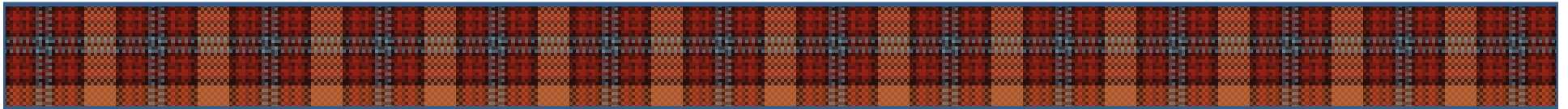


# Holy States Can Save the World!



Brother Jonathan Aldrich  
High Monk of the Plaid Brotherhood  
SIGBOVIK '10

# Imperative Programming is Evil

Consider the wanton destruction that is executed on every assignment

$$x = x + 1$$

The carnage must be stopped!



# Functional Programming is Evil

- Proliferation of data structures creates mountains of garbage
- Attempts at “compacting, composting garbage collection” are doomed to fail [Donham, SIGBOVIK '07]

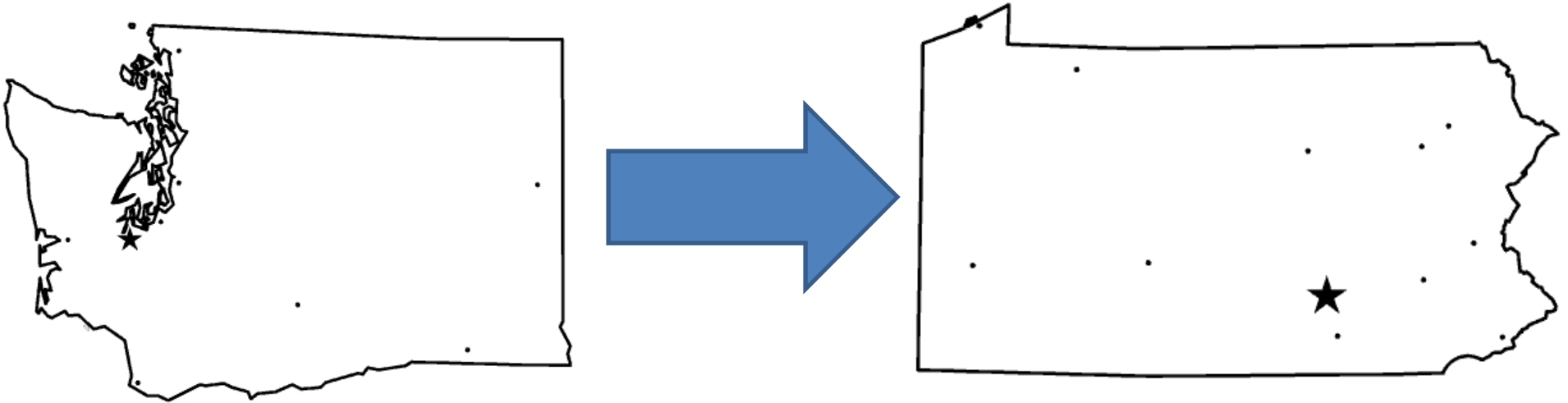


# The One Plaiddish Way

- Banish assignment
- Banish garbage
- But how then can we compute?
- With apologies to the One True Coding Style [SIGBOVIK '09]



# The Holy Sacrament of State Change



- When no longer needed objects can be **born again** in a new state

# St. Turing's Revelation



- St. Turing revealed that all computation can be expressed by changing states. Here, we show how in the holy **PLAID** language

**PLAID**

# Representing Cells with Holy States

```
1  state Cell {
2      method getLeft() {
3          left;
4      }
5      method getRight() {
6          right;
7      }
8      val left;
9      val right;
10
11     method print() { ... }
12 }
```

# Infinite Tapes with Holy States

```
14 state LeftEnd {
15     method getLeft() {
16         val me = this;
17         val myLeft = new LeftEnd with Zero {
18             right = me;
19         };
20         val myRight = this.getRight();
21
22         this <- Cell { left = myLeft; right = myRight; };
23
24         left;
25     }
26     // getRight(), etc. as in Cell
27 }
```

The Holy Sacrament  
of State Change



# Infinite Tapes with Holy States

```
14 state LeftEnd {  
15     method getLeft() {  
16         val me = this;  
17         val myLeft = new LeftEnd with Zero {  
18             right = me;  
19         };  
20         val myRight = this.getRight();  
21  
22         this <- Cell { left = myLeft; right = myRight; };  
23
```

Some infidels might suggest laziness, but Plaid theology holds sloth to be one of the 6 deadly sins.

*in Cell*

**The Holy Sacrament  
of State Change**

# Writing to the Tape without Assignment

```
1 state Zero {  
2     method writeZero() {}  
3     method writeOne() {  
4         this <- One;  
5     }  
6     method printVal() {  
7         java.lang.System.out.print("0");  
8     }  
9 }
```

The Holy Sacrament  
of State Change

# Controlling Turing's Machine

```
1 state Beaver2B {
2   val cell;
3
4   method update() {
5     match (cell) {
6       case Zero {
7         cell.writeOne();
8         val newCell = cell.getLeft();
9         this <- Beaver2A { cell = newCell; };
10      }
11     case One {
12       cell.writeOne();
13       val newCell = cell.getRight();
14       this <- Halt { cell = newCell; };
15     }
16   };
17 }
```

The Holy  
Sacrament of  
State Change

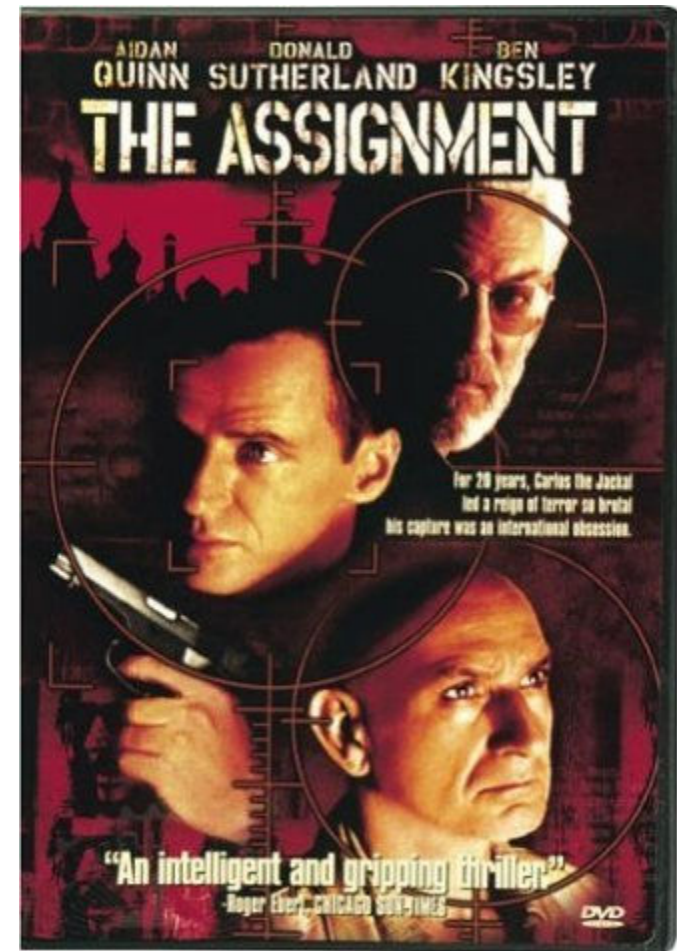
# Beware!

- The creators of the Plaid language have regrettably included both assignment and functional programming in Plaid
- The Supreme Revolutionary Plaid Council of Pittsburgh has declared a Fatwa against these features.
- The penalty for violation is the eternal torment of programming in Cobol

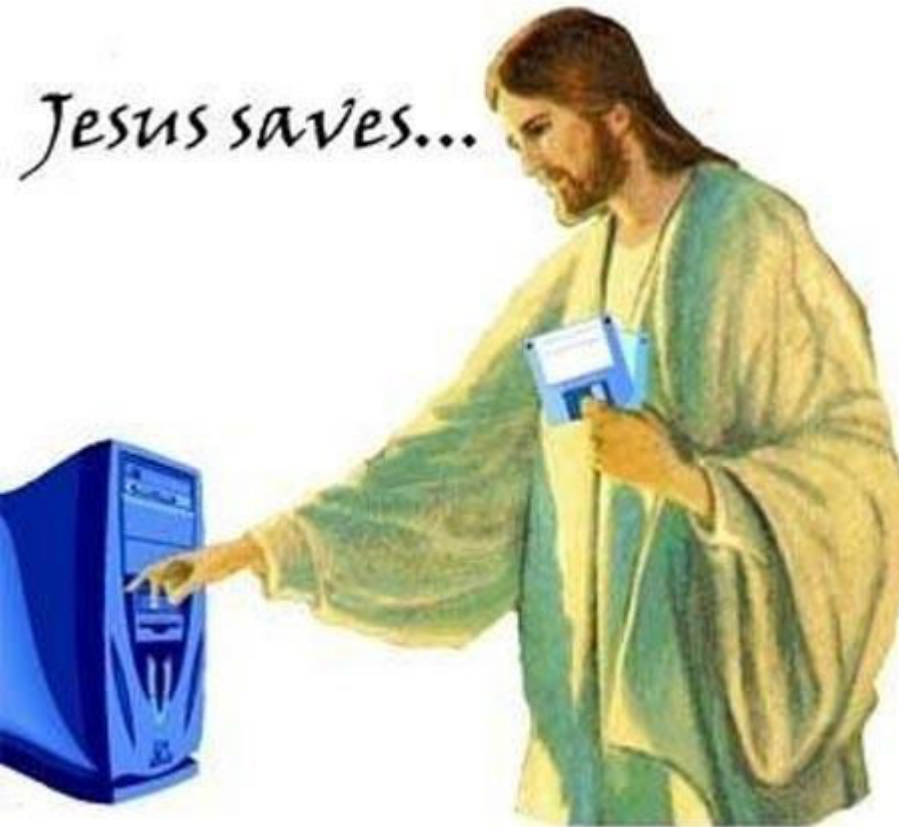


# Related Work

Assignment is well-documented as dangerous



# Previous attempts to Save the World...



*... and makes incremental backups*

evilmlk.com

Have ended badly...



# Other attempts were highly misguided

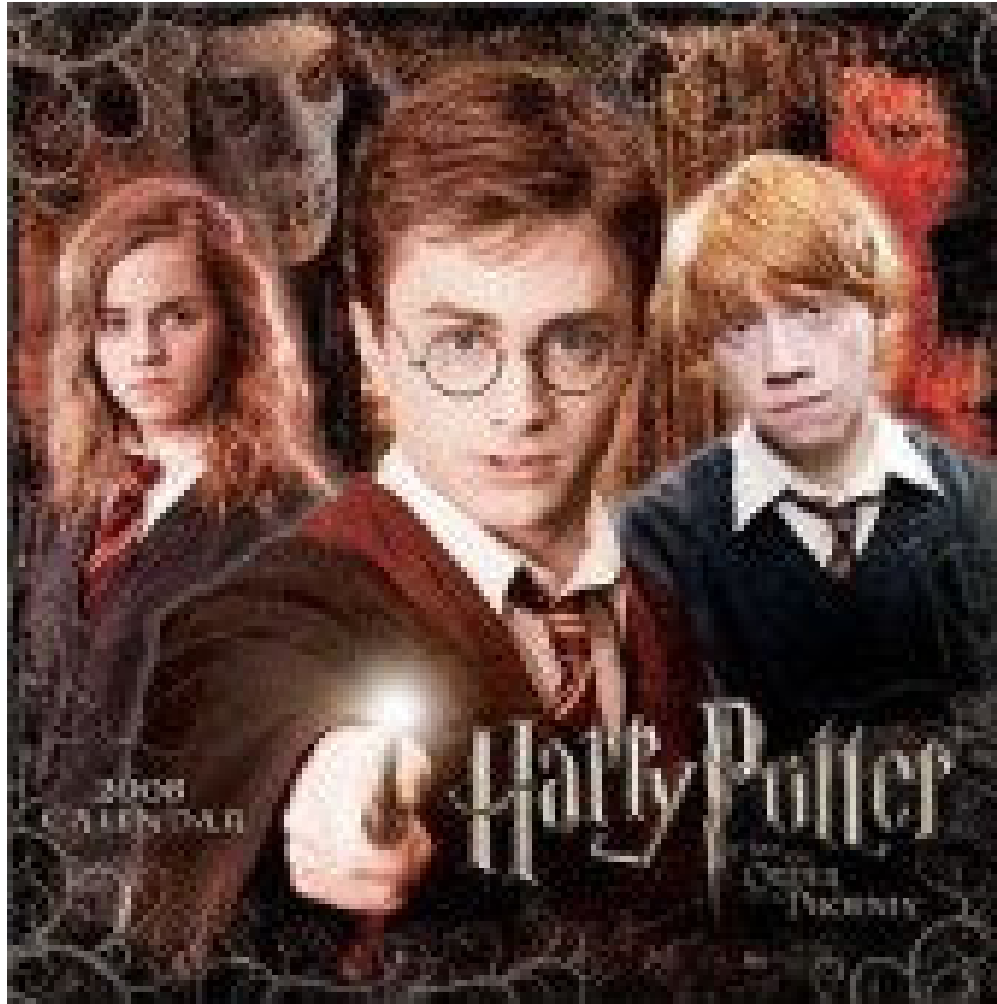


- The Nazis used Standard ML
  - [J. Cette, SIGBOVIK '07 review]
  - Garbage strikes again!





Some attempts have been spellbinding

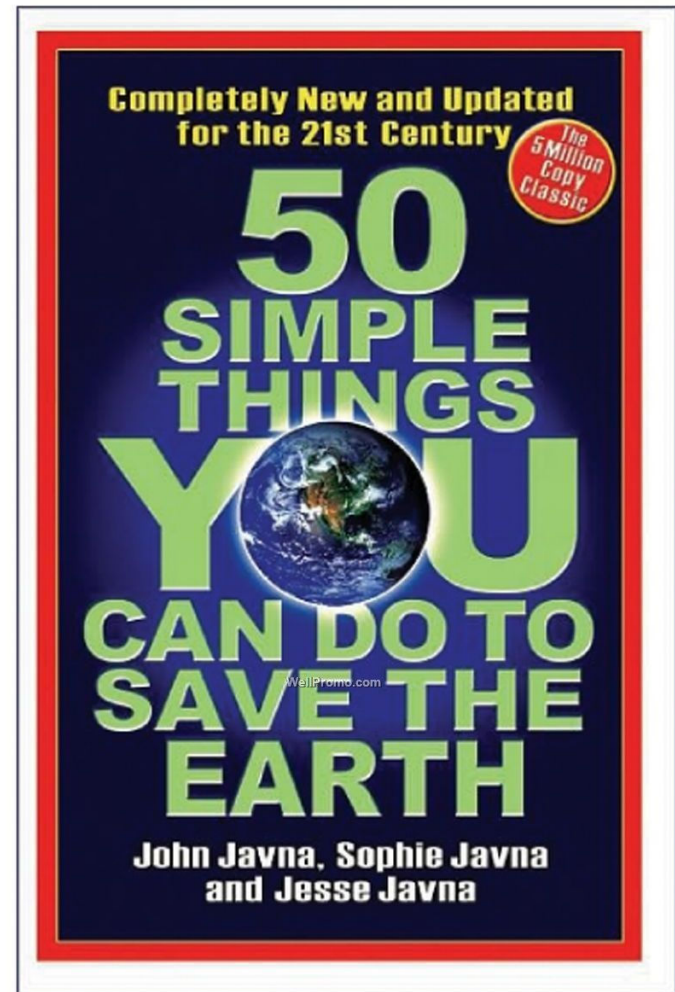


# Recent work considers saving other worlds



# Recycling

- #47 way to save the world
- See also MapReuse, MapRecycle [McGlohon, SIGBOVIK '09]



# Conclusions

- Believe not in the false prophets of imperative and functional programming...holy states **CAN** save the world!

