CS453 Compilers & Systems Software

Lecture:	**Tue and Thu 3:30 to 4:45 in Gould-Simpson**
Recitation:	**Fri 10-11am and 11 to noon in 228**
Instructor:	**Michelle Strout**
	mstrout@cs.arizona.edu
	Gould Simpson 707
	Office hours: Wed 2-3pm
	Thu 5-6pm
TA:	**Erman Gurses**
	egurses@email.arizona.edu
URL:	http://www.cs.arizona.edu/classes/cs453/fall16

Introductions with Index Cards

How we are going to use these cards

- Will be used to call on people in class throughout the semester and anytime a random ordering is needed.
- I will share some of your thoughts about learning compilers anonymously with the whole class.
- I will personally use the cards to associate names with faces.

On the side of the card with lines

- Full name
- NetID
- Preferred name and pronoun
- (Opinion) What is the value of learning compilers?

Collaboration and Anonymity Rules and Guidelines

FERPA law

- The instructor and TAs will keep all of your grade information private.
- D2L is where you can find your current grades.

Piazza

- Please post "Anonymous to Classmates".
- Extra credit is possible for substantive posts, which includes good questions (that have not already been asked).

Sharing Code (PA = Programming Assignment)

- Groups of size 1 to 5 can work together on code.
- Each group will have their own private repository.
- README file for each PA will list partners and ALL web resources used.
- Private class repository will have all anonymized code 24 hours after each PA deadline.

Plan for Today

Meggy Jr demo

Interpreter and Compiler Structure

Goals of Course

- Overview of programming assignments
- The MeggyJava compiler we will be building.

Course Themes

Compilers class and reality

- Why study compilers?

Course Logistics

Example MeggyJava program (see webpage for grammar)

MeggyJava: a Java subset for the Meggy Jr we are using in this course. Example code:

```
import meggy.Meggy;

class PA3Flower {
public static void main(String[] whatever){
  // Upper left petal, clockwise
   Meggy.setPixel( (byte)1, (byte)1, Meggy.Color.WHITE );
   Meggy.setPixel( (byte)2, (byte)1, Meggy.Color.WHITE );
   …

}
```

MeggyJr Demo

MeggyJava language is a subset of Java.

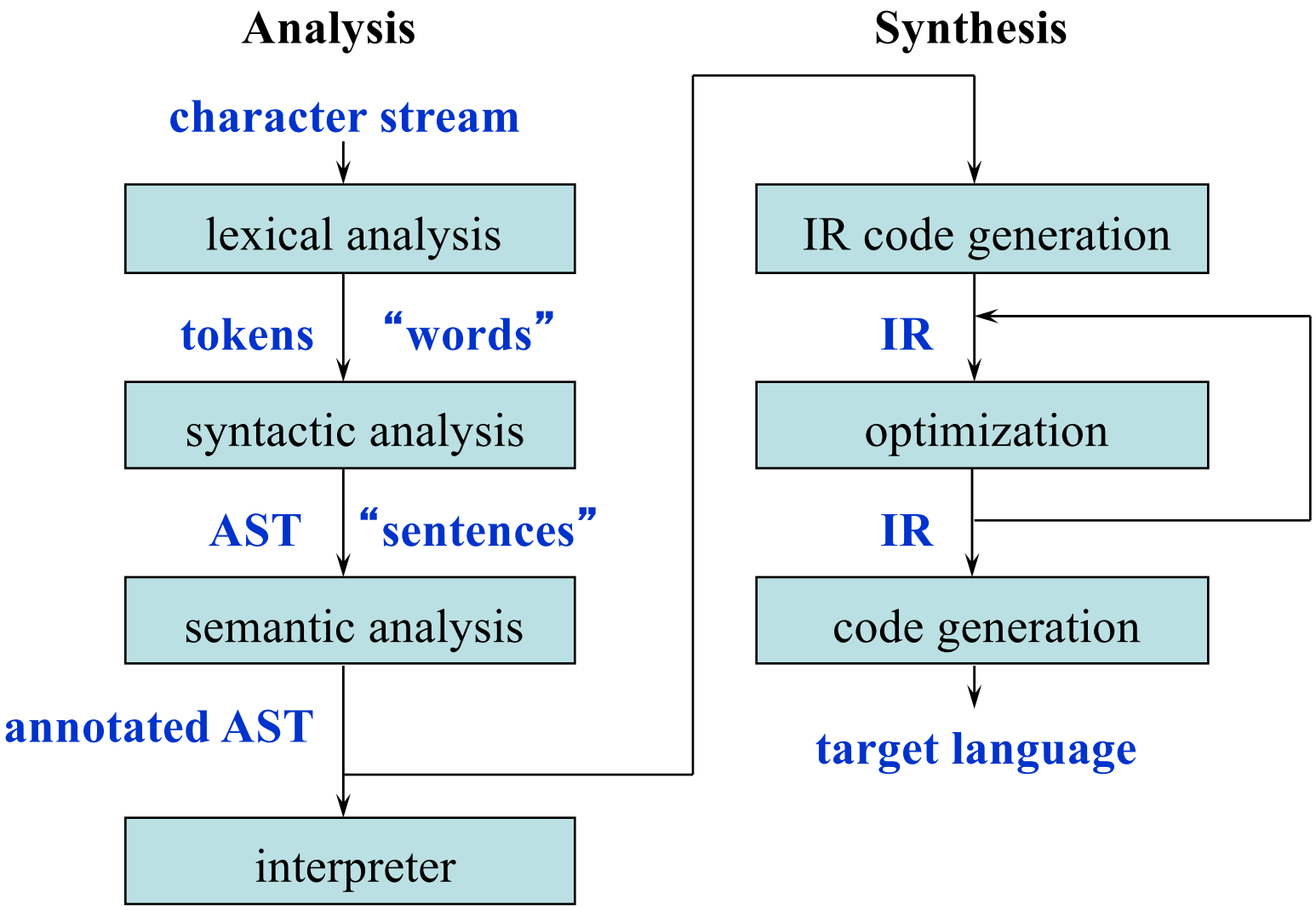
The reference compiler MJ.jar generates AVR assembly code.

AVR-gcc toolchain

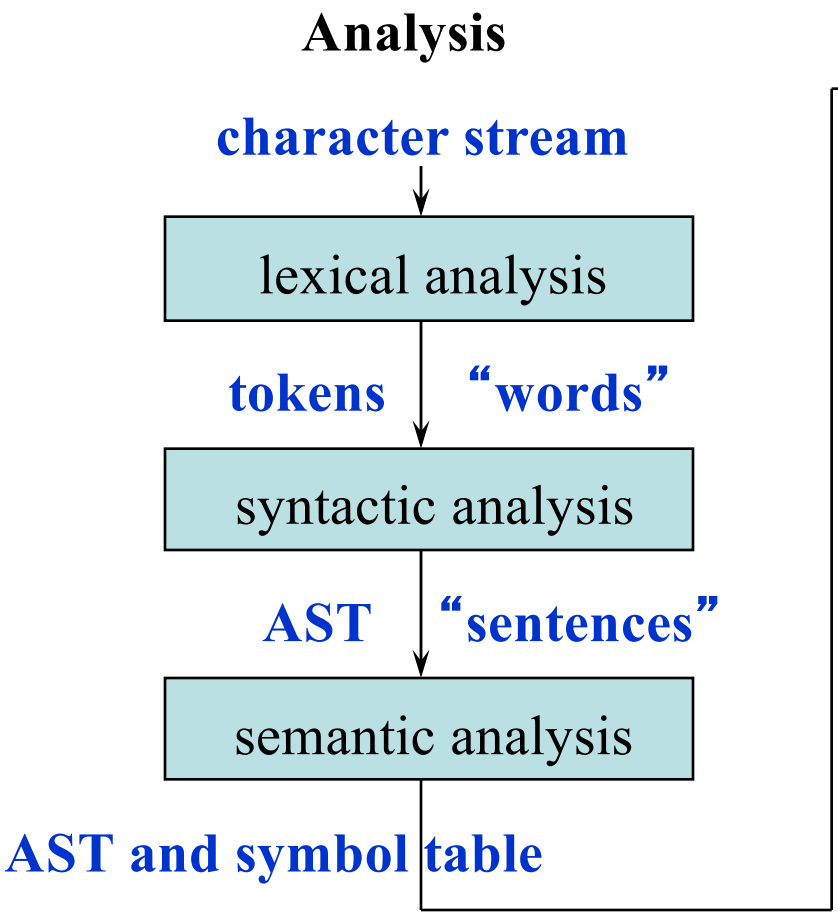
- Links File.java.s with modified Meggy Jr Simple run-time library
- Creates a File.java.hex file.
- Downloads the File.java.hex file to the device with avrdude.

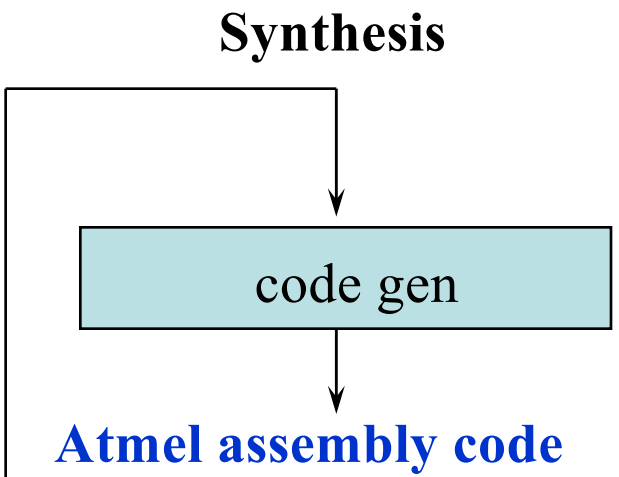
Arduino environment compiles C++ programs to Meggy Jr

Structure of a Typical Compiler



Structure of the MeggyJava Compiler





PA1: Write MeggyJava demo, and AVR warmup
PA2: MeggyJava setPixel
PA3: add exps and control flow (AST)
PA4: add methods (symbol table)
PA5: add objects and arrays

Course Themes

Implementing a broad spectrum of programming language constructs

Iterative development, testing, and revision control

Peer reviewing other people's code

Real world systems programming and limitations

- Infinite while loop is basis of all Meggy Jr programs
- The AVR assembly code has a lot of limitations.
- MeggyJava has a lot of limitations.

Functional and imperative programming constructs

- Functional pure vs. side-effects
- Lazy vs. eager evaluation
- Recursion vs. loops
- Pattern matching vs. the visitor design pattern

A LOT OF CONCEPTS, TOOLS, and CODE

Compilers are large and complex software structures

In this course you will learn a lot of concepts

–Regular and Context Free grammars, pattern matching, architecture

In this course you will use A LOT of tools

–Haskell

–version control (github)

–Makefiles

–Regression testing

–Assemblers

–(Meggy) hardware, graphviz (dot), etc.

In this course you will write a lot of code

–100s approaching thousands of lines

Don't get behind! It will be difficult to catch up.

Why compilers?

Always compilers and translators in industry

- New architectures, programming languages, and applications
- Parsing of input files

General software development skills

- Working with a group on a bigger project and peer review
- Revision control and testing
- REALLY helps your debugging skills

Compilers puts theory to practice

- Regular expressions and context free grammars in action!
- Computer Organization, in the end, it all goes to assembly
- Algorithms, will be discussing complexity of parsing algorithms
- Software Engineering
- Parallel Programming, how can compilers help parallelize?

Course Logistics (Highlights, see web page for more detail)

Schedule Page and Home/News

Read both of these daily. Lots of reading in the first couple of weeks.

HW or PA due each Monday starting this coming Monday Aug 29th.

Resources Page

Syllabus and Grading

Professional Conduct

Do your own work.

Act like a professional in the lab and when working with a group.

Follow the Department of Computer Science Code of Conduct

Participate

Come to class and recitation.

Come to lab and office hours.

Ask questions and post answers on the piazza discussion board.

Before Next Time

Read all the pages of the website.

- Especially PA1 writeup!!
- Go ahead and get started on PA1. PA1 is due Monday September 12th.
- Start working on HW1 (will be posted tonight). HW1 is due Monday the 29th.

Sign up for Piazza

- Possibly post that you are looking for a group for PA1.

Indicate discussion section scheduling info in D2L

- ***Discussion Section Scheduling quiz is due tomorrow!***

Determine group for PA1 and indicate in github

- Create a group repository in github classroom for PA1 and put a README file in the repository with the netids of everyone in the group.
- We can help with this in discussion section on Friday.
- Go ahead and get started on PA1. PA1 is due Monday September 12th.