Compilers

Simon Hua

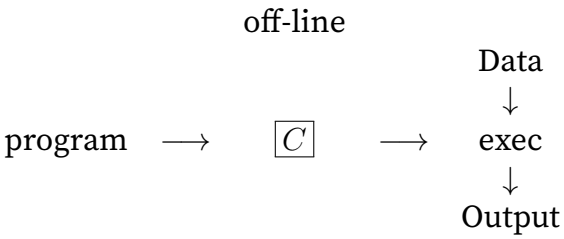
June 2021

Contents

0 Introduction

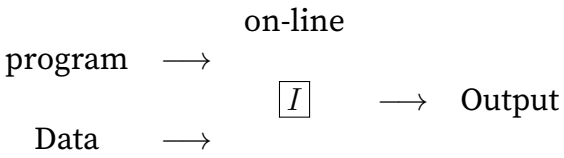
1 Introduction

- Compilers



> 1954 IBM develops the 704
> software > hardware
> "Speedcoding"
>
> - 10-20x slower
> - 300 bytes = 30% memory

- Interpreters



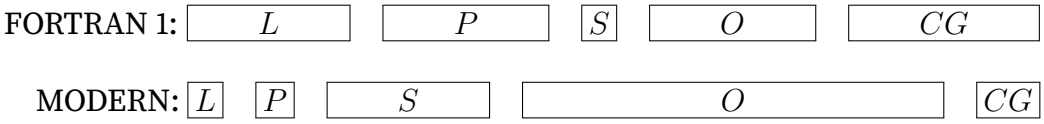
> FORTRAN 1(Formulas Translated)
> 1954-1957
> 1958 50% program in FORTRAN 1

2 Structure of Compiler

5 phases

1. Lexical Analysis: divides program text into "words" or "tokens".
2. Parsing: diagramming sentences.
3. Semantic Analysis: try to understand "meaning". (hard)
   Compilers perform limited senmantic analysis to catch inconsistencies.
   → Programming Languages define strict rules to avoid such ambiguities.

4. Optimization: Antomatically modify prgrams so that they
   → Run faster
   → Use less space
   → Reduce power consumption...
5. Code Generation(Code Gen)
   → Produces assembly code.(usually)
   → A translation int another language.(Analgous to human translation)



3 The Economy of Programming Languages

Question

1. Why are there so many Programming Languages?
   Application domians have distinctive / conflicting needs.

Scientific Computing	→ Good Float Points → Good Arrays → Parallelism	FORTRAN
Business Application	→ Persistence → Report Generation → Data Analysis	SQL
Scientific Computing	→ Control of Resources → Real TimeConstraints	C/C++

2. Why are there new programming languages?
   Claim: **Programmer training** is the dominant cost for a Programming Languages
   (a) widely-used Languages are slow to change.
   (b) Easy to start a new language. ⟶ Productivity > Training Cost
   (c) Languages adopted to fill a void.

   New languages tend to looks like old languages because of the Claim
   → Reducing programming training, like Java vs C++.
3. What is a good programming languages?
   There is no universally accepted metric for language design.

0

The Cool Programming Language

1 Cool Overview