

Grammars and Semantics

Given the following grammar,

$$\begin{aligned} G'' : \langle E \rangle^* & ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle \\ \langle T \rangle & ::= \langle T \rangle * \langle P \rangle \mid \langle P \rangle \\ \langle P \rangle & ::= a \mid b \mid c \end{aligned}$$

Add productions to the grammar that define the right-associative operator = at a lower precedence than any of the other operators.

HW#2:
3.1 a
3.3 b
3.4 b
Due Monday 2/8

Grammars and Semantics

a) Show that the following grammar is ambiguous.

$$\begin{aligned} G : \langle S \rangle & ::= \langle S \rangle \langle S \rangle \\ & \mid (\langle S \rangle) \\ & \mid () \end{aligned}$$

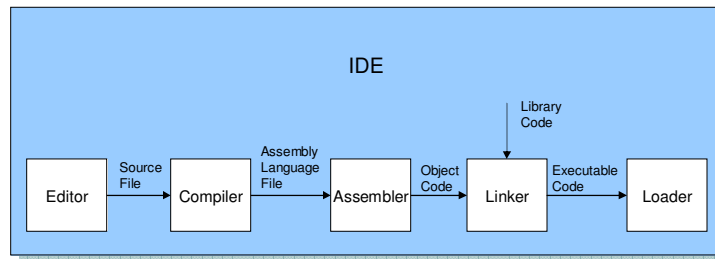
b) Rewrite the above grammar so that it is no longer ambiguous.

Language Systems

What actually happens in your IDE? IDE ≡ Integrated Development Environment

Classical Sequence: C++, C, Fortran

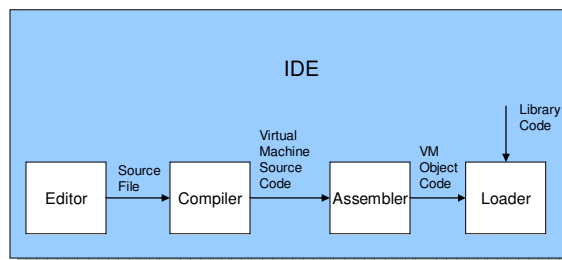
Read Chap 4



Language Systems

Alternatives, depending on Source Language

Java, C#

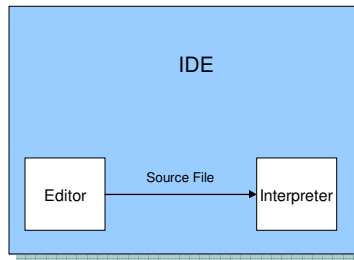


Note: In Java and C# there is no Linker, program is linked at runtime.

Language Systems

Alternatives, depending on Source Language

Lisp



Language Systems

- Some IDEs have compilers, some have interpreters – What is the difference?
- Compilers translate high-level languages (Java, C, C++) into low-level languages (Java Byte Code, assembly language).
- Interpreters execute high-level languages directly (Lisp).

Observation: Virtual machines can be considered interpreters for low-level languages; they directly execute a low-level language without first translating it.

Language Systems

- Why choose compilation over interpretation?
 - Compilers can generate very efficient code and, consequently, the compiled programs run faster than interpreted programs.