Progress report in Pen programming language

August 21, 2022

@raviqqe

Agenda

- Progress report
  - Lambda lifting (partially)
- Next plans

Progress report

Lambda lifting (partially)

- Flatten nested functions into global functions.
- Pen now implements its easy case where closures have no free variable.

References

- Lambda lifting | Wikipedia

Lambda lifting (partially)

Algorithm

Before:

```
f = \(x number) number {
  g = \(y number) number {
    # ...
  }

  # ...
}
```

Lambda lifting (partially)

Algorithm

After:

```
f = \(x number) number {
  g = lifted_g

  # ...
}

lifted_g = \(y number) number {
  # ...
}
```

Benchmark

- Algorithm: Sum of numbers
- A program with a lifted closure takes longer than a program with no closure.
- Why do we have so much difference between "No closure" and "Lifted closure"?

	Time (ms)
No closure	234.6
Lifted closure	768.5
Un-lifted closure	2986

Floating-pointer number optimization in LLVM

Pen:

```
sum = \(x number, i number) number {
  if i == 0 {
    x
  } else {
    sum(x + i, i - 1)
  }
}
```

LLVM:

```
else.i:
  %.tr47.i.int = phi i32 [ 100000000, %phi6 ], [ %.int, %else.i ]
  %.tr36.i = phi double [ 0.000000e+00, %phi6 ], [ %14, %else.i ]
  %indvar.conv = sitofp i32 %.tr47.i.int to double
  %14 = fadd double %.tr36.i, %indvar.conv
  %.int = add nsw i32 %.tr47.i.int, -1
  %15 = icmp eq i32 %.int, 0
```

Summary

- Progress
  - (Partial) lambda lifting
- Next plans
  - (Full) lambda lifting

Appendix

Benchmark result

```
> hyperfine -w 3 ./app ./app-lift ./app-unlift
Benchmark 1: ./app
  Time (mean ± σ):     234.6 ms ±   1.6 ms    [User: 182.2 ms, System: 2.0 ms]
  Range (min … max):   233.0 ms … 239.0 ms    12 runs

Benchmark 2: ./app-lift
  Time (mean ± σ):     768.5 ms ±   4.3 ms    [User: 715.3 ms, System: 3.0 ms]
  Range (min … max):   764.9 ms … 780.1 ms    10 runs

  Warning: Statistical outliers were detected. Consider re-running this benchmark on a quiet PC without any interferences from other programs. It might help to use the '--warmup' or '--prepare' options.

Benchmark 3: ./app-unlift
  Time (mean ± σ):      2.986 s ±  0.070 s    [User: 2.929 s, System: 0.004 s]
  Range (min … max):    2.941 s …  3.136 s    10 runs

  Warning: Statistical outliers were detected. Consider re-running this benchmark on a quiet PC without any interferences from other programs. It might help to use the '--warmup' or '--prepare' options.

Summary
  './app' ran
    3.28 ± 0.03 times faster than './app-lift'
   12.73 ± 0.31 times faster than './app-unlift'
```