

Consider the Monkey and Bananas problem shown on the last slide and described at this page:

[Prolog - Monkey and Banana Problem](#)

In this google doc, discuss the right vocabulary necessary to solve the monkey & bananas problem. Make comments on your fellow student ideas (make sure to keep it positive!). A few well thought out answers will be MUCH better than 50+ individual copies of the same answer.

Idea 1:

For the Monkey Banana problem, there are a few things to consider.

1. Does the monkey know how far it is from the block to grab and drag it?
2. Does the monkey know which direction to pull the block?
3. Is the monkey tall enough to reach the banana once it is on the block?

Obviously, some of these questions are derived from me overthinking the problem but we can now see what vocabulary we need.

Vocabulary:

We can have a Move action that tells the monkey to move and a Direction action to give to the move action to tell the Monkey which direction it needs to move in.

Obviously, we will need a Grasp action for the monkey to actually grab the stool and then we can use the Move/Direction actions to move the stool.

Once the stool is in the right place, we can use a Hop action to have the monkey hop on top of the block.

If the monkey is tall enough, we can simply use the Grasp action to have the monkey grab the banana.

I think one thing that cannot be ignored is obviously the Monkey's intelligence. If the monkey is intelligent enough to know which direction to move in then the Direction action is redundant to include.

It might be interesting to factor in whether the monkey is full and cannot eat bananas anymore.

I would also include a "Let go" action, if the monkey grasps the stool and is able to move with it without any other action, then that implies the action can be performed at the same time as another. We don't want the monkey to try to hop while still holding on to the stool!

I would include a "Consume Banana" action. If there were more than one bananas, we would be able to make sure the current banana has been consumed before trying to get another one.

Using a time limit would not be a bad idea.

An interesting prospect to explore is the possibility of using the door, instead of the block, to reach the bananas in certain problem instances. The motivation for exploring this line of thinking is for cases where the block is immovable for some reason, whether it is too heavy, it is bolted to the floor, or for any other reason. In this instance we need to know the following:

- If the room is small enough such that the middle of the room, where the banana is hung, is close enough to the top of the door such that the monkey can reach the bananas. If so we will describe the door as “viable”.
- If the monkey is nimble enough to climb the door in the first place. We will describe monkeys that can do this as “nimble”, and those that can’t as “not nimble”.
 - If so, can the monkey **safely** jump (taking into consideration if the monkey is nimble enough to perform this action) from the door to get the bananas (if they cannot be reached from the highest part of the door that the monkey can reach), and will this be enough extra distance to reach the bananas. If the monkey can safely jump from the door we will call the door “jumpable”
- If the door is conducive to climbing. If a door can be climbed we will describe it as “climbable”
- If the door can be opened in a way that will make it easier to reach the bananas we will describe it as “openable”

We will need the following new actions:

- “Climb” to climb to the highest part of the door.
- “Jump” to jump from the door to the bananas, if it is safe to do so.
- “Open” to open the door to a point that will make it easier to reach the bananas, if the door is openable.

Idea 2:

When starting the problem we need to consider a few state variables such as:

- Does the monkey have the banana?
- Is the monkey positioned to move the stool?
- Is the monkey on the stool?
- How far is the stool from its required position?
- Is the stool movable?

Given these state variables we could encode the initial state as:

!(haveBanana) AND monkeyCanMoveStool AND !(monkeyOnStool) AND (stoolUnderBanana)

We would also need a set of actions the monkey could perform.

- Monkey Move action
- Monkey grab action (can be generalized to grab either stool or banana)
- Monkey climb action