CSCI 334: Principles of Programming Languages

Lecture 3: What is a language anyway?

Instructor: Dan Barowy Williams

Announcements



Computer Science Movie Night: The Matrix

Tonight at 6:30pm Wege Auditorium

(finishes before fireworks @ Poker Flats)

Benefits:

- Fun!
- Snacks!
- You will finally be able to understand your professor's jokes!
- You will be able to converse fluently with other insufferable nerds!
- You might learn a little computer science!
- Did I mention snacks?!!

Topics

What is a language?

Turing equivalence

WCMA activity

Your to-dos

- 1. Read A F#: The Cool Stuff
- 2. Lab 2, due Wednesday 2/19 (partner lab) Be sure to tell me who your partner is in collaborators.txt file.







What is a language?

In this class, we concern ourselves with a specific formulation of "language," called a **formal language**.

A **formal language** is the set of words whose letters are taken from some **alphabet** and whose construction follows some **rules**.

Example:

```
L = {a, aa, b, bb, ab, ba}

Σ = {a, b}

<expr> ::= <letter> | <letter><letter>

<letter> ::= a | b
```

Interesting fact: language = machine

<expr> ::= <letter> | <letter><letter> <letter> ::= a | b



The above is a machine called a deterministic finite state automaton (DFA) that "accepts" only words $\in L$.

The two definitions above describe the same language, but precisely.

What is a programming language?

A programming language is defined by two machines:

- 1. A syntax machine that determines the set of strings that are in the language.
- 2. A semantics machine that determines what gets done (i.e., what computational work) with an accepted string.

We spend a lot of time in PL thinking about these machines, which we call **language models**.

Surprising fact!

Almost all general purpose programming languages are **equivalent** in computational power to a **Turing machine**.

"Building up a language"

Syntactic sugar for curried functions

Activity

Recap & Next Class

Today we covered:

WCMA

Next class:

F#