THE UNIVERSITY of EDINBURGH
informatics

EPSRC Centre for Doctoral Training in
Pervasive Parallelism

Type-Directed Program Synthesis and Constraint Generation for Library Portability

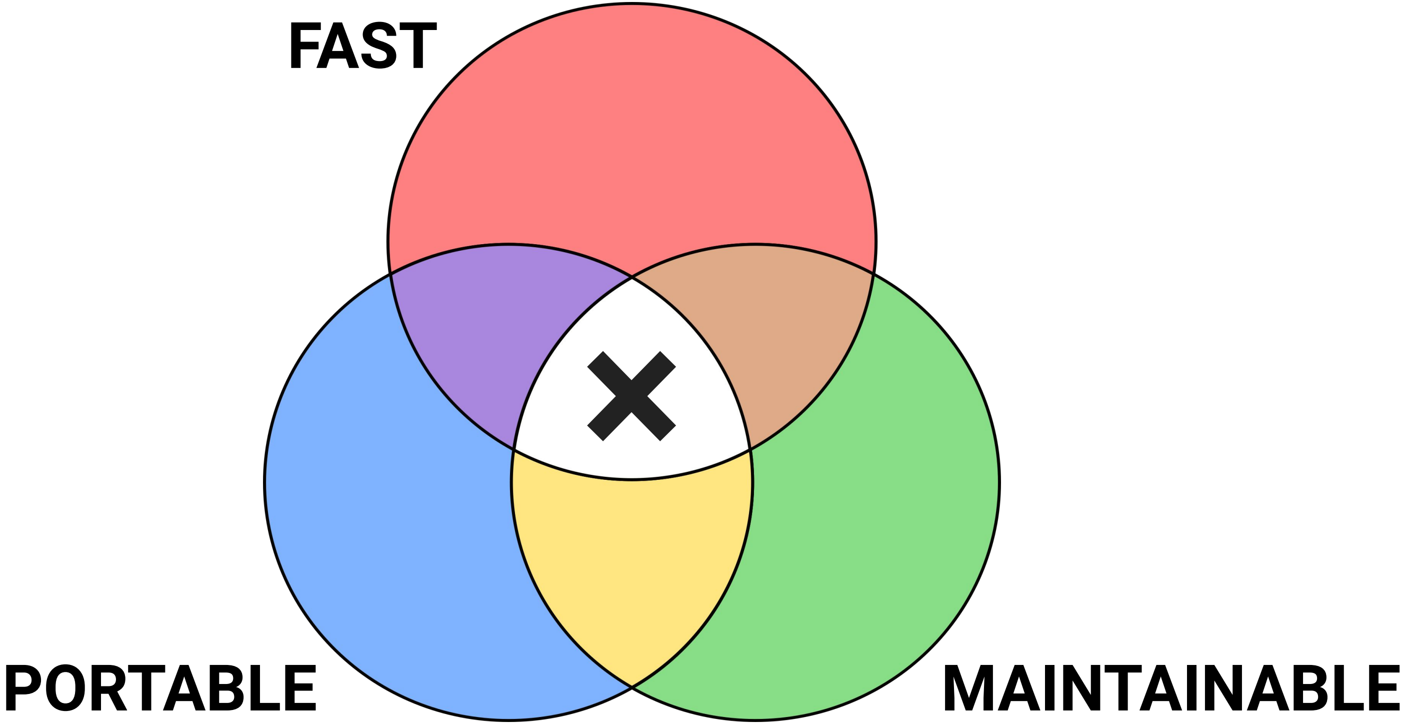
Bruce Collie, Philip Ginsbach and Michael O’Boyle
University of Edinburgh

bruce.collie@ed.ac.uk
baltoli.github.io

PACT 2019

A Heterogeneous World

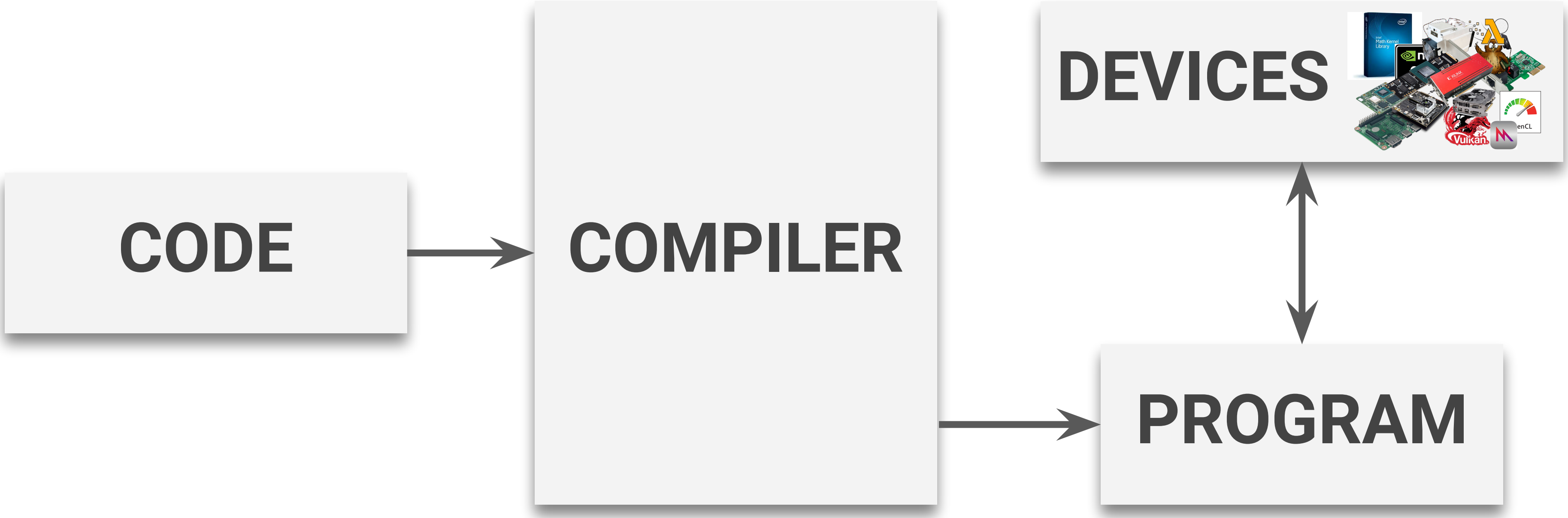
Triple Constraint



A Dire Warning

“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?



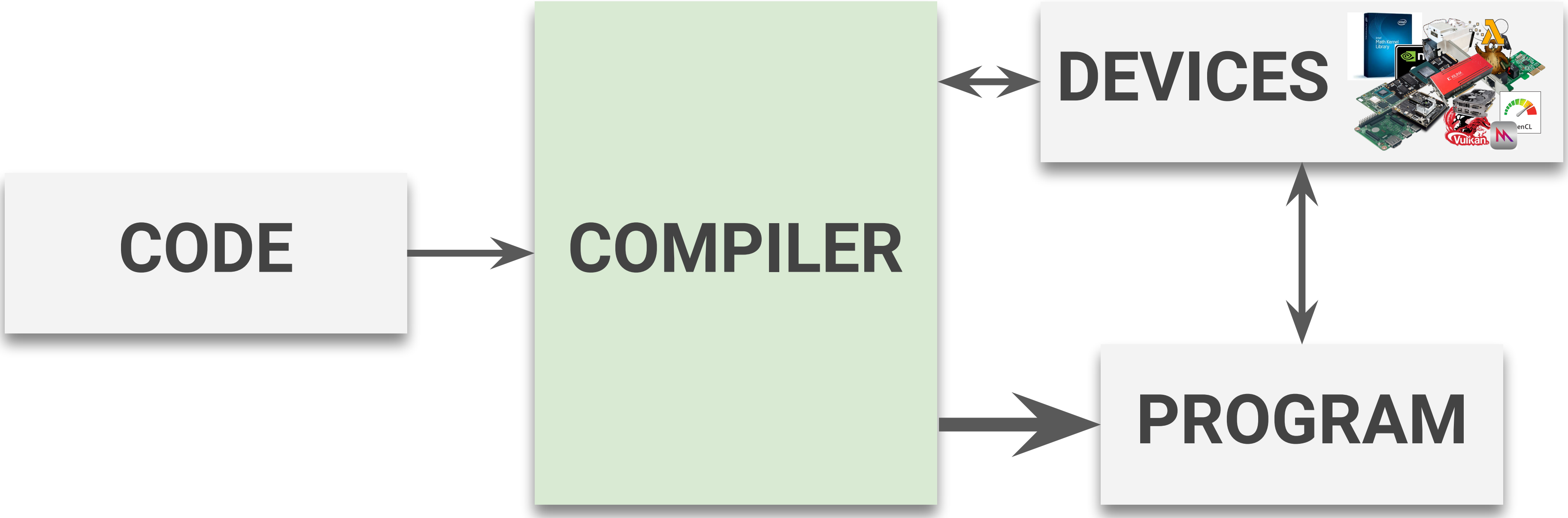
CODE

COMPILER

DEVICES

PROGRAM

Can we do better?



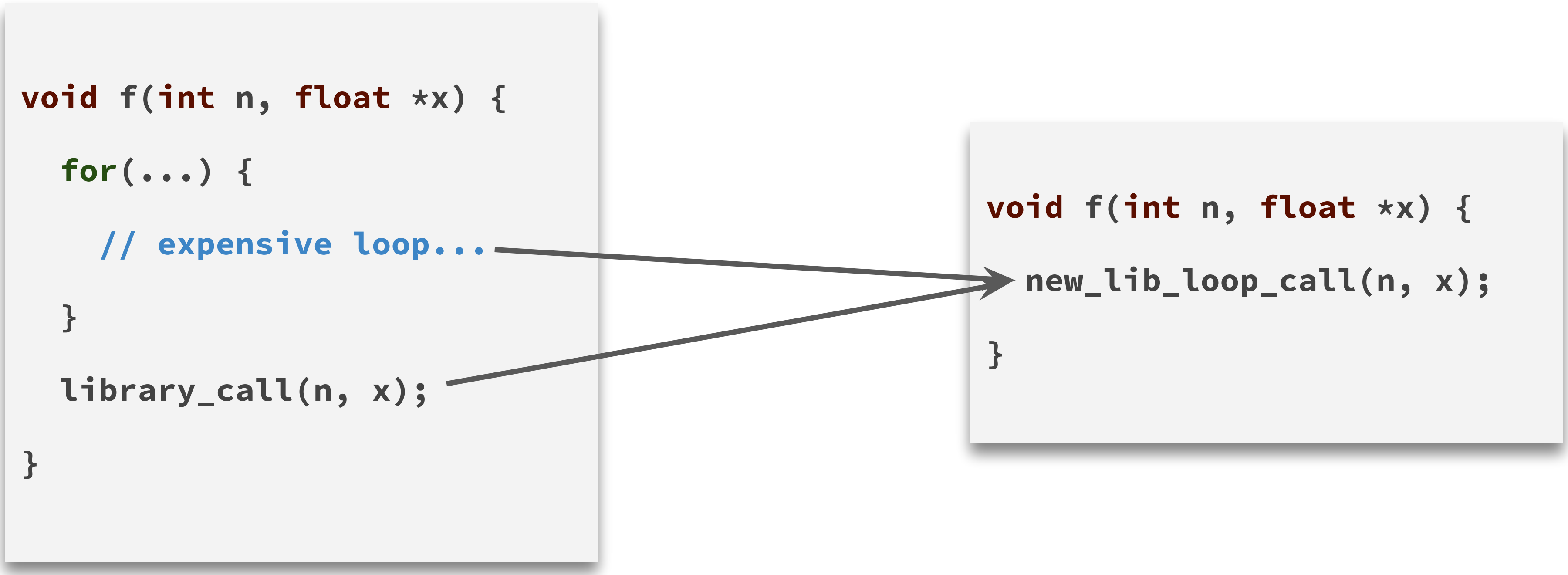
CODE

COMPILER

DEVICES

PROGRAM

Example



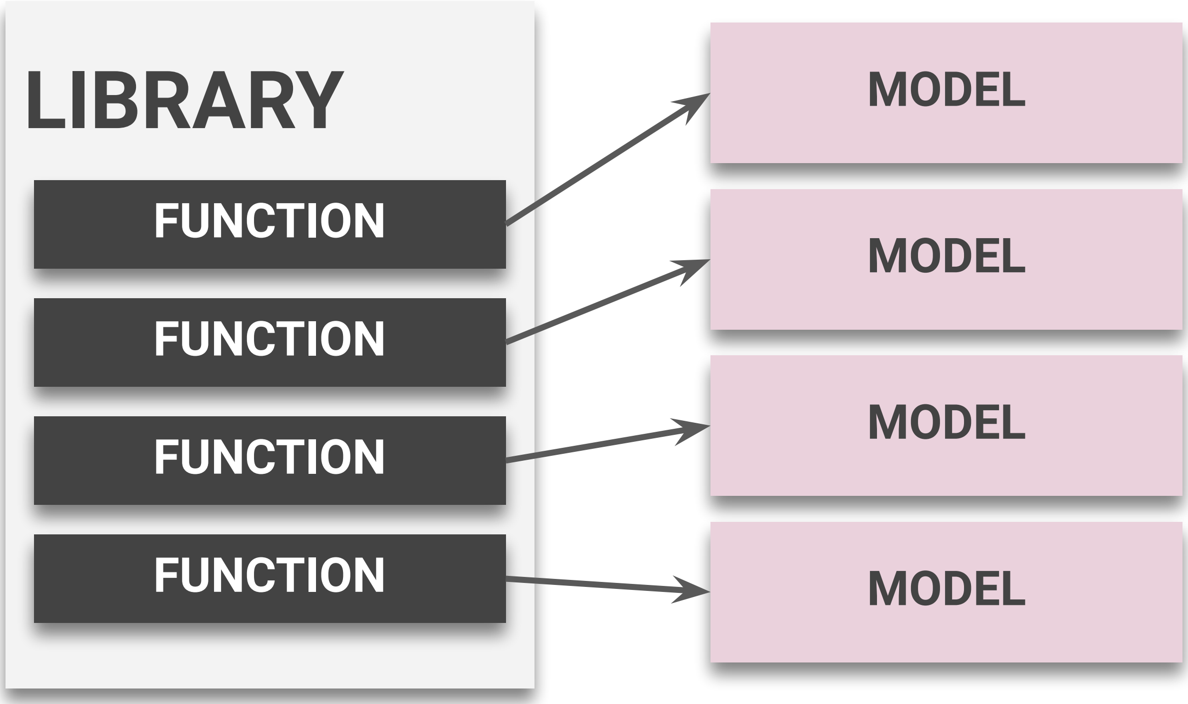
```
void f(int n, float *x) {
  for(...) {
    // expensive loop...
  }
  library_call(n, x);
}
```

```
void f(int n, float *x) {
  new_lib_loop_call(n, x);
}
```

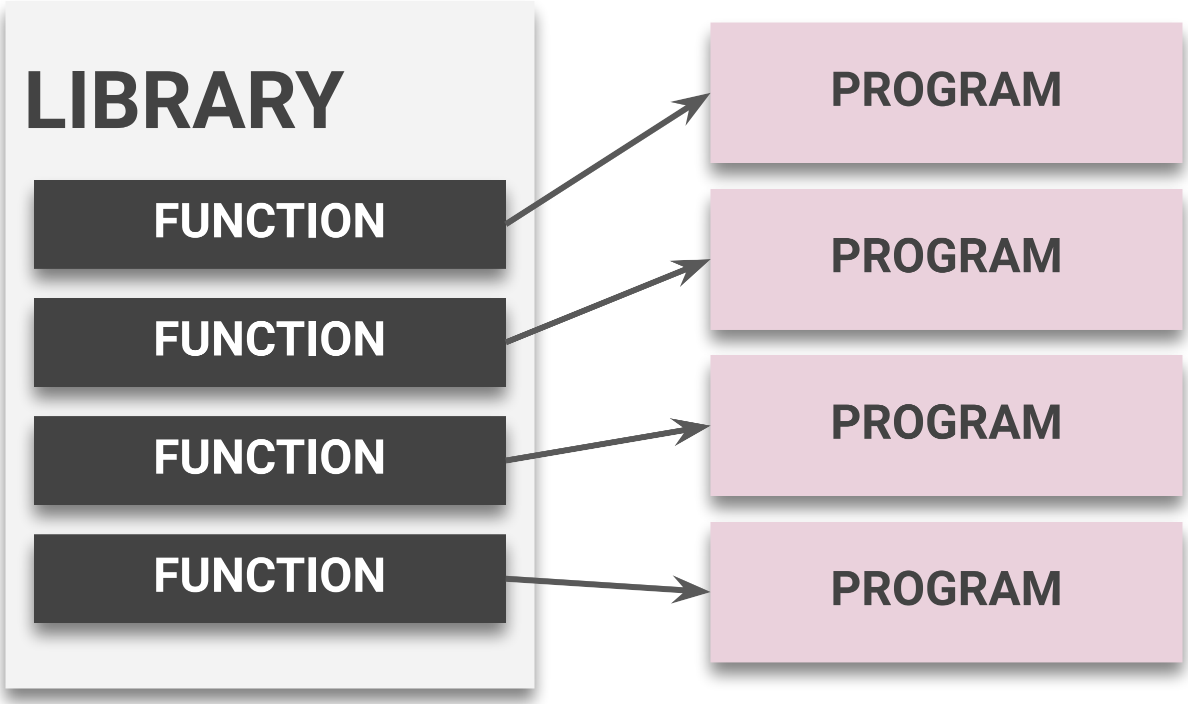
How to achieve this?

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

Learning


LIBRARY
FUNCTION
FUNCTION
FUNCTION
FUNCTION
MODEL
MODEL
MODEL
MODEL

Learning


LIBRARY

FUNCTION

FUNCTION

FUNCTION

FUNCTION

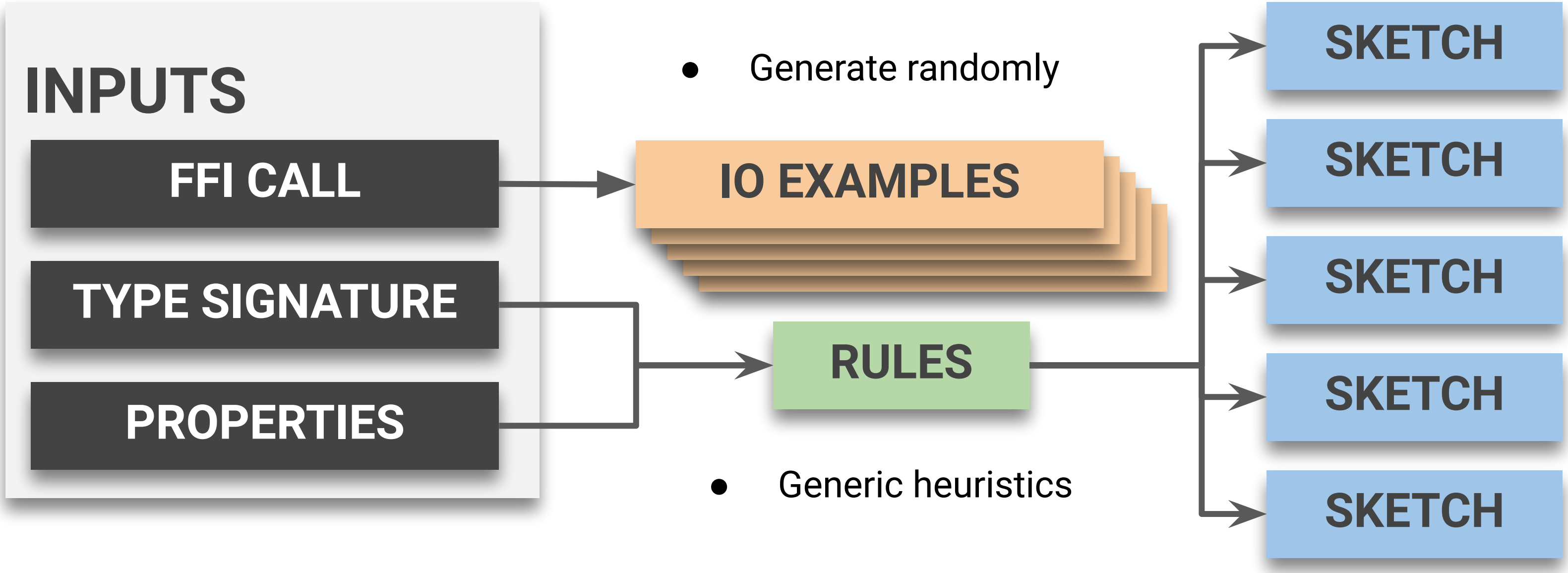
PROGRAM

PROGRAM

PROGRAM

PROGRAM

Synthesis



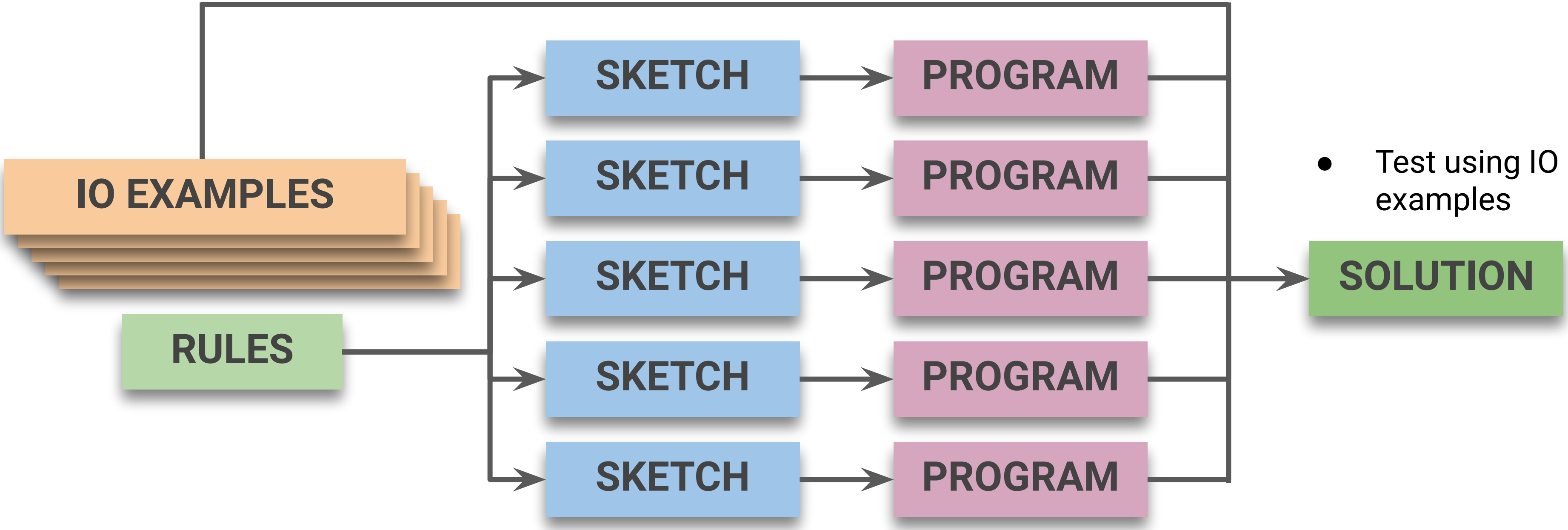
- Generate randomly

- Generic heuristics

- Interface properties
- From documentation

- Control flow
- Program structure

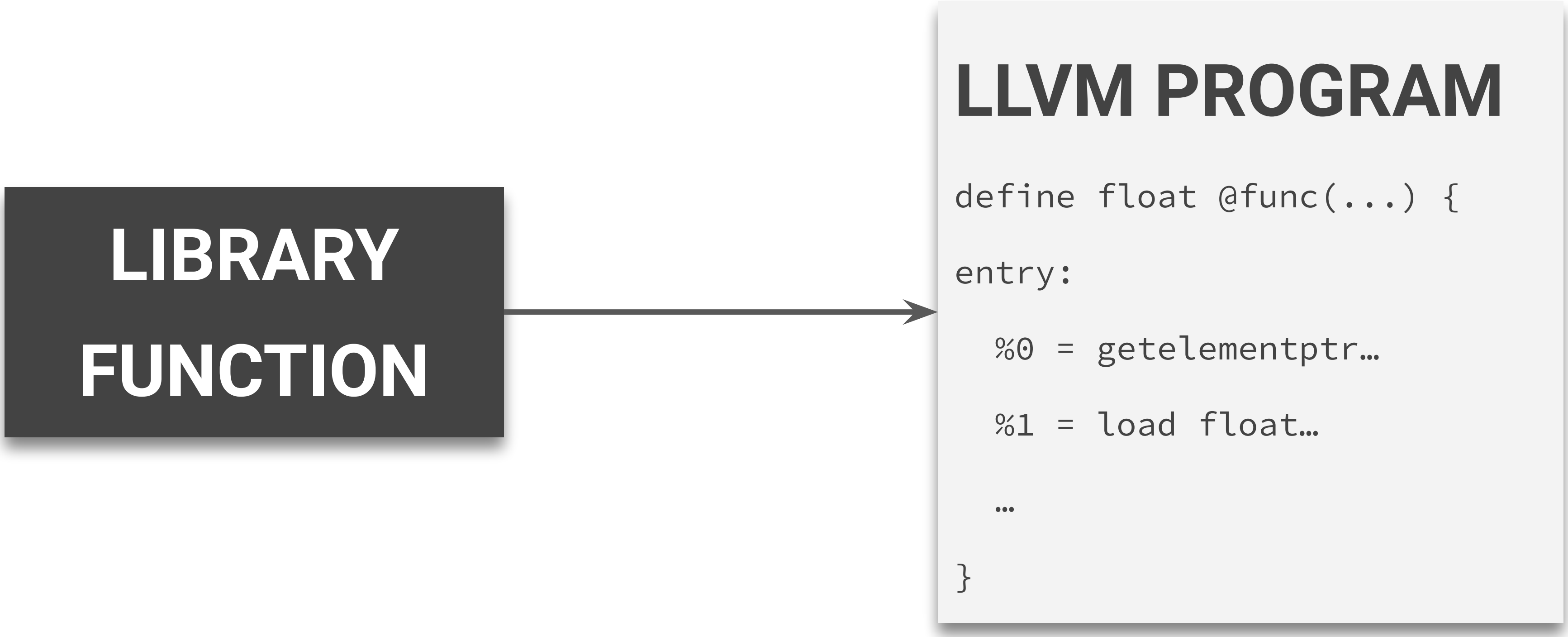
Synthesis



- Test using IO examples

- Search for instructions

Synthesis



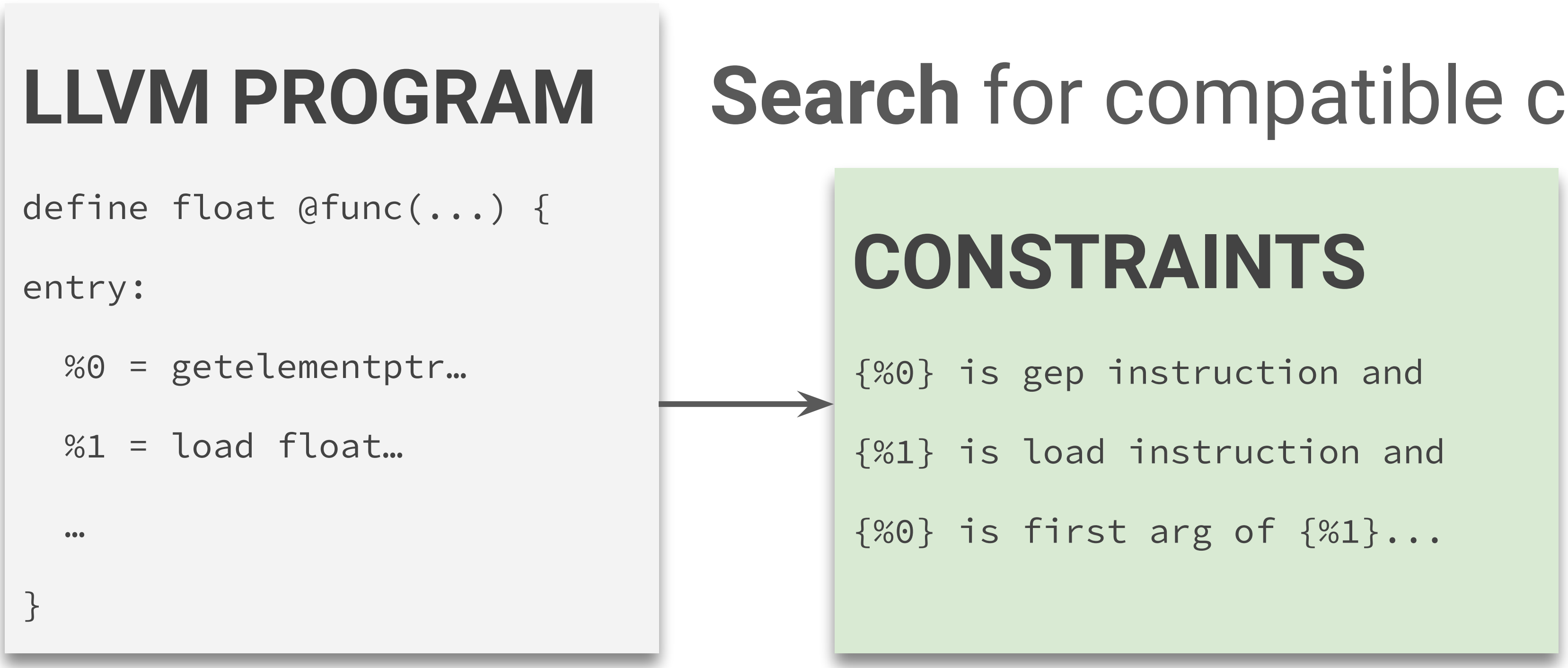
LIBRARY FUNCTION

LLVM PROGRAM

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

Search

Search for compatible code



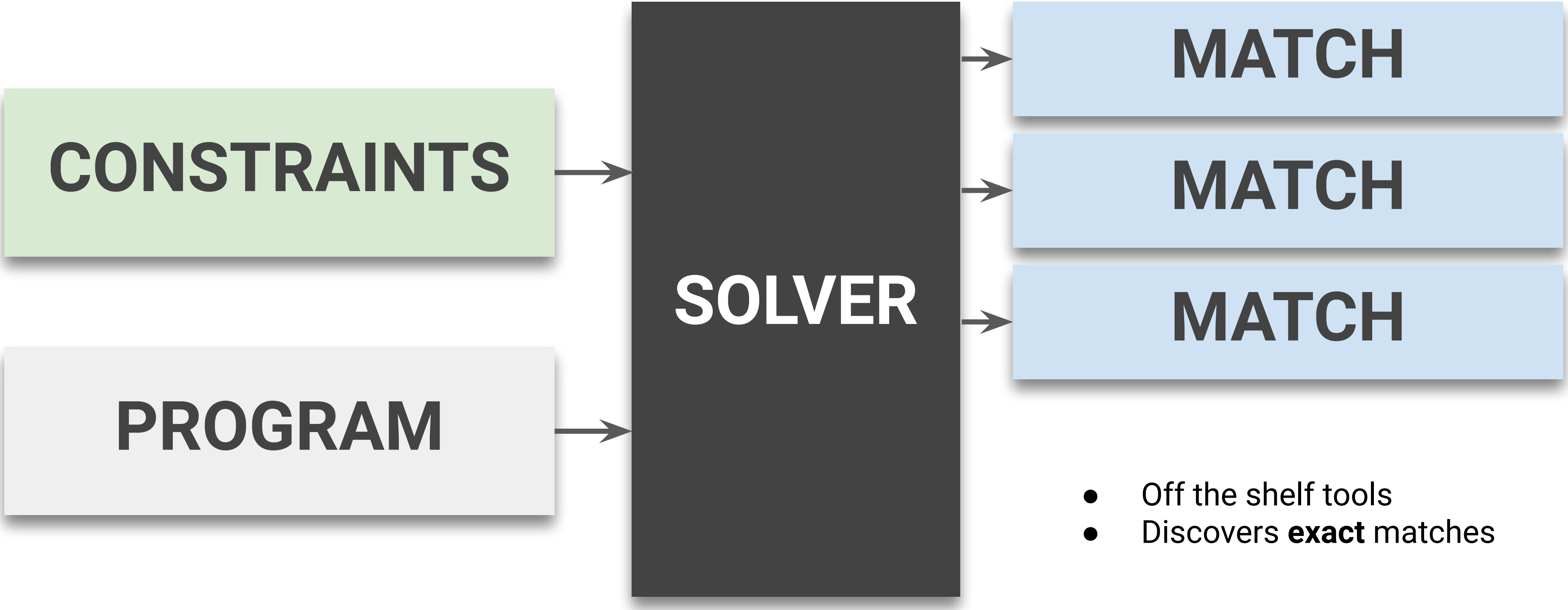
LLVM PROGRAM

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

CONSTRAINTS

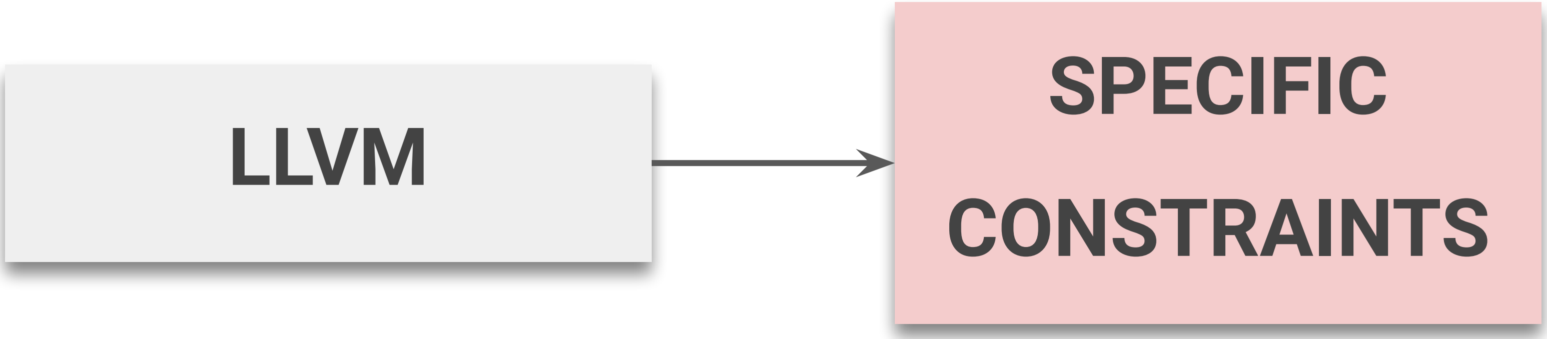
```
{%0} is gep instruction and
{%1} is load instruction and
{%0} is first arg of {%1}...
```

CAnDL



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

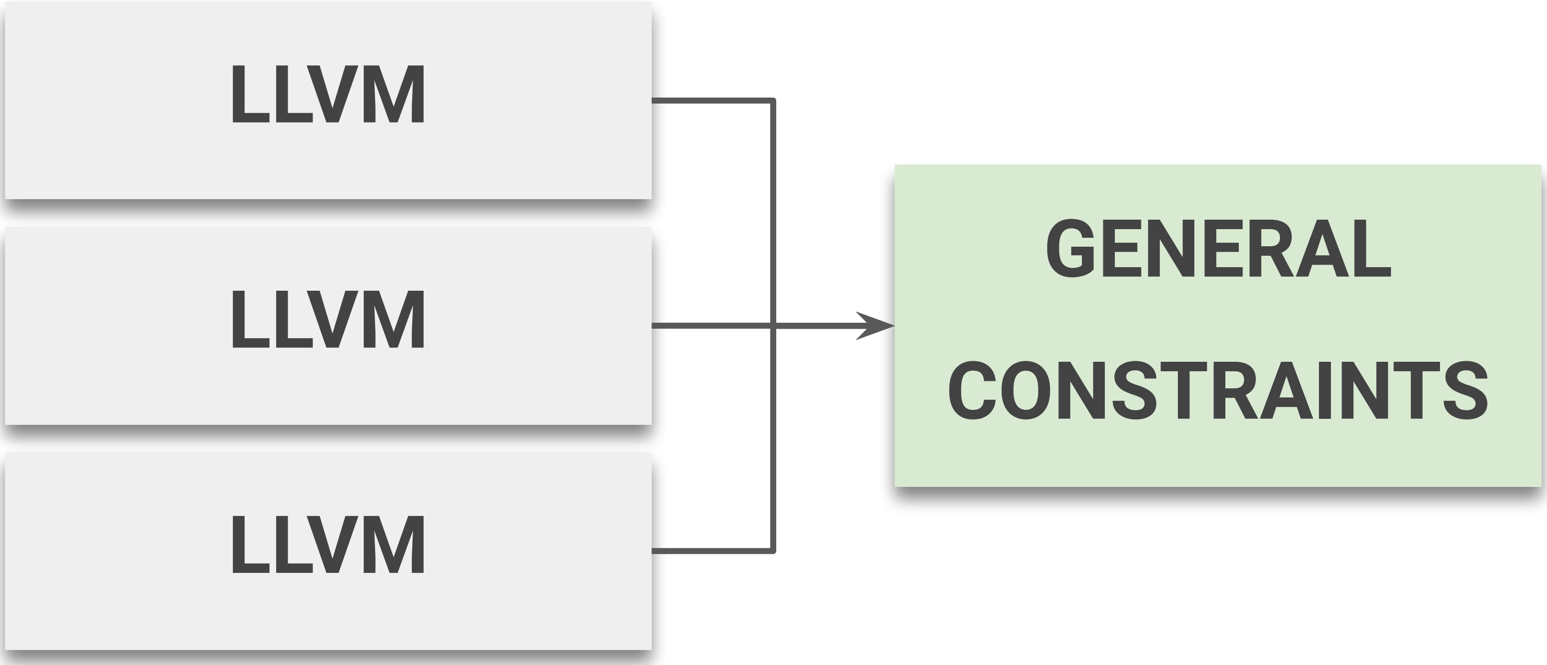
Generalising



LLVM

SPECIFIC CONSTRAINTS

Summary



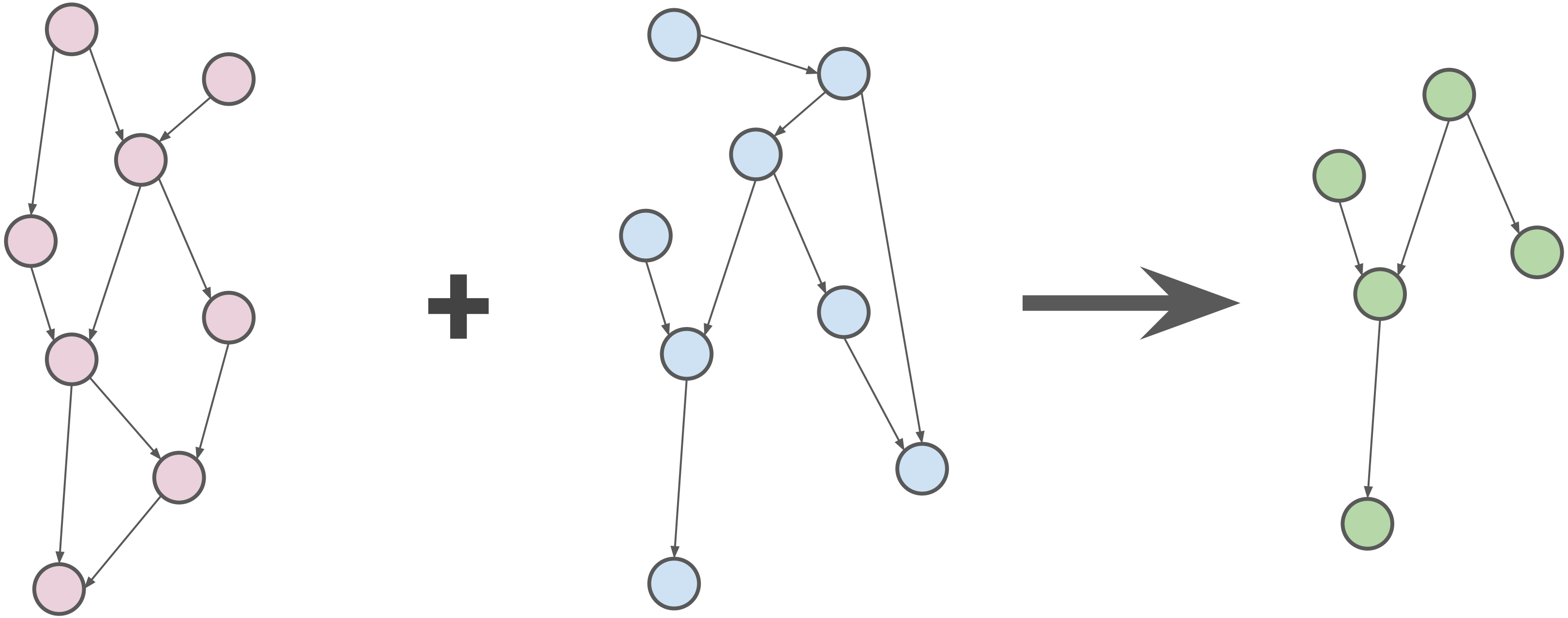
LLVM

LLVM

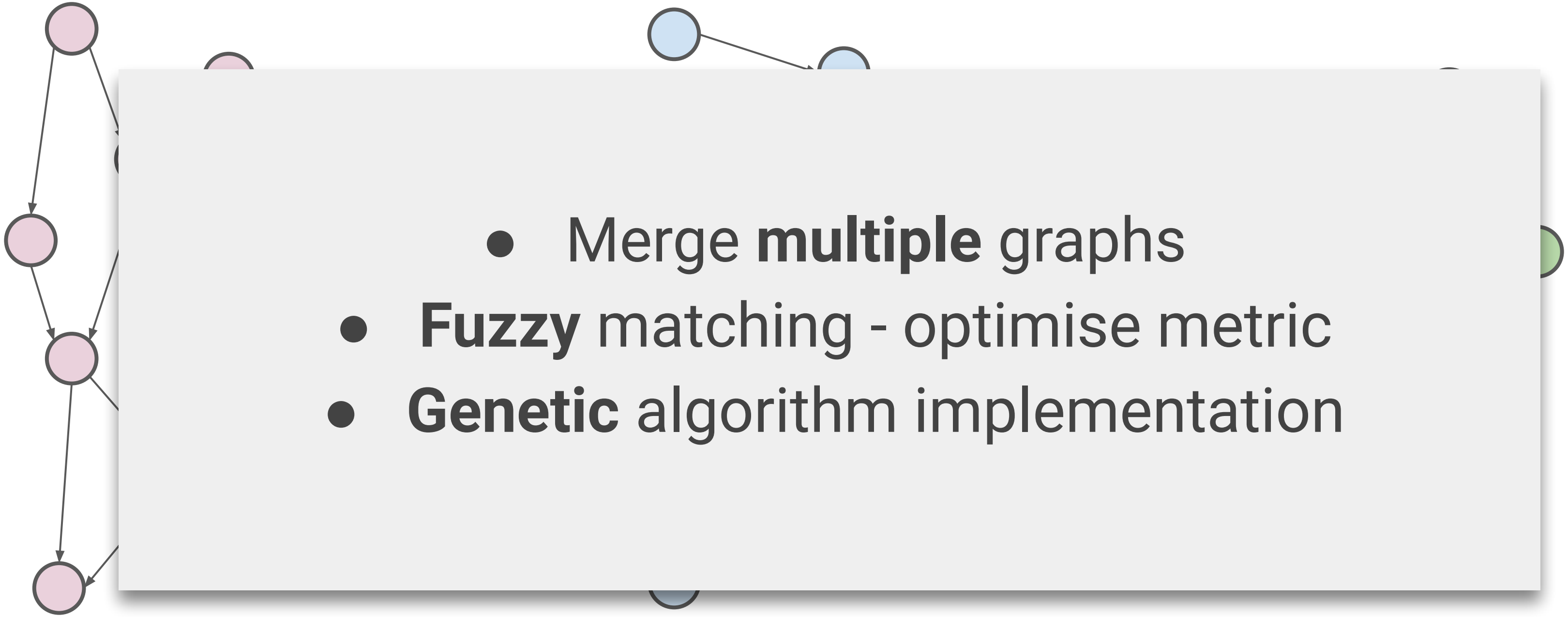
LLVM

GENERAL CONSTRAINTS

Graph Matching

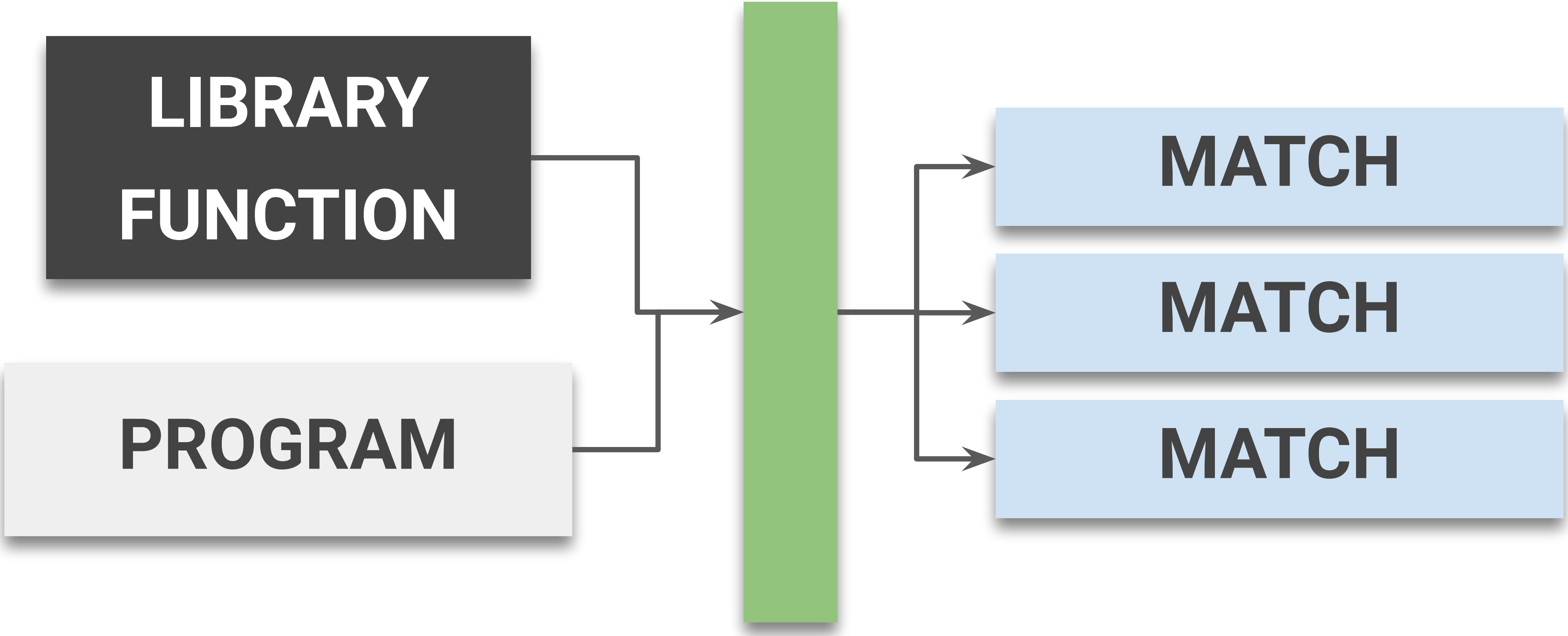


Graph Matching



- Merge **multiple** graphs
- **Fuzzy** matching - optimise metric
- **Genetic** algorithm implementation

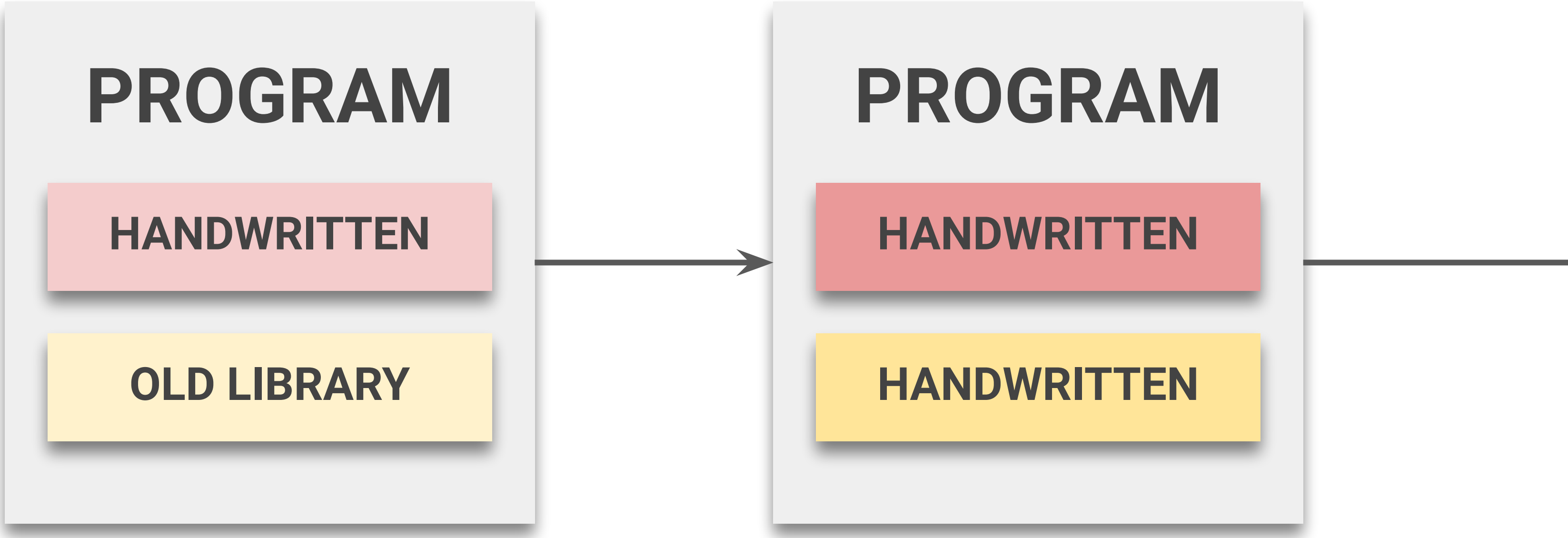
Summary



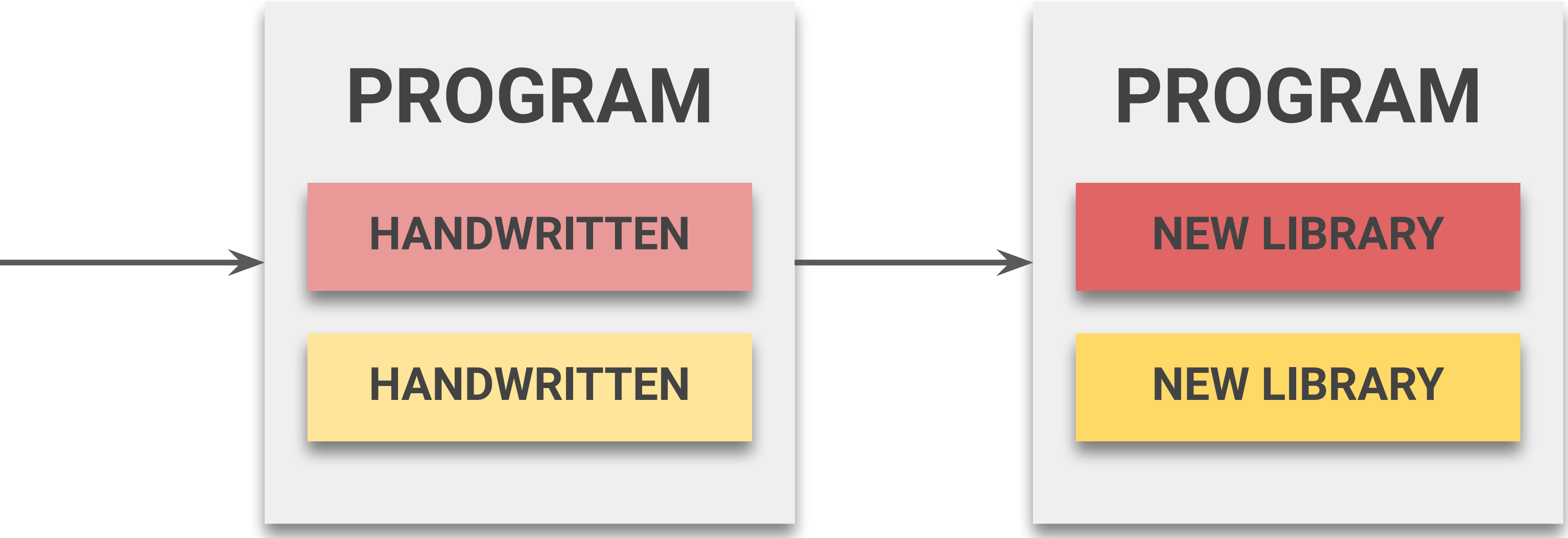
Migrating

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

Inline


PROGRAM
HANDWRITTEN
OLD LIBRARY
PROGRAM
HANDWRITTEN
HANDWRITTEN

Match & Replace



Evaluation

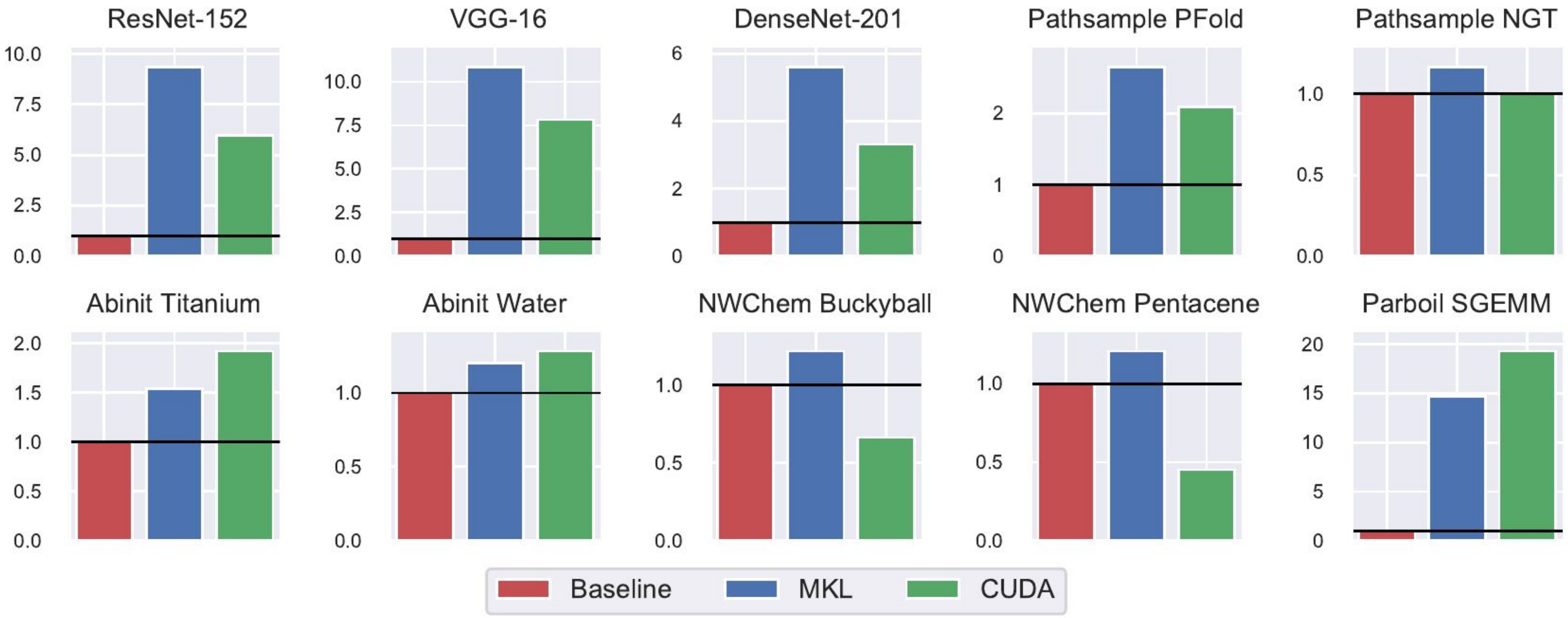
APPLICATIONS

- NWCHEM
- ABINIT
- PATHSAMPLE
- DARKNET
- PARBOIL

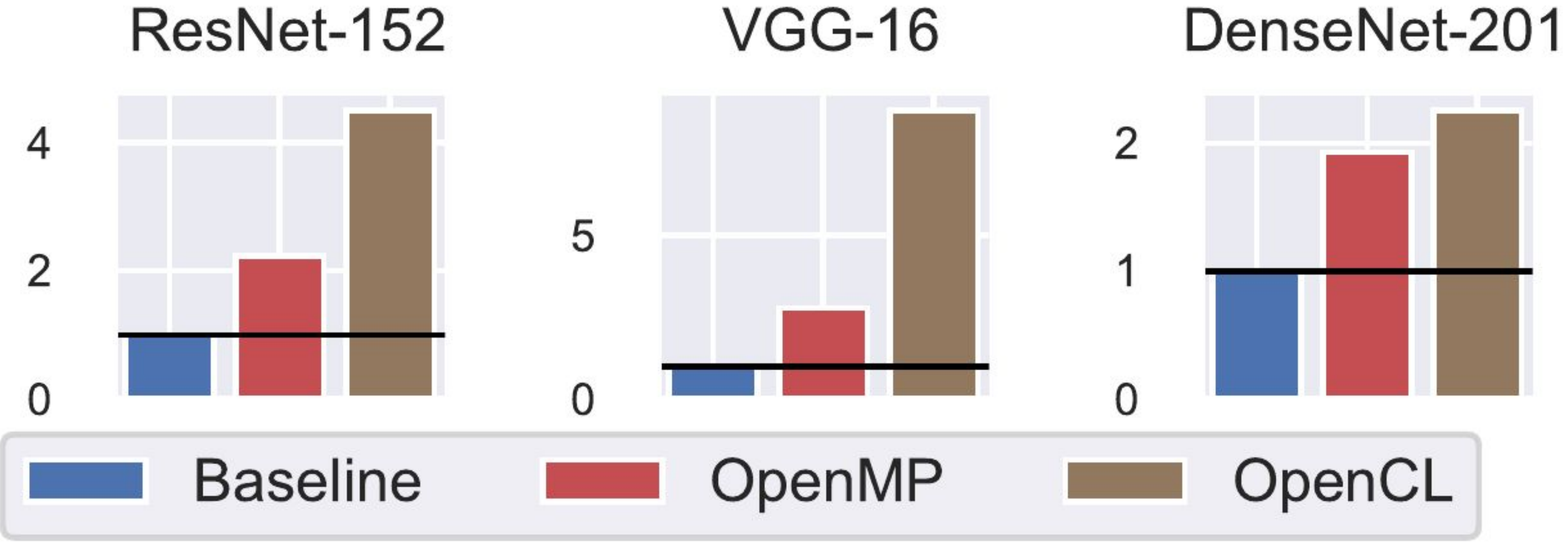
LIBRARIES

- MKL
- CUDA
- OPENCL

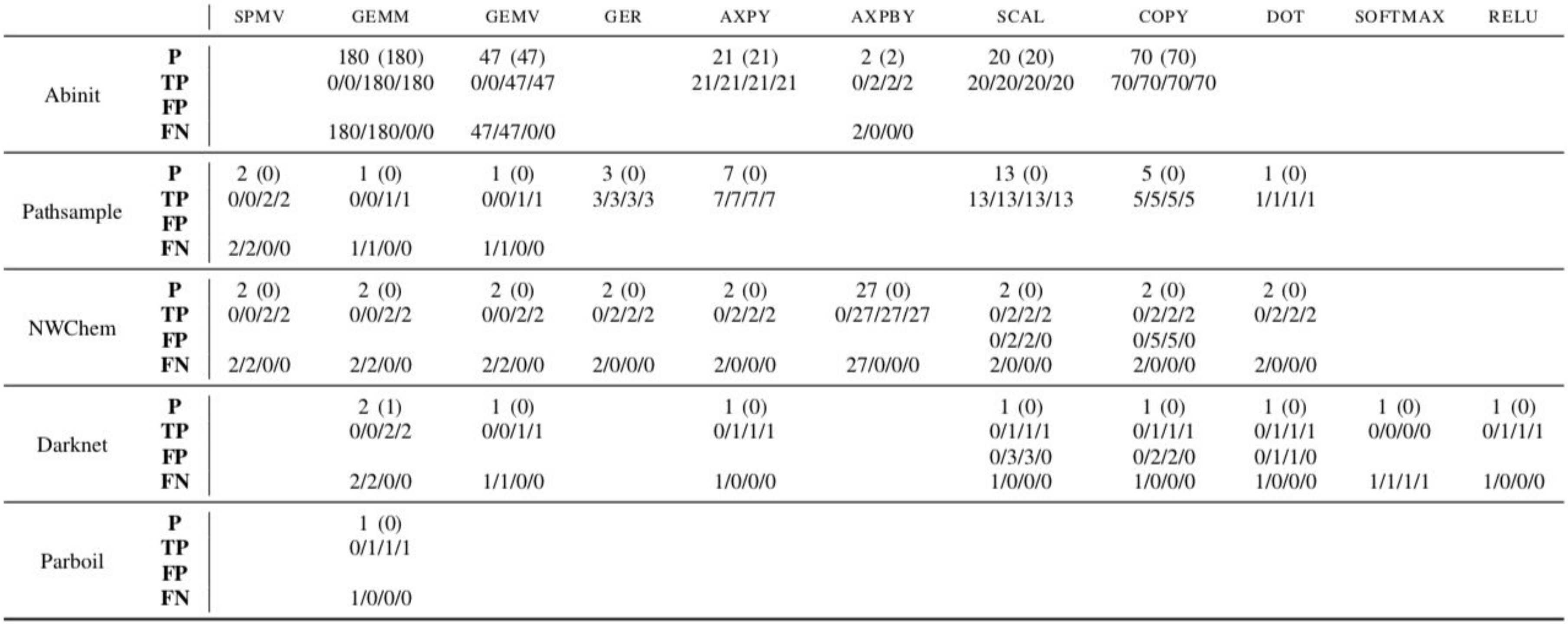
Performance Results



Portability Story

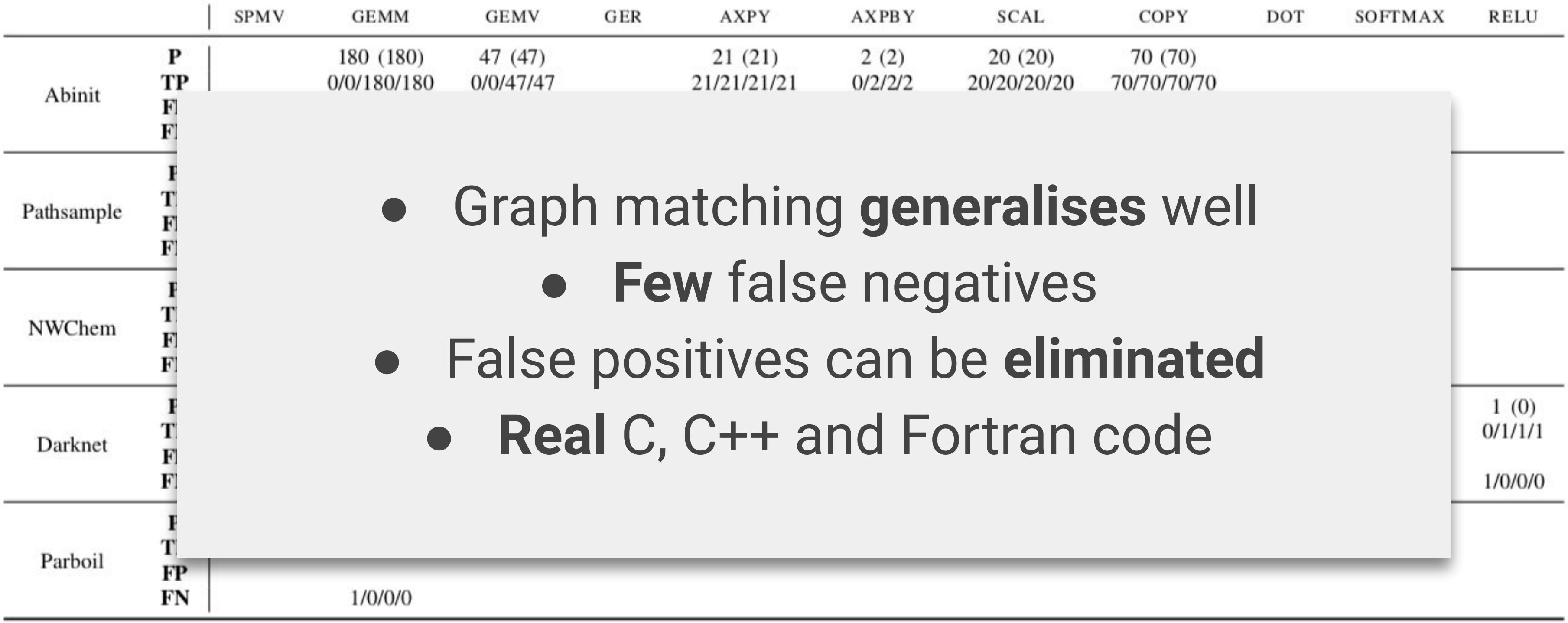


Discovery Results



		SPMV	GEMM	GEMV	GER	AXPY	AXPBY	SCAL	COPY	DOT	SOFTMAX	RELU
Abinit	P		180 (180)	47 (47)		21 (21)	2 (2)	20 (20)	70 (70)			
	TP		0/0/180/180	0/0/47/47		21/21/21/21	0/2/2/2	20/20/20/20	70/70/70/70			
	FP											
	FN		180/180/0/0	47/47/0/0			2/0/0/0					
Pathsample	P	2 (0)	1 (0)	1 (0)	3 (0)	7 (0)		13 (0)	5 (0)	1 (0)		
	TP	0/0/2/2	0/0/1/1	0/0/1/1	3/3/3/3	7/7/7/7		13/13/13/13	5/5/5/5	1/1/1/1		
	FP											
	FN	2/2/0/0	1/1/0/0	1/1/0/0								
NWChem	P	2 (0)	2 (0)	2 (0)	2 (0)	2 (0)	27 (0)	2 (0)	2 (0)	2 (0)		
	TP	0/0/2/2	0/0/2/2	0/0/2/2	0/2/2/2	0/2/2/2	0/27/27/27	0/2/2/2	0/2/2/2	0/2/2/2		
	FP							0/2/2/0	0/5/5/0			
	FN	2/2/0/0	2/2/0/0	2/2/0/0	2/0/0/0	2/0/0/0	27/0/0/0	2/0/0/0	2/0/0/0	2/0/0/0		
Darknet	P		2 (1)	1 (0)		1 (0)		1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	TP		0/0/2/2	0/0/1/1		0/1/1/1		0/1/1/1	0/1/1/1	0/1/1/1	0/0/0/0	0/1/1/1
	FP							0/3/3/0	0/2/2/0	0/1/1/0		
	FN		2/2/0/0	1/1/0/0		1/0/0/0		1/0/0/0	1/0/0/0	1/0/0/0	1/1/1/1	1/0/0/0
Parboil	P		1 (0)									
	TP		0/1/1/1									
	FP											
	FN		1/0/0/0									

Discovery Results

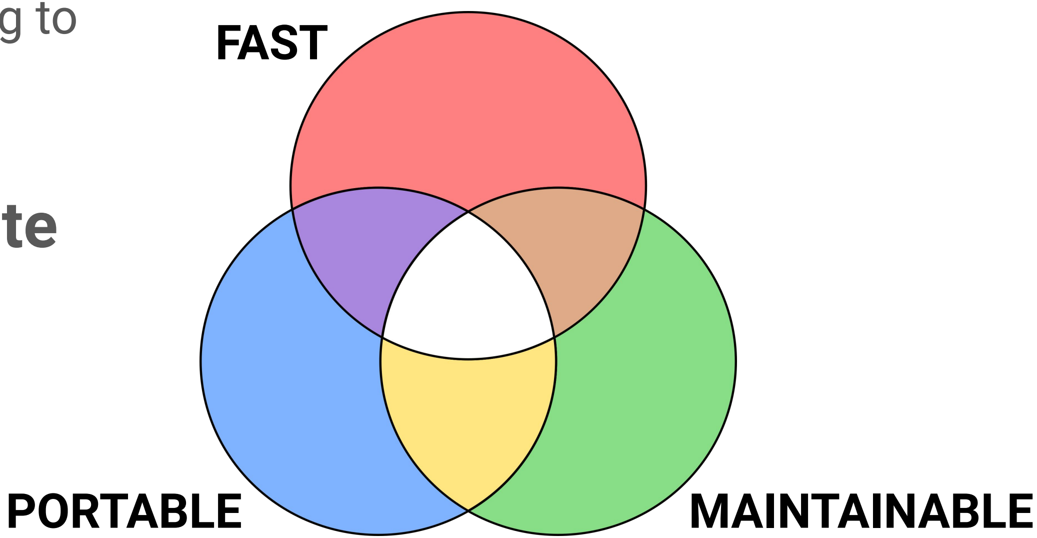


		SPMV	GEMM	GEMV	GER	AXPY	AXPBY	SCAL	COPY	DOT	SOFTMAX	RELU
Abinit	P		180 (180)	47 (47)		21 (21)	2 (2)	20 (20)	70 (70)			
	TP		0/0/180/180	0/0/47/47		21/21/21/21	0/2/2/2	20/20/20/20	70/70/70/70			
	F											
	F											
Pathsample	P											
	T											
	F											
	F											
NWChem	P											
	T											
	F											
	F											
Darknet	P											1 (0)
	T											0/1/1/1
	F											
	F											1/0/0/0
Parboil	P											
	T											
	FP											
	FN		1/0/0/0									

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

Summary

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


FAST
PORTABLE
MAINTAINABLE