

tesi di laurea

Realizzazione di strumenti di misura di metriche di prodotto software con ANTLR

Anno Accademico 2007-2008

Relatore

Ch.mo prof. Porfirio Tramontana

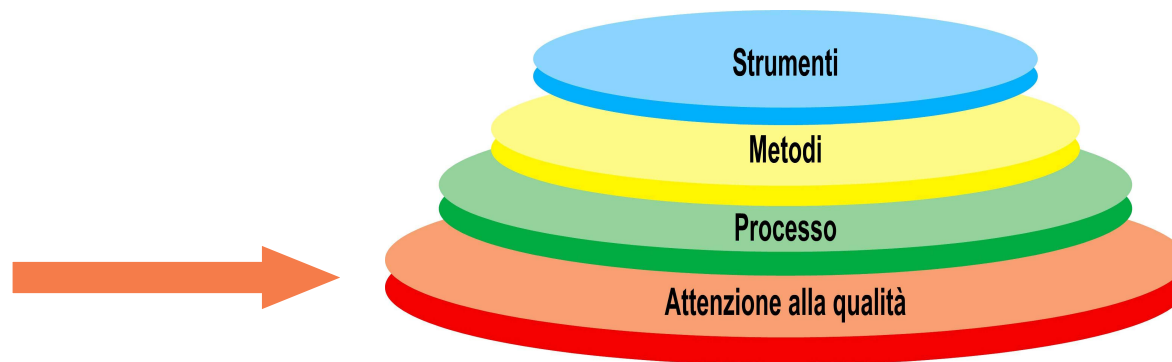
Candidato

Fabio Canova

Matr. 534001505

Obiettivo

- **Analisi statica di codice sorgente, realizzato in javascript, con Antlr**
 - **Data Flow Analysis (verifica del software)**
 - **Valutazione di metriche di prodotto software**
 - complessità ciclomatica
 - lines of code (LOC)

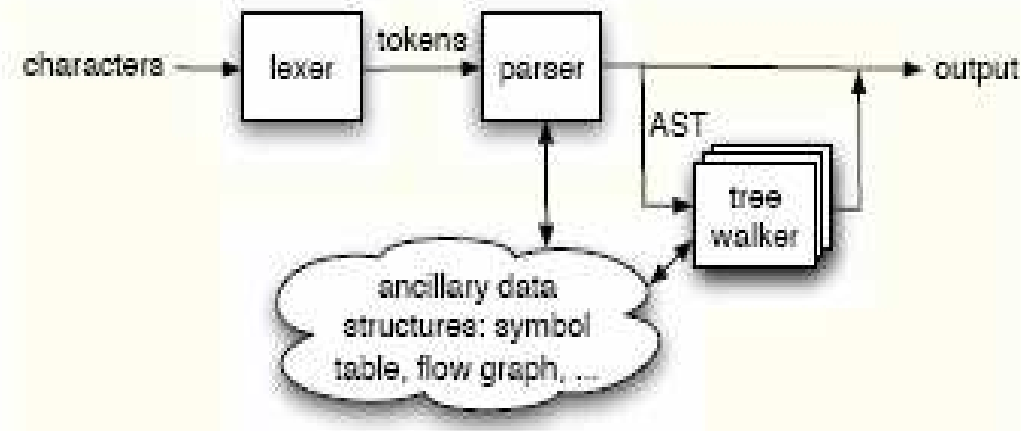




Contesto

- **Data Flow Analysis** : tecnica di verifica del software che si basa sull'evoluzione delle variabili, permettendo di rilevare possibili anomalie. L'analisi è legata alle operazioni eseguite su una variabile :
 - dichiarazione* : enuncia il tipo di variabile
 - definizione* : dichiara il tipo e ne definisce l'allocazione in memoria (alla variabile viene assegnato un valore)
 - uso* : il valore della variabile viene utilizzato in una computazione (C-USE) oppure in un predicato (P-USE)
 - terminazione* : la variabile cessa di esistere
- **Lines of Code** : appartiene alla categoria delle metriche dimensionali in quanto misura la lunghezza di un programma con il numero di linee di codice
- **Complessità ciclomatica** : metrica strutturale che determina la complessità logica di un programma e quindi lo sforzo per realizzarlo; è interamente basata sulla struttura del grafo che rappresenta il programma

Lexer e Parser

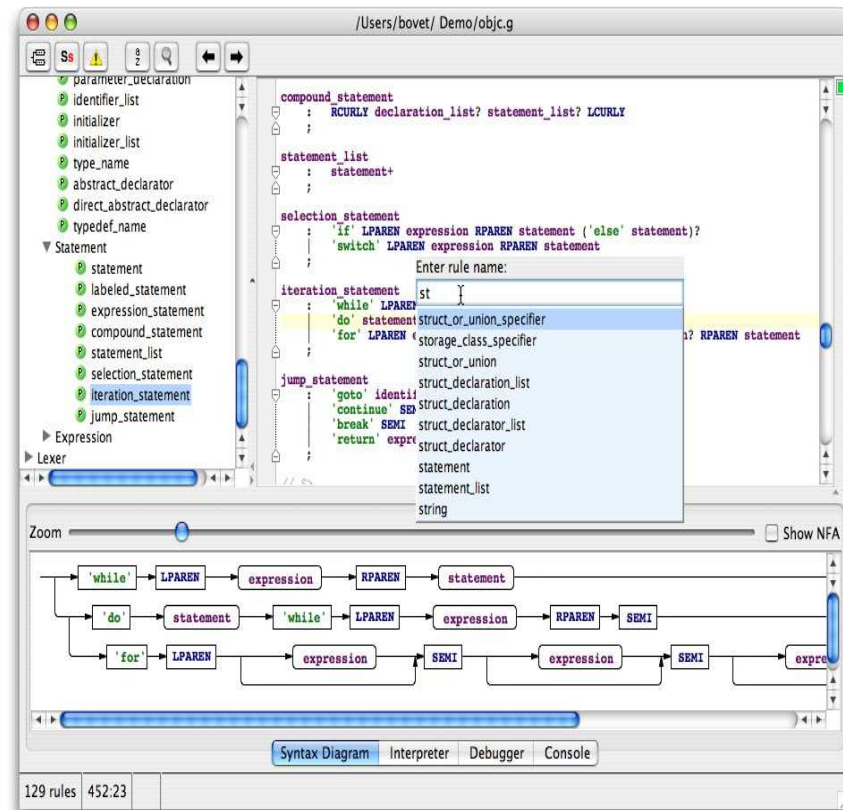


Il *lexer* impacchetta flussi di caratteri in uno o più gruppi detti tokens dell'analisi lessicale. I tokens sono componenti del linguaggio di programmazione in questione come parole chiave, identificatori, simboli ed operatori. Il *parser* analizza la sequenza di tokens fornita in ingresso e riconosce la loro organizzazione sintattica eseguendo anche delle elaborazioni conseguenti al loro riconoscimento.

Antlr ed Antlworks

- **ANTLR** Another Tool for Language Recognition è un sofisticato generatore di parser utilizzato per implementare riconoscitori, compilatori, traduttori a partire dalla descrizione di una grammatica contenente una serie di azioni. Antlr produce automaticamente il codice sia per il lexer che per il parser.
- Antlr ha a sua disposizione un sofisticato ambiente di sviluppo di grammatiche chiamato **ANTLRWorks**.
- Si tratta di un ambiente GUI che integra al suo interno anche Antlr

Realizzazione di strumenti di misura di metriche di prodotto software con antlr





Analisi del problema e soluzione

- Realizzare uno strumento che sia in grado di riconoscere codice javascript ed effettuare un'analisi statica (Data Flow) del codice per la valutazione della complessità ciclomatica, del numero di linee di codice e per la valutazione dei cammini def-use.
- Soluzione con Antlr per la progettazione di un riconoscitore, attraverso la scrittura di regole sintattiche e regole semantiche con Antlrworks, che produca ulteriori elaborazioni (mediante azioni in java) conseguenti al riconoscimento della struttura sintattica.
- Non si tratta di un semplice riconoscitore
- Si può definire come un vero sistema software che :
 - Rappresenti il codice attraverso un grafo (liste di adiacenza)
 - Calcoli le linee effettive di codice mediante una variabile contatore che si incrementa ogni volta che viene riconosciuta un'istruzione
 - Calcoli la complessità ciclomatica $V(G) = P + 1$, con P nodi predicato
 - Effettui l'analisi Data Flow eseguendo l'attraversamento del grafo *depth-first search* e memorizzando le variabili in un database MySQL

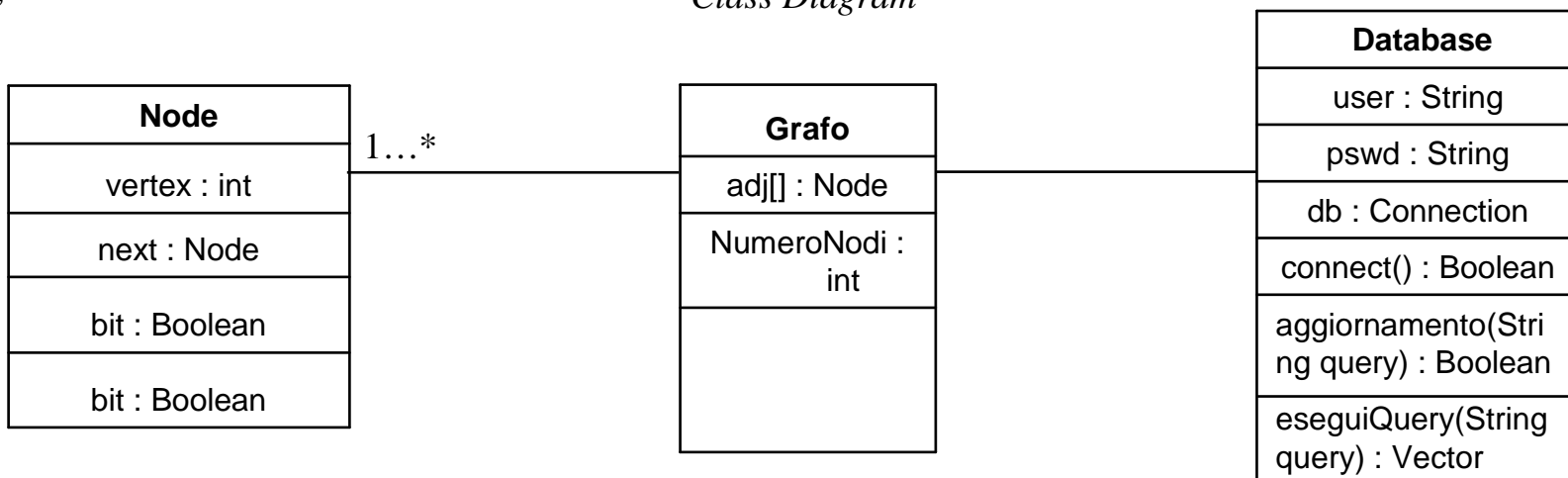
Progettazione Software

- Analisi lessicale e sintattica per la realizzazione di una grammatica JavaScript (Riconoscitore)
grammar JavaScript;

```

    prog : stat+
           ;
    stat : statement
         | start
         | end
    variabili K {gr.push_sequenza(i);} NEWLINE
    ;
  
```

Class Diagram



Un esempio..

- Script che realizza un motore di ricerca interno ad una pagina web

LOC : 36

Complessità ciclomatica : 4

Notazione utilizzata per l'analisi Data Flow

DI → dichiarazione di una variabile

D → definizione di una variabile

U → uso di una variabile

T → terminazione di una variabile

Sequenza della variabile *Found* ad un determinato cammino
sul grafo : DDUT (doppia definizione)

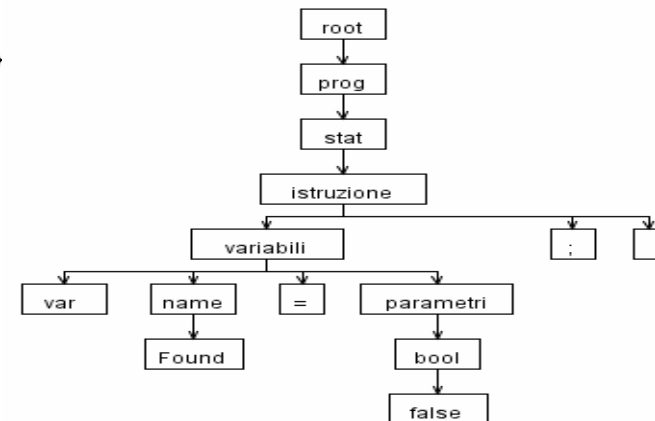
Sequenza della variabile *i* allo stesso cammino sul grafo :

DIDUUT (consentita)

- Script esaminati dettagliatamente : 10

```
function checkDatabase(){ //FUNCTION
var Found=false;
var
Item=document.forms.ricerca.searchfor.value.toL
owerCase();
var i;
MsgBox=window.open("", "msgWindow", 'toolbar=no,
location=no, directories=no, status=no, menubar=no, hei
ght=300, width=300, scrollbars=yes, resizable=yes')
MsgBox.opener=window;
MsgBox.opener.name="opener";
.
}

```





Conclusioni e sviluppi futuri

- Attraverso Antlr :
 - grammatica Javascript (Riconoscitore)
 - elaborazioni per l'analisi Data Flow e la valutazione di metriche di prodotto (Lines of Code e Complessità Ciclomantica)
 - codice sorgente del lexer e del parser
- Ottimo generatore di parser
- Modifica della sintassi e delle azioni senza dover modificare il codice del parser
- Javascript è un linguaggio di scripting usato all'interno di pagine web
- Reale completamento di HTML
- Aggiunta di regole sintattiche per la gestione del linguaggio HTML
- Ulteriori elaborazioni per migliorare l'analisi del codice Javascript :
 - valutazione di ulteriori metriche di prodotto
 - analisi dinamica