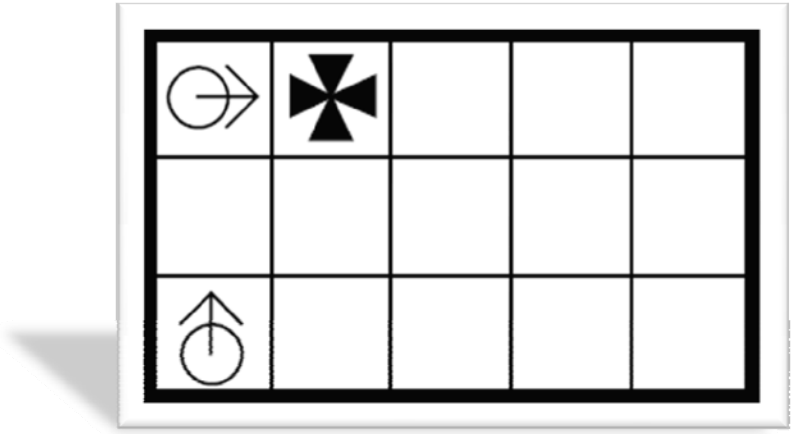
MISSION #11

Conditional Planning

Due: September 16, 2010

Conditional Planning:

You may have already noticed that sequential planning with uncertainty also has limitations. The solution plans are often extremely long and, worse, planning fails altogether in some cases. Consider the following:



There is no sequential solution to this problem. The rotational discrepancy makes the "coercion" performed in the previous example impossible. There is, however, a conditional solution:

```
GTNN(1) GTNN(1)
if (WhatDoISee() == 1100) {TurnTo(90) GTNN(1)}
        else  {TurnTo(-90) TurnTo(-90) GTNN(1)}
```

Assignment 5.2: Conditional Planning

Task: Write a robot function called ***CP()*** *that returning a plan*

Description: Write an optimal conditional planner. When called, this planner attempts to find the shortest conditional plan no longer than ***maxDepth*** that guarantees that the robot will end in a goal state if started in any state specified in the initial conditions. If there is a solution, this function should return the conditional plan and then the robot should execute it. Otherwise, the function should return nil. You may assume that the map is complete.

This is the most challenging assignment in this lab. You will certainly be using a function similar to acts to determine the effect of actions. Since the robot will act differently based upon the percepts it encounters, you will also need to use your sees function from the last lab. [Remember that sees takes a state-set and a percept (the result of WhatDoISee()) and returns the subset of that state-set that is consistent with that percept.] Your searching algorithm will be complex, branching to find the right actions as well as branching across plausible percepts. We will discuss this notion in more detail in class. For those of you in the know, this is AND-OR graph search. For conditional plans, the length of a plan is the maximum number of serial atomic actions (i.e. the number of actions in the longest branch of the plan).