Progress report in the Pen programming language

@raviqqe

Progress report

List comprehension

- An easy way to construct lists with computation of their elements.
- Monadic
  - e.g. Haskell, Python

In Haskell

```
[x + y | x <- xs, y <- ys, x != y]
```

In Pen

The syntax is borrowed from Python.

```
[number x() + y() for y in ys for x in xs if x != y]
```

List comprehension

Concrete examples (1)

Map

```
[number f(x()) for x in xs]
```

Filter

```
[number x() for x in xs if Remainder(x(), 2) == 0]
```

Flatten

```
[number x() for x in xs() for xs in xss]
```

List comprehension

Concrete examples (2)

Permutate

```
[number f(x(), y()) for y in ys() for x in xs]
```

Filter by a type

```
[number
  x()
  for x in if x = x() as number { [number x] } else { [number] }
  for x in xs
]
```

List comprehension

Thoughts

- One of Pen's philosophy is to be a minimal language.
  - Where language features are orthogonal.
  - In the same way as Go
    - https://go.dev/talks/2010/ExpressivenessOfGo-2010.pdf
- Thus, there is no syntax sugar and AST and HIR is one to one.
- It's tiresome to experiment with new language features!
- On the other hand, you just transpile list comprehension with `do` notation or monadic operations in Haskell.

Future work (ideas)

Parallel list comprehension

- Natural extension to list comprehension for zip-ish computation
- Not related to parallel computation

In Haskell

```
[x + y | x <- xs | y <- ys]
```

In Pen

```
[number x() + y() for x, y in xs, ys]
```

Performance optimization

- Lazy lists
  - List fusion
    - Removal of intermediate lists
    - Is this easy to implement for impure languages?
  - Thunk optimization
- Heavy use of thunks
  - Constant propagation
    - Thunk to function conversion
  - Inlining
- Stack operations
  - How much can LLVM understand and optimize tail-called functions?

Near-future work

- More little language features
  - Parallel list comprehension
  - `sort` built-in function
- Code generator
- Language server

Summary

- Pen has monadic list comprehension now!
- I want to make progress...