Fondamenti d'Informatica: linguaggi formali

Barbara Re, Phd

Agenda

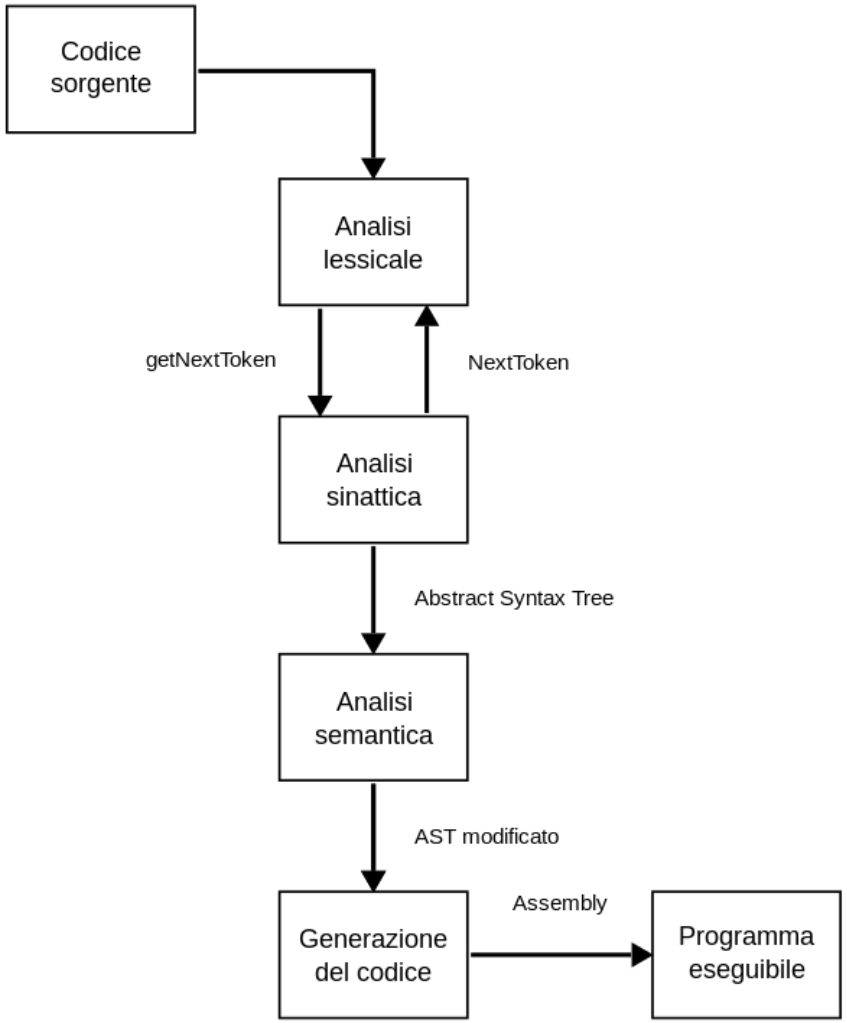
- Introdurremo …
  - La nozione di linguaggio
  - Strumenti per definire un linguaggio
    - Espressioni Regolari

Linguaggio

- Da un punto di vista matematico
  - Insieme di stringhe su un dato alfabeto
- Da un punto di vista informatico
  - E' interessante osservare linguaggi, generalmente infiniti, le cui stringhe sono caratterizzate da qualche particolare proprietà
  - *Esempio: Il linguaggio C è costituito da tutte le stringhe che soddisfano la proprietà di poter essere analizzate da un compilatore C senza che venga rilevato alcun errore sintattico*

Un compilatore, in informatica, è un programma che traduce una serie di istruzioni scritte in un determinato linguaggio di programmazione (codice sorgente) in istruzioni di un altro linguaggio (codice oggetto). Questo processo di traduzione si chiama **compilazione**.

Fasi tipiche della compilazione



- **Analisi lessicale**
  - Attraverso un analizzatore lessicale il compilatore divide il codice sorgente in tanti pezzetti chiamati token. I token sono gli elementi minimi (non ulteriormente divisibili) di un linguaggio, ad esempio parole chiave (for, while), nomi di variabili (pippo), operatori (+, -, «)
- **Analisi sintattica**
  - L'analisi sintattica prende in ingresso la sequenza di token generata nella fase precedente ed esegue il controllo sintattico. Il controllo sintattico è effettuato attraverso una grammatica.
- **Analisi semantica**
  - L'analisi semantica si occupa di controllare il significato delle istruzioni presenti nel codice in ingresso. Controlli tipici di questa fase sono il type checking, ovvero il controllo di tipo, controllare che gli identificatori siano stati dichiarati prima di essere usati e così via. Come supporto a questa fase viene creata una tabella dei simboli che contiene informazioni su tutti gli elementi simbolici incontrati quali nome, scope, tipo (se presente) etc.

Studio formale dei linguaggi: applicazioni

- Studio delle proprietà sintattiche dei linguaggi
  - Studio dei metodi di definizione della sintassi di un linguaggio di programmazione
  - Studio dei metodi di verifica che un programma soddisfi le proprietà sintattiche volute
  - Studio dei metodi di traduzione da un linguaggio all'altro
- Generazione e riconoscimento di linguaggi
  - Stringhe di caratteri aventi struttura particolare vengono frequentemente utilizzate per rappresentare vari aspetti dell'elaborazione dei dati (es. sequenze di operazioni, sequenze di passi, protocolli di comunicazione, sequenze di interazioni sviluppate con l'interfaccia utente)
- Nella formulazione dei problemi trattati con metodi algoritmici ci si riconduce frequentemente al problema standard di decidere l'appartenenza di una stringa ad un dato linguaggio
  - Esempio - Riconoscere il linguaggio $\{a^n b^n \mid n \geq 0\}$

Strumenti per definire un linguaggio

- Espressioni regolari
  - Permettono di **definire linguaggi con una struttura semplice** e non sono idonee a rappresentare linguaggi più complessi come quelli di programmazione
- Approccio generativo
  - Si utilizzano **le grammatiche formali** che consentono di **costruire le stringhe di un linguaggio** tramite un insieme prefissato di **regole** di **produzione**
- Approccio riconoscitivo
  - Consiste nell'utilizzare **macchine astratte**, dette **automi riconoscitivi**, che definiscono algoritmi di riconoscimento dei linguaggi stessi, vale a dire algoritmi che per un dato linguaggio $L \in A^*$ stabiliscono se una stringa $\alpha \in A^*$ appartiene a L o meno

Alcune espressioni comunemente utilizzate

- indirizzo email
- url http
- codice fiscale
- Indirizzo IP

Espressione regolari

- Le espressioni regolari consentono di descrivere tutti i linguaggi appartenenti ad un'importante classe con una struttura abbastanza semplice

Def: Dato un alfabeto A e dato l'insieme di simboli

{+, $\cdot$, *, Ø} si definisce **espressione regolare** sull'alfabeto A una stringa r $\in$ (A U {+, $\cdot$, *, Ø})$^+$ tale che valga le seguenti condizioni

1. r = Ø
2. r = a che è un evento (o azione atomica) dove a $\in$ A
3. r = (s + t), oppure r = (s $\cdot$ t), oppure r = s*, dove s e t sono espressioni regolari sull'alfabeto A

Espressioni Regolari e linguaggi

- Le espressioni regolari consentono di rappresentare linguaggi mediante un'opportuna interpretazione dei simboli che le compongono

Espressioni Regolari		**Linguaggi**
$\emptyset$	Insieme vuoto	Linguaggio che non contiene alcuna stringa
a	Evento o azione atomica	{a}
(s + t)	scelta	L(s) + L(t)
(s · t)	Sequenza (a volte si omette il simbolo)	L(s) · L(t)
s*	Sequenza di n elementi con n maggiore o uguale a zero	(L(s))*

Sono valide le concatenazioni

Semantica

(a + b)* (a*+ b*)*

- Si tratta della stessa espressione?
- Qual'è il significato?
- Un'espressione regolare E denota un linguaggio L(E), dove L rappresenta una funzione da stringhe a linguaggi, pertanto:

$L(\emptyset) = \emptyset$
$L\{a\} = \{a\}$
$L(s + t) = L(s) + L(t)$
$L(s \cdot t) = L(s) \cdot L(t)$
$L(s^*) = (L(s))^*$

SEMANTICA
(data per induzione strutturale)

Costruzione di espressioni regolari

- Esempio: Si faccia riferimento all'espressione regolare per stringhe che constano di 0 e 1 alternati
- Istanze
  - 0101
  - 1010
  - 01010
  - 10101
- 0 e 1 rappresentano i linguaggi L(0) = { 0 }, L(1)= { 1}
- 0 · 1 rappresenta il linguaggio L(0 · 1) = { 0 1}
- (0 · 1)* rappresenta il linguaggio L((0 · 1)*) = {0 1 0 1 0 1……}
- (1 · 0)* rappresenta il linguaggio L((1 · 0)*) = {1 0 1 0 1 0……}

$$(0 \cdot 1)^* + (1 \cdot 0)^* \ ?$$

Costruzione di espressioni regolari

- **Regole di precedenza tra gli operatori:**

  * precede · precede +

Ex: a + b* c è equivalente a a + ((b)*) (c))

Ex: (hot + cold)(apple + blueberry + cherry)(pie + tart)
rappresenta le dodici possibili stringhe date da:

- hot apple pie
- hot apple tart
- ……..
- cold cherry pie
- cold cherry tart

Costruzione di Espressioni Regolari

- Stringhe di a e di b

  $(a + b)^*$

- Stringhe con almeno due a

  $(a + b)^* \; a \; a \; (a + b)^*$

- Stringhe con almeno due b

  $(a + b)^* \; b \; b \; (a + b)^*$

- Stringhe contenenti le sottostringhe aa oppure bb

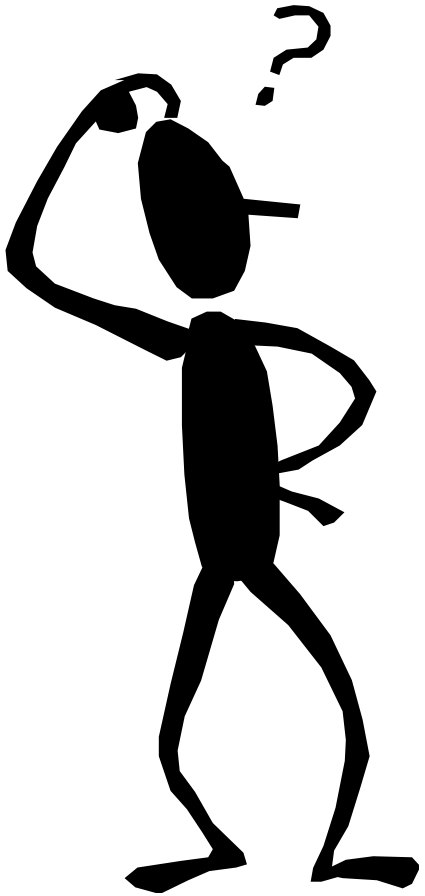
  $(a + b)^* \; a \; a \; (a + b)^* + (a + b)^* \; b \; b \; (a + b)^*$

Esercizi: espressioni comunemente utilizzate

- indirizzo email
- url http
- codice fiscale
- Indirizzo IP

Esercizi Espressioni Regolari

1. Scrivere una espressione regolare per stringhe binarie che descriva l'insieme {0, 11, 101}.
2. Scrivere una espressione regolare per stringhe binarie che descriva l'insieme delle stringhe composte solo da simboli "0".
3. Scrivere una espressione regolare per tutte stringhe binarie.
4. Scrivere una espressione regolare per tutte stringhe binarie che cominciano e finiscono per "1".
5. Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre "1" consecutivi.
6. Scrivere una espressione regolare per le stringhe binarie che contengono almeno tre "1".
7. Scrivere una espressione regolare per le stringhe binarie di lunghezza dispari.
8. Scrivere una espressione regolare per stringhe di testo che descriva le date in formato GG/MM/AAAA.
9. Scrivere una espressione regolare per stringhe di testo che descriva i numeri di telefono.



Questions?