# Appunti di Linguaggi e Computabilità

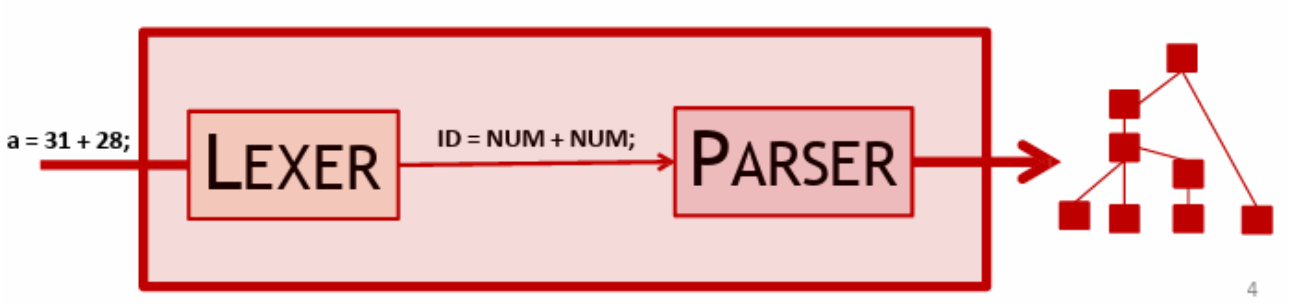
## Domande e Risposte – Laboratorio

### 1) Qual è l'input di un analizzatore lessicale (lexer) e che ruolo possiede?

L'analizzatore lessicale riceve in input una sequenza di caratteri (una stringa) e fornisce in output una sequenza di token. Il lexer analizza l'input, un carattere alla volta, finchè non trovo il prefisso più lungo associato da pi, ed esegue l'azione ai finchè non torna il controllo al parser.

### 2) Qual è l'input di un analizzatore sintattico (parser) e che ruolo possiede?

L'analizzatore sintattico riceve in input una sequenza di token (coppie nome e valore) e fornisce in output una albero sintattico (parse tree). Il parser analizza il testo per determinarne la correttezza della struttura grammaticale. Esso contribuisce all'identificazione degli errori di sintassi e produce un albero sintattico. Essi ricorrono ai lexer per identificare le sequenze di token.




### 3) Qual è la differenza tra la grammatica lessicale e la grammatica sintattica di un linguaggio di programmazione?

L'analisi lessicale è la procedura che consiste nel ricevere in input una stringa (ad esempio un'istruzione) ed analizzarla lessicalmente. Se questa non presenta errori lessicali, tale stringa verrà trasformata in token in modo da poter essere analizzato sintatticamente dal parser (analisi lessicale).

### 4) Quali sono le principali componenti di un compilatore?

I principali componenti di un compilatore sono:

→ lexer: analizza l'input, un carattere alla volta, finchè non trovo il prefisso più lungo associato da pi, ed esegue l'azione ai finchè non torna il controllo al parser;

→ parser: analizza il testo per determinarne la correttezza della struttura grammaticale;

→ type checker: controlla il siginificato delle istruzioni presenti nel codice di input (ad esempio verifica se gli identificatori sono stati dichiarati prima di essere utilizzati) e il risultato prodotto è l'albero sintattico astratto;

→ generazione del codice intermedio: dall'albero sintattico viene generato il codice intermedio.

## 5) Cosa si intende per parse stack?

Il parse stack è una struttura LIFO utilizzata dal parser per controllare la correttezza sintattica del testo di input. Vengono impilati i token letti attraverso l'operazione di shift. Se al termine del parsing la pila viene svuotata si ha che il testo risulta sintatticamente corretto. L'output del generatore del parser è un codice che implementa il PDA.

## 6) Cosa fornisce in output il generatore del lexer?

L'output del generatore del lexer è un codice che implementa il DFA.

```
C:\Users\gianl\OneDrive\Documenti\Uni_Mi_Bicocca\CDL\2_Anno\Attivita_Didattica\PrimoSemestre\Linguaggi_Computabilita\Laboratorio\Laboratorio1\Esercitazione1\materialeLaboratorio2>jflex calc.l
Reading "calc.l"

Warning: Macro "LOGE" has been declared but never used.
Constructing NFA : 67 states in NFA
Converting NFA to DFA :
...........................
32 states before minimization, 19 states in minimized DFA
Old file "Yylex.java" saved as "Yylex.java~"
Writing code to "Yylex.java"
```

## 7) Cosa si intende per bottom – up parser?

Data una frontiera o prodotto dell'albero di derivazione, il bottom – up parser costruisce il parse tree dalle foglie verso i padri fino a giungere la root (simbolo iniziale della CFG). Un esempio di bottom – up parser è il left – right parser.

## 8) LR parser: significato

Legge la stringa da sinistra a destra e cerca nell'input elaborato le sottostringhe che corrispondono al corpo di una produzione partendo da destra e applica quella produzione al contrario.