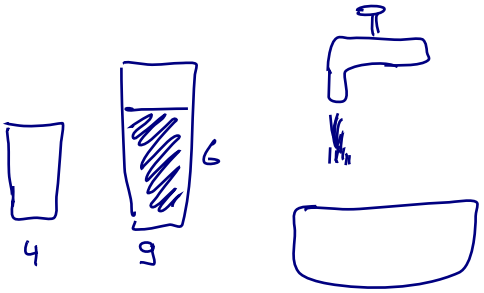
Case Study

The Water Pouring Problem



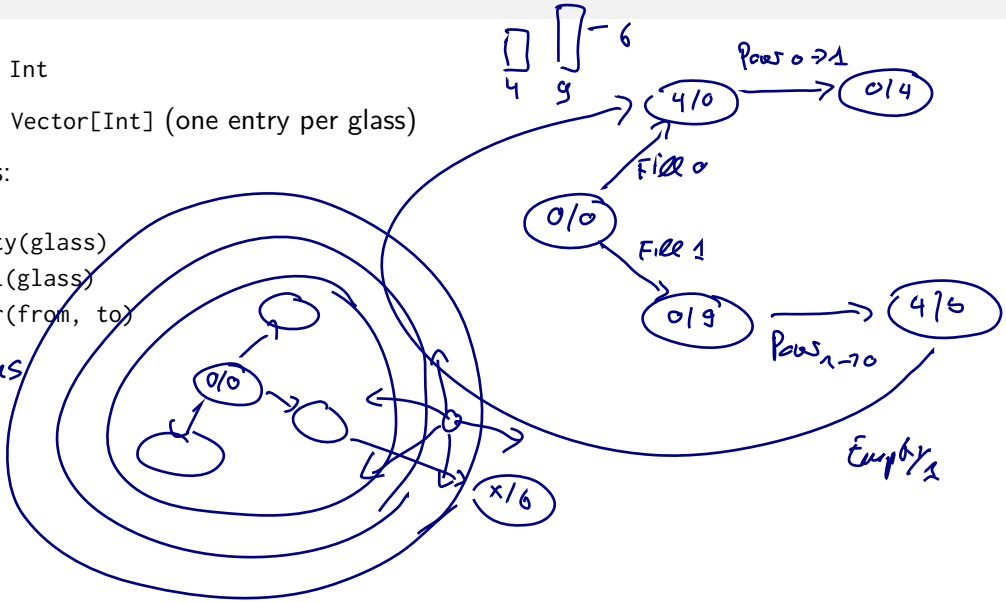
States and Moves

Glass: `Int`

State: `Vector[Int]` (one entry per glass)

Moves:

```
Empty(glass)
Fill(glass)
Pour(from, to)
```



Variants

In a program of the complexity of the pouring program, there are many choices to be made.

Choice of representations.

- Specific classes for moves and paths, or some encoding?
- Object-oriented methods, or naked data structures with functions?

The present elaboration is just one solution, and not necessarily the shortest one.

Guiding Principles for Good Design

- Name everything you can.
- Put operations into natural scopes.
- Keep degrees of freedom for future refinements.