

Compiler Design Mini Project

Recursive Descent Parser

Using Python

Samriddho Das (119)

Madura Sankar S. (114)

Rini Ravindran (081)

Nanditha (080)



Parsing is the process to determine whether the start symbol can derive the program or not. If the Parsing is successful then the program is a valid program otherwise the program is invalid.

There are generally two types of Parsers:

- **Top-Down Parsers**
- **Bottom-Up Parsers**

Top-Down Parsers

- In this Parsing technique we expand the start symbol to the whole program.
- Recursive Descent and LL parsers are the Top-Down parsers.





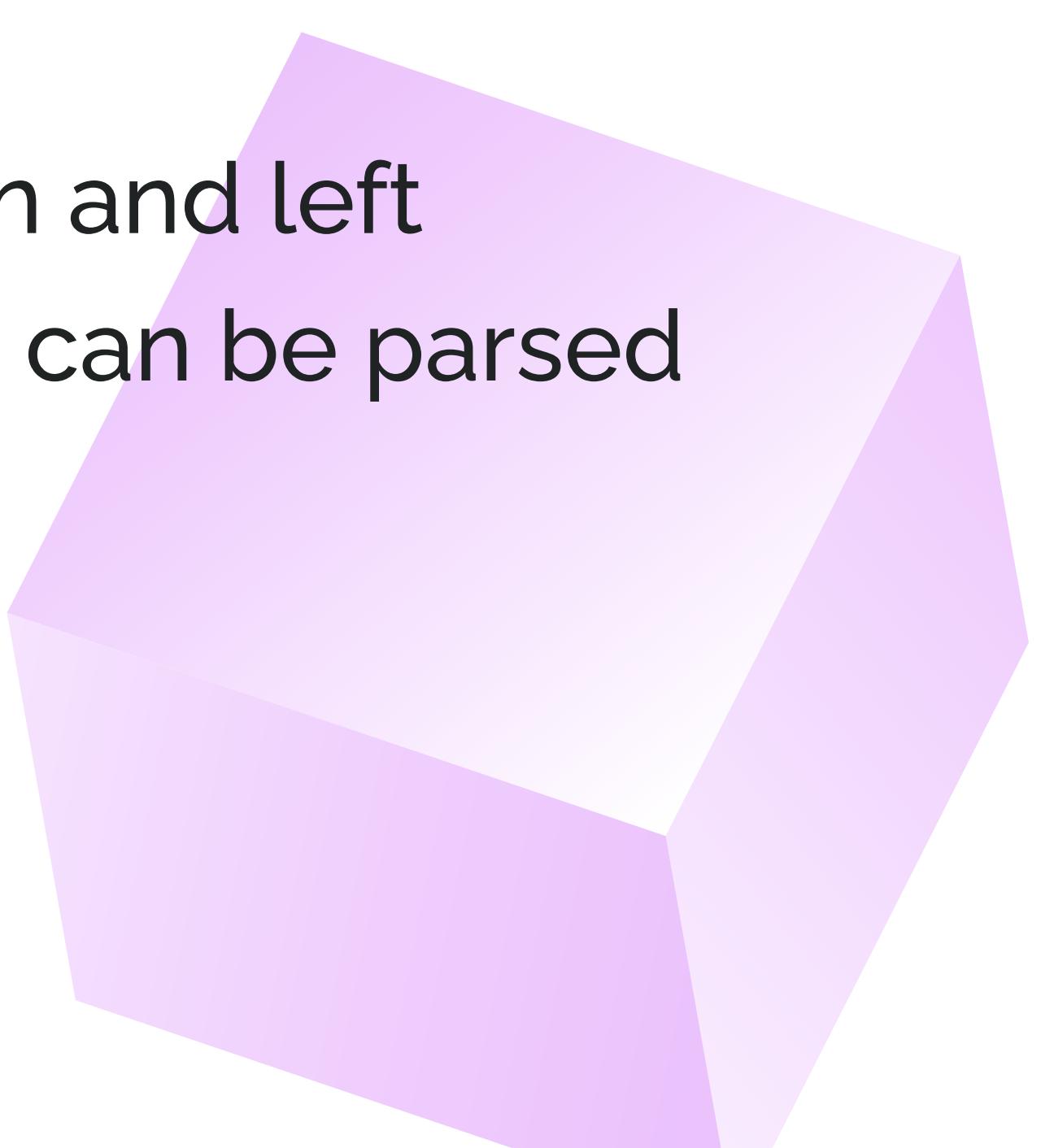
Bottom-Up Parsers

- In this Parsing technique we reduce the whole program to start symbol.
- Operator Precedence Parser, LR(0) Parser, SLR Parser, LALR Parser and CLR Parser are the Bottom-Up parsers.

What is **Recursive Descent Parser**



It is a kind of **Top-Down Parser**. A top-down parser builds the parse tree from the top down, starting with the start non-terminal. A **Predictive Parser** is a special case of **Recursive Descent Parser**, where **no Back Tracking is required**.



By carefully writing a grammar means eliminating left recursion and left factoring from it, the resulting grammar will be a grammar that can be parsed by a recursive descent parser.

GDB online Debugger | Compiler x +

onlinegdb.com

Gmail YouTube Maps

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

151K

Recursive Desent Parsing For following grammar

```
E->TE'  
E'->+TE'/@  
T->FT'  
T'->*FT'/@  
F->(E) /i  
  
Enter the string want to be checked  
  
i+i  
String is accepted
```

...Program finished with exit code 0
Press ENTER to exit console.

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Type here to search

30°C Mostly clear 23:22 21-04-2022

GDB online Debugger | Compiler x +

onlinegdb.com

Gmail YouTube Maps

OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

151K

Recursive Desent Parsing For following grammar

```
E->TE'  
E'->+TE'/@  
T->FT'  
T'->*FT'/@  
F->(E) /i  
  
Enter the string want to be checked  
  
[i+  
string is not accepted
```

...Program finished with exit code 0
Press ENTER to exit console.

About • FAQ • Blog • Terms of Use • Contact Us • GDB
Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Type here to search

30°C Mostly clear 23:23
21-04-2022

Thank you!!!

Our Teammates

- Samriddho Das (RA1911003010119)
- Madura Sankar S. (RA1911003010114)
- Rini Ravindran (RA1911003010081)
- Nanditha (RA1911003010080)

