CSEN 1003: Compilers

Tutorial 1 - Introduction and Regular Languages

2/2/2020 - 5/2/2020

This Instructor

Name	Nourhan Ehab Azab
Email	nourhan.ehab@guc.edu.eg
Office	C7.305
Office Hours	Wednesday 5^{th}

Today's Plan

1. Administrivia
2. Motivation and Course Plan
3. Regular Languages
4. Recap

Today's Plan

1. Administrivia
2. Motivation and Course Plan
3. Regular Languages
4. Recap

Grading Scheme and Communication Channels

Quizzes (2/3)	25%
Assignments (2/2)	10%
Midterm	25%
Final	40%

- Course Material: http://met.guc.edu.eg/Courses/CourseEdition.aspx?crsEdId=996
- Piazza Course Page: piazza.com/guc.edu.eg/spring2020/csen1003/home

Today's Plan

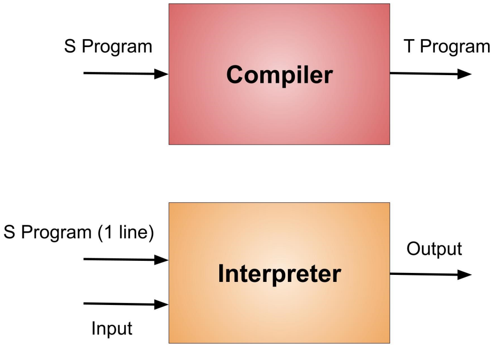
1. Administrivia
2. Motivation and Course Plan
3. Regular Languages
4. Recap

What Do You Expect We Will Do?

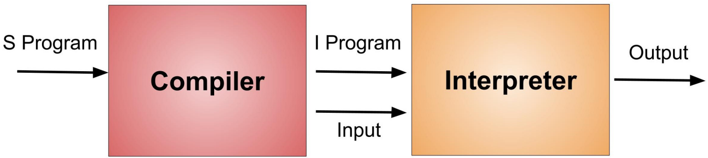
What Do You Expect We Will Do?

What is a Compiler?

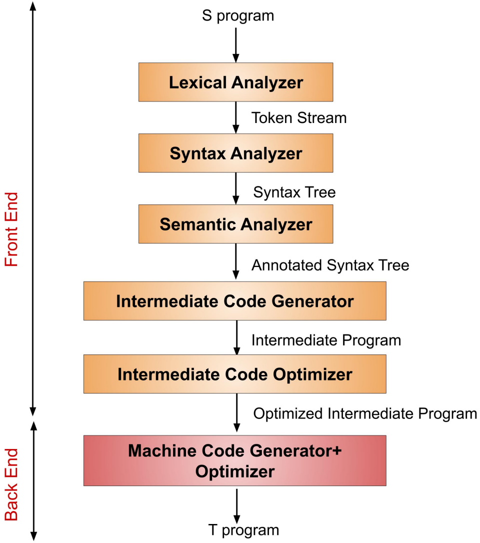
Compilers vs Interpreters



Hybrid Approaches



Overall structure of a Compiler



But Why Study Compiler Construction?

But Why Study Compiler Construction?

1. A compiler can be viewed as a general language processor.

But Why Study Compiler Construction?

1. A compiler can be viewed as a general language processor.
2. Studying compilers will involve studying a lot of useful data structures and algorithms.

But Why Study Compiler Construction?

1. A compiler can be viewed as a general language processor.
2. Studying compilers will involve studying a lot of useful data structures and algorithms.
3. It is a very successful realization of computer science.

Today's Plan

1. Administrivia
2. Motivation and Course Plan
3. Regular Languages
4. Recap

Lexical Analysis and Regular Languages

- The lexical analyzer needs to scan the input to identify valid tokens.
- For programming languages, tokens can be expressed by regular languages.

Regular Expressions - Exercise 1-3

Example

Write a regular expression for each of the following regular languages. The alphabet $\Sigma = \{0, 1\}$.

a $L_1 = \{w \mid w \text{ begins with a 1 and ends with a 0}\}$.

b $L_2 = \{w \mid w \text{ contains the substring 0101}\}$.

c $L_3 = \{w \mid \text{every odd position of } w \text{ is a 1 }\}$

d $L_4 = \{w \mid w \text{ contains at least two 0 and at most one 1}\}$

DFA and NFA Design - Exercise 1-2

Example

Give the DFA and NFA state diagrams recognizing each of the following regular languages. The alphabet $\Sigma = \{0, 1\}$.

- a $L_1 = \{w \mid w \text{ begins with a 1 and ends with a 0}\}$.
- b $L_2 = \{w \mid w \text{ contains the substring 0101}\}$.
- c $L_3 = \{w \mid \text{every odd position of } w \text{ is a 1 }\}$
- d $L_4 = \{w \mid w \text{ contains at least two 0 and at most one 1}\}$

From Regular Expressions to NFAs - Thomson's Construction

Example

Convert the regular expression you wrote in 1-3 (a) to an NFA.

From Regular Expressions to NFAs to DFAs

Example

Convert the NFA you got to a DFA.

Today's Plan

1. Administrivia
2. Motivation and Course Plan
3. Regular Languages
4. Recap

Points to Take Home!

1. Compilers, Interpreters, and Hybrid Approaches.
2. Stages of Compilation.
3. Recap of Regular Languages.
    - Regular Expressions.
    - From Regular Expressions to NFAs.
    - From NFAs to DFAs.

Next Week: Lexical Analysis!