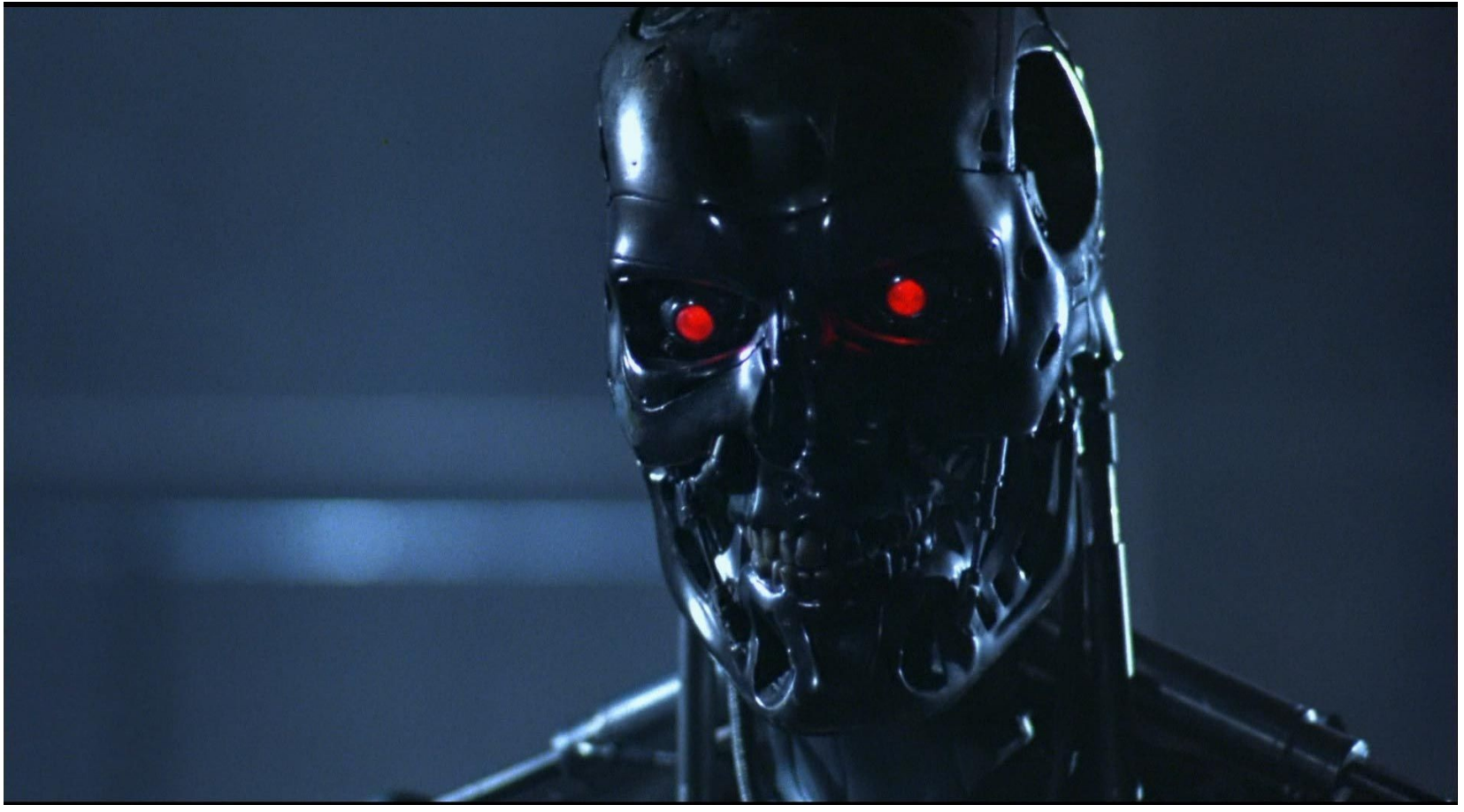
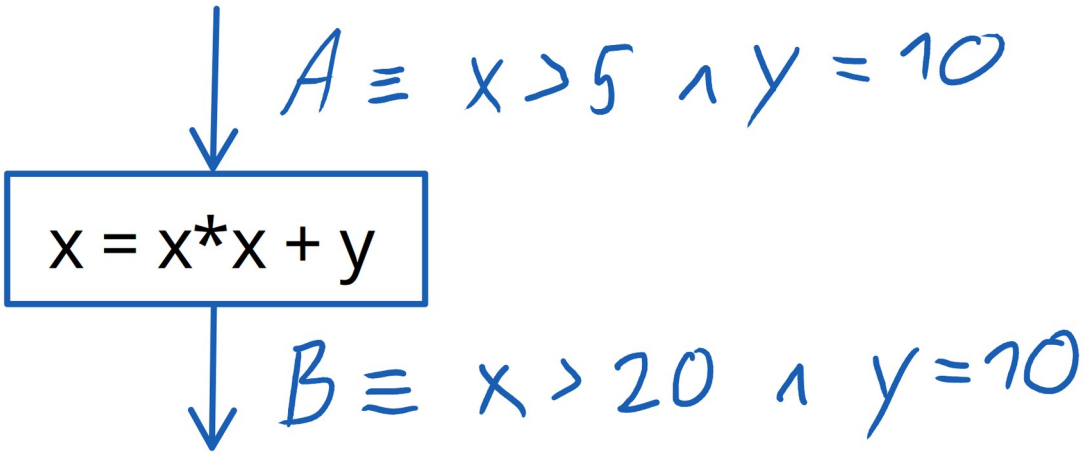
Week 4: Termination



Local Consistency revisited



Idea

- Make sure that each loop is executed only finitely often ...
- For each loop, identify an indicator value r, that has two properties

  (1) $r > 0$ whenever the loop is entered;

  (2) r is decreased during every iteration of the loop.
- Transform the program in a way that, alongside ordinary program execution, the indicator value r is computed.
- Verify that properties (1) and (2) hold!

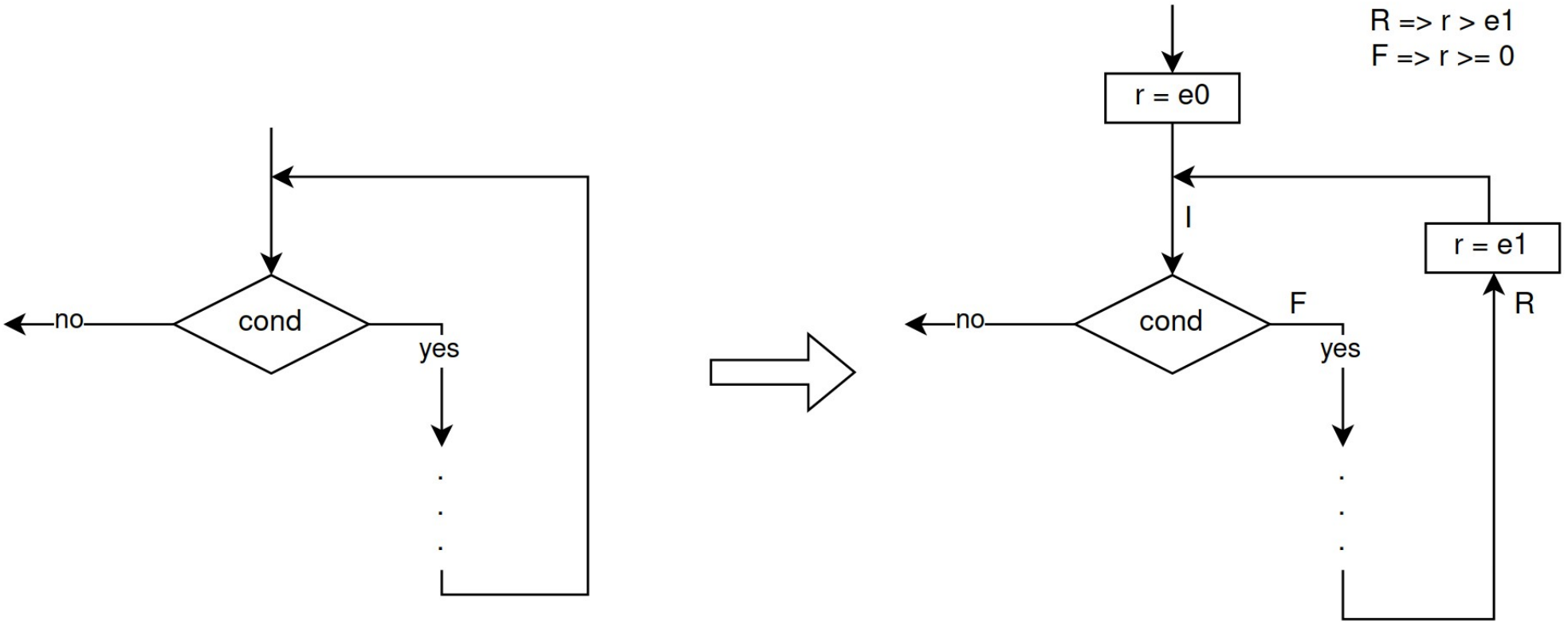
General Method

- For every occurring loop `while (b) s` we introduce a fresh variable `r`.
- Then we transform the loop into:

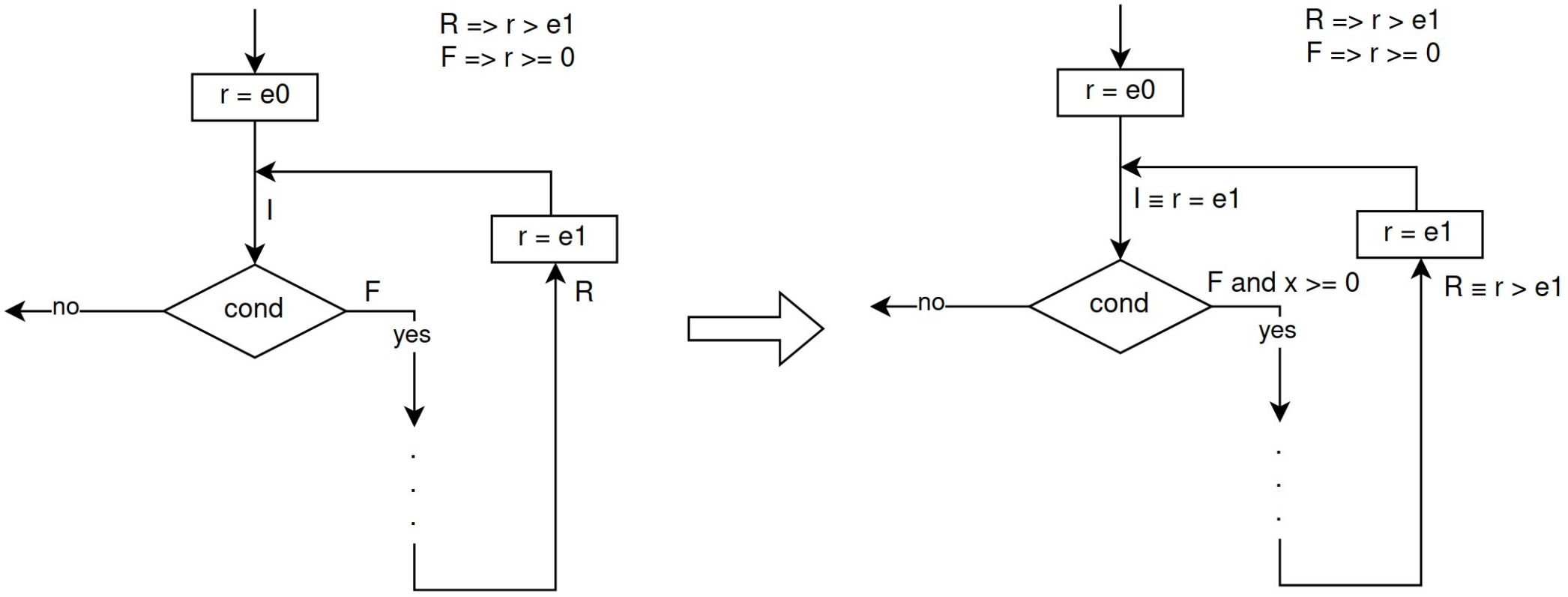
```
1  r = e0;
2  while(b) {
3          assert(r > 0);
4          s;
5          assert(r > e1);
6          r = e1;
7  }
```

for suitable expressions `e0`, `e1`.

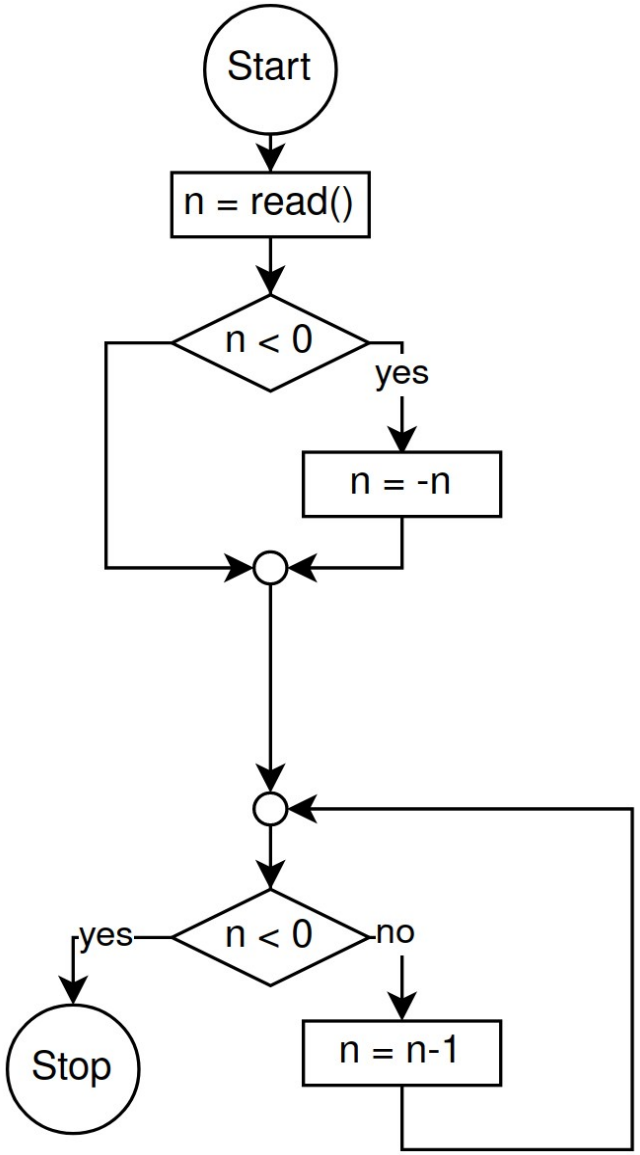
How to prove termination



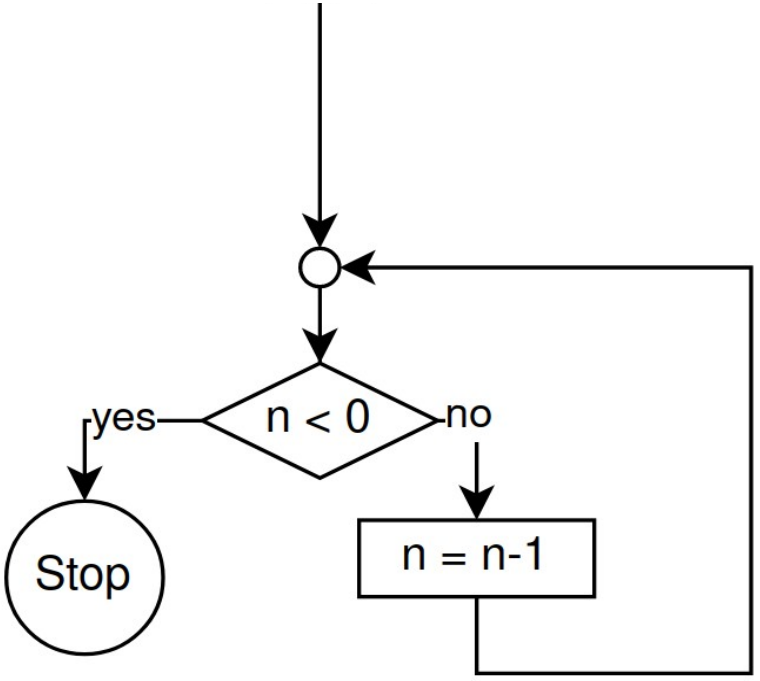
How to prove termination



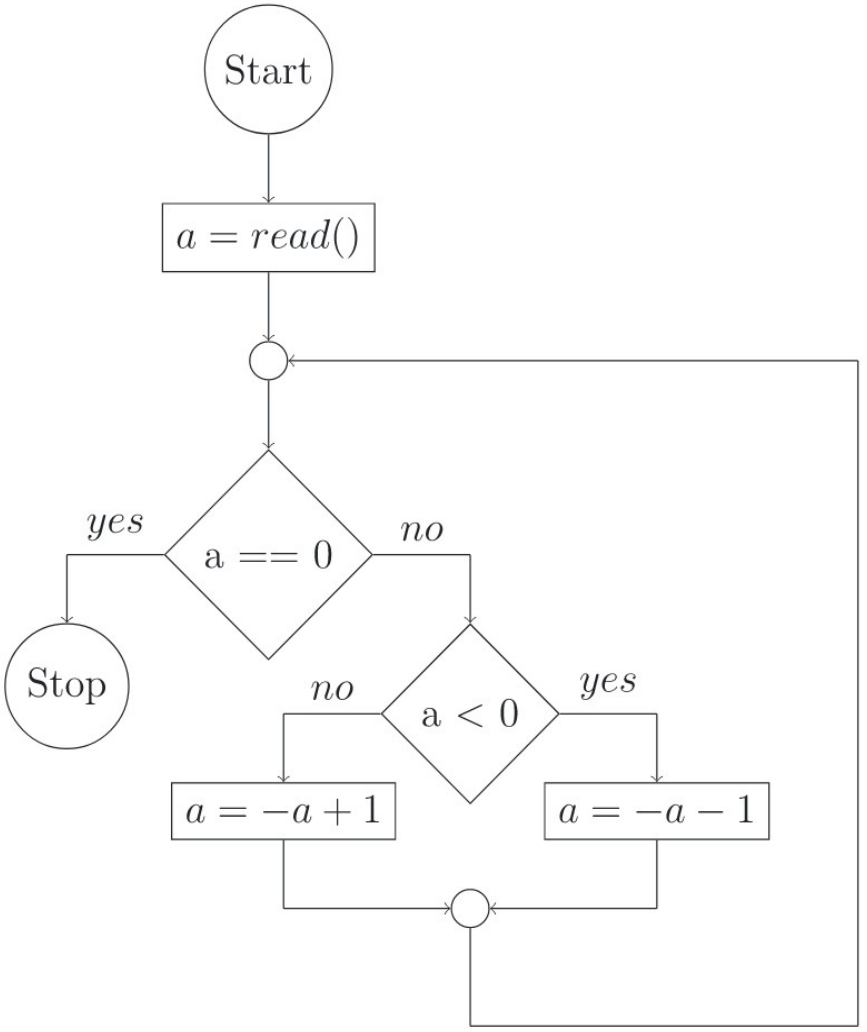
How to prove termination



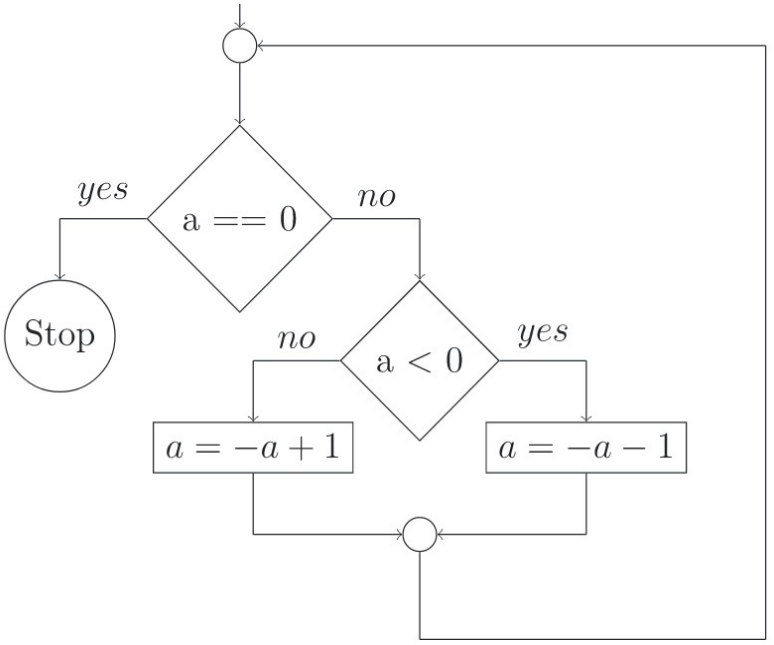
How to prove termination



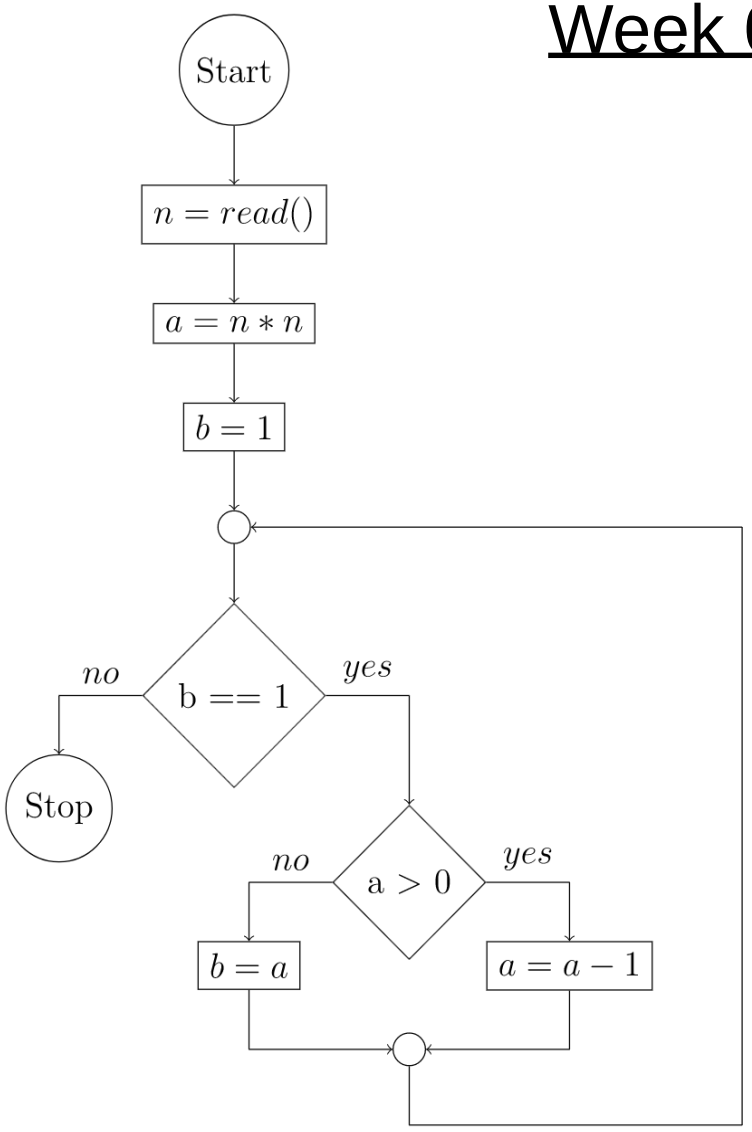
Week 04 Tutorial 03 — A Wavy Approach



Week 04 Tutorial 03 — A Wavy Approach



Week 04 Task 4: Why is this not on Artemis 2.0



Week 04 Task 4: Why is this not on Artemis 2.0