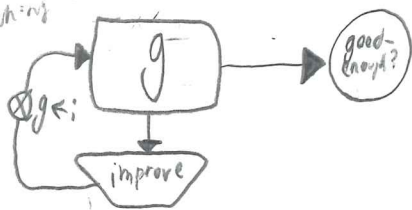


5.3 | a) Assuming good-enough? & improve as primitives

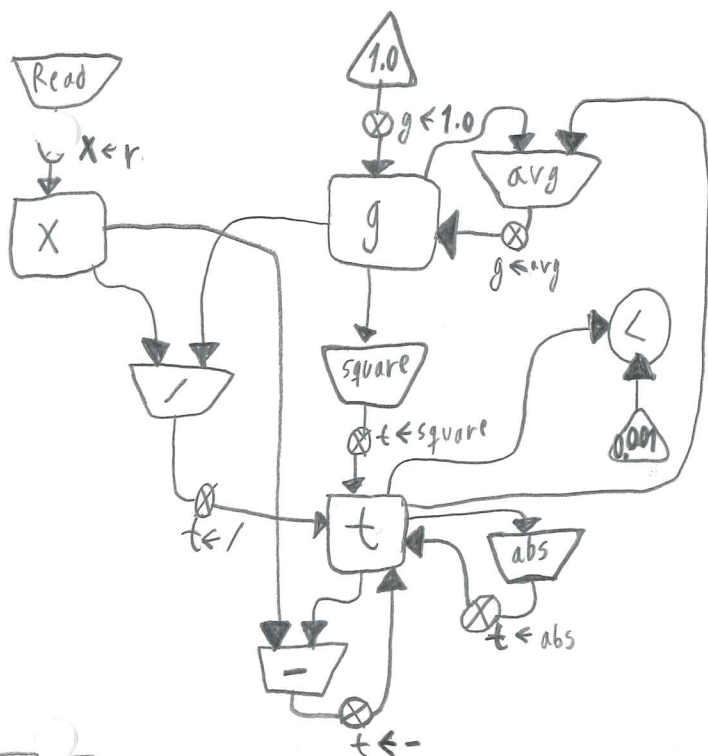
Note: diagrams would've been generated from branching data paths. e.g:



(controller
test-g

```
(test (op good-enough?) (reg g))
(branch (label sqrt-done))
(assign g (op improve) (reg g))
(goto (label test-g))
sqrt-done)
```

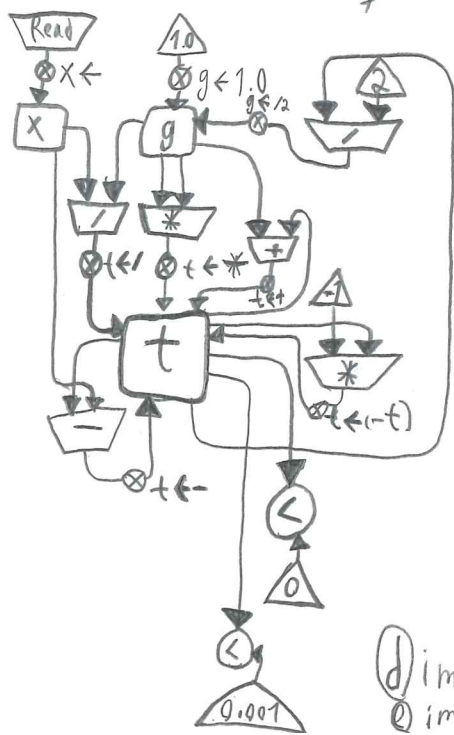
b) Assuming good-enough? & improve, while civilized, are made up of primitives like abs, square, average...



(controller

```
(assign X (op read))
(assign g (const 1.0))
test-g
(assign t (op square) (reg g))
(assign t (op -) (reg t) (reg x))
(assign t (op abs) (reg t))
(test (op <) (reg t) (const 0.001))
(branch (label sqrt-done))
(assign t (op 1) (reg x) (reg g))
(assign g (op avg) (reg g) (reg t))
(goto (label test-g))
sqrt-done)
```

c) Assuming abs, square & avg are also civilized.



(controller

```
(assign x (op read))
(assign g (const 1.0))
test-g
(assign t (op *) (reg g) (reg g))
(assign t (op -) (reg t) (reg x))
(test (op <) (reg t) (const 0))
(branch (label absed-t))
(assign t (op *) (reg t) (const -1))
absed-t
(test (op <) (reg t) (const 0.001))
(branch (label sqrt-done))
(assign t (op 1) (reg x) (reg g))
(assign t (op +) (reg t) (reg g))
(assign g (op 1) (reg t) (const 2))
(goto (label test-g))
sqrt-done)
```

d) implement * & / in terms of + & -
e) implement + & - in terms of peano arithmetic. ∴