Compilers & Lab

Assignment #1 (Theory)

- **0 Score** for the corresponding assignment **for both the copy and source**.
- **Clearly indicate** the source if you get some from other places or if it is not your original idea or thoughts. **(Otherwise, the score will be cut by 10%)**

1. What is the difference between a compiler and an interpreter?

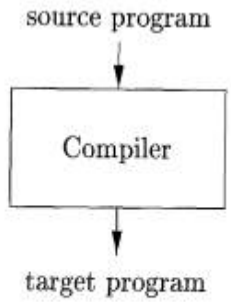


Figure 1.1: A compiler

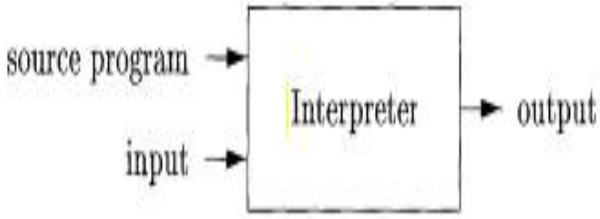


Figure 1.3: An interpreter

Compiler and interpreter are both types of programming language processors.

The compiler converts source code into object code, in other words source program into target program.

However interpreter appears to directly execute the operations specified in the source program on inputs supplied by the user. In other words, interpreter produces output –not object code- as a soruce code's translation.

2. What are important qualities of compilers?

The important elements to improve qualities of compilers are correct recognization of program constructs, speed of compilation and cognifition of source code, preservation the meaning of the code, Good error reporting and handling, Code debugging help etc.

3. Why are compilers commonly split into multiple passes?

The kind of compilers's split method is three – one pass, two pass, multiple pass. Advantage of multiple passes compared to other methods is machine Independent and more expressive. So multiple passes method is commonly used in many of compilers.

4. What are the typical responsibilities of the different parts of a modern compiler?

Compilers convert source code into object code. Modern compilers have typical responsibilities that function Lexical / Syntax / Semantic analysis, Intermediate(IR) Generation / Optomization, Final code generation, Optimization IR code to final code.

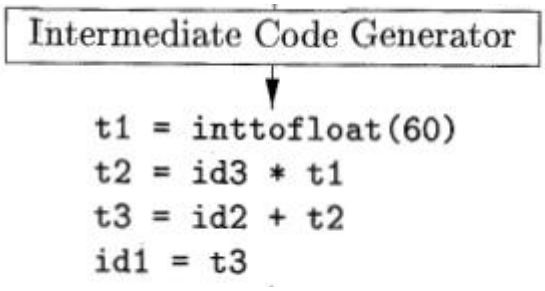
5. How are context-free grammars specified?

Context-free grammar is a formal grammar that usually specified with Backus Naur form(BNF). BNF is kind of metasyntax notation to discribe programing language by language or symbol. And they formally collection of four objects-set of non-terminal symbols, termial symbols, production rules and start symbol.

6. What is "abstract" about an abstract syntax tree?

Mean of "abstract" in abstract syntax tree is it does not detailly represent than it appears in real syntax. Real syntax trees -like concrete syntax trees, parse trees- are built by a parser during the source code translation and compiling process. However abstract syntax trees contain an abstract representation of a program's syntax.

7. What is intermediate representation and what is it for? Why is optimization a separate activity?



Intermediate representation(IR) is result of Intermediate Code Generator.

Intermdiate code is that used to improve better target code. The reason of that optimization is separated is a resonable way to generate good target code. It improves the running time of the target program without slowing down compilation too much. So it is efficient that optimization separate.

8. Is Java compiled or interpreted? What about Smalltalk? Ruby? PHP? Are you sure?

Java is both Interpreted and compiled. Java is a compiled language because its source code is first compiled into a binary byte code. And Java is an interpreted language because its binary bite code running on a virtual machine, the Java Virtual Machine (JVM), which is usually a software-based interpreter.

Also Smalltalk, Ruby, PHP are both interpreted and compiled. They usually compiled to bytecode, which is then interpreted by a virtual machine or dynamically translated into machine native code.

9. What are the key differences between modern compilers and compilers written in the 1970s?

When 1970s, FORTRAN is main programming language, and theory of computer science and other programming language started to make. So it is key differences between modern and 1970s. Nowadays there are many function related to computer science also modern compilers have to reflect this situation.

10. **Why is it hard for compilers to generate good error messages?**

It is different that recognize error messages and that generate error messages. In other words, It is easy to know what is wrong in our source code by error messages, but It is hard to generate and classify what is wrong in lots of source code. Because kinds of exception is so a lot, and we could not know all that error or exception. So, It is hard for compilers to generate good error messages because it is hard to know and classify kind of error or exception.

11. **What is "context-free" about a context-free grammar?**

In merriam-webster mean of context-free is that.

Mean of "context-free" in context-free grammar is it is regardless of the context of a nonterminal if that can be applied production rules.

<The End of the Assignment>