# 03-60-214 Computer Languages, Grammars, and Translators

Jianguo Lu
School of Computer Science
University of Windsor

## Instructors

- Professor Jianguo Lu
  - Office: Lambton Tower 5111
  - Phone: 519-253-3000 ext 3786
  - Email: jlu at uwindsor
  - Web: http://cs.uwindsor.ca/~jlu/214

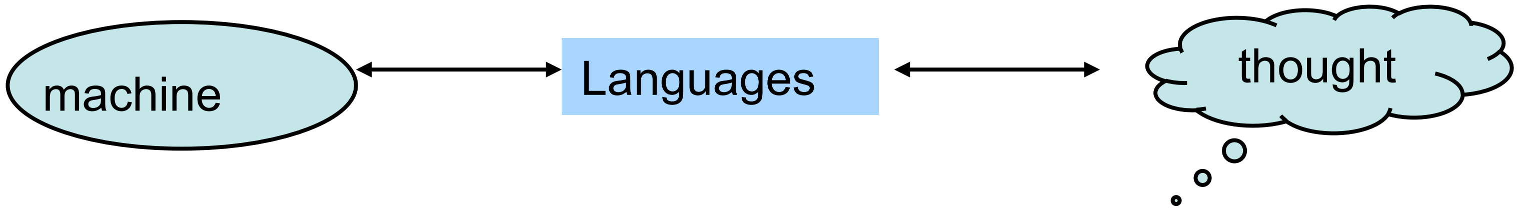
- GAs

## Course description

- Computer languages, grammars, and translators
- Prerequisite: 60-100, **03-60-212**
  - Assignments will be implemented in Java.
- Objective
  - Knowledge of computer languages and grammars
  - Able to analyze programs written in various languages
  - Able to translate languages
- Contents
  - Regular expressions, finite automata and language recognizers;
  - Context free grammar;
  - Languages parsers.
- Software tools used
  - Programming language: Java (including tokenizer, regular expression package)
  - Lexical analyzer: JLex,
  - Parser generator: JavaCup

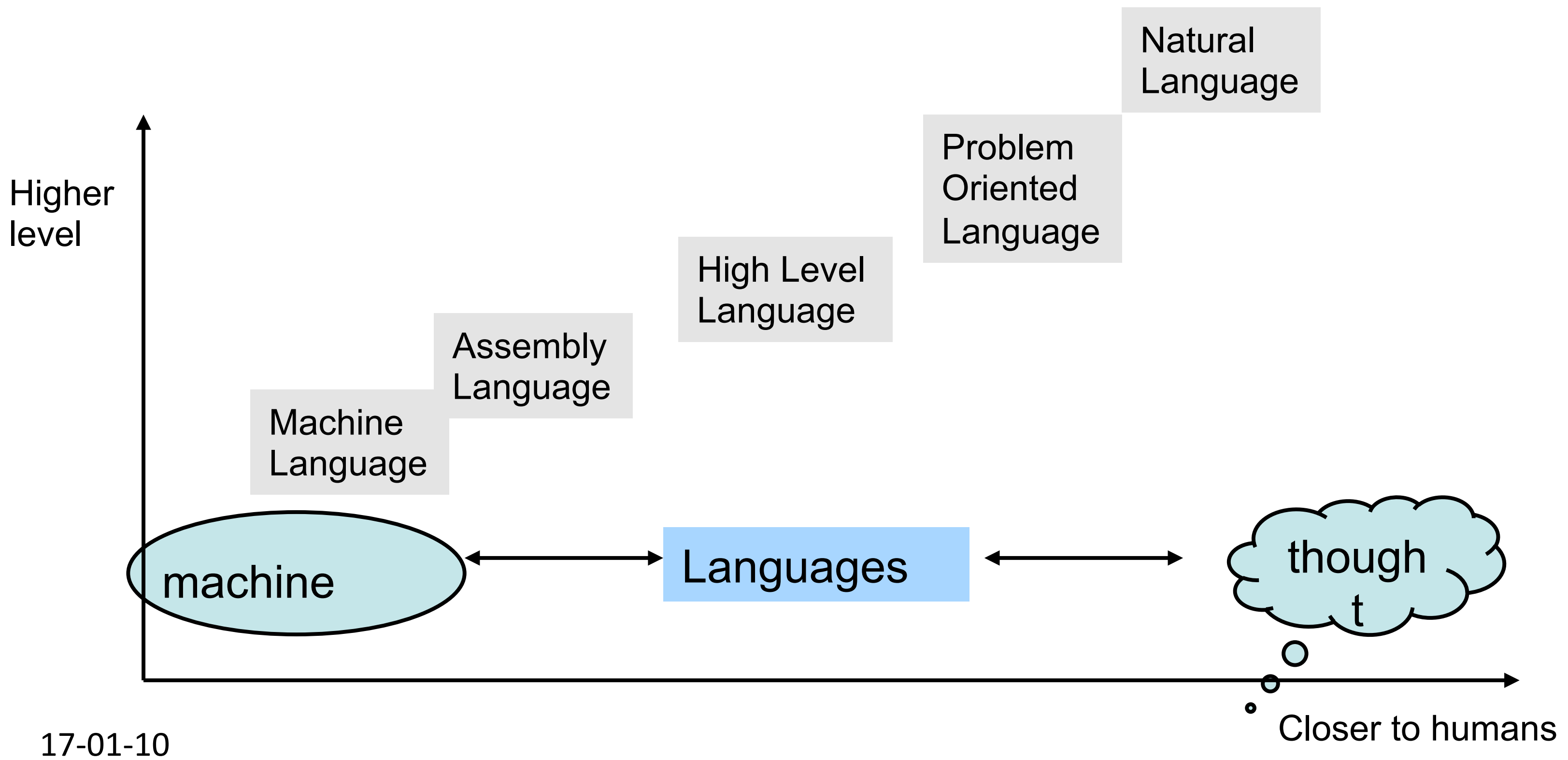
## What is Language

- Language: “any system of formalized symbols, signs, etc., used or conceived as a means of communication.”
  - Communicate: to transmit or exchange thought or knowledge.
- Programming language: communicate between a person and a machine
  - Programming language is an intermediary



## Hierarchy of (programming) languages

- Machine language;
- Assembly language: mnemonic version of machine code;
- High level language: Java, C#, Pascal;
- Problem oriented;
- Natural language.



## Grammar

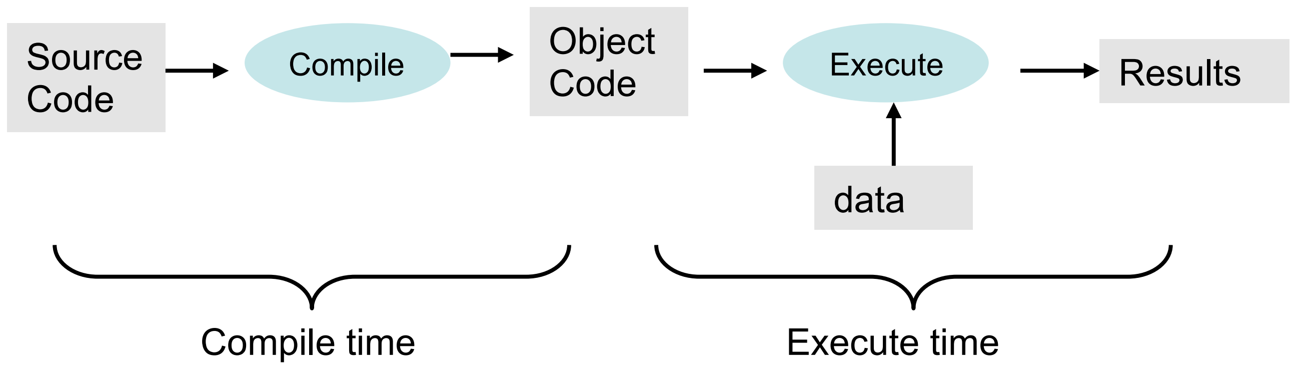
- Grammar: the set of structural rules that govern the composition of sentences, phrases, and words in any given natural language. --wikipedia
- Formal grammar: rules for forming strings in a formal languages
- Computer language grammar: rules for forming tokens, statements, and programs.
- Different layers of grammar:
  - Regular grammar (for words, tokens)
  - Context free grammar (for sentences, programs)
  - …

## Language Translators

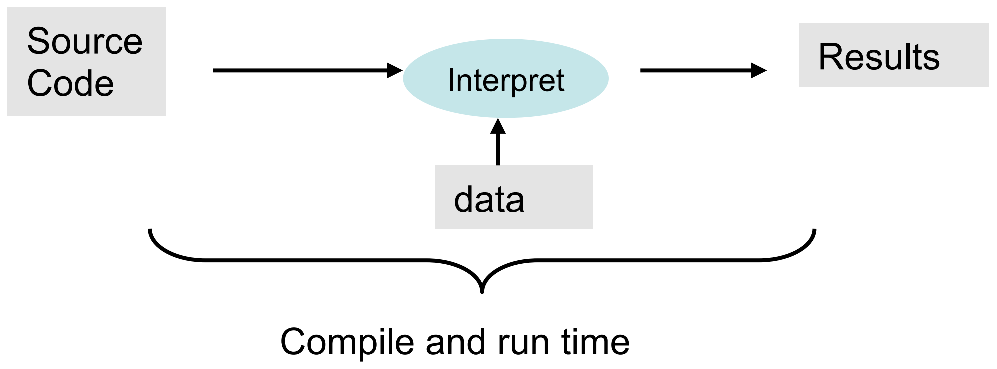
- Translator: Translate one language into another language (e.g., from C++ to Java)
  - A generic term.
- For high level programming languages (such as java, C):
  - *Compiler*: translate high level programming language code into host machine's assembly code and execute the translated program at run-time.
  - *Interpreter:* process the source program and data at the same time. No equivalent assembly code is generated.
- Assembler: translate an assembly language to machine code.

## Compiler and Interpreter

- Compiler



- Interpreter



## How does a compiler work

- A compiler performs its task in the same way how a human approaches the same problem
- Consider the following sentence:

  "Write a translator"

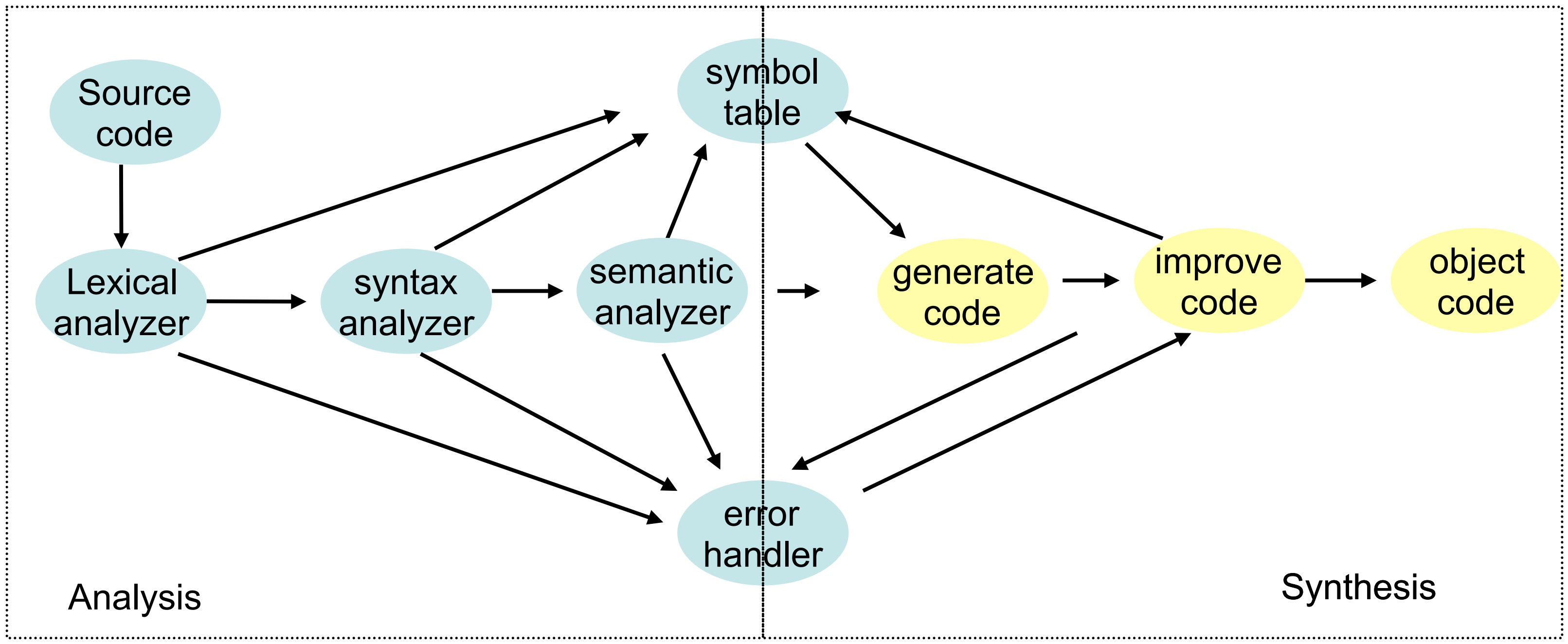
- We all understand what it means. But how do we arrive at the conclusion?

## The process of understanding a sentence

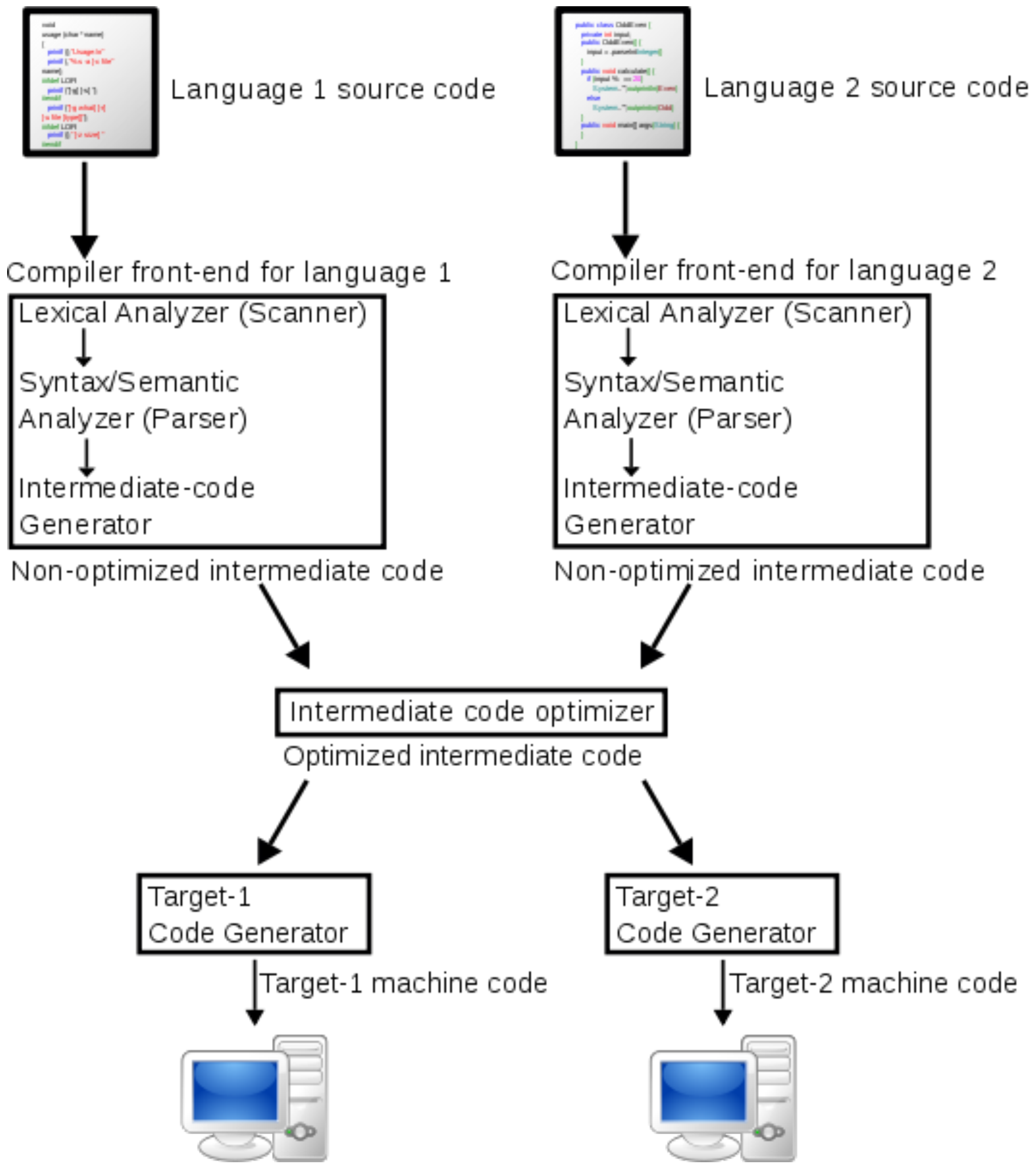
1. Recognize characters (alphabet, mathematical symbols, punctuations).
   - 16 explicit (alphabets), 2 implicit (blanks)
2. Group characters into logical entities (words).
   - 3 words.
   - Lexical analysis
3. Check the words form a structurally correct sentence
   - "translator a write" is not a correct sentence
   - Syntactic analysis
4. Check that the combination of words make sense
   - "dig a translator" is a syntactically correct sentence
   - Semantic analysis
5. Plan what you have to do to accomplish the task
   - Code generation
6. Execute it.

"Write a translator"

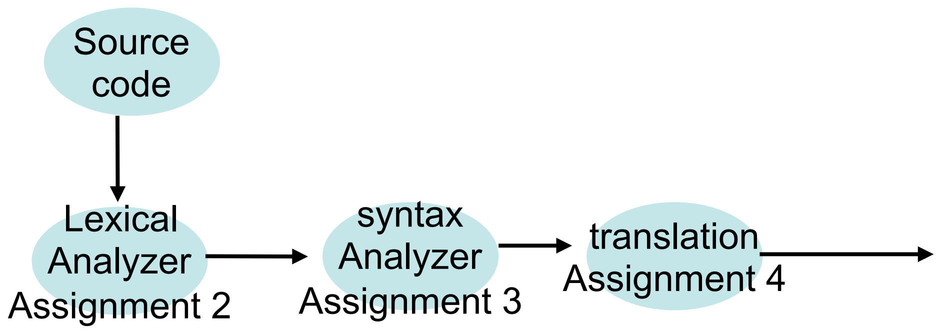
## The structure (phases) of a compiler



- Front end (analysis): depend on source language, independent on machine
  - This is what we will focus (mainly the blue parts).
- Back end (synthesis): dependent on machine and intermediate code, independent of source code.


Language 1 source code
Language 2 source code
Compiler front-end for language 1
Lexical Analyzer (Scanner)
Syntax/Semantic Analyzer (Parser)
Intermediate-code Generator
Non-optimized intermediate code
Compiler front-end for language 2
Lexical Analyzer (Scanner)
Syntax/Semantic Analyzer (Parser)
Intermediate-code Generator
Non-optimized intermediate code
Intermediate code optimizer
Optimized intermediate code
Target-1 Code Generator
Target-1 machine code
Target-2 Code Generator
Target-2 machine code

## Assignments overview



- Our focus is the front end
  - Automated generation of lexical analyzer
  - Automated generation of syntax analyzer

## Assignments (28%)

- Assignment 1 (warm up): Regular expression in Java (5%)
  - Use StringTokenizer in JDK to tokenize the strings.
  - Use regular expressions to match strings
  - You will see the difficulty to analyse programs without advanced tools such as Jlex and Java Cup.
- Assignment 2 (6%)
  - Use JLex to build a lexical analyzer for tiny program
- Assignment 3 (6%)
  - Manually write a recursive descendent parsing
  - Use JavaCup to generate a parser
- Assignment 4 (6%)
  - Translate the tiny program to Java and actually run it.
- Assignment 5 (5%)
  - Manually write a recursive descendent parsing

## Why this course

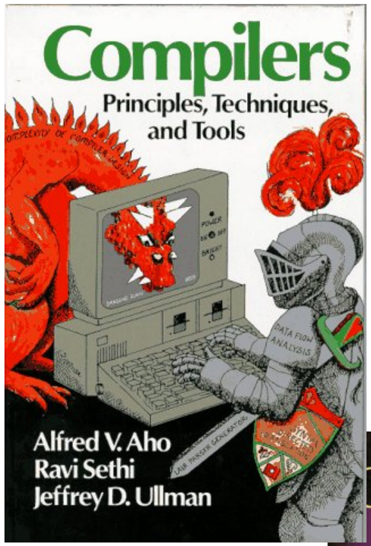
- Every university offers this type of courses.
- Skills learnt
  - write a parser
  - process programs
  - re-engineer and migrate programs
    - Migrate from C++ to C#
    - …
  - process data
    - Xml, web logs, social networks, …

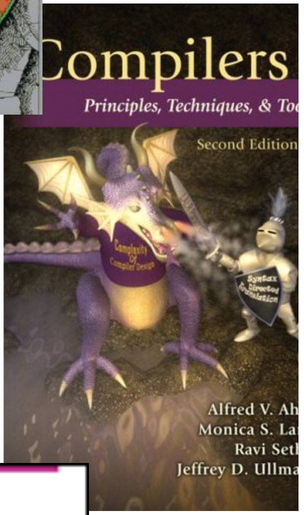
## Why this course (cont.)

- Theoretical aspects of programming
- The science of developing a large program
  - Not handcraft the program
- How to
  - define whether a program is valid
  - Determine whether a program is valid
  - Generate the program

## Course materials

- Reference books (not required)
  - Compilers: Principles, Techniques, and Tools (2nd Edition) by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman (Aug 31, 2006)
  - Or A.V. Aho, R. Sethi, and J.D. Ullman, *Compilers: Principles, Techniques, and Tools*, Addison-Wesley, 1988. (Chapter 1-5)
  - John R. Levine, Tony Mason, and Doug Brown, Lex & Yacc, O'Reilly & Associates, 1992.
- Online manual
  - JavaCup, www.cs.princeton.edu/~appel/modern/java/CUP/
  - JLex, www.cs.princeton.edu/~appel/modern/java/JLex/





## Marking scheme

| | | | |
|---|---|---|---|
| Exams | 72% | Midterm 1 | 12 % |
| | | Midterm 2 | 20 % |
| | | Final | 40% |
| Assignments | 28% | assignment 1 | 5% |
| | | assignment 2 | 6% |
| | | assignment 3 | 6% |
| | | assignment 4 | 6% |
| | | Assignment 5 | 5% |
| **Total** | **100%** | | **100%** |

## Assignments (28%)

- Assignment submission
- All assignments must be completed individually.
  - All the assignments will be checked by a copying detection system.

- Academic dishonesty
  - Discussion with other students must be limited to general discussion of the problem, and must never involve examining another student's source code or revealing your source code to another student.

## Exams (72%)

- Two midterm exams
- Final exam
- Close book exams
- Exams cover topics in *lectures*
  - Class attendance is important.
- Exams will cover topics in assignments
  - Finishing assignments is also important.
- **What if you missed exam (s)**
  - A missed exam will result in a mark of zero. The only valid excuse for missing an exam is a documented medical emergency.

**Student Medical Certificate [1]**
**Faculty of SCIENCE**

**A. TO BE COMPLETED BY THE STUDENT:**

I, ______________________________ , hereby authorize Dr. _________________________________ to provide the following information to the University of Windsor and, if required, to supply additional information to support my request for special academic consideration for medical reasons. My personal information is being collected under the authority of the *University of Windsor Act 1962* and will be used for administrative and academic record-keeping, academic integrity purposes, and the provision of services to students. For questions in connection with the collection of this information, the Associate Dean of my Faculty may be contacted at 519-253-3000.

___________________________________ _____________________ _____________________

Signature Student No. Date

**B. TO BE COMPLETED BY THE PHYSICIAN:**

1. I hereby certify that I provided health care services to the above-named student on _____________________________________________.
**(insert date(s) student seen in your office/clinic)**
2. The student could not reasonably be expected to complete academic responsibilities for the following reason (in broad terms):
_________________________________________________________________________________
3. This is an acute / chronic problem for this student.
4. Date(s) during which student claims to have been affected by this problem:

_________________________________________________________________________________________

5. Unable to complete academic responsibilities for:

| | |
|---|---|
| 24 hours | 2 days |
| 3 days | 4 days |
| 5 days | Other (please indicate) ___________________________ |

6. If the student is permitted to continue his/her course of study, is the medical problem likely to recur and affect his/her studies again? Yes No

Reason: _________________________________________________________________________________

**PHYSICIAN VERIFICATION**

Name: (please print) _______________________________ Registration No. __________________________
Signature: _________________________________________ Telephone No. ___________________________
Address: _______________________________________________________________________________________ (stamp, business card, or letterhead acceptable)

**PLEASE RETAIN COPY FOR THE PATIENT'S CHART.** ***Note:*** **Cost of certificate to be paid by student.**

[1] This form has been adapted, with permission, from the University of Windsor Faculty of Law Student Medical Certificate and the University of Western Ontario Student Medical Certificate.

## Labs

- You can attend all the lab sections
- Labs are mainly used to help you with the assignments

## Interaction with professor

- During lectures
- During labs
- During office hours: Wednesday 1:00-3:00
- Emails: jlu at uwindsor
  - Subject line must start with “214”
  - Example:   Subject: 214--About assignment 1
  - Mails without proper subject may not be read (and hence not answered)
  - Attach detailed error messages
  - Write your name in the email

## Web contents

- Course plan;
- Slides for lectures;
- Assignment descriptions;
- Links to tools, manuals, tutorials;
- List of marks;
- Announcements;

## Important note

- Please note that no student is allowed to take a course more than two times without permission from the Dean.

## Front desk job at Leddy Library

- Contact Grace Liu at gliu@uwindsor.ca
- Starting next week

## Introduction to grammar

Jianguo Lu
School of Computer Science
University of Windsor

## Formal definition of language

- A language is a set of strings
  - English language

    {“the brown dog likes a good car”, … …}

    {sentence | sentence written in English}
  - Java language {program |program written in Java}
  - HTML language {document |document written in HTML}
- How do you define a language?
- It is unlikely that you can enumerate all the sentences, programs, or documents

## How to define a language

- How to define English
  - A set of words, such as brown, dog, like
  - A set of rules
    - A sentence consists of a subject, a verb, and an object;
    - The subject consists of an optional article, followed by an optional adjective, and followed by a noun;
    - … …
  - More formally:
    - Words ={a, the, brown, friendly, good, book, refrigerator, dog, car, sings, eats, likes}
    - Rules:
      1) SENTENCE → SUBJECT VERB OBJECT
      2) SUBJECT → ARTICLE ADJECTIVE NOUN
      3) OBEJCT → ARTICLE ADJECTIVE NOUN
      4) ARTICLE → a | the| EMPTY
      5) ADJECTIVE → brown | friendly | good | EMPTY
      6) NOUN → book| refrigerator | dog| car
      7) VERB → sings | eats | likes

# Derivation of a sentence

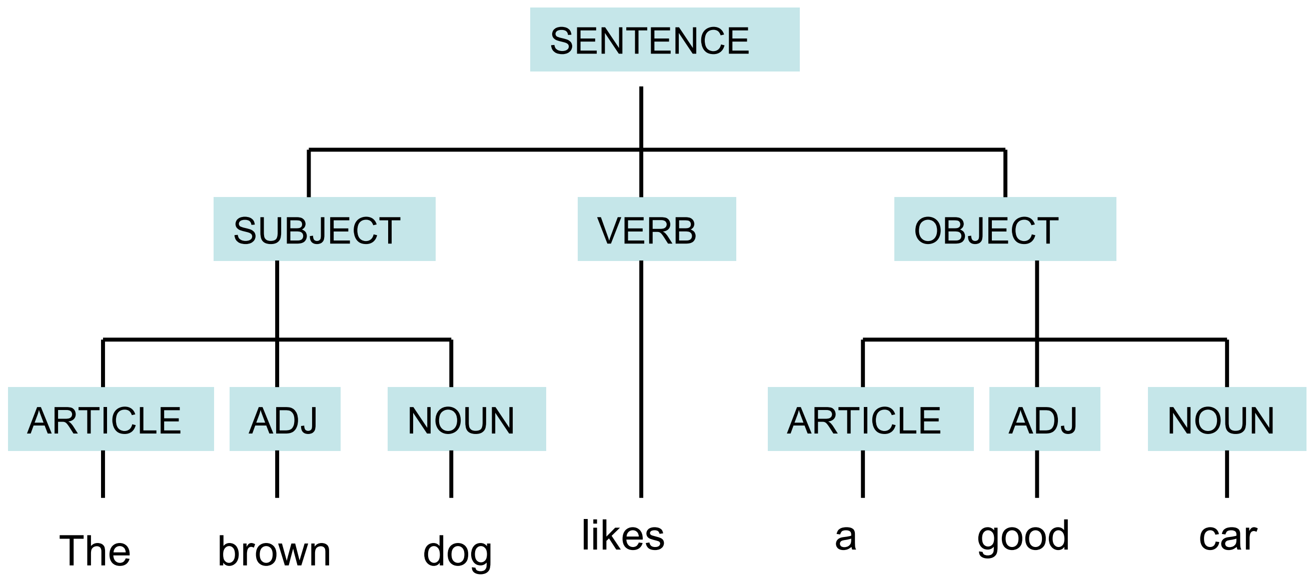
- Derivation of a sentence “*the brown dog likes a good car*”

  SENTENCE
  →SUBJECT VERB OBJECT
  →ARTICLE ADJECTIVE NOUN VERB OBJECT
  →the brown dog VERB OBJECT
  →the brown dog likes ARTICLE ADJECTIVE NOUN
  →the brown dog likes a good car

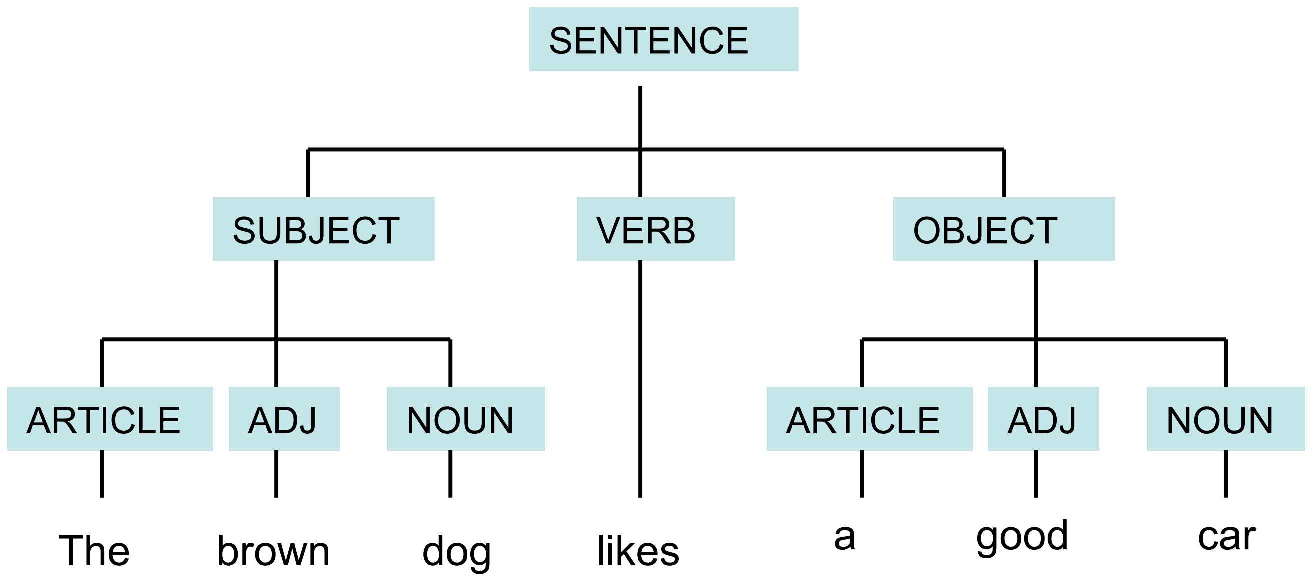
- Rules:
  1) SENTENCE → SUBJECT VERB OBJECT
  2) SUBJECT → ARTICLE ADJECTIVE NOUN
  3) OBEJCT → ARTICLE ADJECTIVE NOUN
  4) ARTICLE → a | the| EMPTY
  5) ADJECTIVE → brown | friendly | good | EMPTY
  6) NOUN → book| refrigerator | dog| car
  7) VERB → sings | eats | likes

## The parse tree of the sentence

Parse the sentence: “the brown dog likes a good car”
The top-down approach



## Top down and bottom up parsing



## Types of parsers

- Top down
  - Repeatedly rewrite the start symbol
  - Find the left-most derivation  of the input string
  - Easy to implement
- Bottom up
  - Start with the tokens and combine them to form interior nodes of the parse tree
  - Find a right-most derivation of the input string
  - Accept when the start symbol is reached
- Bottom up is more prevalent

## Formal definition of grammar

- A grammar is a 4-tuple G = (Σ, N, P, S)
  - Σ is a finite set of terminal symbols;
  - N is a finite set of nonterminal symbols;
  - P is a set of productions;
  - S (from N) is the start symbol.
- The English sentence example
  - Σ ={a, the, brown, friendly, good, book, refrigerator, dog, car, sings, eats, likes}
  - N={SENTENCE, SUBJECT, VERB, NOUN, OBJECT, ADJECTIVE, ARTICLE}
  - S={SENTENCE}
  - P={rule 1) to rule 7) }

## Recursive definition

- Number of sentence can be generated:

| ARTICLE | ADJ | NOUN | VERB | ARTICLE | ADJ | NOUN | sentences |
|---|---|---|---|---|---|---|---|
| 3 * | 4* | 4* | 3* | 3* | 4* | 4* | = 6912 |

- How can we define an infinite language with a finite set of words and finite set of rules?
- Using recursive rules:
  – SUBJECT/OBJECT can have more than one adjectives:
    1) SUBJECT → ARTICLE ADJECTIVES NOUN
    2) OBEJCT → ARTICLE ADJECTIVES NOUN
    3) ADJECTIVES→ ADJECTIVE | ADJECTIVES ADJETIVE

  – Example sentence:

    "the good brown dog likes a good friendly book"

## Chomsky hierarchy

- Noam Chomsky hierarchy is based on the form of production rules
- General form

  $\alpha_1\ \alpha_2\ \alpha_3 \ldots \alpha_n \rightarrow \beta_1\ \beta_2\ \beta_3 \ldots \beta_m$

  Where $\alpha$ and $\beta$ are from terminals and non terminals, or empty.
- Level 3: Regular grammar
  - Of the form $\alpha \rightarrow \beta$ or $\alpha \rightarrow \beta_1\ \beta_2$
  - n=1, and $\alpha$ is a non terminal.
  - $\beta$ is either a terminal or a terminal followed by a nonterminal
  - RHS contains at most one non-terminal at the right end.
- Level 2: Context free grammar
  - Of the form $\alpha \rightarrow \beta_1\ \beta_2\ \beta_3 \ldots \beta_m$
  - $\alpha$ is non terminal.
- Level 1: Context sensitive grammar
  - n<m. The number of symbols on the lhs must not exceed the number of symbols on the rhs
- Level 0: unrestricted grammar

## Context sensitive grammar

- Called context sensitive because you can construct the grammar of the form
  - $A\alpha B \rightarrow A\ \beta\ B$
  - $A\alpha C \rightarrow A\ \gamma\ B$
- The substitution of $\alpha$ depending on the surrounding context A and B or A and C.

## Review

- Languages
- Language translators
  - Compiler, interpreter
  - Lexical analysis
  - Parser
    - Top down and bottom up
- Grammars
  - Formal definition, Chomsky hierarchy
  - Regular grammar
  - Context free grammar