

CS 160 Compilers

程序代写代做 CS编程辅导



Lecture 0 Introduction to Parsing

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

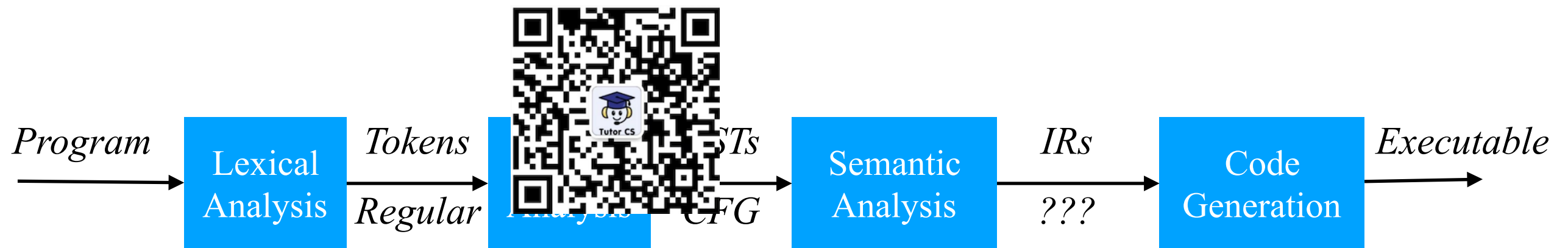
QQ: 749389476

<https://tutorcs.com>

Yu Feng
Fall 2021

A typical flow of a compiler

程序代写代做CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Chomsky hierarchy

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



<https://en.wikipedia.org/wiki/File:Chomsky-hierarchy.svg>

Lexical analysis



- Consider the following program:

```
if x > y
```

```
then 10
```

```
else 8
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

- This program is just a string of characters

QQ: 749389476

```
if x > y\nthen\t10\nelse\t8
```

<https://tutorcs.com>

- Goal: Portion the input string into substrings where the substrings are *tokens*

The role of a parser



Phase	Input	Output
Lexer	String of characters	String of tokens
Parser	String of tokens	Parse tree

- Input: sequence of tokens from the lexer
- Output: parse tree (Abstract Syntax Tree) of the program

QQ: 749389476

<https://tutorcs.com>

Example

程序代写代做CS编程辅导



- Input: Consider the Patina expression: **if** $x > y$ **then** 10 **else** 8
- Parse Input: TOKEN_ID("x") TOKEN_GT
TOKEN_ID("y") TOKEN_THEN TOKEN_INT(10) TOKEN_ELSE
TOKEN_INT(8)

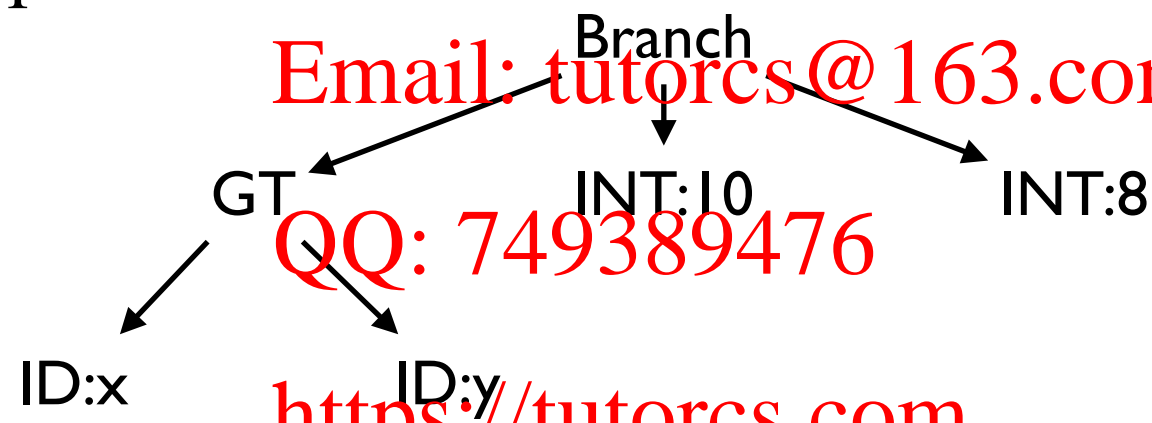
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



The role of a parser



- Not all strings of tokens are valid programs...
- Parser must distinguish between valid and invalid strings of tokens
- What we need:
 - A language for describing valid strings of tokens
 - A method for recognizing if a string of tokens is in this language or not

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Context free grammar (CFGs)



- Programming language constructs have *recursive* structure

- Example: A Patina expression is

Assignment Project Exam Help

- $expression + expression$,

Email: tutorcs@163.com

- **if** $expression$ **then** $expression$ **else** $expression$, ...

QQ: 749389476

- Context free grammars are a natural notation for this recursive structure

<https://tutorcs.com>

CFGs in more detail



- A CFG consists of
 - A set of terminals T
 - A set of non-terminals N
 - A start symbol S (non-terminal)
 - A set of productions: $X \rightarrow Y_1 Y_2 \dots Y_n$

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

where $X \in N$ and $Y_i \in (T \cup N \cup \{\varepsilon\})$

CFGs example



- Recall the earlier CFG of Patina:

$EXPR \rightarrow \text{if } EXPR \text{ then } EXPR \text{ else } EXPR$

| $EXPR + EXPR$

| ID

WeChat: cstutorcs

Assignment Project Exam Help

- Some strings in this language:

ID QQ: 749389476

$IF ID THEN ID ELSE ID$

$ID + ID$

$IF ID THEN ID+ID ELSE ID$

$IF IF ID THEN ID ELSE IF THEN ID ELSE ID$

<https://tutorcs.com>

From derivations to parse trees



- A derivation is a sequence of productions: $S \rightarrow \dots \rightarrow \dots \rightarrow \dots$

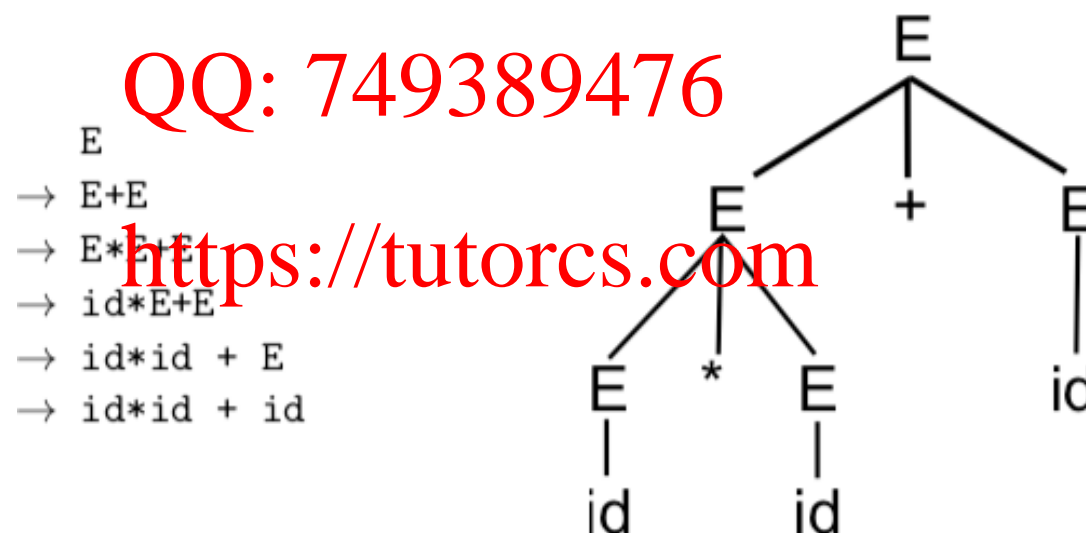
- A derivation can be drawn as a tree

WeChat: cstutorcs

- Start symbol is the tree's root

Assignment Project Exam Help

- For a production $X \rightarrow Y_1 \dots Y_n$ add children $Y_1 \dots Y_n$ to node X



Ambiguity

程序代写代做CS编程辅导

- Consider this grammar



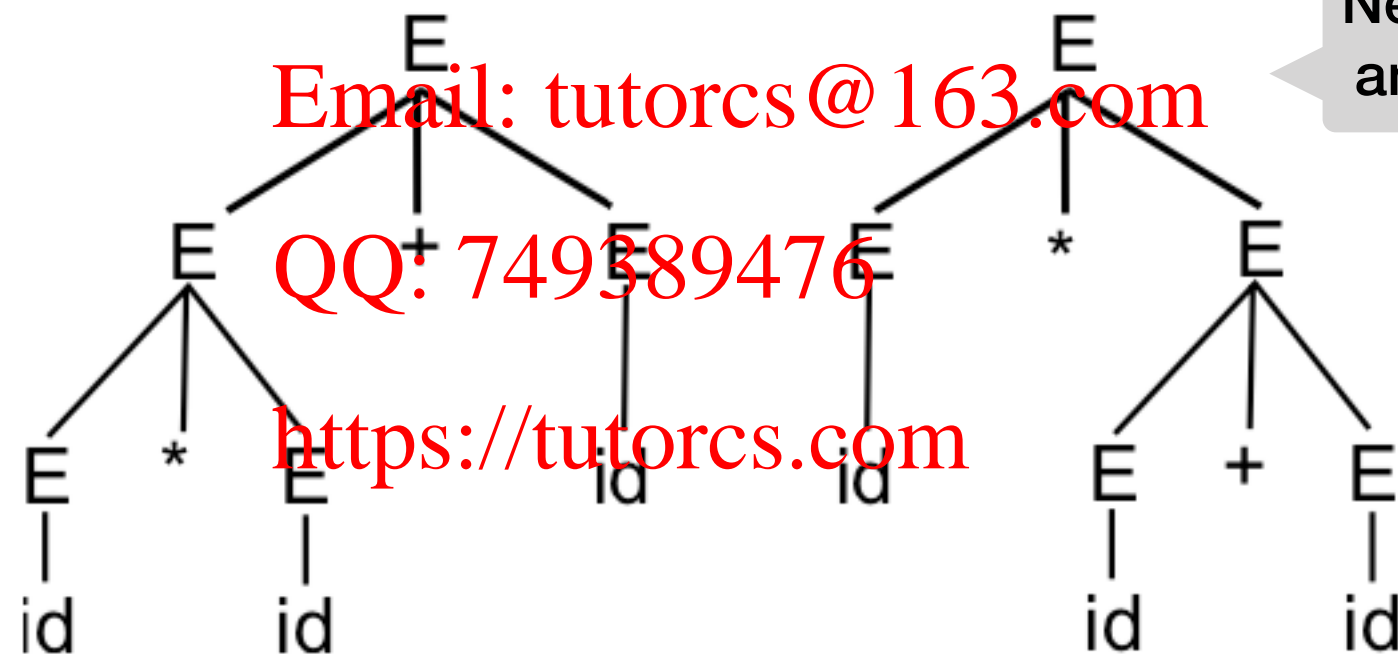
$EXPR \rightarrow E * E$

$| E + E | (E)$

$| id$

WeChat: cstutorcs

- Now, this string $id * id + id$ has two parse trees!



Need Precedence
and Associativity

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

TODOs by next lecture



- Hw2 is out. Please submit by!
- Come to the discussion session if you have questions

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>