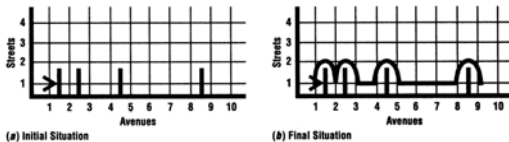


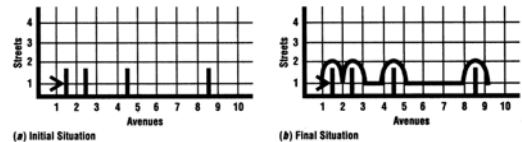
## Mile-long hurdle race

Suppose that we want to program *Reeborg* to run a one-mile long hurdle race, where vertical wall sections represent hurdles. The hurdles are only one block high and are randomly placed between any two corners in the race course. One possible race course is shown below.

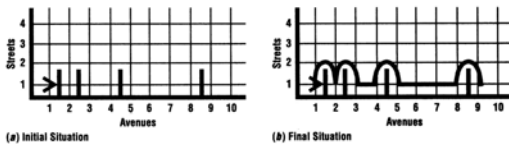


## Strategy?

*Reeborg* could run this race by jumping up between every pair of corners, but is this an appropriate strategy?

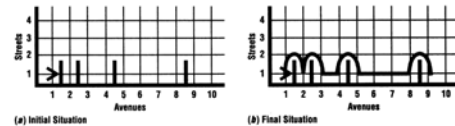


No. Jumping up between corners, whether a hurdle was between them or not, would slow *Reeborg* down. Instead, program *Reeborg* to move straight ahead when it can, and to jump over hurdles only when it must. The program could then consist of 8 `advance_a_corner()` instructions. The definition of `advance_a_corner()` can be written using stepwise refinement as follows:



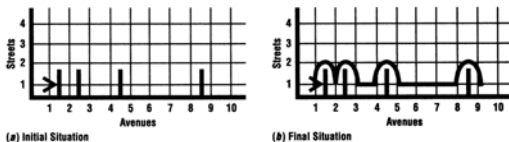
```
def advance_a_corner():
    if front_is_clear():
        move()
    else:
        jump_hurdle()
```

Notice we have used a new command `jump_hurdle()` that must now be defined.



We continue our refinement by defining `jump_hurdle()` to be

```
def jump_hurdle():
    jump_up()
    move()
    jump_down()
```



To finish the problem, we write `jump_up()` and `jump_down()`

```
def jump_up():
    turn_left()
    move()
    turn_right()

def jump_down():
    turn_right()
    move()
    turn_left()
```

