

Computer Science 131, Spring 2002

Assignment 8:

Structured Programming with `go to` Statements

Out: Monday, March 25

Due: Monday, April 1, in class (no fooling!)

The main goal of this assignment is to get you thinking about Donald Knuth's paper "Structured Programming with GOTO Statements", in preparation for an in-class discussion on Monday, April 1. As before, please type up your submission; you should aim for around 2 pages.

Background

The paper you are about to read was published in 1974. Here are a few facts that might help you better understand the context of the paper.

1. Given extremely limited memories and slow processors, there was an especially large incentive to squeeze every drop of performance out of code, even at the expense of clarity. Compilers were not terrific at generating optimized code (or didn't even exist for new architectures) and so a lot of programming was still done in assembly.
2. Fortran 66, Cobol, and Algol were around and reasonably widespread use. Lisp had also been around since the 1950's. These were the primary "high-level" languages Knuth was probably thinking of. Much published code was in Algol or Algol-like syntax, as in this paper. [A syntactic difference between Algol and C was that blocks of code were surrounded by `begin` and `end` instead of braces. Algol-68 syntax explicitly marked the end of conditionals, loops, etc., by spelling the main keyword backwards (i.e., `if...then...else...fi` and `do...od`).]
3. What languages weren't around? The design of Pascal had been published in 1970, but it didn't attain widespread use until the release of the USCD p-System in the late 70's [which, by the way, translated Pascal programs into an efficient, portable bytecode called p-code, just as Java systems often generate JVM bytecode today]. Although the (pre-ANSI) C language development occurred around 1972, it didn't become widespread until Unix did in the late 70's. The object-oriented language Simula

had been designed in the 60's, but never obtained widespread use. [However Simula did strongly influence the design of later OOP languages such as Smalltalk and C++ in the late 70's/early 80's.]

4. At the time of the paper “structured programming” was still a relatively new idea, introduced when people realized how hard it really was to write and maintain correct programs. The ideas could be summarized as the notion that better programs result from writing code that is clear than by code that is clever. More specifically, structured programming includes ideas such as using multiple levels of abstraction [shades of CS 60!], top-down design and solving a problem by decomposing it into modular subproblems, and using special-purpose control flow constructs like `if` or `while` instead of arbitrary `goto`'s.
5. The only way to alter control in assembly language is typically via branch/jump instructions. These could be conditional or unconditional, and they could act like a one-way `goto` or a procedure call (by saving the address to return to), but that's it. Many programmers, being fluent in assembly, thought in these terms, which matched up well with documentation techniques such as flowcharts.

Fortran 66 had loops built in, but didn't have the modern `if-then-else`. What it did have was the “arithmetic if” that let you `goto` one of three labels depending on whether an expression was negative, zero, or positive. `else` was added only in Fortran 77, released three years after Knuth's paper was written.

Algol 60 did support more modern control flow constructs as well as `goto`, as you'll see in the paper.

If you find yourself asking “why didn't he just use *xxx*?” where *xxx* is some language feature such as objects, higher-order functions, etc., the answer is very likely to be that such things weren't in the programming languages of the time.

Assignment

Several modern languages like Java and Python have entirely dropped support for `goto`, but they also include many features that were unknown or at least uncommon in 1974. Do Knuth's arguments apply to such languages, or has history made the paper obsolete, or is the truth somewhere in-between? Be sure to carefully explain the points in the paper to which you are responding, and provide convincing arguments for your beliefs.

Additionally, list any questions you have about the paper, and any specific topics in the paper you'd like to have discussed in class. [There should be at least a couple of responses to this part.]