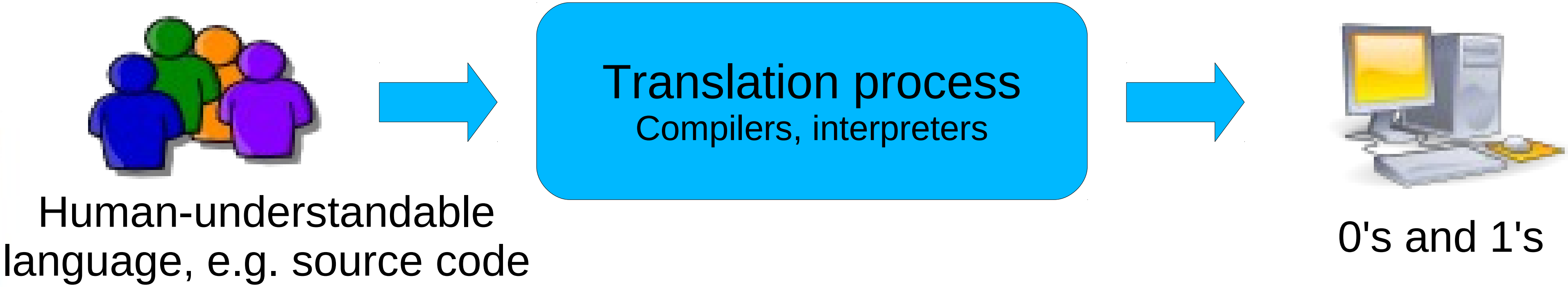
Languages and Operating Systems

Nasser Giacaman
SoftEng 206: Software Engineering Design

Outline for the first few weeks

- Intro:
  - Languages, compilers, interpreters
- Linux
  - Command line, BASH
- Test

Outline for the first few weeks



Human-understandable language, e.g. source code

Translation process
Compilers, interpreters

0's and 1's

- As humans, we have languages that we understand..
- But machines ultimately only understand 0's and 1's, eg:
  - 1 means yes, 0 means no
  - 00 means add, 01 means subtract, 10 means divide, 11...
  - 010110011010 means ???

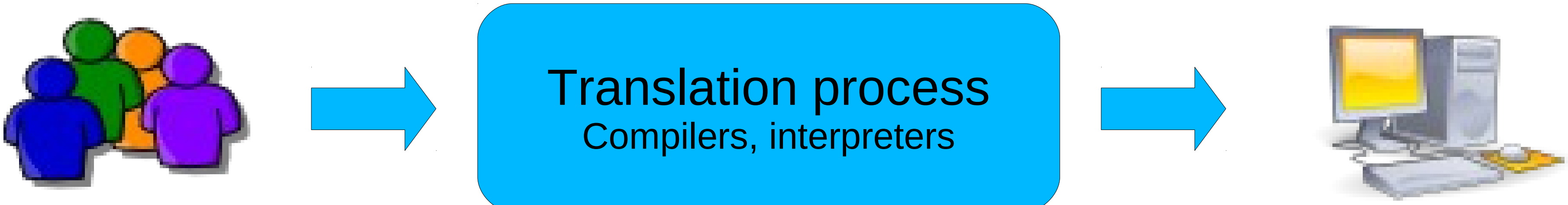
Consider the following human example

- Mr Okonomiyaki only reads & writes Japanese
- Mrs Falafel only reads & writes Arabic
- How can they communicate together? They'll need the help of a third person:
  - Interpreter... at “conversation time” (ie. Runtime), the interpreter translate one sentence at a time from Japanese to Arabic
  - Compiler... beforehand (ie. Compile time), the translator compiles a complete Arabic letter/email from the Japanese document

Consider the following human example

- What's the advantage of a compiler?
- What's the advantage of an interpreter?
- For example:
  - Re-running the program? (Re-interpreting vs already compiled)
  - Execution time: “native language” vs interpreting
  - Development time: making changes to interpreted vs compiled languages
  - Memory space? (The interpreter itself is a program)
  - Optimisation?

Back to machines...



Translation process

Compilers, interpreters

Human-understandable language, e.g. source code

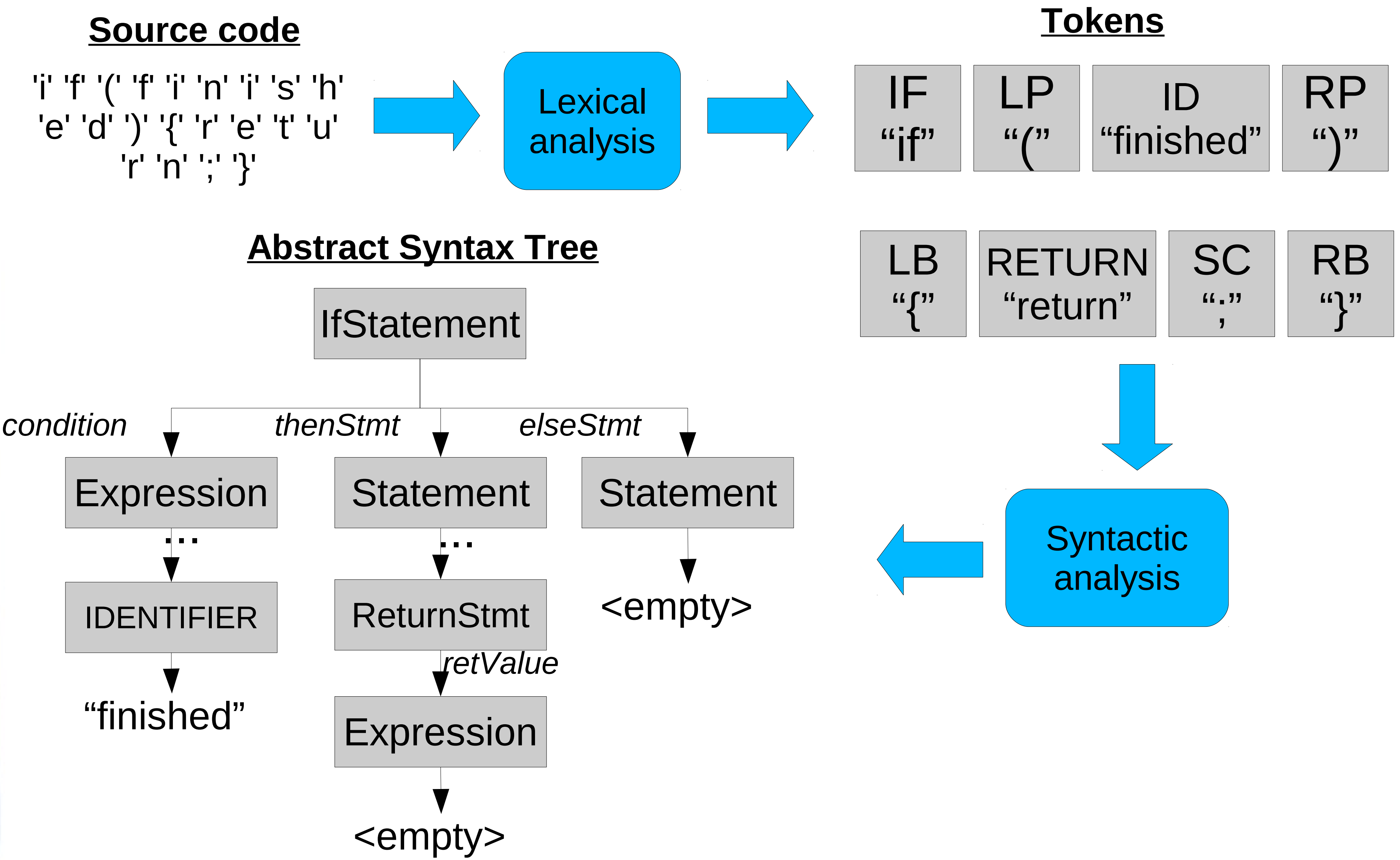
0's and 1's

- **Compiler:** translates the *complete* source program into machine code, creating a compiled version. Only then can the program be executed once the translation is complete.
- **Interpreter:** processes the source by translating and executing the source *one line at a time*.

So what does the interpreter/compiler do?

- Before translating, it “parses” the source code to make sure the following make sense:
  - Lexical analysis: It parses the “source code” into tokens to make sure correct words/symbols (e.g. “if”, “while”, “{“, etc) are being used
  - Syntactic analysis: It parses the tokens to ensure they confer to the correct “grammar” of the source language (e.g. IfStatement → “if” “(“ Expression “)” Statement
- Following this parsing, a tree is produced, where the code is presented in a meaningful manner

Parsing



Source code

'i' 'f' '(' 'f' 'i' 'n' 'i' 's' 'h'
'e' 'd' ')' '{' 'r' 'e' 't' 'u'
'r' 'n' ';' '}'

Lexical analysis

Tokens

IF "if"
LP "("
ID "finished"
RP ")"
LB "{"
RETURN "return"
SC ";"
RB "}"

Syntactic analysis

Abstract Syntax Tree

IfStatement

condition

Expression

…

IDENTIFIER

"finished"

thenStmt

Statement

…

ReturnStmt

retValue

Expression

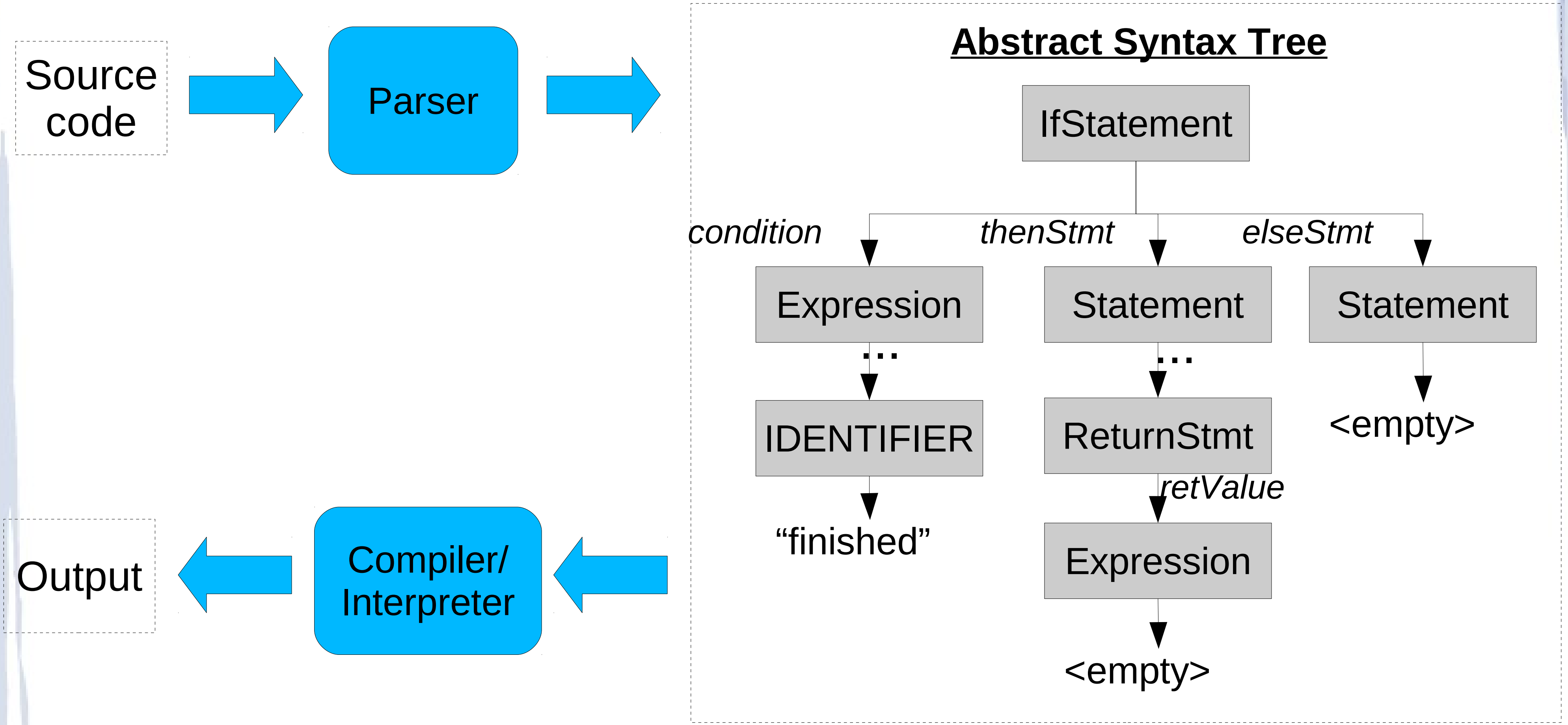
<empty>

elseStmt

Statement

<empty>

Back to interpreters and compilers



Software and hardware

- So what is a program? A sequence of instructions doing some task .. source code is that sequence of instructions “in action”.. ie the program is the living execution of source code on the CPU
- The interpreter/compiler is also a program! Just like any other program, it executes, performs some task, and itself is written using source code like any other program
  - The purpose of the compiler/interpreter is to take your instructions (i.e. the source code) and execute it

Operating system

- What is the operating system? It is ultimately a program too. The OS however, has responsibility of managing the hardware and managing other programs
- Windows
- Mac OS X
- UNIX ($$)
- Linux
  - Open-source
  - Free

Linux

- UNIX-like
- Who uses Linux?
  - The 10 most powerful computers in the world (TOP500)
  - 90% of the TOP500 computers
  - Most servers
  - Android: The world's #1 smartphone platform is Linux
  - In this department, we have a 16-core and 64-core system, both running Linux
  - I'm using Linux right now!
- It's free and easy to install

Kernel and distribution

- There is one Linux “kernel”
  - This is the actual “brains” of the operating system, the “real operating system” that manages processes, memory, devices, I/O, etc
- Linux distributions
  - Includes the Linux kernel, along with a collection of other applications (e.g. media players, graphics apps, word editor, file browser, etc)
  - 100s of different distros out there!
    - Ubuntu, OpenSUSE, Fedora, …
- Desktop environments
  - The graphical component (windows, toolbars, icons, etc)
    - e.g. KDE or Gnome in Linux, Aqua in Mac OS X, Luna in Windows XP, Aero in Windows 7

How to get Linux?

- In the labs: reboot (Ubuntu is installed)
- Live CDs and installation CDs
  - Download online, or from ECE's IT support page
- VirtualBox
  - Easy, don't need to repartition your existing system (but won't run as fast as having a real installation)