The Mars Rover is a robot designed to explore Mars. Its task is to collect a rock sample from as many different locations as possible. In this discussion forum, you should consider the following questions.

* How might you design a general utility function (i.e., not a predicate task specification) for this Mars Rover? What are the challenges around this?

If the Mars Rover arrived at all obstacles or useless points(The way to the collected location, which consumes the power), then will have a simple punishment, on the contrary, if it arrived at an explorable location and successfully collected the rock samples, it will need to be rewarded. If all the points in a run are obstacles or useless points or just the combinations of the two, then = -1. If all the points in a run are explorable locations where succeed in the collection, then = 1. If the points are the combinations of the three, then will be given a range from -1 to 1 based on their proportion.

The above is a general utility function for the Mars Rover I designed, it will give a performance measure from the range of -1 to 1.

The challenges are the environment on Mars might be dynamic, so the calculated utilities might not be accurate or calculable, because of many other affecting factors.

* How might you design a predicate task specification for this Mars Rover? What are the challenges around this?

If I would like to define a predicate task specification for the Mars Rover, I will choose the utility function as , and Ѱ (*r*) to denote it, when the Mars Rover arrives at the designated locations and collects the rock samples, then = 1 and *r*) is true, and in other = 0 and *r*) is false.

The challenges are the same as general utility function, when defining a , there are many factors to influence how to form a specific formula.