Vars not used from Fable 1

visionObstacles : $\mathbb{P}$ Pos
safeLocation : $(x,y), x \in \mathbb{N}, y \in \mathbb{N}^+$
return2Normal operation : bool
prioritisedLocations : $\mathbb{P}$ Pos
Chargers : $\mathbb{P}$ Pos
invalidMap : Message
noMoreViablePlans : Message
failureModeInvoked : Message
noPlan : Bool
systemState : (failure, noPlan, messages)
plan2C : seq Pos
plan2O : seq Pos
recheck : Bool
communication Data : (messages, data collected, ~~~) — used this
GSObstacles : $\mathbb{P}$ Pos
failed2Reconnect : Bool
Completed : Message
identifiedFailure : Message
Command : Message.

Deliverables:
[~~Scenario1~~, ~~Scenario2~~,
~~SL1~~, ~~SL2~~, ~~SL3~~, ~~SL4~~,
~~V2~~,
~~G6~~, ~~G7~~,
~~HI1~~, ~~HI2~~
~~CR3~~]

Clarifications:

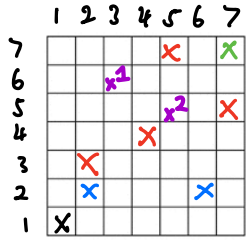
- SL2 is 0 allowed? or is 1 lowest
- SL2 how to verify this, can't do in Z schema, need some algo to confirm,
- HI1, currently set to recharge when 0, is this okay? If not need algo to keep track of ~~~.
- Do we need to model the helper rovers? I don't think so. Focus should be main rover.
- Simplified case study doesn't say how to handle if not correct ~~~
  -> can assume is correct?

Types

Pos == 0…7 × 0…7

Max_Battery == 5

Message is a String



RoverState

currentPosition	: Pos	
obstacles	: ℙ Pos	(V2)
failure	: Bool	
failureReboot	: ℙ Pos	(new variable)
failureHelp	: Pos ⇸ helperIO	(new variable)
goal	: ℙ Pos	
batteryLevel	: ℕ	
chargingComplete	: Bool	
recharge	: Bool	
atGoal	: Bool	
dataAllocated	: Message	(part of communication data)
helperID	: ℕ	

RoverState INIT

currentPosition	:	(1,1)
obstacles	:	{(2,3), (4,4), (5,7), (7,5)}
failure	:	False
failureReboot	:	{(2,2), (6,2)}
failureHelp	:	{(2,6) ⟼ 1, (5,5) ⟼ 2}
goal	:	(7,7)
batteryLevel	:	5
chargingComplete	:	True
recharge	:	False
atGoal	:	False
dataAllocated	:	"This is mock data"
helperID	:	0

This means at (5,5) the rover with id of 2 will come to help

(failureReboot ⩤ obstacles) ∧ (dom(failureHelp) ⩤ obstacles)

∧ (failureReboot ∩ dom(failureHelp) = ∅)

batteryLevel ∈ 0…Max_Battery (SL1)

chargingComplete ⇔ batteryLevel = Max_Battery

currentPosition ∉ obstacles (SL4)

goal ∉ obstacles (SL2) (V2)

helperID ∈ {0,1,2}, 0 means no help (or main rover ID) (SL3)

dom(failureHelp) ⊆ Pos

ran(failureHelp) ⊆ helperID

N.B is variable not stated, implicitly means retain original value

Scenario 1

move

ΔRoverState
next? : Pos

failure = False
rCharge = False (HI2)
batteryLevel > 0 (SL1)
next? ∉ obstacles (SL2) (SL4)

currPosition' = next?
batteryLevel' = batteryLevel − 1
atGoal' = True ⟷ (currPosition' = goal)

notifyComplete (G6)

= RoverState
dataEmit! : Message

atGoal = True
dataEmit! = dataCollected

setRecharge (HI1)

ΔRoverState

failure = False
rCharge = False
batteryLevel = 0

rCharge' = True

charge (HI2)

ΔRoverState

failure = False
rCharge = True
batteryLevel < 5

batteryLevel' = batteryLevel + 1

finishCharge (HI2)

ΔRoverState

failure = False
rCharge = True
batteryLevel = 5

rCharge' = False
chargingComplete = True

Scenario 2

triggerRebootFailure (SL3)

ΔRoverState

failure = False
currPosition ∈ failureReboot

failure' = True (G7)
helperID' = 0

rebootRecover (SL3)

ΔRoverState

failure = True
currPosition ∈ failureReboot

failure' = False
helperID' = 0

triggerHelpFailure (SL3) (CR2)

ΔRoverState

failure = False

(currPosition ∈ dom(failureHelp))

failure' = True (G7)

helperID = 0

requestHelp (SL3) (G7) (CR3)

=_RoverState

failure = True

(currPosition ∈ dom(failureHelp))

recieveHelperID (SL3) (CR3)

ΔRoverState

failure = True

(currPosition ∈ dom(failureHelp))

helperID' = failureHelp(currPosition)

authenticateAndRecover (SL3) (CR3)

ΔRoverState

arrivingRoverID? : ℕ

failure = True

helperID ≠ 0

arrivingRoverID = helperID

failure = False

helperID = 0