CS4430/7430
Compilers

Prof. Bill Harrison

Spring 2019

Today's Lecture

- **What is this course about?**
  - Programming Language Implementation
    - Overview of implementation styles
      - Interpreters, compilers, formal semantics…
    - High-level view of compiler structure
      - Lexing, parsing, code generation & optimization
- **Administrivia**: grading, textbook, syllabus, …

What is a Programming Language?

```
class Foo {
      String name; int age; Widget w;
      int alg1 (int a, Widget w) { … }
                    …
      void algk (String n, Widget w) { … }
}
```

- Syntax for describing data and associated algorithms
- And, there are many such syntax:
  - Java, C++, ML, Scheme, Haskell, Prolog, Perl, …

Language implementation

- After you have typed in a program, what have you got?

c	l	a	s	s		p	u	b	l	i	c		F	o	o		{	…

- This sequence of characters must be given some “meaning” or definition to be useful
  - Translation into machine code, JVM code,…
  - Evaluation by a program in another high-level programming language
  - Mathematical specifications of some kind

Varieties of Language Definition

- Mathematical (aka “denotational”) semantics
  - Precise language definition
  - Suitable for proving properties of programs
- Interpreter
  - An “evaluator” program for the new language
  - Usually written in another, existing high-level PL
  - Relatively easy to write, but
    - Doesn’t run as fast as possible
- Compilers
  - Translate programs into “stand-alone executables”
  - Efficiency of executable is usually the biggest concern
    - take a long time to write,
    - are notoriously tricky to get correct,
    - are large and difficult to maintain
    - Gnu GCC-1750 (version 1.0) C++ compiler has **278,949** lines of code in **168** separate files

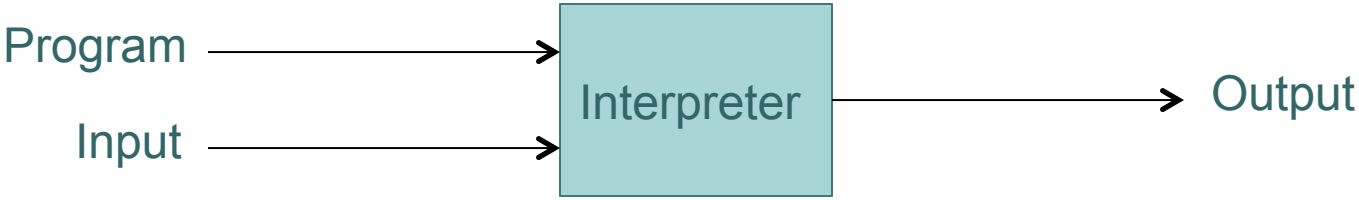
What is a compiler?

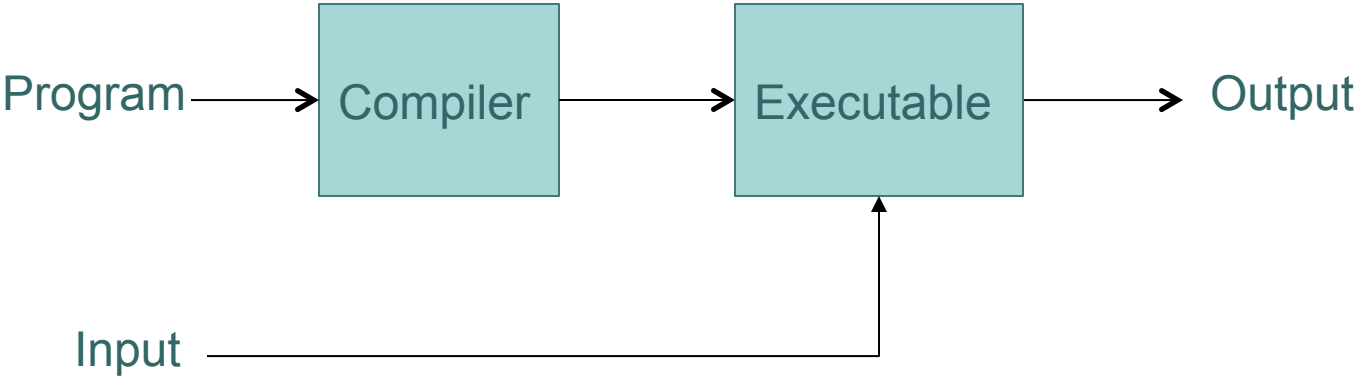
- A translator from one formal language (the “source”) to another (the “target”).
- Also, the source language is generally more human-friendly (i.e., “higher-level”) than the target language.
- Examples?

So why study compilers?

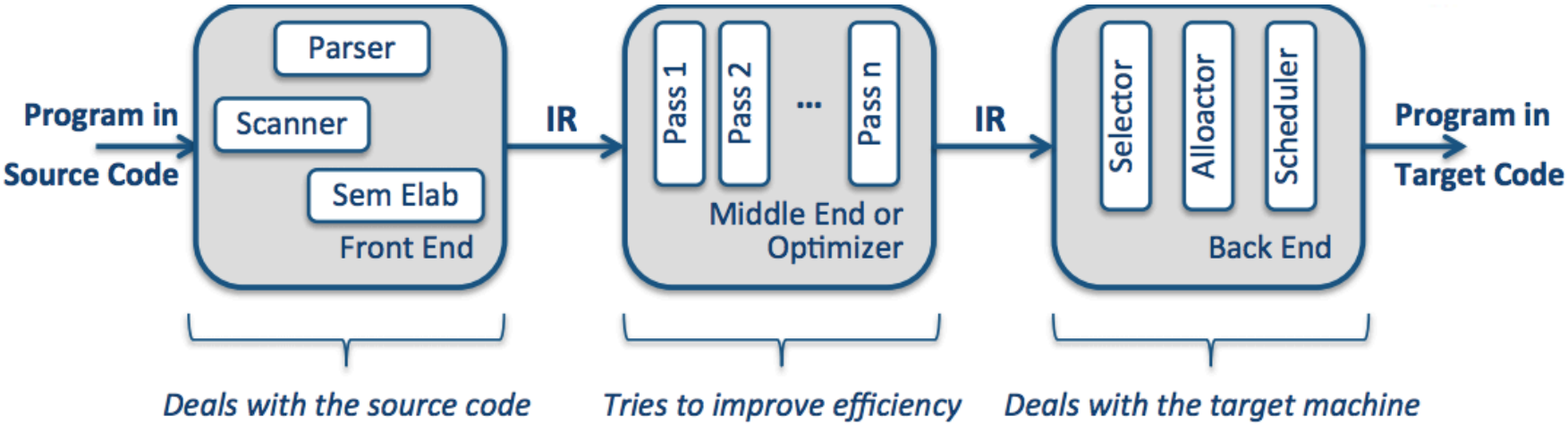
- To make us better programmers!
- Compilers are a great case-study in software engineering.
- Formal languages are everywhere–e.g., C, Java, Python, etc., but also HTML, LaTeX, protocols, APIs, file formats, etc.
- Compilers are fascinating blend of the theoretical and the practical—and they’re fun to write.

Interpreters vs. Compilers



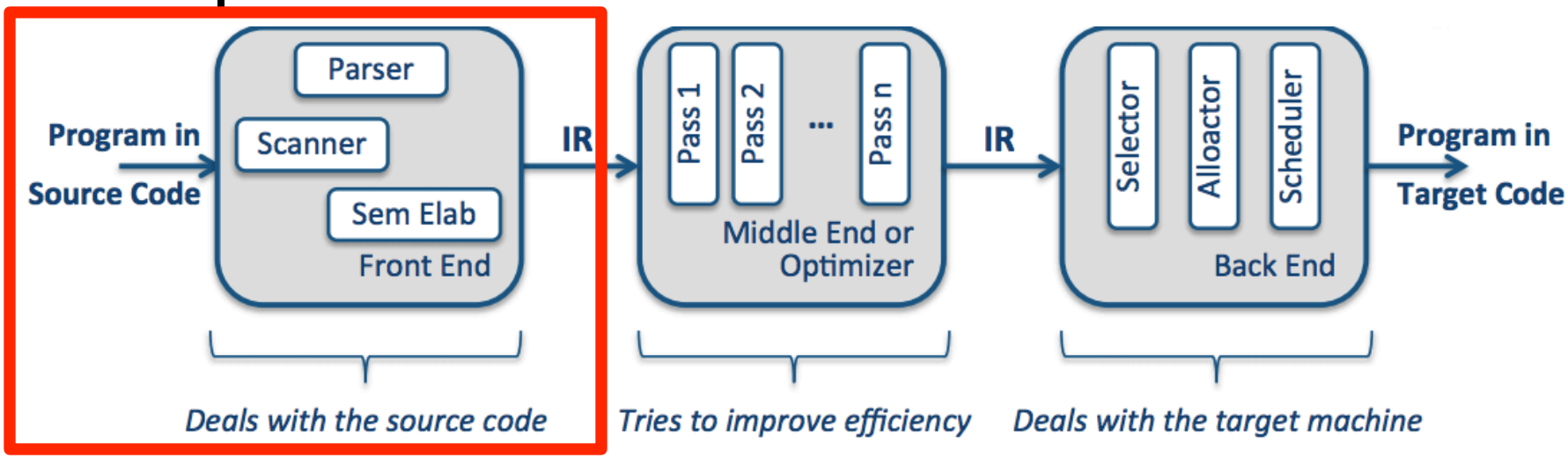


Inside a Compiler



*graphic © Keith Cooper & Linda Torczon.

Inside a Compiler



Front end

- Lexical analysis (the lexer or scanner)
- Parsing
- Semantic analysis

Front End: Lexical Analysis

ascii form

c	l	a	s	s		p	u	b	l	i	c		F	o	o		{		i	n	t	...

lexer

symbolic form

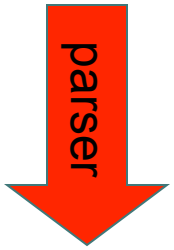
class	public	name(“Foo”)	left-brack	type-int	…

Key Concept: regular expressions

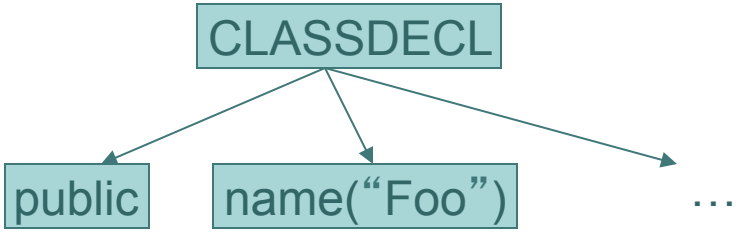
Front End: Parsing

symbolic form

class | public | name(“Foo”) | left-brack | type-int



abstract syntax tree



Key Concept: “Backus-Naur form” grammars (BNF)

Front End: Semantic Analysis

For example, type checking and violations of scope rules:

```
string x = “abc”;
int y = 2;
int z = x + y;
```

```
int x = 1;
{
        int y = 2;
}
int z = x + y;
```

The Front End: Semantic Analysis

For example, type checking and violations of scope rules:

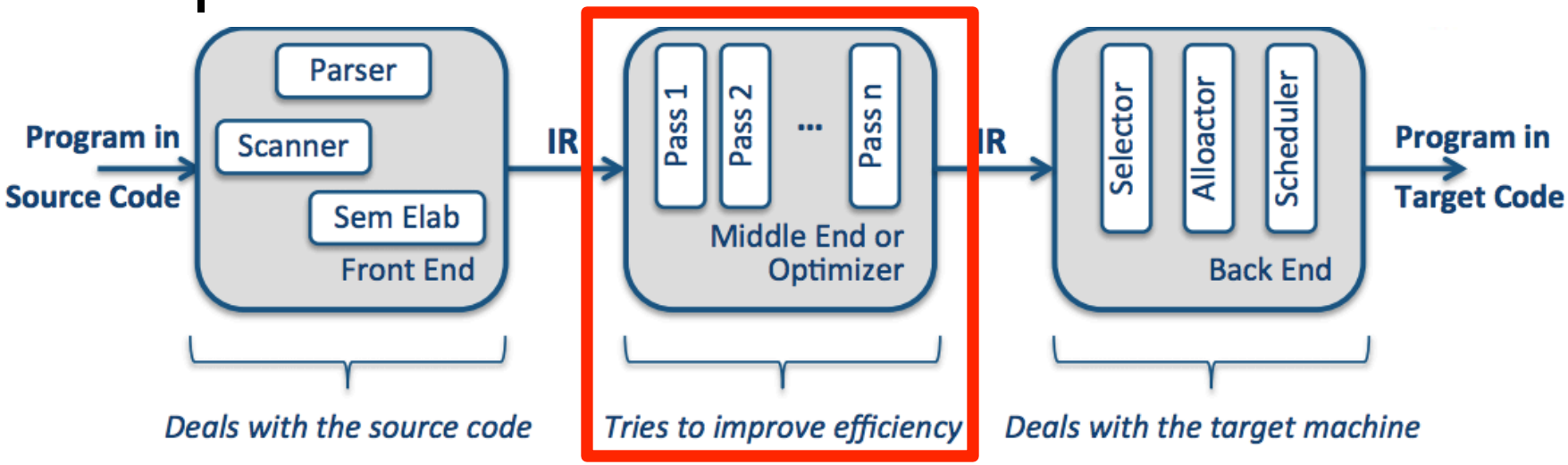
```
string x = “abc”;
int y = 2;
int z = x + y;
```

Error! Attempting to add an int to a string.

```
int x = 1;
{
        int y = 2;
}
int z = x + y;
```

Error! “y” is not in scope.

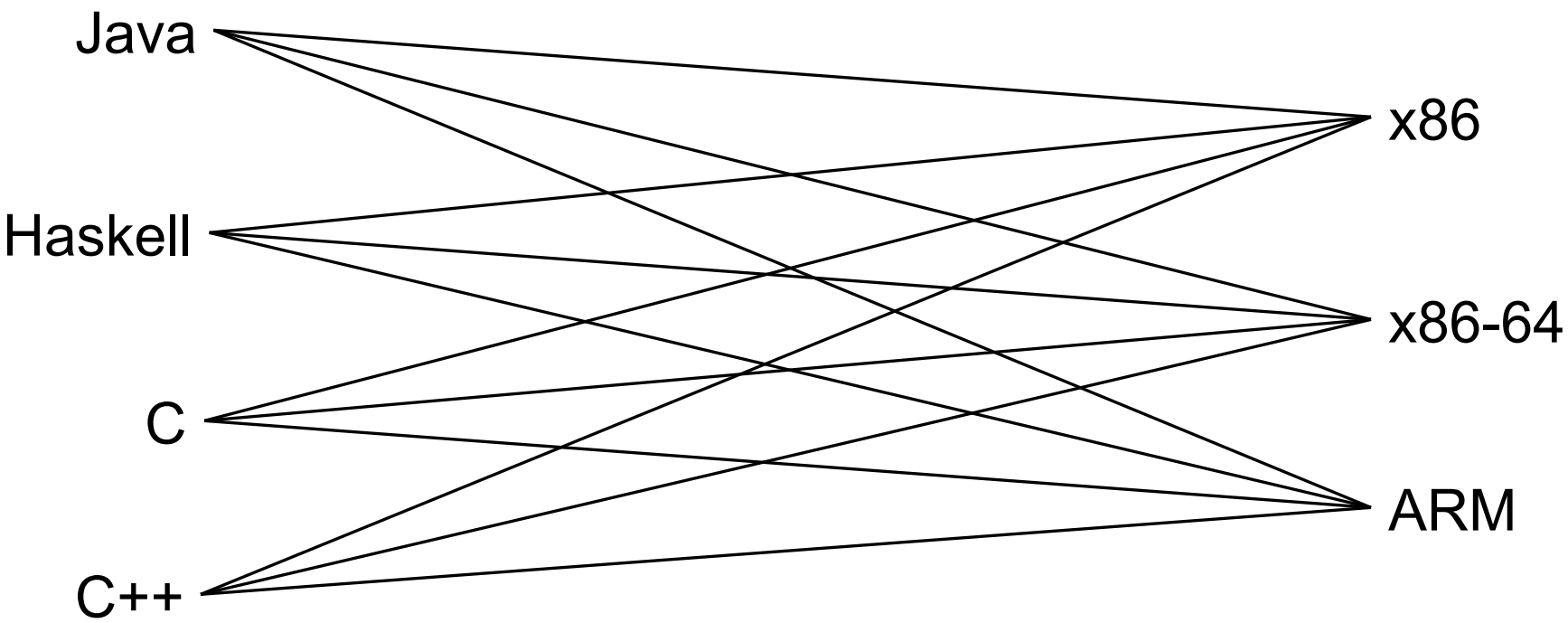
Inside a Compiler



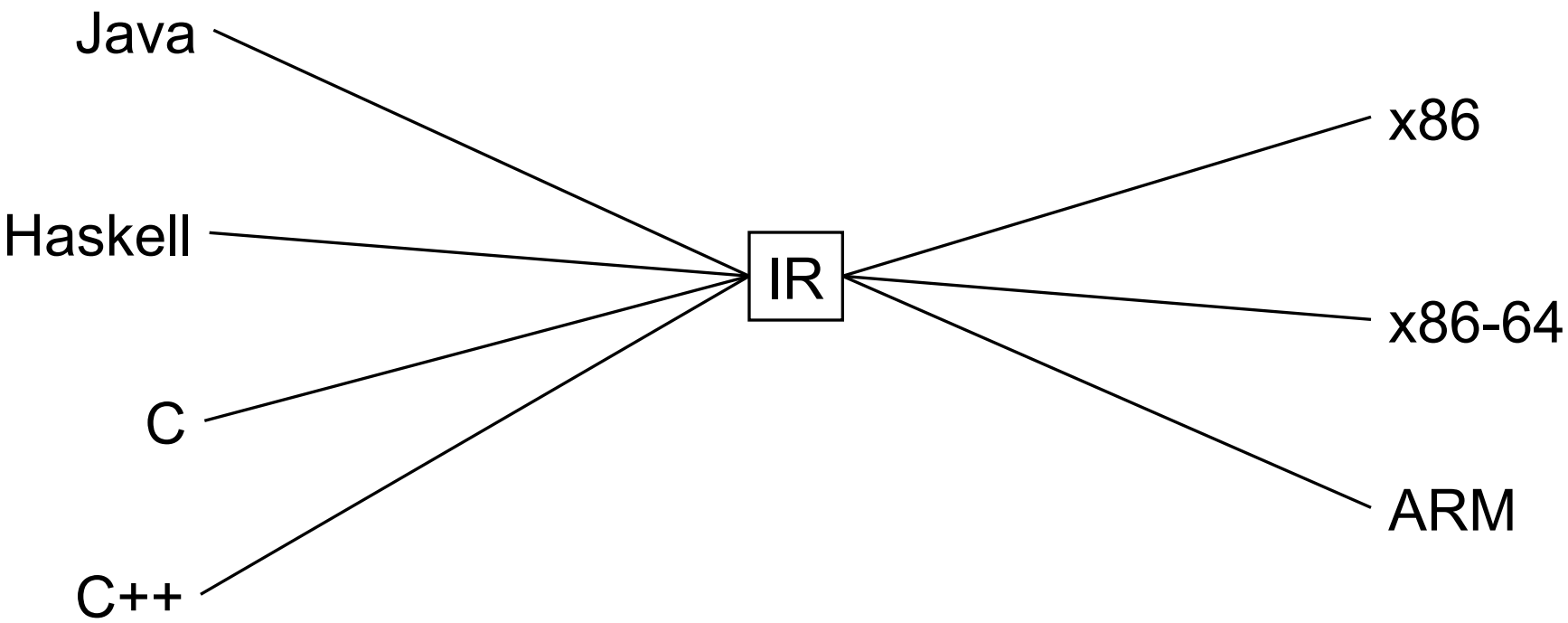
Middle end

- Language-independent optimizations and analysis
- Transformations on a series of “intermediate representations” (IRs)

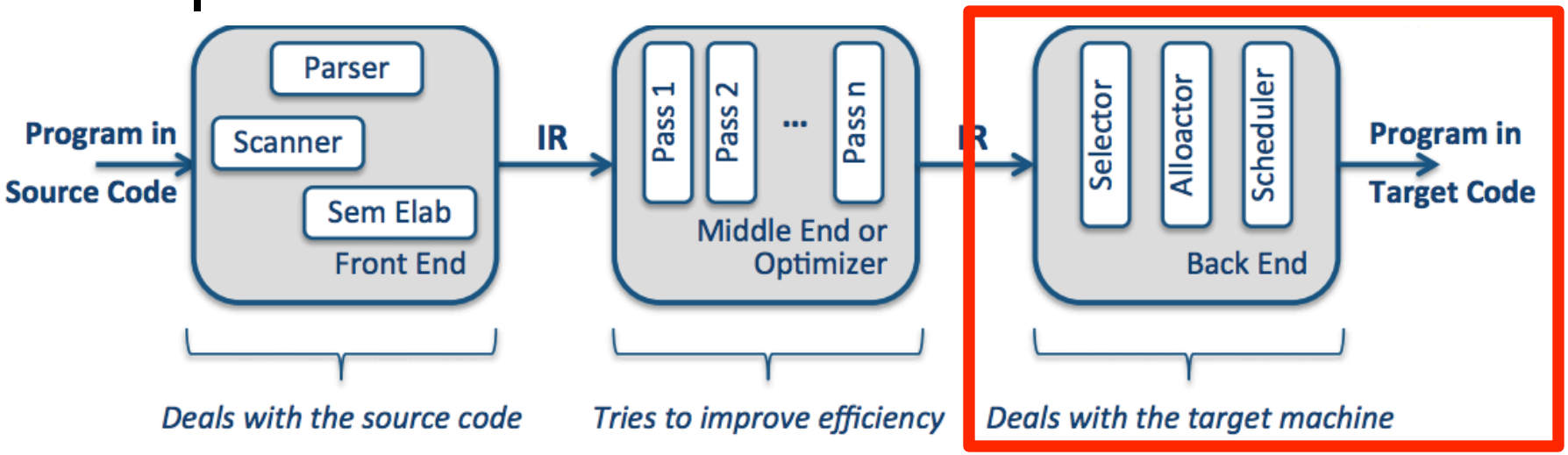
Middle End



Middle End



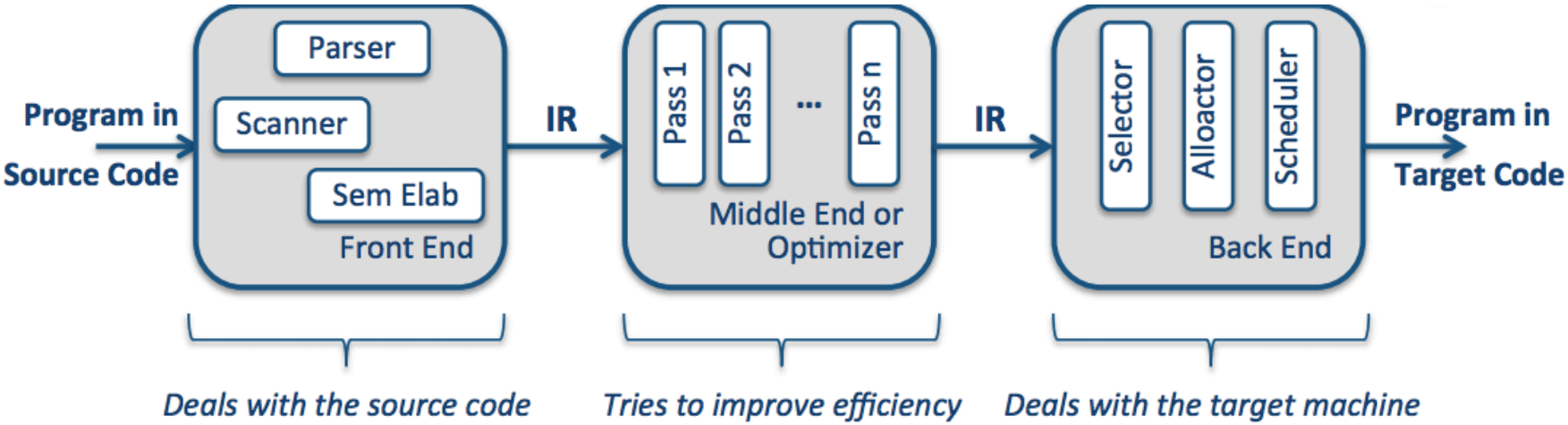
Inside a Compiler



Back end

- Instruction selection
- Register allocation
- Instruction scheduling
- Back end-specific optimization

Inside a Compiler



*graphic © Keith Cooper & Linda Torczon.

Grading

- Grading
  - Programming assignments (100%)
- Re-grades: requests for re-grades must be made **within 7 days** of receiving the graded assignment or test in question.

Academic Honesty

- A group's work must be their own
  - Discussion with other groups on assignments is NOT allowed
  - If you consult/copy something from the web then you must cite the source
- Consequences of academic dishonesty:
  - 1st offense: Receive a zero on that assignment or test
  - 2nd offense: Automatic "F" grade in the class and I forward the evidence to the Dean
- Continued enrollment in this class implies your consent to these rules.

More administrivia

- Office hours:
  - By appointment **only**: feel free to email me to set up an appointment
  - 318 EBN
- Course website:
  - https://harrisonwl.github.io/doc/cs4430.html
  - Can get to it off of my github page, too.
  - I will not use canvas to distribute grades or for anything else.