

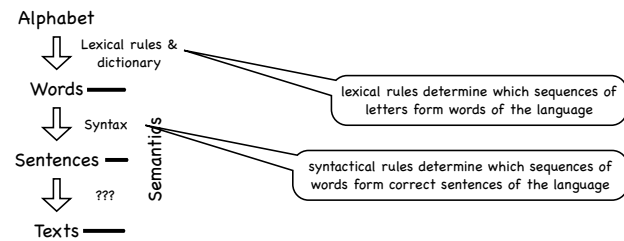
What Is a Programming Language?

©Andrey Kruglyak, 2010

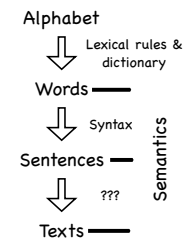
What is a programming language?

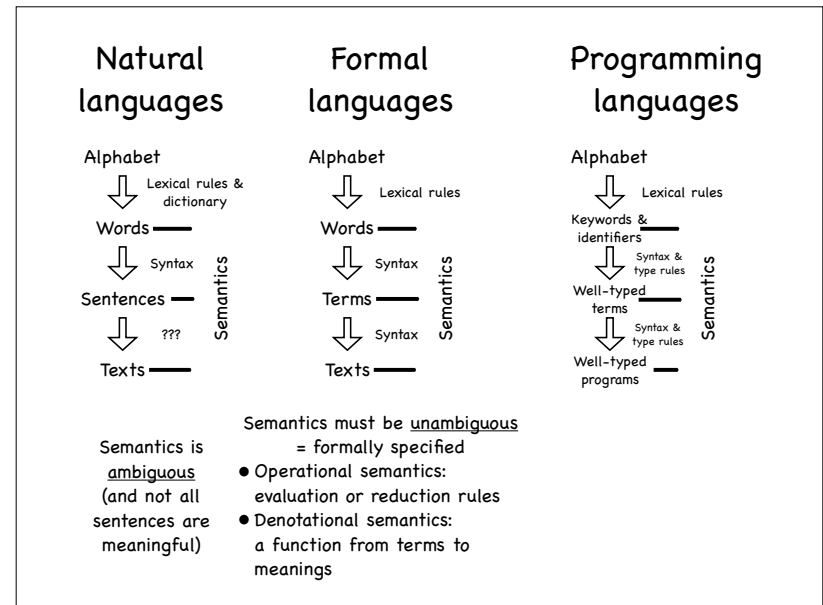
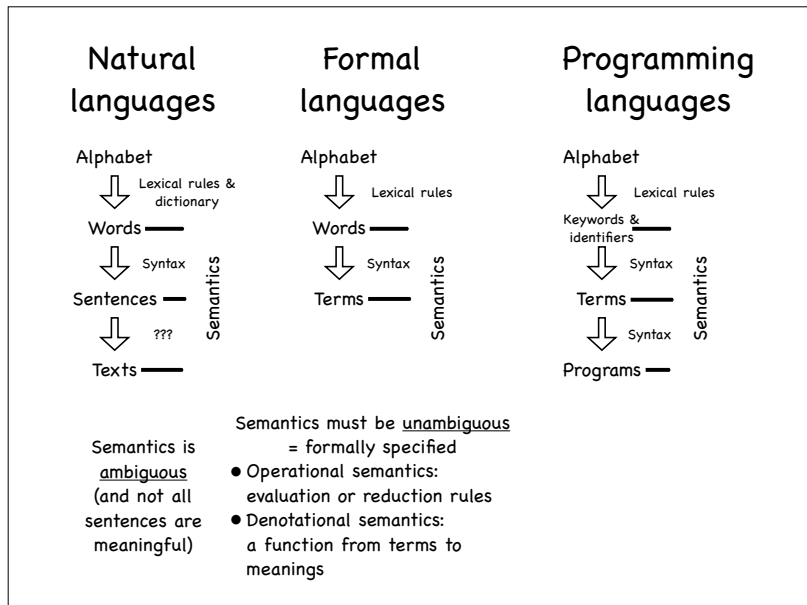
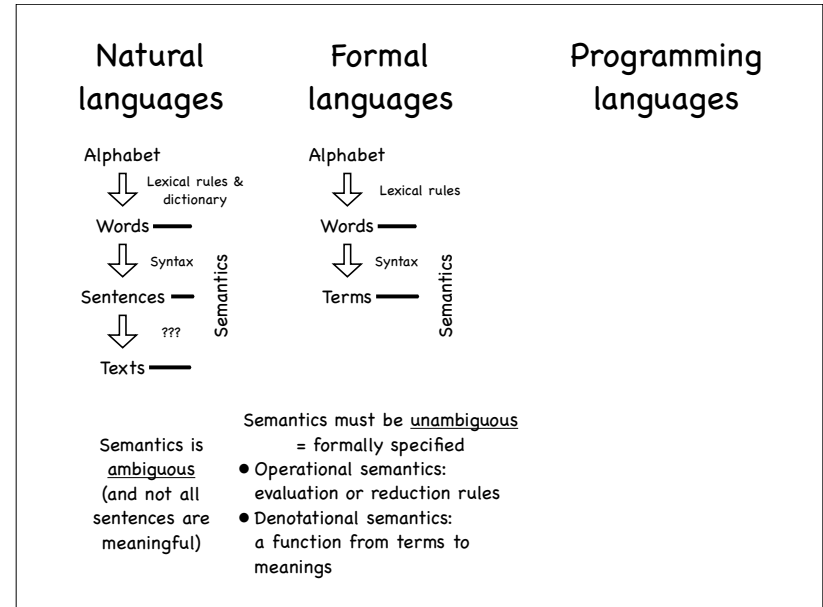
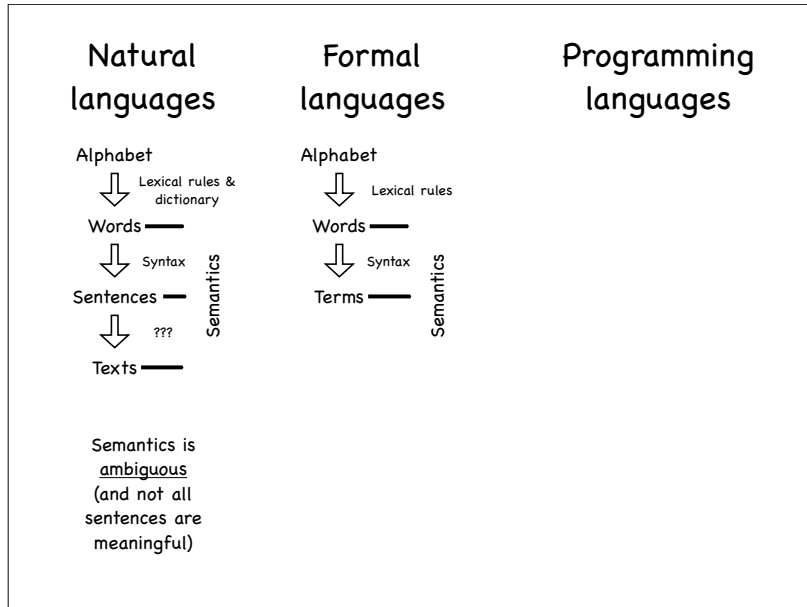


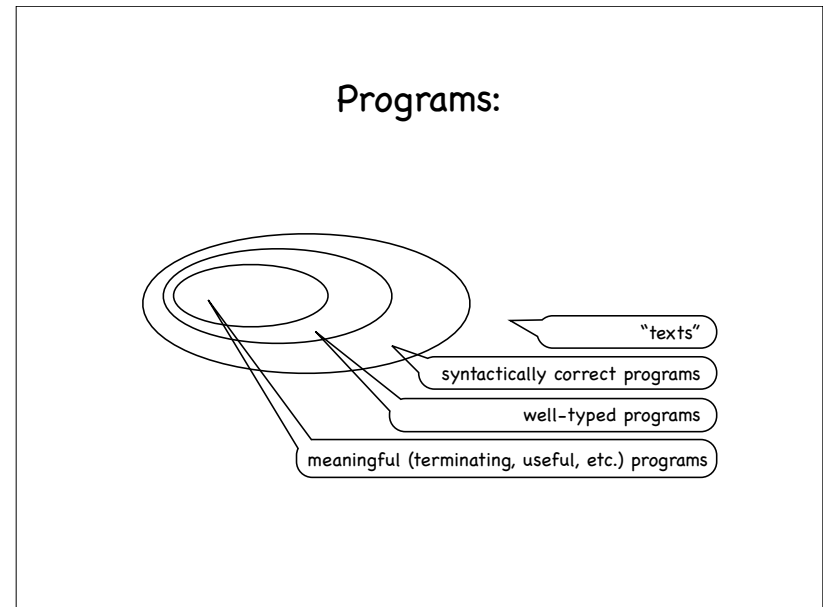
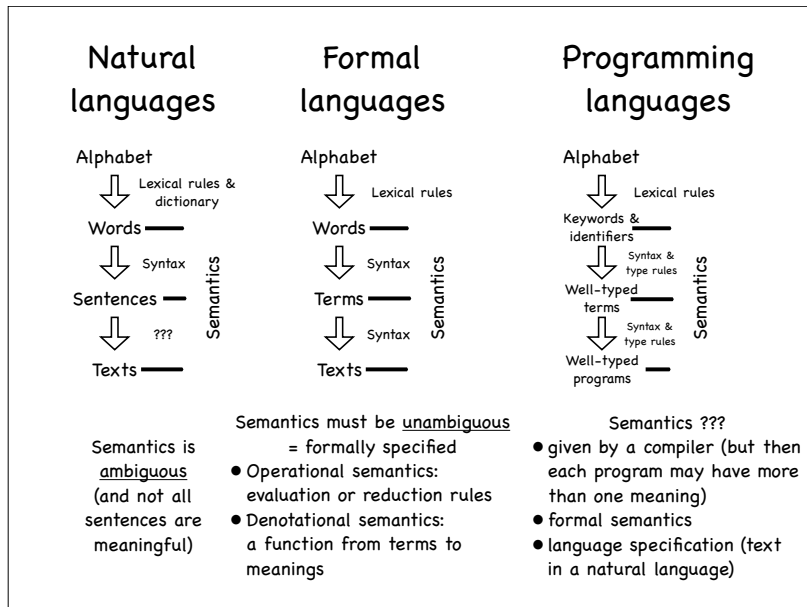
Natural languages Formal languages Programming languages



Natural languages Formal languages Programming languages







- ## For a programming language we need:
- ◁ An alphabet
 - ◁ Lexical rules for identifiers + a list of keywords & special symbols
 - ◁ Syntactical rules to make terms & programs
 - ◁ Typing rules to eliminate some meaningless programs (optional)
 - ◁ Semantics, e.g. small-step operational semantics: rules for evaluation of terms into other terms and ultimately into values

- ## How do languages differ?
- ◁ We will be dealing with this question throughout the course, but here are some examples:
 - ◁ Evaluation strategies (deterministic vs. non-deterministic order of evaluation, with and without backtracking, call-by-value vs. call-by-name, etc.)
 - ◁ How restrictive are the syntactic and typing rules (compile-time vs. run-time checks)
 - ◁ Expressive power
 - ◁ Fancy language features

Do we need different programming languages?



Do we need different programming languages?

- ◀ Different languages to address different classes of problems (domain-specific languages have been quite successful, e.g. ColdFusion Markup Language, Erlang)
- ◀ Different languages for different hardware (e.g. for massively parallel computations)
- ◀ Languages with different levels of abstraction
- ◀ Different kinds of solutions (imperative, functional, logic - we will discuss that later)

Something to think about...

- ◀ The same functionality can be offered as:
 - ◀ a language feature - can be used directly by programmers, implementation is the responsibility of the compiler constructor (e.g. concurrent objects)
 - ◀ a library - reusable code that provides services to different programs (e.g. POSIX thread library, with mutex locks)
 - ◀ a design pattern - a general reusable solution to a commonly occurring problem, implementation is the responsibility of the programmers (e.g. a certain use of mutex locks to guarantee state consistency)

Something to think about...

- ◀ Language designers need to make their choice, different choices give us different languages.
- ◀ In the end, this is very much a matter of taste.