

Program 3 Preview: Prolog Two-Room Planner

Dr. Demetrios Glinos
University of Central Florida

CAP4630 – Artificial Intelligence

Outline

- The Two-Room Blocks World Problem Domain
- The Assignment in a Nutshell
- Program Development
- Test Cases to Consider
- Two-Room Planner Demos

Two-Room Blocks World Problem Domain

- Two-room blocks world consists of
 - **2 rooms**, each with a **table** in it
 - **a set of blocks**
 - a block can be on a table or stacked on top of other blocks, or it can be held by the robot arm
 - an *initial configuration* of blocks on the tables and robot arm location
 - desired *final configuration* of blocks on the tables and robot arm location
 - **a robot arm** that can shuttle between the rooms
 - it can pick up a block in the same room if the block is uncovered
 - it can put down the block it is holding onto another uncovered block or onto the table, if they are in the same room as the robot arm
- **Planning goal:** Determine the sequence of robot arm actions that are needed to rearrange the blocks from the initial configuration to the final configuration and to put the robot arm in the proper room at the end

The Assignment in a Nutshell

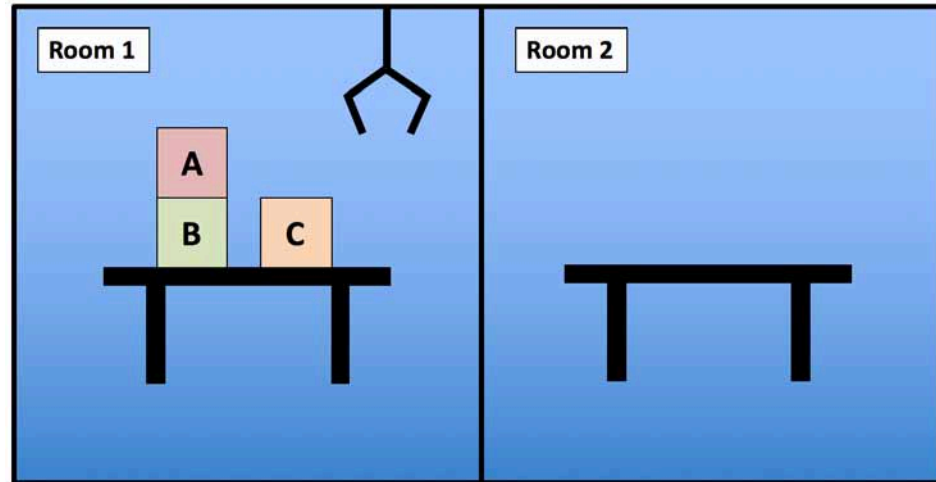
- **Basic task:** Develop a two-room blocks world planner in Prolog
- **Extra credit task:** Implement IDS to find optimal move sequences
- Your planner must determine the sequence of moves that takes a two-room situation from an initial state to a goal state
 - Initial configuration can involve any number of blocks
 - Initially, the blocks can be in either or both rooms
 - The blocks in the final configuration can be in either or both rooms
 - The robot arm can start and end in either room
- Must have a go/2 predicate (go/3 for IDS)
 - first argument is the initial configuration
 - second argument is goal configuration
 - for IDS option: third argument is IDS depth
- Must have prescribed test1/0 and test2/0 predicates
 - test1/0: a 1-room scenario using your 2-room predicates
 - test2/0: a 2-room scenario using your 2-room predicates

Program Development

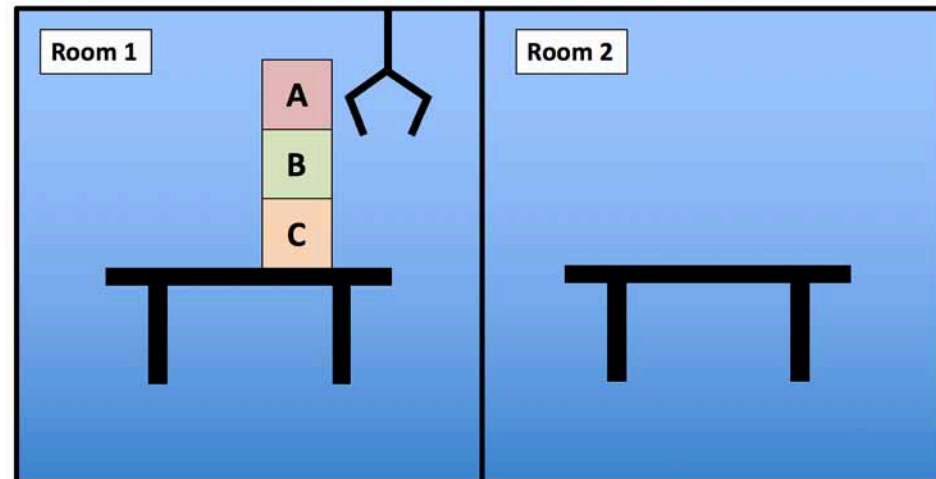
- Files provided (and discussed in lecture)
 - `planner.pl` - one-room planner
 - `utils.pl` - utility predicates
- Development approach
 - suggest extend the one-room planner to achieve two-room functionality
 - modify some predicates
 - add some new predicates
 - recommend using 0-arity predicates "`goroom1`" and "`goroom2`" for moving the robot arm from one room to another
- IDS option
 - OK to add another argument to the go predicate to keep track of the depth, provided it counts up from 1, not down from a magic value maximum that might be a different value depending on the particular planning problem

Required Test Predicate: test1/0

Start state:

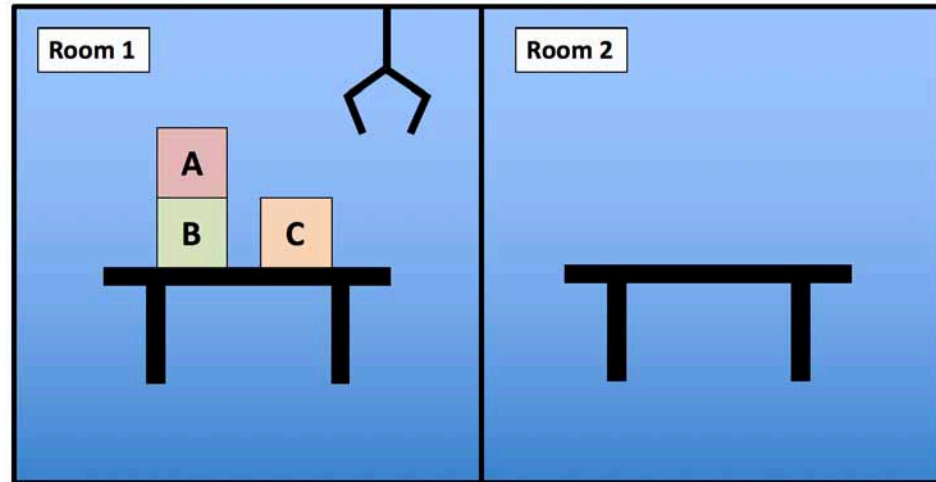


Goal state:

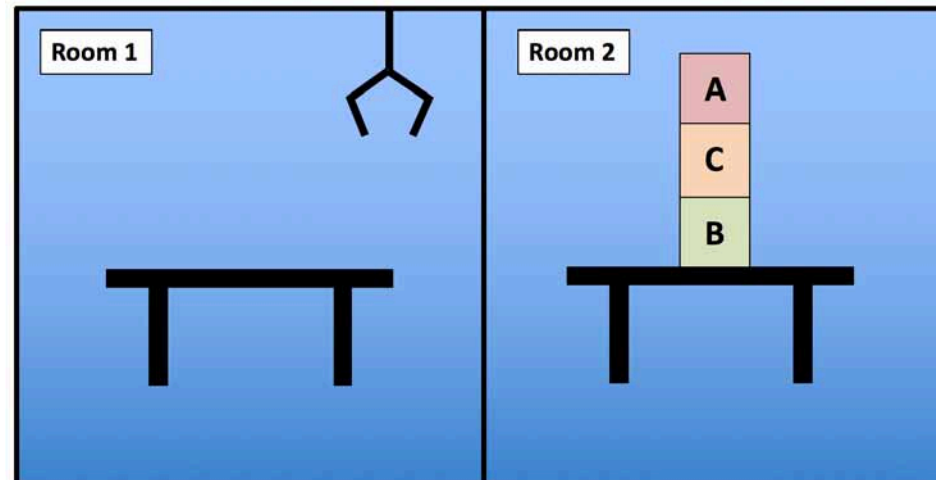


Required Test Predicate: test2/0

Start state:



Goal state:



Two-Room Planner Demos

```
[?- test1.
moves are
pickup(a)
putdown(a)
pickup(b)
stack(b,a)
pickup(c)
goroom2
putdown(c)
goroom1
pickup(b)
putdown(b)
pickup(a)
goroom2
putdown(a)
pickup(c)
stack(c,a)
goroom1
pickup(b)
goroom2
putdown(b)
pickup(c)
putdown(c)
pickup(b)
stack(b,c)
pickup(a)
goroom1
putdown(a)
goroom2
pickup(b)
putdown(b)
pickup(c)
goroom1
putdown(c)
pickup(a)
goroom2
putdown(a)
pickup(b)
goroom1
stack(b,c)
goroom2
pickup(a)
goroom1
stack(a,b)
true.
```

test1 basic and IDS

```
[?- test1.
moves are
pickup(a)
putdown(a)
pickup(b)
stack(b,c)
pickup(a)
stack(a,b)
true .
```

Note: multiple solutions exist

demos: blocks.pl and idsblocks.pl

test2 basic and IDS

```
[?- test2.
moves are
pickup(a)
putdown(a)
pickup(b)
goroom2
putdown(b)
goroom1
pickup(c)
goroom2
stack(c,b)
goroom1
pickup(a)
goroom2
stack(a,c)
goroom1
true .
```

```
[?- test2.
moves are
pickup(a)
putdown(a)
pickup(b)
stack(b,a)
pickup(c)
goroom2
putdown(c)
goroom1
pickup(b)
putdown(b)
pickup(a)
goroom2
putdown(a)
pickup(c)
stack(c,a)
goroom1
pickup(b)
goroom2
putdown(b)
pickup(c)
putdown(c)
pickup(b)
stack(b,c)
pickup(a)
goroom1
putdown(a)
goroom2
pickup(b)
putdown(b)
pickup(c)
stack(c,b)
goroom1
pickup(a)
goroom2
stack(a,c)
goroom1
true.
```