If the world pauses right now, the following facts will hold true:

- There is a Martian environment that is made of:
    1. A starting point
    2. A goal point
    3. Obstacles (except at the goal location)
    4. 1 master rover
    5. 2 side rovers
    6. Any other point

$N, M : \mathbb{N}$
$Position ::= (0 \ldots N-1) \times (0 \ldots M-1)$

Martian Environment
$start, goal : Position$
$valid : \mathbb{P}\ Position$
$obstacles : \mathbb{P}\ Position$

$start, end \in valid$
$valid \cap obstacles = \phi$

Martian Environment$_{Init}$
$goal = x \in Position$
$valid = y \subseteq (\mathbb{N} \times \mathbb{N})$
$obstacles = z \subseteq (\mathbb{N} \times \mathbb{N})$

Rover-Related Facts

- There is a master rover at some point in the Martian environment
- The master rover is not located on:
    - an obstacle
- The master rover has a battery level between 0 (non-inclusive) and a maximum value

The master rover is in exactly one of the following states:

1. Failed (Failure Mode)
    - Rebooting
    - Requesting help
        - Notify ground station
        - Awaiting side rover arrival
        - Authenticating a side rover
          (e.g. establishing communication between rover IDs)
2. OK
    - Traversing
    - Setting up communication with ground control at goal location
3. Charging (Recharge Mode)
    - Battery is not at maximum

```
Act   ::= traversing | comms |
          rebooting | requestingHelp |
          awaitingHelper | authenticating |
          stall | charging

RoverID ::= {0,1,2}
```

```
MasterRover ──────────────
  currentPosition : Position
  maxBattery      : ℕ
  batteryLevel    : ℕ
  neededBattery   : ℕ
  recharge        : Bool
  completedNotified : Bool
  failureNotified : Bool
    failureMode   : Bool
  command         : Act
  helperID        : RoverID
──────────────────────────
  batteryLevel > 0
  batteryLevel ≤ MaxBattery
((failureMode = True ∧
  act ∈ {rebooting, requestingHelp,
         awaitingHelp, authenticating,
         Stall})
  ∨
 (helperId = 0 ∧ failureMode = False
  ∧ recharge = False ∧
  act ∈ {Traversing, comms})
  ∨
 (recharge = True ∧
  act ∈ {charging} ∧ helperId = 0))

((SystemState = charging ∧ recharge = True
  ∧ batteryLevel < maxBattery) ∨
  (systemstate ≠ charging))
((recharge = True ∧ batteryLevel' < maxBattery
  ∧ batteryLevel < neededBattery) ∨
  (recharge = False))

 ~(completedNotified = True ∧
    failureNotified = True)
```

```
SideRover 1 ──────
  sid1  : RoverID
  spos1 : Position
──────────────────
```

```
sideRover 2 ──────
  sid2  : RoverID
  spos2 : Position
──────────────────
```

```
System ──────────────
  MartianEnvironment
  MasterRover
  sideRover1
  sideRover2
─────────────────────
  currentPosition, spos1, spos2 ∈ Valid
  sid1 != sid2
  currentPosition != spos1
  currentPosition != spos2
  spos1 != spos2
─────────────────────
```

$MasterRover_{Init}$

currentPosition = $x \in \mathbb{P}(\mathbb{N} \times \mathbb{N})$

maxBattery = $x \in \mathbb{Q}_{>0}$

batteryLevel = $x \in \mathbb{Q}_{>0}$

recharge = False

completedNotified = False

failureNotified = False

failureMode = False

command = traversing

helpersd = 0

$SideRover1_{Init}$

sid1 =

spos1 = $x \in \mathbb{P}(\mathbb{N} \times \mathbb{N})$

$sideRover2_{Init}$

sid2 = 2

spos2 = $x \in \mathbb{P}(\mathbb{N} \times \mathbb{N})$

$System_{Init}$

$MartianEnvironment_{Init}$

$MasterRover_{Init}$

$sideRover1_{Init}$

$sideRover2_{Init}$

```
Move
  ΔMasterRover
  next: Position
  ----------------
  command = traversing
  recharge = False
  failureMode = False
  next ∈ valid
  currentPosition' = next
  batteryLevel' = batteryLevel - 1
  neededBattery' = neededBattery - 1
```

SL4: rover does not collide with obstacles because currentPosition and next are in valid

G7: Rover is stationary when charging

G7: FailureMode entered

```
BeginFailure
  ΔMasterRover
  ----------------
  failureMode = False
  failureMode' = True
  command' = still
```

```
AttemptReboot
  ΔMasterRover
  ----------------
  failureMode = True
  command' = Rebooting
```

```
RebootSuccess
  ΔMasterRover
  ----------------
  failureMode = True
  command = Rebooting
  systemState' = ok
  command' = traversing
```

```
RebootFail
  ΔMasterRover
  ----------------
  failureMode = True
  command = Rebooting
  systemState' = Failure
  command' = RequestingHelp
```

SL3: rover either reboots or request help on failure.

```
AttemptRequestHelp
  ΔMasterRover
  ----------------
  failureMode = True
  command' = RequestingHelp
```

```
NotifyFailure
  ΔMasterRover
  ΔsideRover1
  ΔsideRover2
  ----------------
  command = RequestingHelp
  helperId = 0
  failureNotified' = True
  command' = awaitingHelp
  helperId' ∈ {sid1, sid2}
```

```
FailureSolved
  ΔMasterRover
  ----------------
  failureMode = True
  command' = Traversing
  helperId' = 0
  failureMode' = False
```

CR3: If help requested, helperId set to one of the rovers. Authentication ensure that roverId match

```
HelperArrives
  ΔMasterRover
  ΔsideRover1
  ΔsideRover2
  ----------------
  failureMode = True
  command = awaitingHelp
  failureNotified = True
  command' = Authenticating
  (spos1' = currentPosition ∨
  spos2' = currentPosition)
```

```
AuthenticateHelper
  ΔMasterRover
  ΔsideRover1
  ΔsideRover2
  ----------------
  failureMode = True
  command = Authenticating
  failureNotified = True
  helperId ∈ {sid1, sid2}
  failureNotified' = False
```

G7: Recharge set to True

```
InitiateRecharge
  ΔMasterRover
  ----------------
  recharge = False
  batteryLevel < maxBatteryLevel
  batteryLevel < neededBattery
  recharge' = True
```

```
BeginRecharge
  ΔMasterRover
  ----------------
  recharge = True
  command' = Charging
```

```
chargeUptick
  ΔMasterRover
  ----------------
  recharge = True
  batteryLevel < maxBatteryLevel
  batteryLevel' = batteryLevel + 1
```

```
EndRecharge
  ΔMasterRover
  ----------------
  recharge = True
  batteryLevel = maxBattery
  command' = traversing
  recharge' = False
```

G7: Charging ends when battery full

G6: notification sent on reaching goal

```
GoalReached
  ΔMasterRover
  ----------------
  currentPosition = goal
  command = Traversing
  completedNotified = False
  command' = Comms
```

```
NotifyCompleted
  ΔMasterRover
  ----------------
  currentPosition = goal
  command = comms
  completedNotified' = True
```