Affecting States

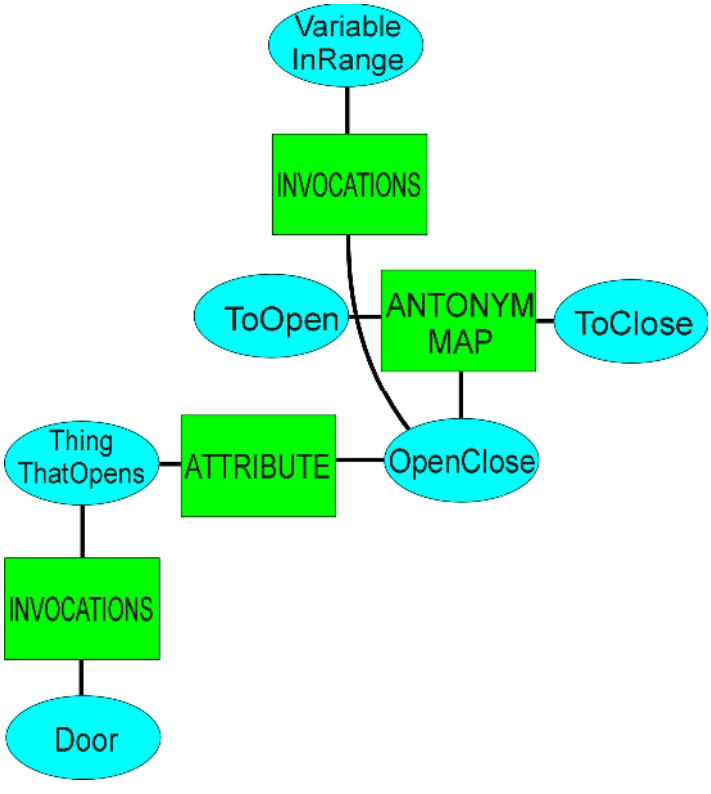
He opened the door.

Something about the door has changed – let's assume a state, which is capable of having a value between extreme states:

The door is fully open
The door is partly open
The door is ajar
The door is fully closed

How do we make the ToOpen relation see the state?

We can have an ANTONYM map – much like SYNONYMMAP, but with a state which the two relations operate on:



"VariableInRange" means that the state can take any value in range – "partly open"

ThingThatOpens is synonymous with ThingThatCloses – a parameter of one meaning of the ToClose relation.

He closed the door
He closed in on his quarry.

When we have an ToOpen relation, such as

He opened the door.

We search around the relation to see if there is a state it operates on – we find OpenClose, we see whether the object (or any parent) has such an attribute, we create an invocation of it and attach it to the invoked door. For OpenClose, it can be a door, a valve or a container (including a file or document).

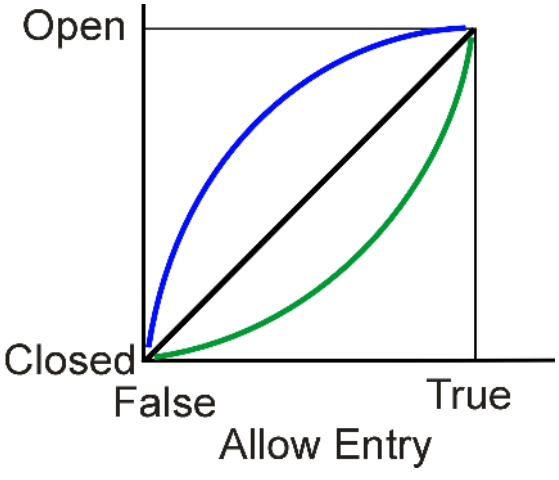
The state object becomes a repository of the present state, and allows us to say

He opened the door slightly.
He closed the door (meaning it hit one end of the range)

The state cannot directly control whether relations are possible.

He opened the door to allow entry – but
The door was slightly open – was it open enough to allow entry?

We will need a conversion between the state and an existential logic variable – the endpoints are the same, but the middle allows a curve, as



A large door only needs to be partly open, a small door needs to be fully open.

Other states are either on or off.

He logged on the computer
Hours passed
He logged off the computer

The "logged on" state only has endpoints (he can't be slightly logged on), and maintains its state until some antonymical action occurs.

We need a default end state – if we are not told he has logged on, we assume he is logged off.

We will need the state to do things:

The door has an automatic closer

He opened the door

The door closes after the duration of a door opening – say 10 seconds – whereas normally the door stays open

He held the door open

The closer could not operate – but how did we know that? What is it about ToHold that maintains the state of the action?