

# The FORTRAN Automatic Coding System

Chris Gregory & Matthew Lyons

## History

- First complete compiler
- Influenced BASIC and C
  - BASIC has numbers and FORTRAN also has numbers
- Designed for math

## FORTRAN Language Specification

- Variable assignments are similar to modern assignments
- Cannot use integer values in floating point expressions
- Functions end with F
- Functions must be declared first
- The only accessible library functions are those on the library tape
- Builtin I/O functions: PRINT, READ, FORMAT
- Has control flow: DO, IF, GO TO
- Maintains similarities to assembly; for example, IF branches instead of having its own block
- No explicit types

## FORTRAN Translator

- Single pass
- Six steps

## Stage One

Compiles arithmetic, puts others into tables

$$A + B * * C * (E + F).$$

$$+(*(*A))+(*(*B**C)*[+(*(*E))+(*(*F))]).$$

## Stage One Optimization

$$u_0 = + u_1 + u_3$$

$$u_1 = * u_2$$

$$u_2 = ** A$$

$$u_3 = * u_4 * u_5$$

$$u_4 = ** B * * C$$

$$u_5 = + u_6 + u_8$$

$$u_6 = * u_7$$

$$u_7 = ** E$$

$$u_8 = * u_9$$

$$u_9 = ** F.$$

$$u_0 = + A + u_3$$

$$u_3 = * u_4 * u_5$$

$$u_4 = ** B * * C$$

$$u_5 = + E + F.$$

## Phase Two

Compiles DO and subscripted variables

$$A(2 * I + 1, 4 * J + 3, 6 * K + 5),$$
$$(c_1 i - 1) + (c_2 j - 1) d_i + (c_3 k - 1) d_i d_j + 1,$$

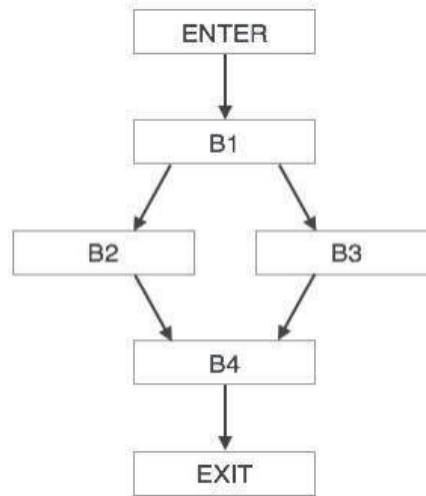
## Part Three

Merges the previous two steps; at this point, the program is "complete" for a computer with an infinite register file



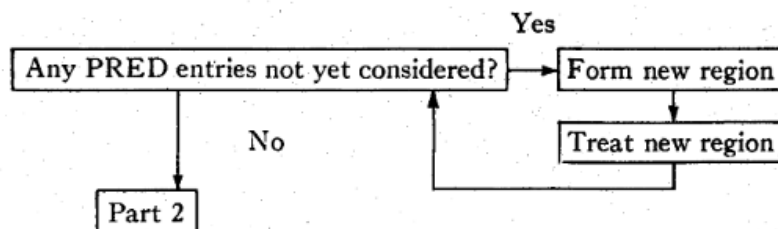
## Step Four

Analyzes the program flow



## Stride Five

Modifies the program to use only 3 registers



## Period Six

Assembles the program and outputs the binary



## Conclusion

- FORTRAN's design prioritizes:
  - Flexibility
  - Conciseness
  - Convenience
  - Low learning curve
  - Easy debugging
- Reduced a three day task into a mere four hours
  - That's mad gains, bruh 🤪



Compiler for  
FORTRAN?

Yep

Hello, ma'am, is your  
fortress running?

...Well you better go catch  
it!



questions?