Università degli Studi di Salerno

Dipartimento di Informatica



Progetto di Compilatori

MyFun

# Analisi lessicale

Implementazione aderente alle indicazioni fornite.

# Analisi sintattica

### Risoluzione dei conflitti della grammatica in Cup

1. Una volta inserite le regole della grammatica data, ho provato a generare il parser con Cup ed ho analizzato il file "dump.txt" prodotto in output: sono stati individuati dei conflitti.
2. Ho aggiunto le **regole di precedenza e associatività**, seguendo le specifiche date, ed il numero dei conflitti è sceso ma non si è azzerato; nello specifico ho notato che la totalità dei conflitti è causata dalle produzioni *StatList -> Stat*e *Stat -> /\*empty\*/*.
3. Ho risolto il conflitto causato da ***StatList -> Stat*** e***Stat -> /\* empty \*/***rimuovendo quest'ultima produzione e cambiando la prima con ***StatList -> /\*empty\*/***, in questo modo invece di arrivare a Stat per andare in **/\*empty\*/** è possibile anticipare restituendo la lista vuota di Stat. A seguito di ciò, il numero dei conflitti si è **azzerato**.

# PATTERN JAVA

Pattern Java per la visita dei nodi Abbiamo deciso di gestire la visita dell’AST servendoci di due interfacce: Visitor e Visitable. Le classi XMLVisitor, SemanticVisitor e CLangVisitor implementano un’interfaccia Visitor che obbliga queste ad implementare un metodo visit(). Queste classi sono quelle che “visitano” l’AST. Le classi corrispondenti ai nodi dell’AST invece implementano un’interfaccia Visitable che obbliga loro ad implementare un metodo accept(). Queste classi sono quelle “visitate”. Prendendo in esame un nodo che si vuole visitare, si chiama su di esso il metodo accept() passandogli il visitor. All’interno del metodo accept() si va a chiamare il metodo visit del visitor passato fornendogli in input il nodo stesso.

# Analisi semantica

### Regole di type checking implementate nel formato regole di inferenza

sono definite le seguenti regole di type checking:

### Typing relations

|  |  |  |
| --- | --- | --- |
| Op1 | Operando | Risultato |
| - | integer | integer |
| - | real | real |
| ! | boolean | boolean |

|  |  |  |  |
| --- | --- | --- | --- |
| Op2 | Operando1 | Operando2 | Risultato |
| + - \* / ^ | integer | integer | integer |
| + - \* / ^ | integer | real | real |
| + - \* / ^ | real | integer | real |
| + - \* / ^ | real | real | real |
| div int | integer | integer | integer |
| div int | integer | real | integer |
| div int | real | integer | integer |
| div int | real | real | integer |
| && || | bool | bool | boolean |
| = <> < <= > >= | bool | bool | boolean |
| = <> < <= > >= | integer | integer | boolean |
| = <> < <= > >= | integer | real | boolean |
| = <> < <= > >= | real | integer | boolean |
| = <> < <= > >= | real | real | boolean |
| = <> < <= > >= | string | string | boolean |
| String Concat | string | string | string |
| String Concat | string | real | string |
| String Concat | real | string | string |
| String Concat | string | integer | string |
| String Concat | integer | string | string |
| String Concat | string | boolean | string |
| String Concat | boolean | string | string |

Tabella di compatibilità tra tipi

|  |  |  |
| --- | --- | --- |
| Tipo | Tipi compatibili | |
| integer | integer | |
| real | real | integer |
| boolean | boolean | |
| string | string | |