

CS 160 Compilers

程序代写代做 CS编程辅导



Lecture 5: Lexical Analysis

WeChat: estutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

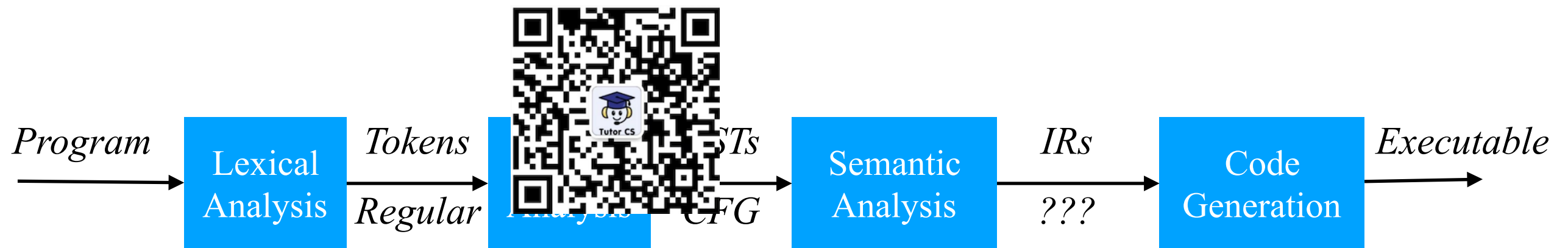
QQ: 749389476

<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

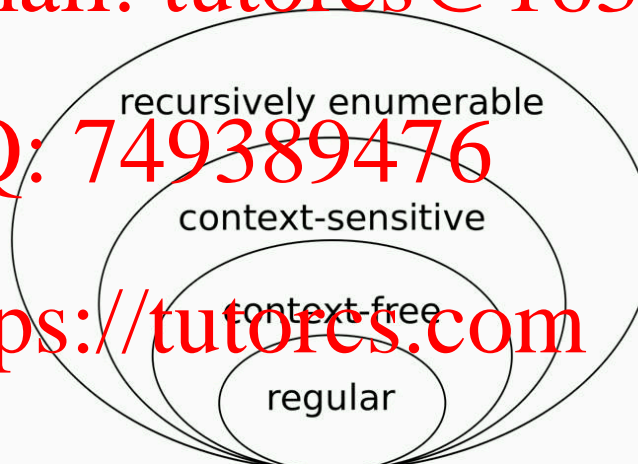
Assignment Project Exam Help

Chomsky hierarchy

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



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

A typical flow of a compiler

程序代写代做CS编程辅导

Source Code
(Character stream)

```
if (b == 0) { a = 1;
```

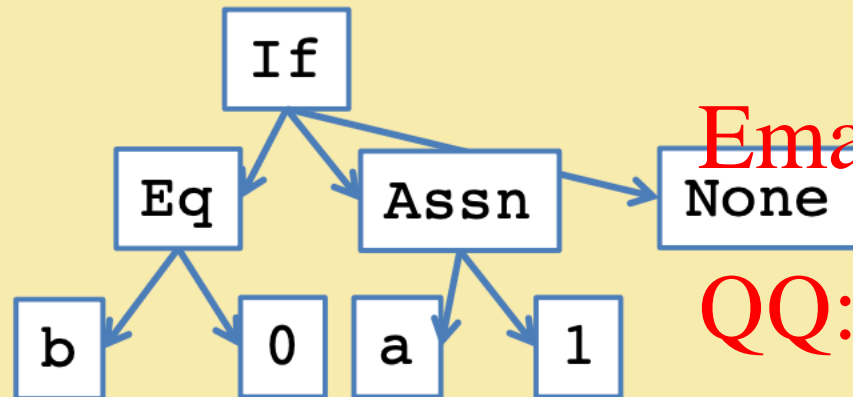


Token stream:

| | | | | | | | | | | | |
|----|---|---|----|---|---|---|---|---|---|---|---|
| if | (| b | == | 0 |) | { | a | = | 0 | ; | } |
|----|---|---|----|---|---|---|---|---|---|---|---|

WeChat: cstutorcs

Abstract Syntax Tree:



Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Intermediate code:

```
11:
    %cnd = icmp eq i64 %b,
    0
    br i1 %cnd, label %12,
    label %13
12:
    store i64* %a, 1
    br label %13
13:
```

Lexical Analysis

Parsing

Analysis &
Transformation

Backend

Assembly Code

```
11:
    cmpq %eax, $0
    jeq 12
    jmp 13
12:
```

Lexical analysis



- Main Question: How to convert a structure to strings

- Analogy: Understanding an English sentence

WeChat: cstutorcs

- First, we separate a string into words

Assignment Project Exam Help

- Second, we understand sentence structure by diagramming the sentence

QQ: 749389476

- Separating a string into words is called *lexing*

<https://tutorcs.com>

- Note that lexing is not necessarily trivial

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*

What is a Token?



- Token is a syntactic category
- Example in English: noun, verbs, adjectives,...
- In a programming language: constants, identifiers, keywords, whitespaces...

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Tokens in Patina



- Tokens correspond to strings
- Identifier: strings of letters, digits and '_' starting with a letter
- Integer: a non-empty string of digits
- Keywords: "let", "if", ...
- Whitespace: a non-empty sequence of blanks, newlines, and tabs

WeChat: dstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

What are tokens for?



- Classify program substrings according to their role

WeChat: cstutorcs

- Output of lexical analysis is a stream of tokens...

Assignment Project Exam Help

- ...which is input to the parser

Email: tutorcs@163.com

- Parser relies on token distinction

QQ: 749389476

- An identifier is treated different than a keyword

<https://tutorcs.com>

Regular language/expressions



- We could specify tokens in many ways

WeChat: cstutorcs

- Regular Languages are the most popular

Assignment Project Exam Help

- Simple and useful theory

Email: tutorcs@163.com

- Easy to understand

QQ: 749389476

- Efficient to implement

<https://tutorcs.com>

Languages

程序代写代做CS编程辅导



- Definition: Let Σ be a set of characters, A language over Σ is a set of strings from characters drawn from Σ

WeChat: cstutorcs

- Alphabet: English characters \Rightarrow Language: English sentences

Assignment Project Exam Help

- Languages are sets of strings

Email: tutorcs@163.com

- Need some notation for specifying which sets we want

QQ: 749389476

- The standard notation for regular languages is regular expressions

<https://tutorcs.com>

Regular expressions



- Atomic Regular Expression

- Single character: $c = \{c\}$

WeChat: cstutorcs

- Epsilon: $\epsilon = \{\epsilon\}$

Assignment Project Exam Help

- Compound Regular Expressions

Email: tutorcs@163.com

- Union: $A+B = \{s \mid s \in A \text{ or } s \in B\}$

QQ: 749389476

- Concatenation: $AB = \{ab \mid a \in A \text{ and } b \in B\}$

<https://tutorcs.com>

- Iteration: $A^* = \bigcup_{i \geq 0} A^i$ where $A^i = A \dots i \text{ times } A$

Regular expressions



- ▶ The **regular expressions** over Σ are the smallest set of expressions

- ▶ ε

WeChat: cstutorcs

- ▶ $'c'$ where $c \in \Sigma$

Assignment Project Exam Help

- ▶ $A + B$ where A, B are regular expressions over Σ

Email: tutorcs@163.com

- ▶ AB where A, B are regular expressions over Σ

QQ: 749389476

- ▶ A^* where A is a regular expression over Σ

<https://tutorcs.com>

- ▶ Regular expressions are simple, but **very useful**

Example: Integers



- Integer: non-empty string of digits.
WeChat: cstutorcs
- $\text{digit} = '0' + '1' + '2' + '3' + '4' + '5' + '6'$
Assignment Project Exam Help
- $\text{integer} = \text{digit digit}^*$
Email: tutorcs@163.com
- Abbreviation: $A^+ = AA^*$
QQ: 749389476
<https://tutorcs.com>

Example: Identifier



- Identifier: strings of letters or digits, starting with a letter

WeChat: cstutorcs

- letter = 'A'+...+'Z'+ 'a'+...+'z'+ '_'

Assignment Project Exam Help

- identifier = letter (letter + digit)*

Email: tutorcs@163.com

QQ: 749389476

- How about (letter* + digit*)?

<https://tutorcs.com>

Example: Whitespace



WeChat: cstutorcs

- Whitespace: a non-empty sequence of blanks, newlines and tabs

- Whitespace = $(\text{' '} + \text{'\n'} + \text{'\t'})^+$

QQ: 749389476

<https://tutorcs.com>

Last example: email



- Consider UCSB cs emails: anyone@cs.ucsb.edu format

- $\Sigma = \text{letters} \cup \{., @\}$

- name = letter⁺

- address = name '@' name '.' name '.' name

<https://tutorcs.com>

TODOs by next lecture

程序代写代做CS编程辅导



- Come to the discussion or office hour if you have questions
- Continue with your good work on HW1

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>