Type-Directed Program Synthesis and Constraint Generation for Library Portability

Bruce Collie, Philip Ginsbach and Michael O’Boyle
University of Edinburgh

bruce.collie@ed.ac.uk
baltoli.github.io
A Heterogeneous World
Triple Constraint

- FAST
- PORTABLE
- MAINTAINABLE
"IMPORTANT: GPU support is currently highly EXPERIMENTAL and should be used by experienced developers only. In particular, DO NOT TRY TO WILDLY AND DIRTILY HACK THE BUILD SYSTEM, EVEN IF YOU ARE A PHYSICIST!"
Can we do better?

CODE → COMPILER

COMPILER → PROGRAM

PROGRAM → DEVICES
Can we do better?

- CODE → COMPILER → PROGRAM
- COMPILER ↔ DEVICES
Example

```c
void f(int n, float *x) {
    for(...) {
        // expensive loop...
    }
    library_call(n, x);
}
```

```c
void f(int n, float *x) {
    new_lib_loop_call(n, x);
}
```
How to achieve this?

1. **Learn** model of behaviour
2. **Search** for compatible code
3. **Migrate** to new library
Learning
Synthesis

INPUTS
- FFI CALL
- TYPE SIGNATURE
- PROPERTIES
  - Interface properties
  - From documentation

IO EXAMPLES
- Generate randomly

RULES
- Generic heuristics

SKETCH
- Control flow
- Program structure
Synthesis

- Search for instructions
- Test using IO examples

IO EXAMPLES

RULES

SKETCH

PROGRAM

SOLUTION
LLVM PROGRAM

define float @func(...) {
entry:
  %0 = getelementptr...
  %1 = load float...
  ... 
}

LLVM PROGRAM

```c
define float @func(...) {
entry:
  %0 = getelementptr...
  %1 = load float...
  ...
}
```

Search for compatible code

CONSTRAINTS

- {%0} is gep instruction and
- {%1} is load instruction and
- {%0} is first arg of {%1}...
CAnDL

CONSTRANTS

PROGRAM

SOLVER

MATCH

MATCH

MATCH

- Off the shelf tools
- Discovers **exact** matches
Generalising

LLVM -> SPECIFIC CONSTRAINTS
Graph Matching
- Merge **multiple** graphs
- **Fuzzy** matching - optimise metric
- **Genetic** algorithm implementation
Summary

LIBRARY
FUNCTION

PROGRAM

MATCH
MATCH
MATCH
Migrating

- **Inline** every synthesised library call
- **Match** fully inlined code
- **Replace** match results
Match & Replace

PROGRAM

HANDWRITTEN

HANDWRITTEN

PROGRAM

NEW LIBRARY

NEW LIBRARY
Evaluation

APPLICATIONS
- NWChem
- Abinit
- PATHSAMPLE
- DARKNET
- PARBOIL

LIBRARIES
- MKL
- CUDA
- OPENCL
Performance Results

- ResNet-152
- VGG-16
- DenseNet-201
- Pathsample PFold
- Pathsample NGT
- Abinit Titanium
- Abinit Water
- NWChem Buckyball
- NWChem Pentacene
- Parboil SGEMM

Legend:
- Baseline
- MKL
- CUDA
Portability Story

- **ResNet-152**
  - Baseline: 0
  - OpenMP: 2
  - OpenCL: 4

- **VGG-16**
  - Baseline: 0
  - OpenMP: 5

- **DenseNet-201**
  - Baseline: 0
  - OpenMP: 1
  - OpenCL: 2
## Discovery Results

<table>
<thead>
<tr>
<th></th>
<th>SPMV</th>
<th>GEMM</th>
<th>GEMV</th>
<th>GER</th>
<th>AXPY</th>
<th>AXPBY</th>
<th>SCAL</th>
<th>COPY</th>
<th>DOT</th>
<th>SOFTMAX</th>
<th>RELU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abinit</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>180 (180)</td>
<td>47 (47)</td>
<td>21 (21)</td>
<td>2 (2)</td>
<td>20 (20)</td>
<td>70 (70)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0/0/180/180</td>
<td>0/0/47/47</td>
<td>0/2/2/2</td>
<td>20/20/20/20</td>
<td>70/70/70/70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>180/180/0/0</td>
<td>47/47/0/0</td>
<td>2/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>2/0/0/0</td>
<td>1/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Pathsampling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2 (0)</td>
<td>1 (0)</td>
<td>3 (0)</td>
<td>7 (0)</td>
<td>13 (0)</td>
<td>5 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0/0/2/2</td>
<td>0/0/1/1</td>
<td>3/3/3/3</td>
<td>7/7/7/7</td>
<td>13/13/13/13</td>
<td>5/5/5/5</td>
<td>1/1/1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>2/2/0/0</td>
<td>1/1/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NWChem</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2 (0)</td>
<td>2 (0)</td>
<td>2 (0)</td>
<td>2 (0)</td>
<td>27 (0)</td>
<td>2 (0)</td>
<td>2 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0/0/2/2</td>
<td>0/0/2/2</td>
<td>0/2/2/2</td>
<td>0/2/2/2</td>
<td>0/2/2/2</td>
<td>0/2/2/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>2/2/0/0</td>
<td>2/2/0/0</td>
<td>2/0/0/0</td>
<td>27/0/0/0</td>
<td>2/0/0/0</td>
<td>2/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Darknet</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>2 (1)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0/0/2/2</td>
<td>0/0/1/1</td>
<td>0/1/1/1</td>
<td>0/1/1/1</td>
<td>0/1/1/1</td>
<td>0/1/1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>2/2/0/0</td>
<td>1/1/0/0</td>
<td></td>
<td>1/0/0/0</td>
<td>1/0/0/0</td>
<td>1/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Parboil</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>1 (0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>0/1/1/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN</td>
<td>1/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discovery Results

- Graph matching *generalises* well
- **Few** false negatives
- False positives can be **eliminated**
- **Real** C, C++, and Fortran code

<table>
<thead>
<tr>
<th></th>
<th>SPMV</th>
<th>GEMM</th>
<th>GEMV</th>
<th>GER</th>
<th>AXPY</th>
<th>AXPBY</th>
<th>SCAL</th>
<th>COPY</th>
<th>DOT</th>
<th>SOFTMAX</th>
<th>RELU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abinit</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0/0/180</td>
<td>47 (47)</td>
<td>21 (21)</td>
<td>2 (2)</td>
<td>20 (20)</td>
<td>70 (70)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pathsam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NWChem</td>
<td></td>
<td>I</td>
<td>I</td>
<td>F</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Darknet</td>
<td></td>
<td>I</td>
<td>I</td>
<td>F</td>
<td>I</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parboil</td>
<td>FP</td>
<td>FN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1/0/0/0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

- Getting all 3 is a **hard** problem
- Program synthesis to **model**
- Constraints and graph matching to **search**
- **Inline** and replace
- **Performant and accurate**