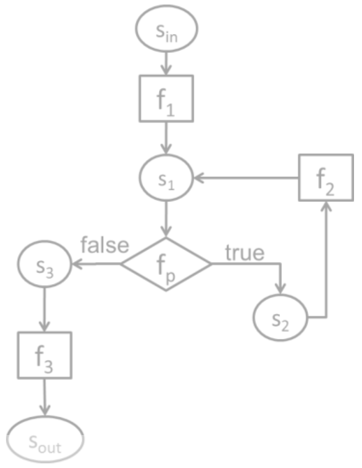
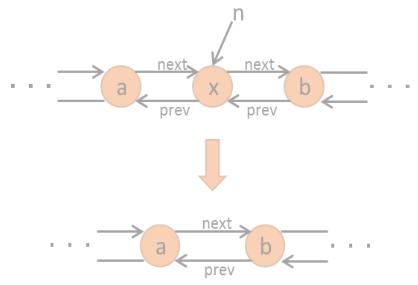
$$\exists c \forall in\ Q(c, in)$$

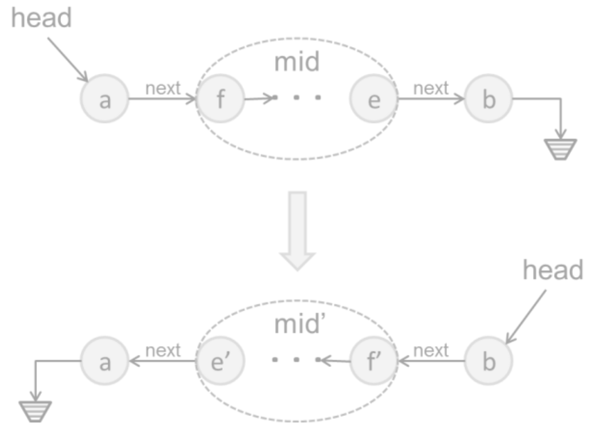
```
/* Average of x and y without using x+y (avoid overflow)*/
int avg(int x, int y){
  int t = expr({x/2, y/2, x%2, y%2, 2 }, {PLUS, DIV});
  assert t == (x+y)/2;
  return t;
}
```

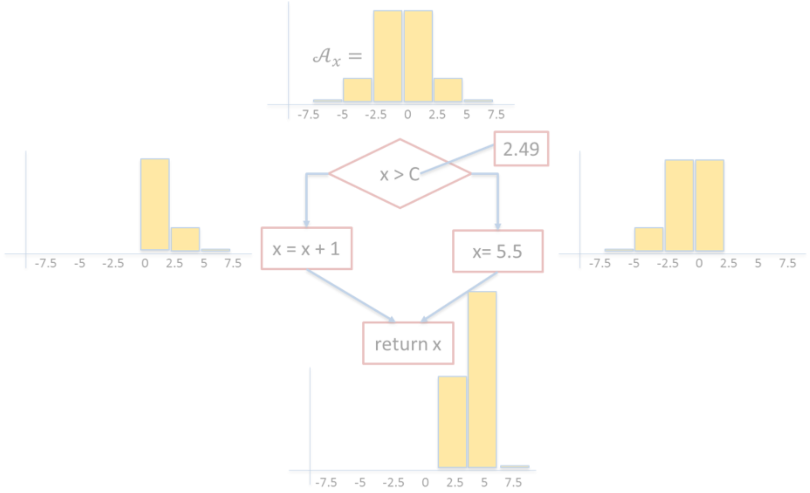




```
{
   s  = n.succ;
   p = n.pred;
   p.succ = s;
   s.pred = p;
}
```

Advanced Topics in Language and Translation





$$\varphi(p)$$

$$Sk[c](in)$$

Lecture 1
Course Overview

Kirshanthan (Krish) Sundararajah

Who are we?

Kirshanthan (Krish) Sundararajah

kirshanthans@vt.edu

Assistant Professor of Computer Science

Research interests: Compilers, Programming Languages, HPC, Systems

How about you?

What is this course about?

- This course focuses on the techniques used by advanced compilers to optimize and analyze programs.

- First 1/3 course: Core Topics
- Second 1/3 course: Presentations of Papers
- Third 1/3 course: Project presentations
- *No final exam*

Logistics: When & Where

- Lecture: MW 02:30-03:45 (WHIT 349)
  - But you already knew that ☺
- Office Hours
  - Answer questions, discuss your project
  - In my office (Gilbert Place 4103)
  - By Appointment
- Course Website
  - Website: kirshanthans.github.io/cs6304/
  - Piazza: https://piazza.com/class/lrhz6pau3ly6ag

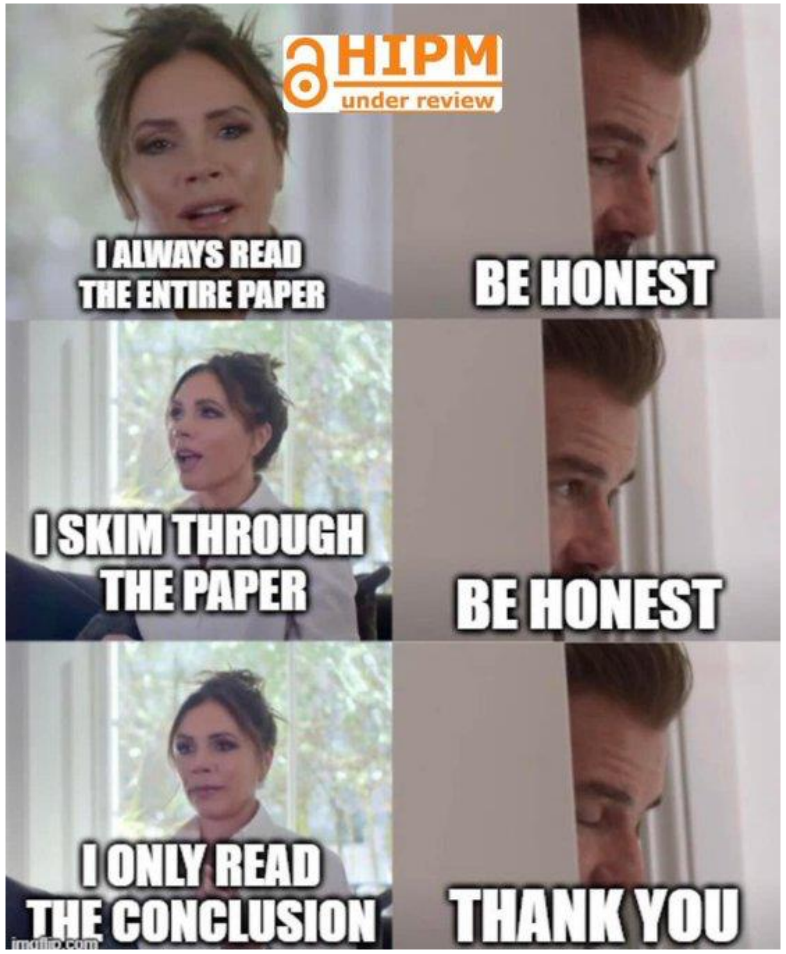
Logistics: Grading

- 15% -- Paper Responses (3 x 5%)
- 25% -- Paper presentation
- 50% -- Project
  - 10% -- Proposal
  - 15% -- Presentation
  - 25% -- Report

Logistics: Paper Presentation

- **How it works:**
  - A list of papers will be posted on the course website
  - Pick a paper from the list, hopefully, related to your project
  - Present the paper as a full lecture
  - You may suggest a paper out of the list (you need to justify your choice)

Logistics: How not to read a paper!! :D



Logistics: Project

- 1-page project proposal (10%)
- Final presentation (15%)
- Project report (25%)
  - Hopefully, it will be at the level of a conference publication.
  - It will be judged in terms of,
    - quality of execution,
    - originality, and
    - scope

What is a Compiler?

Traditionally …

- A program that *translates* from a high-level language (e.g., C++) to low-level assembly language that can be executed by hardware

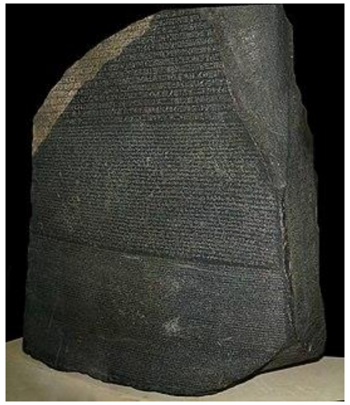
```
int a, b;
a = 3;
if (a < 4) {
   b = 2;
} else {
   b = 3;
}
```

$\Rightarrow$

```
    var a
    var b
    mov 3 a
    mov 4 r1
    cmpi a r1
    jge l_e
    mov 2 b
    jmp l_d
l_e: mov 3 b
l_d: ;done
```

… but really

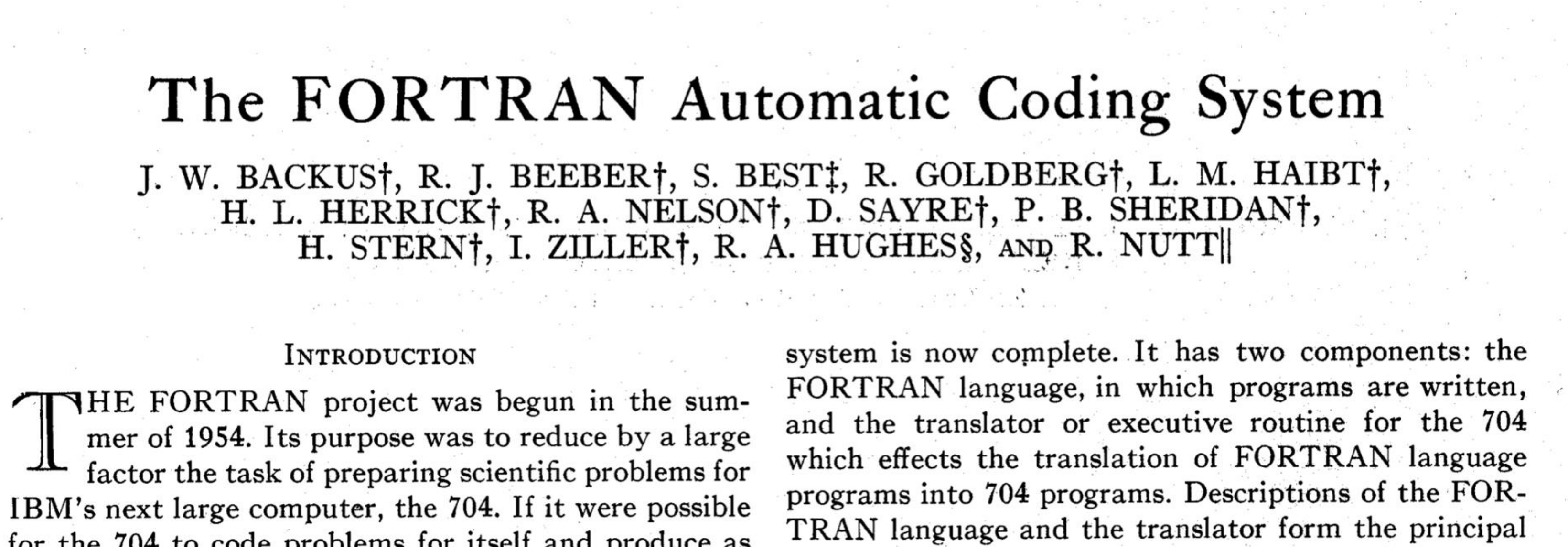
- A program that translates or transforms one *representation* of a program to *another* [and perhaps does some analysis along the way]
  - Fortran
  - C
  - C++
  - Java
  - Text processing language
  - HTML/XML
  - Command & Scripting Languages
  - Natural language
  - Domain specific languages



$\Rightarrow$

- Machine code
- Virtual machine code
- Transformed source code
- Augmented source code
- Low-level commands
- Semantic components
- Abstract syntax trees
- Another language

Historically (65 years ago)



The FORTRAN Automatic Coding System

J. W. BACKUS†, R. J. BEEBER†, S. BEST‡, R. GOLDBERG†, L. M. HAIBT†, H. L. HERRICK†, R. A. NELSON†, D. SAYRE†, P. B. SHERIDAN†, H. STERN†, I. ZILLER†, R. A. HUGHES§, AND R. NUTT||

INTRODUCTION

THE FORTRAN project was begun in the summer of 1954. Its purpose was to reduce by a large factor the task of preparing scientific problems for IBM's next large computer, the 704. If it were possible for the 704 to code problems for itself and produce as

system is now complete. It has two components: the FORTRAN language, in which programs are written, and the translator or executive routine for the 704 which effects the translation of FORTRAN language programs into 704 programs. Descriptions of the FORTRAN language and the translator form the principal

"AI is whatever hasn't been done yet." – Tesler's Theorem

Philosophically

Chris Lattner (designer of LLVM and Swift programming language):

> “The most important part of a compiler is: the design, representation, validation and translation of structured data. The mentality and design center crosscuts all of computing, from the simplest json payload to the most fiddly compiler IR
>
> Twitter, 2/16/20

Philosophically

Chris Lattner (designer of LLVM and Swift programming language):

> "The most important part of a compiler is: the design, representation, validation and translation of structured data. The mentality and design center crosscuts all of computing, from the simplest json payload to the most fiddly compiler IR

We will touch on different aspects of this in this course