



Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

# Introducing Formal Languages and Compilers Laboratory

Alessandro Barengi   Ettore Speziale   Michele Tartara

Politecnico di Milano



# Contents

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

**1** Introduction

**2** Compiler Structure

**3** Advice

**4** Bibliography



# Contents

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

## 1 Introduction

## 2 Compiler Structure

## 3 Advice

## 4 Bibliography



# Topics

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

In this 5 lessons we will see:

- how theoretical concepts (e.g. regular expressions) are exploited in compiler development
- how a compiler is internally organized and how it works
- how to modify a simple compiler

Some concepts can be applied in **everyday work**.



# Exam

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

The lab is  $\frac{1}{5}$  of the exam score:

- you need **to pass** the lab exam in order to pass the whole exam
- you need at least  **$\frac{15}{30}$**  to pass the lab test

Usually the lab exam is performed before the theory exam.  
You can consult anything you want during the exam:

- except your classmates and your laptops/phones



# Contents

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

1 Introduction

2 Compiler Structure

3 Advice

4 Bibliography



# Basic Assumptions

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barenghi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

This is a 4-th year course, so we require:

- a good knowledge of C language
- usage of compiler-related tools (e.g. `gcc`, `make`, ...)
- your brain



# A Tiny and Nice Compiler I

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barenghi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

Compiler purpose is:

- translating a program written with language  $L_0$  into a **semantically equivalent** program expressed with language  $L_1$

A compiler is organized like a pipeline:

- each stage applies transformation to the input program producing an output program





# A Tiny and Nice Compiler II

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

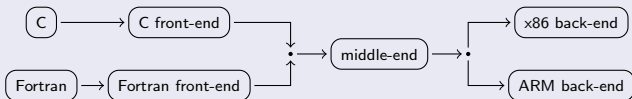
Compiler  
Structure

Advice

Bibliography

A simple compiler contains at least three stages:

## Simple compiler structure



Different stages for different purposes:

- front-end** abstract from the hardware
- middle-end** abstract from both high-level language and hardware
- back-end** abstract from the high-level language



# Front-end

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

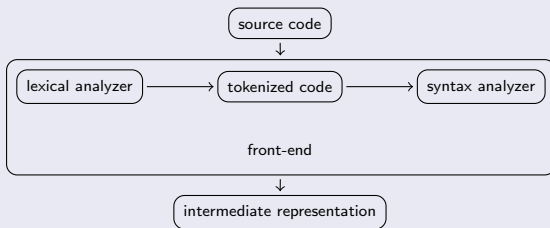
Compiler  
Structure

Advice

Bibliography

Front-end purpose is to translate code into a *intermediate form*.

## Front-end structure



Main actions:

- recognize language constructs
- find syntax error



# Back to Real World: GCC

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

Many front-ends:

- most of them target the *TREE* language

Common lowering to intermediate representation:

- *GIMPLE* and *GIMPLE-SSA* languages

At last:

- translation to *RTL* language
- back-ends emit native instructions

The dark side:

- language hooks



# Contents

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

1 Introduction

2 Compiler Structure

3 Advice

4 Bibliography



# Think First

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

We will see very few-concepts:

- tokens
- statements
- control structures
- ...

You already know *how to use* them:

- you only need to understand how to *recognize* and *compile* them

Many statements are just a variation of a common idiom:

- *syntactic sugar* around a concept



# UNIX is your friend

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Bareghi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

Every UNIX-derived OS contains a lot of compiler-related tools:

- to automate compilers development
- to automate tedious tasks

Few will works on compilers, almost all, soon or later, will find a tedious task:

- count the occurrences of a pattern
- substitute a parametric sentence with another
- ...

Tools (grep, sed, awk) can automate your work!



# Contents

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography

1 Introduction

2 Compiler Structure

3 Advice

4 Bibliography



# Bibliography

Introducing  
Formal  
Languages  
and Compilers  
Laboratory

Alessandro  
Barengi,  
Ettore  
Speziale,  
Michele  
Tartara

Introduction

Compiler  
Structure

Advice

Bibliography



Formal Languages and Compilers Group.  
Formal Languages and Compilers – CorsiOnline.  
<http://corsi.metid.polimi.it>, 2010.