regolari e con $\mathbf{FIN}$ la famiglia di tutti i linguaggi finiti (cioè con cardinalità finita).

Allora possiamo dire che:

$$\mathbf{FIN} \subset \mathbf{REG}$$

(intuibile: un linguaggio finito può sempre essere visto come l'unione di un numero finito di stringhe, ognuna delle quali concatenazione di un numero finito di simboli dell'alfabeto)

Derivare il linguaggio dalla RE

Per derivare il linguaggio dobbiamo definire alcuni concetti supplementari.

Sottoespressione

Definiamo **sottoespressione** (in inglese subexpression o SE) una *ben parentizzata* sottostringa di una RE che si presenta nelle parentesi più esterne.

Chiariamo con un esempio. Sia data la RE:

$r = (s \cup (t \cdot (u \cup z)^+))$

questa RE ha due SE: s e $(t \cdot (u \cup z)^+)$, mentre t e $(u \cup z)^+$ NON sono SE di r, ma sono SE di $(t \cdot (u \cup z)^+)$.

Versione numerata

Definiamo '*versione numerata* di una RE, la RE a cui vengono aggiunti i numeri alle lettere che compongono la RE, in modo da differenziale le lettere uguali. Anche qui chiariamo il concetto con un esempio:

$$(aa)^* \cup (b \cdot ((cc)^+ \cdot a))$$

la sua versione numerata è:

$$(a_1 a_2)^* \cup (b_1 \cdot ((c_1 c_2)^+ \cdot a_3))$$

Questa notazione è importante per definire l'ambiguità di un linguaggio (introdotta nelle sezioni successive).

Scelta e derivazione

Diciamo che una RE è una **scelta** (in inglese **choice**) di un'altra RE nei seguenti casi:

- $e_k,\ 1 \leq k \leq m$ è una scelta di $(e_1 \cup e_2 \cup \ldots \cup e_k)$
- $e_m = \underbrace{e \ldots e}_{m},\ m \geq 1$ è una scelta di e^+ e e^*
- ε è una scelta di e^*

Diciamo che una SE e' **deriva** da e'' (scritto come $e' \Rightarrow e''$ se:

- e'' è una scelta di e';
- *oppure,* e''_i è una scelta di e'_i per ogni $1 \leq i \leq m$

La derivazione può avvenire più volte allo stesso modo. In questo caso scriviamo:

- $e_0 \overset{n}{\Rightarrow} e_n$
  - se $e_0 \Rightarrow e_1, e_1 \Rightarrow e_2$, ..., $e_{n-1} \Rightarrow e_n$ con n fisato
- $e_0 \overset{+}{\Rightarrow} e_n$
  - se $e_0 \Rightarrow e_1, e_1 \Rightarrow e_2$, ..., $e_{n-1} \Rightarrow e_n$ con $n \geq 1$
- $e_0 \overset{*}{\Rightarrow} e_n$
  - se $e_0 \Rightarrow e_1, e_1 \Rightarrow e_2$, ..., $e_{n-1} \Rightarrow e_n$ con $n \geq 0$

Esempi

- $a^* \cup b^+ \Rightarrow a^*$
- $a^* \cup b^+ \Rightarrow a^+$
- $a^* \cup b^+ \Rightarrow a^* \Rightarrow \varepsilon$ o equivalentemente $a^* \cup b^+ \overset{2}{\Rightarrow} \varepsilon$ o ancora $a^* \cup b^+ \overset{+}{\Rightarrow} \varepsilon$
- $a^* \cup b^+ \Rightarrow b^+$
- $a^* \cup b^+ \Rightarrow b^+ \Rightarrow bbbb$ o equivalentemente $a^* \cup b^+ \overset{2}{\Rightarrow} bbbb$ o ancora $a^* \cup b^+ \overset{+}{\Rightarrow} bbbb$

Linguaggio definito da un RE

Il **linguaggio definito da una espressione regolare** r è:

$$L_r = \{x \in \Sigma^* | r \overset{*}{\Rightarrow} x\}$$

Diciamo che due RE sono **equivalenti** se definiscono lo stesso linguaggio.

Ambiguità delle RE

Una stringa di un linguaggio regolare può essere derivato dalla RE in modi differenti, cioè attraverso distinte derivazioni. Diciamo che una RE è **ambigua** se esiste una stringa derivabile attrverso due distinte derivazioni *che non differiscono solo dall'ordine di applicazione*.

Esempio:

$$a*\cup(b*\cup a)$$

Ambigua, due modi di derivazione di a:

- $a*\cup(b*\cup a) \Rightarrow a* \Rightarrow a$
- $a*\cup(b*\cup a) \Rightarrow (b*\cup a) \Rightarrow a$

Condizione **sufficiente** affinchè una RE sia ambigua, se il linguaggio generato dalla RE in versione numerata include due stringhe che coincidono a meno dei numeri.

Esempio:

$$a_1*\cup(b_1*+\cup a_2)$$

Genera:

1. ε
2. a_1
3. $a_1 a_1$
4. $a_1 a_1 a_1$
5. b_1
6. $b_1 b_1$
7. a_2
8. ...

Come si vede eliminando i numeri, la stringa 2 coincide con la stringa 7, perciò la RE è ambigua.

Proprietà di chiusura

> « Un insieme è chiuso rispetto a un'operazione se e solo se ogni insieme ottenuto applicando l'operazione ai membri dell'insieme originale, l'insieme ottenuto è contenuto nell'insieme originario »

REG è chiuso rispetto alla concatenazione, unione e star (quindi anche per gli altri operatori sopra descritti).

Link e riferimenti

Esempi pratici - https://www.evemilano.com/2014/07/come-funzionano-le-espressioni-regolari-regex/

Altri progetti

- **Wikibooks** contiene testi o manuali sulle **espressioni regolari**
- **Wikipedia** contiene informazioni sulle **espressioni regolari**
- **Wikimedia Commons (https://commons.wikimedia.org/wiki/?uselang=it)** contiene immagini o altri file sulle **espressioni regolari (https://commons.wikimedia.org/wiki/Category:Regex?uselang=it)**