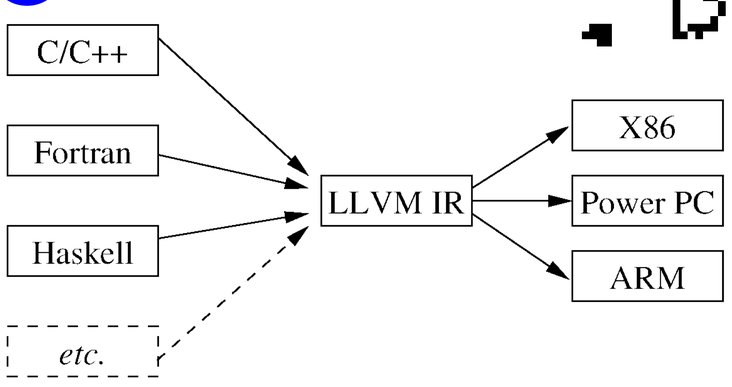
William B. Langdon, Afnan Al-Subaihin, Aymeric Blot, David Clark
Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK (Email: w.langdon@cs.ucl.ac.uk)

Genetic Improvement of LLVM Intermediate Representation

1 What is LLVM?



The LLVM project supports many programming languages. Compiling them to device independent intermediate code. LLVM-IR can be optimised. Intermediate code is compiled into device specific machine code.

LLVM-IR

```
size_t OLC_CodeLength(const char* code, size_t size) {
  CodeInfo info;
  analyse(code, size, &info);
  return code_length(&info);
}

; Function Attrs: noinline nounwind optnone uwtable
define dso_local i64 @OLC_CodeLength(i8* noundef %0, i64 noundef %1) #0 {
  %3 = alloca i8*, align 8
  %4 = alloca i64, align 8
  %5 = alloca %struct.CodeInfo, align 8
  store i8* %0, i8** %3, align 8
  store i64 %1, i64* %4, align 8
  %6 = load i8*, i8** %3, align 8
  %7 = load i64, i64* %4, align 8
  %8 = call i32 @analyse(i8* noundef %6, i64 noundef %7, %struct.CodeInfo* noundef %5)
  %9 = call i64 @code_length(%struct.CodeInfo* noundef %5)
  ret i64 %9
}
```

Local variables

- Strongly typed (eg i32, i8*, double)
- Single-Static Assignment
- Numbered registers and labels (must be in order)
- `define }` delimit scope. Local registers start again at 0 in next function

2 What is Genetic Improvement?

Genetic Improvement uses evolution to modify existing software. Typically GI is applied to human written source code but it can be applied to anything. E.g. C, C++, Java, Java byte code, assembler, even machine code. Non-program software could include comments, documentation and specifications.

3 What is Google's OLC

Open Location Code (OLC) plus codes can identify anywhere on Earth. OLC is open source

```
latlngolc.exe 57.101474 -2.242851
9C9V4Q24+HVJXR32
```

4 What is Uber's H3?

H3 is another geospatial indexing system. It uses a hexagonal grid that can be (approx- imately) subdivided into finer and finer levels. Like OLC it is an open source C industry standard.

```
h3.exe -r 15 --lat -2.242851 --lng 57.101474
8f8512959c55cb5
```

5 Delete lines of LLVM IR

By deleting local registers (i.e. set to zero) or forcing conditional branches, IR remains legal and so compiles and runs.

6 Pass 1: which lines can be deleted?

37-63% of mutants do not change output on test cases. These are used by pass 2.

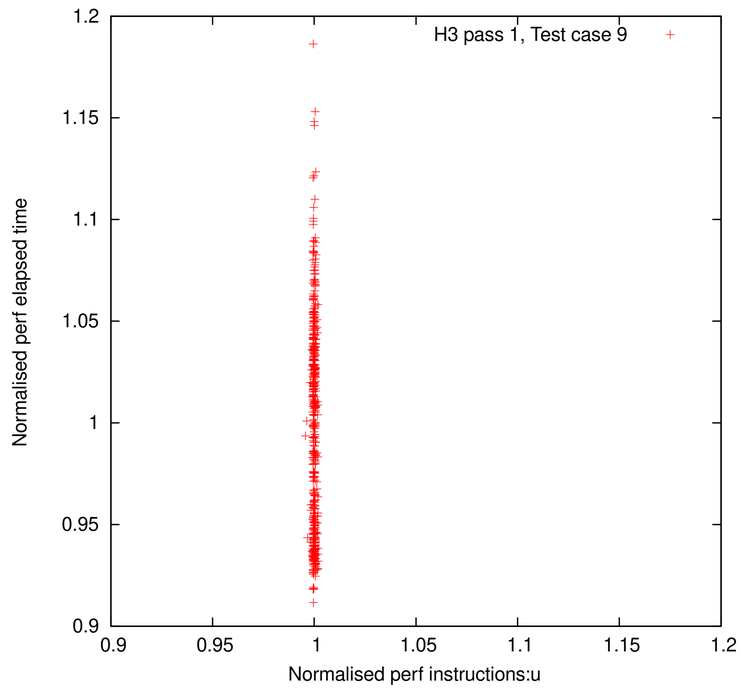
7 Pass 2: Use hill climber to join mutants

Start with fastest ok pass 1 mutant and add others only if they increase speed.

8 Fitness Function

- Is the mutated binary code different
- Does the mutant program run ok
- Are outputs same as unmutated code
- How long does perf say it took

```
timeout 2, limit cputime 2, limit filesize 1M
 perf stat -e instructions -x, -o perfout \
   mutant.exe 57.101474 -2.242851 >& output
```



perf instruction count more stable than elapsed time. Note large value tail

9 Post hill climb tidy: Bloat removed

Although in pass 2, each additional mutant makes the whole faster, noisy interactions mean after pass 2 the total change can be slimmed without loosing speedup.

10 Results

C	files	LOC	(used)	LLVM IR total	LLVM IR mutable	no output-change	Mutant size	Mutant speed up	Mutant holdout	GI duration
OLC	4	586	(127)	2546	294	141	2	698	682	5 minutes
-O3	4	586	(127)	2248	219	82	5	683	681	7 minutes
H3	43	5708	(1615)	19415	2113	955	51	2897	2631[a]	2.5 hours
-O3	43	5708	(1615)	15680	1762	1108	46	3272	2985	3.25 hours

[a] One holdout test failed

Example: H3 mutation 10508%74 saved 872 instructions by causing clang -O3 to remove condition before function doCoords (which must be called).

11 Overfitting Co-Evolution, Profile

H3 is larger and needed better training cases. There are almost unlimited possible test cases, perhaps co-evolution or white box fuzzing could help. LLVM supports profiling, which GI often uses

12 Summary

Evolving LLVM intermediate representation is widely applicable, as LLVM supports an increasing range of processors and programming languages. Genetic Improvement on IR in a few minutes or hours gave 0.5% (Google's OLC) and 2% (Uber's H3) speed up even on compiler optimised code for two industrial open source C programs.

Reference: Genetic Improvement of LLVM Intermediate Representation. W. B. Langdon, A. Al-Subaihin, A. Blot, D.Clark, EuroGP-2023. G. Pappa *et al*. Eds., Brno, Czech Republic. Springer LNCS 13986. doi:10.1007/978-3-031-29573-7_16