1

# Intelligent Autonomous Agents
# ICS 606 / EE 606

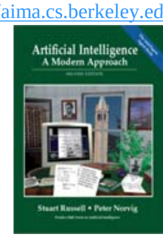
Nancy E. Reed

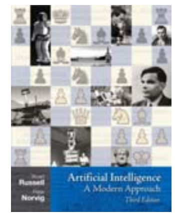
nreed@hawaii.edu

2

## Assignment #1
## Vacuum Cleaner World

- The vacuum cleaner world and code from Russell and Norvig (2nd edition, 2003 or 3rd edition, 2009)
  - http://aima.cs.berkeley.edu/




3

## Vacuum World Code

- Go to http://aima.cs.berkeley.edu
- Download code.tar.gz and unzip it
- Modify the file aima.lisp with the location of
  1. the code directory on your machine, and
  2. name of Lisp installed
- Start your Lisp
- (load “aima”) ; use directory if necessary
- (aima-load ‘agents) ; loads agent functions
- Example tests are in “test-agents.lisp” file

4

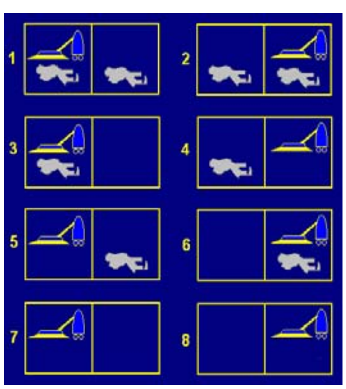
## Assignment #1

- http://www2.hawaii.edu/~nreed/ics606/programming.html
- Vacuum cleaner world – modify code from AIMA.
- Online code/comments: http://www2.hawaii.edu/~nreed/aima/doc
- Due Friday, Midnight, Week 6

5

## A Vacuum World Example

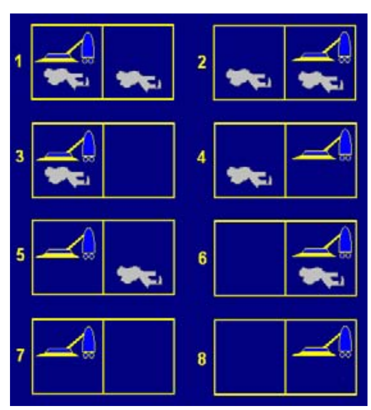
- 2 cell room
- 8 different states possible w/ 1 vacuum, 0-2 dirty cells
- Moves possible:
  - Right
  - Left
  - Suck
- Start in position #5
- Solution??




6

## Example: Vacuum World

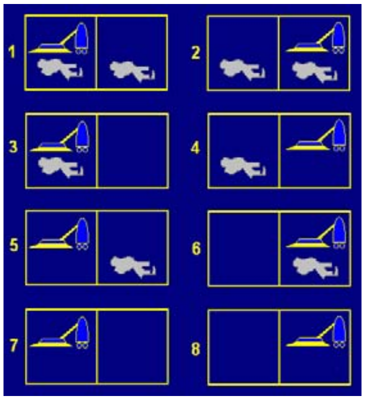
- Start in position #5
  - Solution: [Right, Suck]
- If started in cells {1,2,3,4,5,6,7,8 }
  - E.g., right goes to {2,4,6,8}



## Vacuum World Example

7

- Start in cell #5
  - Solution [right, suck]
- Start in {1,2,3,4,5,6,7,8}
  - Move to {2,4,6,8}
- Solution [right, suck, left, suck]
- Murphy's law – suck will dirty a clean spot!
- Solution: Add sensor for dirt in local cell (only).
  - [right, if dirt then suck, left, if dirt then suck]




## State Space Problem Formation

8

- A problem is defined by four items:
- Initial state
  - E.g., "left square"
- Successor function S(x) = set of action-state pairs
  - E.g., S(left) = {<left→right>, <right→right>,…}
- Goal test, can be
  - Explicit, e.g., "at right"
  - Implicit, e.g., NoDirt(x)
- Path cost (additive)
  - E.g., a sum of distances, number of actions executed, etc.
  - C(x,a,y) is the step cost, assumed to be non-negative
- A solution is a sequence of actions leading from the initial state to a goal state

## State Space

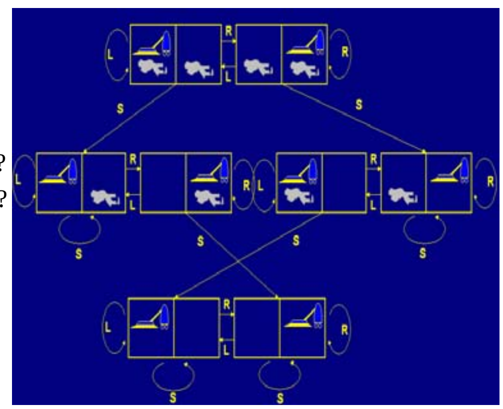
9

- Real world is absurdly complex ➔ state space must be abstracted for problem solving
  - (Abstract) state = set of real states
  - (Abstract) action = complex combination of real actions, e.g., "Left→Right" represents a complex set of possible routes, detours, rest stops, etc.
  - (Abstract) solution = set of real paths that are solutions in the real world
- Each abstract action should be "easier" than the original problem!

## Example: Vacuum World state space graph

10

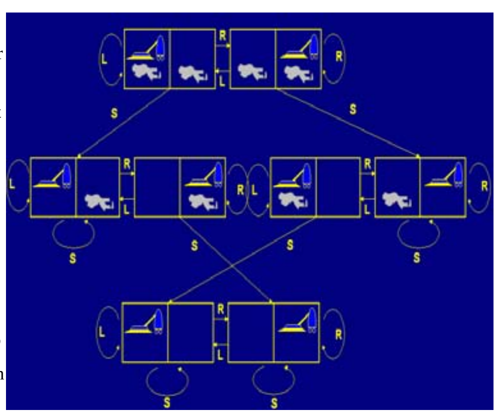
- States?
- Actions?
- Goal test?
- Path cost?



## Example: Vacuum World state space graph

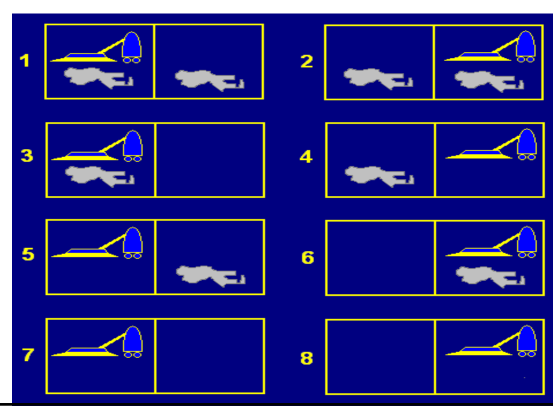
11

- States
  - Integers for dirt, robot locations (ignore dirt amount)
- Actions?
  - Left
  - Right
  - Suck
  - No op
- Goal test?
  - No dirt
- Path cost?
  - 1 per action



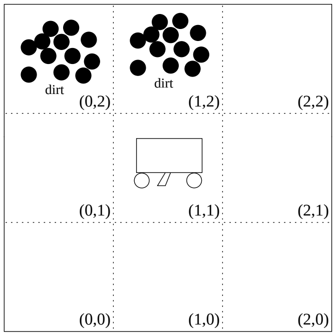
## The Super Simple Vacuum World

12



13

## A 3 by 3 Cell Vacuum World




14

## What is the State Space Representation for a 3 by 3 Vacuum World?

- States
- Moves
- ….