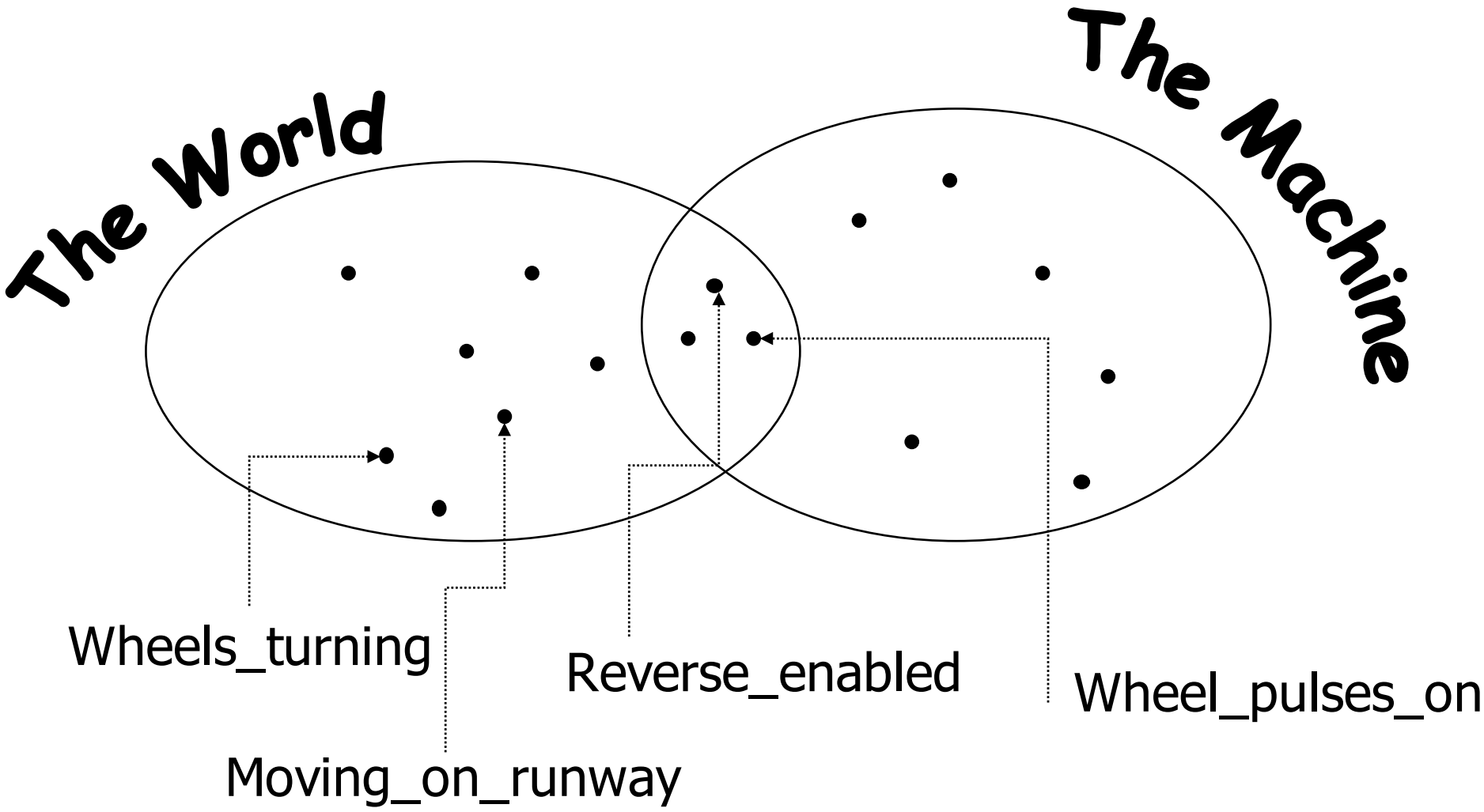
Alloy, the world and the machine, and requirements

Alloy and requirements

- Are the previous examples describing requirements?
- What about the Line example?
  - We are modeling the structure of geometric figures and
  - Some constraints on this structure
  - Some operations
  - A constraint on one of the operations: the last point of the last segment of the line and the first point of the segment must coincide
  - Where are requirements?

Example – Airbus A320 Braking Logic



Modeling the Airbus braking logic with Alloy


```
abstract sig Bool {}
one sig True extends Bool {}
one sig False extends Bool {}

abstract sig AirCraftState {}
sig Flying extends AirCraftState {}
sig TakingOff extends AirCraftState {}
sig Landing extends  AirCraftState {}
sig MovingOnRunaway extends AirCraftState {}
```

… for landing, we are not considering the movement due to takeoff as it is not relevant to our analysis

Modeling the Airbus braking logic with Alloy


```
sig Weels {
        enabled: Bool,
        turning: Bool
}{turning = True implies enabled = True}

sig Aircraft {
        status: one AirCraftState,
        weels: one Weels,
        weelsPulsesOn: one Bool,
        reverseThrustEnabled: one Bool
}{status = Flying implies Weels.enabled = False}
```

Modeling the Airbus braking logic with Alloy


```
fact domainAssumptions {
all a: Aircraft | a.weelsPulsesOn = True <=> a.weels.turning = True
all a: Aircraft | a.weels.turning = True <=> a.status = MovingOnRunaway}

fact requirement {
all a: Aircraft | a.reverseThrustEnabled = True <=> a.weelsPulsesOn =
True}

assert goal {
all a: Aircraft | a.reverseThrustEnabled = True <=> a.status =
MovingOnRunaway}
check goal
```

No counter examples are found!

But note that, still, this is the wrong model of our world: the spec is internally coherent but does not correctly represent the world