BuckCal

for when you need something smarter to Calculate your Bucks ($$$), amongst other things

The Team

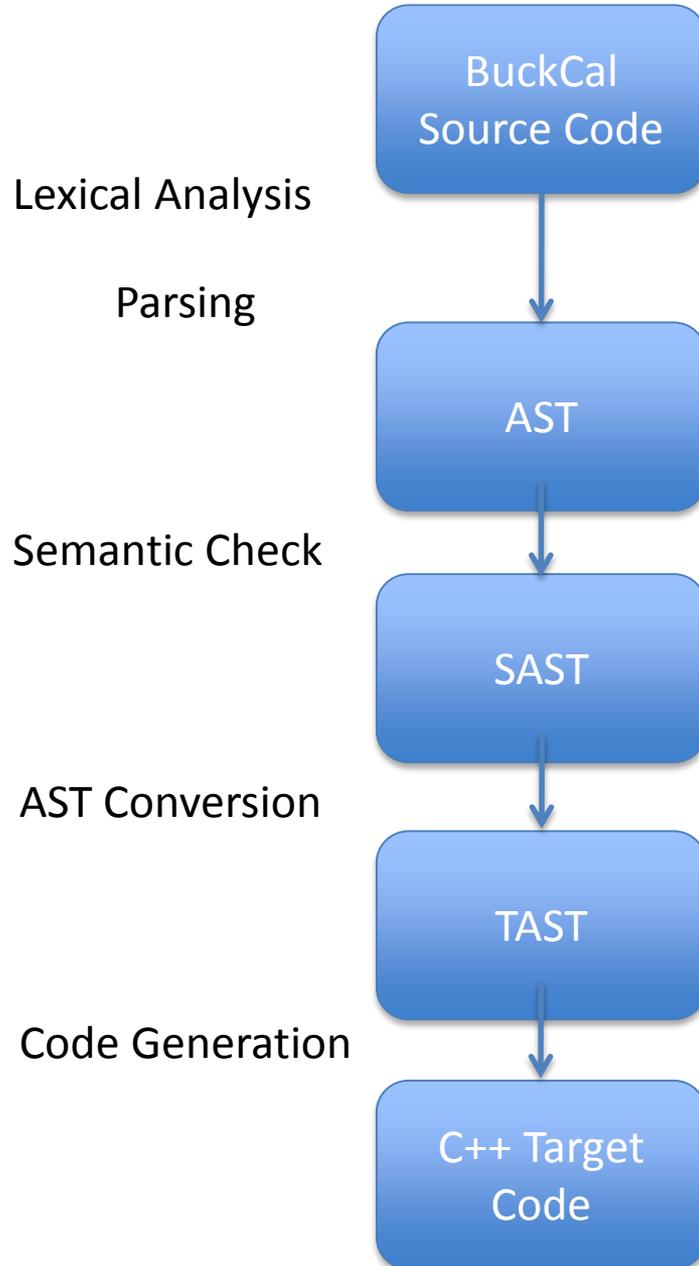Ahmad Maruf

Lan Yang

Lingyuan He

Meng Wang

Prachi Shukla

The Overview

- BuckCal is a matrix manipulation language
- Essentially a scripting-style language with full support for mathematical matrix operation and spreadsheet calculation

The Structure

BuckCal
Source Code

Lexical Analysis

Parsing

AST

Semantic Check

SAST

AST Conversion

TAST

Code Generation

C++ Target
Code

The Implementation in OCaml and C++

Matrix literal

BuckCal

```
disp {1, 2, x};
```

C++

```
int T_0[] = {1, 2, x};
int mat TT_1 = int_mat(T_0, 1, 3);
cout<< TT_1 << endl;
```
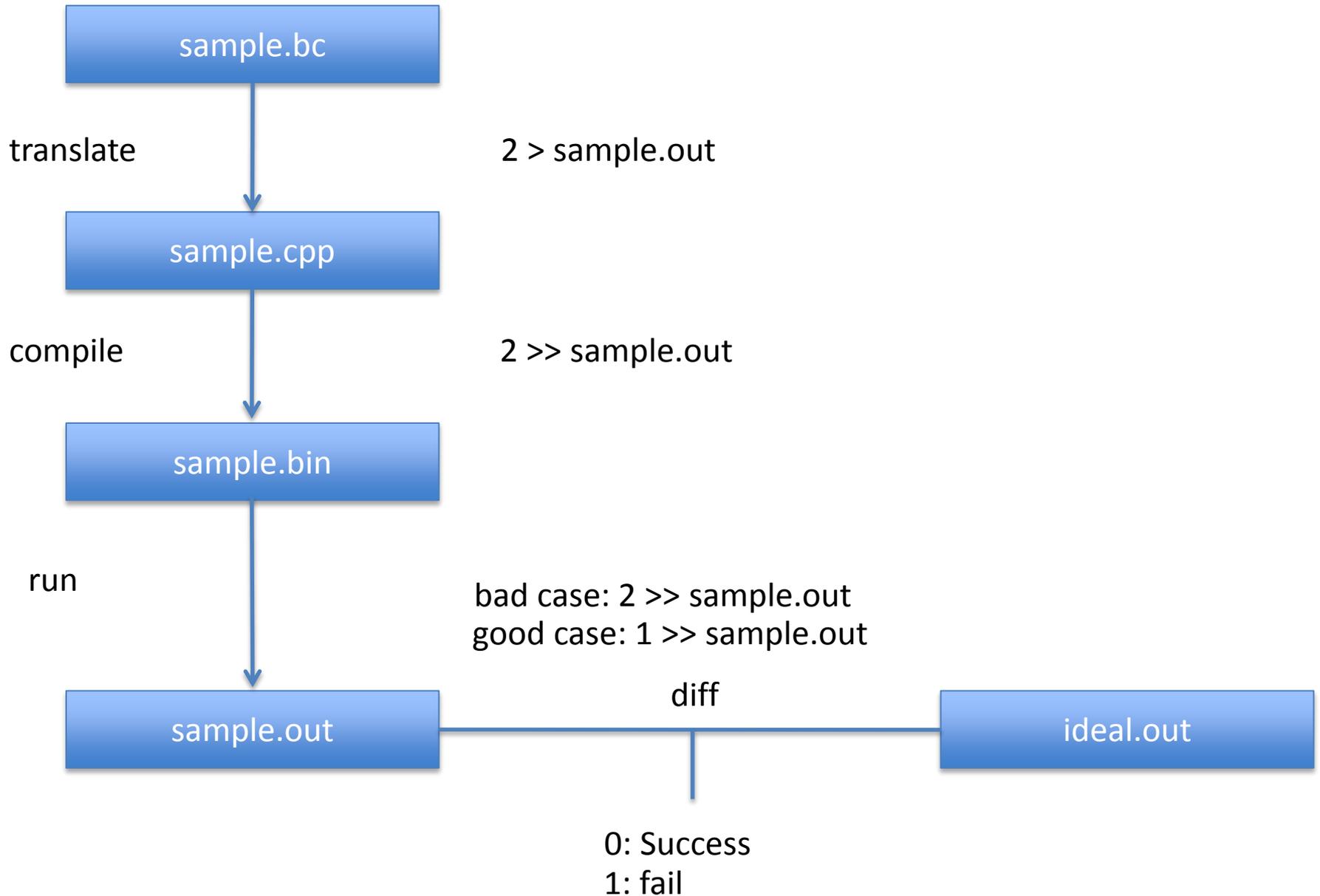
Implementation in C++

- Why C++?
  - One abstraction level down
  - Typed with primitives similar to BuckCal
  - Class abstraction with operator overload to easily add operation

Implementation in C++

- Structure:
  - Class int_mat / double_mat / string_mat;
  - Operators (+ - * /) in between int/int_mat double/double_mat
  - Operator << for matrix output
  - Built-in functions concerning concatenation, row/col names and conversion

Testing

sample.bc

translate   2 > sample.out

sample.cpp

compile   2 >> sample.out

sample.bin

run   bad case: 2 >> sample.out
                   good case: 1 >> sample.out

                              diff

sample.out   ideal.out

                   0: Success
                   1: fail

Combat Issues

The Lessons Learned

- Delivering good work is not dictatorship
- It's more about making a timeline
- A complete testing framework should be built ASAP
- Get people to work in small teams
- Keep the modules small

The Demos