Genetic Algorithm

Anh Huynh

Problem Overview

Given the digits 0 through 9 and the operators +, -, * and /, find a sequence that will represent a given target number. The operators will be applied sequentially from left to right as you read.

Example: 10 = 7 * 6 - 2 / 4

^ Solve this using genetic algorithm + Python + Pymonad.

Genetic Algorithm Overview

1. Generate a random first generation.
2. Pick out the best ones (using roulette selection).
3. Breed them together to generate a new generation.
4. Repeat step 2-3 until a solution is found.

Chromosomes

Crossover:

000000000 + 111111111 = 000000111 111111000

Mutation:

000001111 = 010001101

Decoding

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	+
1011	-
1100	*
1101	/
1110	%
1111	_

Decoding

0000	0	1000	8
0001	1	1001	9
0010	2	1010	+
0011	3	1011	-
0100	4	1100	*
0101	5	1101	/
0110	6	1110	%
0111	7	1111	_

```
0111 1100 0110 1011 0010 1101 0100

 7    *    6    -    2    /    4


4 _ 2 = 42
```

Functional Programming with Python

```
[chrom_map[''.join([str(x) for x in xs])] for xs in split_four(chrom)]
```

```
{to_str(child):fitness(child, target) for child in generation}
```

```
reduce_tokens(ys[0], ys[1:])
```

```
return (xs[0:point] + ys[point:], ys[0:point] + xs[point:]) \
       if random() < rate else (xs, ys)
```

Pymonad

```
@curry
def evaluate_values(xs):
```

```
evaluate = v * evaluate_values * decode
```

Values

- ➔ (Just 6, +) → 6 +
- ➔ (Nothing, +) → +

- ➔ Identity - (N, _)

- ➔ (6, -) + (2, *) = (4, *)
- ➔ 6 - 2 * = 4 *

```
def evaluate_values(xs):
    return reduce(lambda x, y: x + y, xs)
```

Demo Time!

Randomness

Summary + Q&A

Genetic algorithm

Functional programming with Python

Pymonad