Honors Compilers

The Course Project
Feb 28th 2002
The Goal

Write a compiler for a subset of Algol-68

The compiler to be written in its own language (i.e. in the chosen subset)

Targetted to MMIX

Bootstrapped (Compiler can compile itself)
The Language to be Compiled

A subset of Algol-68 (roughly Algol-68S)
You get to choose the exact subset

Required elements

- int, bool, char primitive modes
- ref, [], struct, proc modes
- Normal declaration and statement forms
- Including if/endif and do/od loops
- All ordinary expression forms
- Compile a single procedure (acts as main)
Optional Language Elements

1-d arrays required, 2-d etc optional
Union modes optional
Floating-point processing optional
Garbage collection optional
Case/esac optional
Implement separately compiled modules
Good error messages
Optimization

Aim for a reasonable level of code quality
At least minimal register allocation
Preferably attempt at least some global optimization
Input/Output

You must be able to read and write files (treated as a sequence of char)

Exact provision of facilities is up to you
Lexical Structure

Standard Algol-68 lexical elements
Use upper case for bold, lower case for non-bold.
Lexical structure to be defined by your second assignment.
Third assignment

Due right after the break
Exact description of subset you will implement
Indicate possible optional additions
Provide BNF of language grammar
Fourth Assignment

Take grammar you devised for your subset.

Modify grammar to be acceptable as bison input, and use output bison to parse a small program.
Fifth Assignment

Provide complete design document for compiler
Exact details of what is to be implemented
Internal data structures
Format of intermediate files
Details of algorithms used
Implementation Language

To be implemented in its own language
Final version must be able to compile itself
So you define the implementation language as well as the language to be compiled.
The Target Language

Generate MMIX assembly language
To be assembled and run on MMIX assembler
Compiler Organization

Organize compiler as set of separate programs, e.g. parser, semantic analyzer, optimizer, code generator.

Programs communicate by means of files.

Carefully define format of these files.
Bootstrapping

Some different methods

- Write crude translator from Algol-68 subset to C (e.g. PERL script)
- Write crude translator from C to Algol-68
- Write in Algol-68 and use some existing A68 compiler or translator
What to Hand In

Commented source code
Set of sample programs used to test compiler, with output from MMIX run, and MMIX assembly language.
Standard class bench mark (to be supplied later) run on instrumented MMIX simulator.
Due Date

Last Day of the Exam Period
Extensions possible
But not if you want to use grade to avoid exam!