#### **Assignment 3**

#### **Problem Statement**:

#### Implement goal stack planning for the following configurations from the blocks world.

## Theory

### Planning in AI

#### The planning in Artificial Intelligence is about the decision making tasks per- formed by the robots or computer programs to achieve a specific goal. The execution of planning is about choosing a sequence of actions with a high like- lihood to complete the specific task.

#### The blocks-world problem is known as Sussman Anomaly. Noninterleaved planners of the early 1970s were unable to solve this prob- lem, hence it is considered as anomalous.

•

#### When two subgoals G1 and G2 are given, a noninterleaved planner pro- duces either a plan for G1 concatenated with a plan for G2, or vice-versa.

•

#### In blocks-world problem, three blocks labeled as ’A’, ’B’, ’C’ are allowed to rest on the flat surface.

•

#### The given condition is that only one block can be moved at a time to achieve the goal.

•

## Goal Stack Planning

#### This is one of the most important planning algorithms, which is specifically used by **STRIPS**. The stack is used in an algorithm to hold the action and satisfy the goal. A knowledge base is used to hold the current state, actions. Goal stack is similar to a node in a search tree, where the branches are created if there is a choice of an action.

### Algorithmic Steps for Goal Stack Planning

#### Start by pushing the original goal on the stack. Repeat this until the stack becomes empty. If stack top is a compound goal, then push its unsatisfied subgoals on the stack.

#### If stack top is a single unsatisfied goal then, replace it by an action and push the action’s precondition on the stack to satisfy the condition.

#### If stack top is an action, pop it from the stack, execute it and change the knowledge base by the effects of the action.

#### If stack top is a satisfied goal, pop it from the stack.

## Source Code

/\*

stack(X,Y):putblockXonblockY Pre:CLEAR(y)^HOLDING(x) Add:ARMEMPTY^On(x,y)^CLEAR(x) Del:CLEAR(y)^HOLDING(x)

unstack(X,Y):removeblockXfromblockY Pre:On(x,y)^CLEAR(x)^ARMEMPTY() Add:HOLDING(x)^CLEAR(y) Del:On(x,y)^CLEAR(x)^ARMEMPTY()

pickup(X):pickupblockXfromthetable Pre:ONTABLE(x)^CLEAR(x)^ARMEMPTY() Add:HOLDING(x) Del:ONTABLE(x)^CLEAR(x)^ARMEMPTY()

putdown(X):putblockXonthetable Pre:HOLDING(x) Add:ONTABLE(x)^ARMEMPTY^CLEAR(x) Del:HOLDING(x)

\*/

import java.util.\*; classPlanner{

staticStack<String> goalStack =newStack<>();

//staticStringstart\_state="ONTABLE(B)^ONTABLE(A)^ON(C,A)^ ARMEMPTY^CLEAR(C)^CLEAR(B)";

//staticStringgoal\_state="ONTABLE(C)^ON(B,C)^ON(A,B)^ ARMEMPTY^CLEAR(A)";

//staticStringstart\_state="ON(B,A)^ONTABLE(A)^ONTABLE(C)^ ONTABLE(D)^ARMEMPTY^CLEAR(B)^CLEAR(C)^CLEAR(D)";

//staticStringgoal\_state="ON(B,D)^ON(C,A)^ONTABLE(A)^ ONTABLE(D)^CLEAR(B)^CLEAR(C)";

staticString start\_state="ONTABLE(B)^ONTABLE(C)^ON(A,C)^ ON(D,B)^ARMEMPTY^CLEAR(D)^CLEAR(A)";

staticString goal\_state="ONTABLE(A)^ON(B,A)^ON(C,B)^ON(D,C)^ ARMEMPTY^CLEAR(D)";

//staticStringgoal\_state="ON(C,A)^ON(B,D)^ONTABLE(A)^ ONTABLE(D)^CLEAR(B)^CLEAR(C)";

staticArrayList<String> current\_state; staticArrayList<String> action\_list=newArrayList<>();

publicstaticvoidlist\_operate(ArrayList<String> lp,ArrayList<String> la,ArrayList<String> ld,boolean alltrue){ if(!alltrue){

//System.out.println(":::PRE:::"); for(String as:lp){

//ADDPreconditionstoGoalStack goalStack.push(as);

}

for(String as:la){

//System.out.println(as);

//IFallpreconditonspopadditemstostate if(!current\_state.contains(as.trim())){

current\_state.add(as.trim());

}

}

//System.out.println(":::DEL:::"); for(String as:ld){

//System.out.println(as);

//IFallpreconditonspop,deleteitemsfromcurrentstate if(current\_state.contains(as.trim())){

current\_state.remove(as.trim());

}

}

}

}

publicstaticvoidstack(charx,chary,boolean alltrue){ String Pre="HOLDING("+x+")^CLEAR("+y+")";

String Add="ARMEMPTY^ON("+x+","+y+")^CLEAR("+x+")"; String Del="CLEAR("+y+")^HOLDING("+x+")";

ArrayList<String> lp=list\_assertion(Pre); ArrayList<String> la=list\_assertion(Add); ArrayList<String> ld=list\_assertion(Del); if(!alltrue){

goalStack.push("STACK("+x+","+y+")"); goalStack.push(Pre);

}

list\_operate(lp,la,ld,alltrue);

}

publicstaticvoidunstack(charx,chary,boolean alltrue){ String Pre="ON("+x+","+y+")^CLEAR("+x+")^ARMEMPTY"; String Add="HOLDING("+x+")^CLEAR("+y+")";

String Del="ON("+x+","+y+")^CLEAR("+x+")^ARMEMPTY";

ArrayList<String> lp=list\_assertion(Pre); ArrayList<String> la=list\_assertion(Add); ArrayList<String> ld=list\_assertion(Del); if(!alltrue){

goalStack.push("UNSTACK("+x+","+y+")"); goalStack.push(Pre);

}

list\_operate(lp,la,ld,alltrue);

}

publicstaticvoidpickup(charx,boolean alltrue){

String Pre="ONTABLE("+x+")^CLEAR("+x+")^ARMEMPTY"; String Add="HOLDING("+x+")";

String Del="ONTABLE("+x+")^CLEAR("+x+")^ARMEMPTY";

ArrayList<String> lp=list\_assertion(Pre); ArrayList<String> la=list\_assertion(Add); ArrayList<String> ld=list\_assertion(Del); if(!alltrue){

goalStack.push("PICKUP("+x+")"); goalStack.push(Pre);

}

list\_operate(lp,la,ld,alltrue);

}

publicstaticvoidputdown(charx,boolean alltrue){ String Pre="HOLDING("+x+")";

String Add="ONTABLE("+x+")^ARMEMPTY^CLEAR("+x+")"; String Del="HOLDING("+x+")";

ArrayList<String> lp=list\_assertion(Pre); ArrayList<String> la=list\_assertion(Add); ArrayList<String> ld=list\_assertion(Del); if(!alltrue){

goalStack.push("PUTDOWN("+x+")"); goalStack.push(Pre);

}

list\_operate(lp,la,ld,alltrue);

}

publicstaticArrayList<String> list\_assertion(String compound){ ArrayList<String> maker =newArrayList<>();

String assertion[] = compound.split("[’^’]"); for(String as:assertion){

//System.out.println(sg.trim()); maker.add(as.trim());

}

returnmaker;

}

publicstaticvoidmain(String args[]){

//ListofAssertionsthatareTruewhichmakethestate description

current\_state=list\_assertion(start\_state);

//PushFinalStateandSubGoalsontoStack goalStack.push(goal\_state);

String subgoals[] = goal\_state.split("[’^’]"); for(String sg:subgoals){

goalStack.push(sg.trim());

}

boolean b=true; String s="ON(a,b)";

//charp=’b’;

//System.out.println(s.matches("ON\\(.,"+p+"\\)"));if(b)return;

//stack(’a’,’b’,false);

//show\_stack(); chart=’\0’; charx=’\0’; chary=’\0’; String temp="";

ArrayList<String> simple; while(!goalStack.empty()){

show\_stack();

//try{Thread.sleep(5000);}

//catch(Exceptione){e.printStackTrace();} String top=goalStack.pop();

//System.out.println("Checkif"+top+"holdstrue"); if(current\_state.contains(top)){

System.out.println(top+"TruepushedoffStack");

//show\_stack();

}

elseif(top.contains("^")){

//SPLITANDRESTACK

System.out.println(top+"CompoundSplitBeforeRe-entry"); simple=list\_assertion(top);

for(String sp:simple){ goalStack.push(sp);

}

}

else{

if(top.contains("ONTABLE")){ x=top.charAt(8); putdown(x,false);

System.out.println(top+"FalseReplacedwithaction");

}

elseif(top.contains("ON")){ x=top.charAt(3); y=top.charAt(5);

stack(x,y,false); System.out.println(top+"FalseReplacedwithaction");

//show\_stack();

}

elseif(top.contains("ARMEMPTY")){

//checkforholdingandputdownORSTACK for(String assertion:current\_state){

if(assertion.contains("HOLDING")){ temp=assertion;

}

}

x=temp.charAt(8); putdown(x,false);

System.out.println(top+"FalseReplacedwithaction");

}

elseif(top.contains("HOLDING")){ x=top.charAt(8);

if(current\_state.contains("ONTABLE("+x+")")){ pickup(x,false);

}

else{

for(String assertion:current\_state){ if(assertion.contains("ON("+x+",")){

temp=assertion;

}

}

y=temp.charAt(5); unstack(x,y,false);

}

System.out.println(top+"FalseReplacedwithaction");

}

elseif(top.contains("CLEAR")){

//INCOMPLETE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

y=top.charAt(6);

//lookforon(?,t)

//unstack(?,t)

for(String assertion:current\_state){ if(assertion.matches("ON\\(.,"+y+"\\)")){

temp=assertion;

}

}

x=temp.charAt(3); unstack(x,y,false);

System.out.println(top+"FalseReplacedwithaction");

//goalStack.push(top);

}

elseif(top.contains("UNSTACK")){

x=top.charAt(8); y=top.charAt(10); unstack(x,y,true);

action\_list.add("UNSTACK("+x+","+y+")"); System.out.println("ActionPerformed

:::::UNSTACK("+x+","+y+")");

}

elseif(top.contains("STACK")){ x=top.charAt(6); y=top.charAt(8); stack(x,y,true);

action\_list.add("STACK("+x+","+y+")"); System.out.println("ActionPerformed

:::::STACK("+x+","+y+")");

}

elseif(top.contains("PICKUP")){ x=top.charAt(7); pickup(x,true);

action\_list.add("PICKUP("+x+")"); System.out.println("ActionPerformed

:::::PICKUP("+x+")");

}

elseif(top.contains("PUTDOWN")){ x=top.charAt(8); putdown(x,true);

action\_list.add("PUTDOWN("+x+")"); System.out.println("ActionPerformed

:::::PUTDOWN("+x+")");

}

else{goalStack.push(top);}

}

//System.out.println(goalStack.pop());

}

System.out.println("--Stepstoperfrom--"); for(String action:action\_list){

System.out.println(action);

}

//show\_stack();

//show\_stack();

}

publicstaticvoidshow\_stack(){

System.out.println("-----------"); Stack<String> tStack =newStack<>(); String temp=""; while(!goalStack.empty()){ temp=goalStack.pop();

System.out.println(temp); tStack.push(temp);

}

while(!tStack.empty()){ goalStack.push(tStack.pop());

}

System.out.println("-----------");

}

}

## Output

----------- CLEAR(D) ARMEMPTY ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(D) True pushed off Stack

----------- ARMEMPTY ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,C) False Replaced with action

----------- CLEAR(C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) False Replaced with action

----------- ARMEMPTY CLEAR(A) ON(A,C)

ON(A,C)^CLEAR(A)^ARMEMPTY UNSTACK(A,C)

HOLDING(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(A) ON(A,C)

ON(A,C)^CLEAR(A)^ARMEMPTY UNSTACK(A,C)

HOLDING(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(A) True pushed off Stack

----------- ON(A,C)

ON(A,C)^CLEAR(A)^ARMEMPTY UNSTACK(A,C)

HOLDING(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(A,C) True pushed off Stack

----------- ON(A,C)^CLEAR(A)^ARMEMPTY UNSTACK(A,C)

HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(A,C)^CLEAR(A)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(A) ON(A,C) UNSTACK(A,C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(A) ON(A,C) UNSTACK(A,C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(A) True pushed off Stack

----------- ON(A,C) UNSTACK(A,C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(A,C) True pushed off Stack

----------- UNSTACK(A,C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::UNSTACK(A,C)

----------- HOLDING(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(D) False Replaced with action

----------- ARMEMPTY CLEAR(D) ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY False Replaced with action

----------- HOLDING(A) HOLDING(A) PUTDOWN(A) CLEAR(D) ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(A) True pushed off Stack

----------- HOLDING(A)

PUTDOWN(A) CLEAR(D) ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(A) True pushed off Stack

----------- PUTDOWN(A) CLEAR(D) ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::PUTDOWN(A)

----------- CLEAR(D) ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,B)

ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,B) True pushed off Stack

----------- ON(D,B)^CLEAR(D)^ARMEMPTY UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,B)^CLEAR(D)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(D) ON(D,B) UNSTACK(D,B)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ON(D,B) UNSTACK(D,B)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,B) UNSTACK(D,B)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,B) True pushed off Stack

-----------

UNSTACK(D,B) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::UNSTACK(D,B)

----------- HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(D)^CLEAR(C) Compound Split Before Re-entry

----------- CLEAR(C) HOLDING(D) STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) True pushed off Stack

----------- HOLDING(D) STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(D) True pushed off Stack

----------- STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::STACK(D,C)

----------- ON(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(C,B) False Replaced with action

----------- CLEAR(B) HOLDING(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(B) True pushed off Stack

----------- HOLDING(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(C) False Replaced with action

----------- ARMEMPTY CLEAR(C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) False Replaced with action

----------- ARMEMPTY CLEAR(D) ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,C) True pushed off Stack

----------- ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,C)^CLEAR(D)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(D) ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(D,C) True pushed off Stack

----------- UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY

PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::UNSTACK(D,C)

----------- ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(C) True pushed off Stack

----------- ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(C)^CLEAR(C)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY False Replaced with action

----------- HOLDING(D) HOLDING(D) PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(D) True pushed off Stack

----------- HOLDING(D) PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(D) True pushed off Stack

----------- PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::PUTDOWN(D)

----------- CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) True pushed off Stack

----------- ONTABLE(C) PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(C) True pushed off Stack

----------- PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::PICKUP(C)

----------- HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(C)^CLEAR(B) Compound Split Before Re-entry

----------- CLEAR(B) HOLDING(C) STACK(C,B) ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(B) True pushed off Stack

----------- HOLDING(C) STACK(C,B) ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(C) True pushed off Stack

----------- STACK(C,B) ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::STACK(C,B)

----------- ON(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(B,A) False Replaced with action

----------- CLEAR(A)

HOLDING(B) HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(A) True pushed off Stack

----------- HOLDING(B) HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(B) False Replaced with action

----------- ARMEMPTY CLEAR(B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(B) False Replaced with action

----------- ARMEMPTY CLEAR(C) ON(C,B)

ON(C,B)^CLEAR(C)^ARMEMPTY UNSTACK(C,B)

ONTABLE(B) ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(C) ON(C,B)

ON(C,B)^CLEAR(C)^ARMEMPTY UNSTACK(C,B)

ONTABLE(B) ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) True pushed off Stack

----------- ON(C,B)

ON(C,B)^CLEAR(C)^ARMEMPTY UNSTACK(C,B)

ONTABLE(B) ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(C,B) True pushed off Stack

----------- ON(C,B)^CLEAR(C)^ARMEMPTY UNSTACK(C,B)

ONTABLE(B) ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(C,B)^CLEAR(C)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(C) ON(C,B) UNSTACK(C,B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B)

HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(C) ON(C,B) UNSTACK(C,B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(C) True pushed off Stack

----------- ON(C,B) UNSTACK(C,B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ON(C,B) True pushed off Stack

----------- UNSTACK(C,B) ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::UNSTACK(C,B)

----------- ONTABLE(B)

ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(B) True pushed off Stack

----------- ONTABLE(B)^CLEAR(B)^ARMEMPTY PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A)

ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(B)^CLEAR(B)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(B) ONTABLE(B) PICKUP(B)

HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ARMEMPTY False Replaced with action

----------- HOLDING(C) HOLDING(C) PUTDOWN(C) CLEAR(B) ONTABLE(B) PICKUP(B)

HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(C) True pushed off Stack

----------- HOLDING(C) PUTDOWN(C) CLEAR(B) ONTABLE(B) PICKUP(B)

HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(C) True pushed off Stack

----------- PUTDOWN(C) CLEAR(B)

ONTABLE(B) PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::PUTDOWN(C)

----------- CLEAR(B) ONTABLE(B) PICKUP(B)

HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(B) True pushed off Stack

----------- ONTABLE(B) PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(B) True pushed off Stack

----------- PICKUP(B) HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::PICKUP(B)

----------- HOLDING(B)^CLEAR(A) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(B)^CLEAR(A) Compound Split Before Re-entry

----------- CLEAR(A) HOLDING(B) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

CLEAR(A) True pushed off Stack

----------- HOLDING(B) STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

HOLDING(B) True pushed off Stack

----------- STACK(B,A) ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

Action Performed :::::STACK(B,A)

----------- ONTABLE(A)

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(A) True pushed off Stack

-----------

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D)

-----------

ONTABLE(A) ^ ON(B,A) ^ ON(C,B) ^ ON(D,C) ^ ARMEMPTY ^ CLEAR(D) Compound

Split Before Re-entry

----------- CLEAR(D) ARMEMPTY ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(D) True pushed off Stack

----------- ARMEMPTY ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- ON(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ON(D,C) False Replaced with action

----------- CLEAR(C) HOLDING(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(C) True pushed off Stack

----------- HOLDING(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(D) False Replaced with action

----------- ARMEMPTY CLEAR(D) ONTABLE(D)

ONTABLE(D)^CLEAR(D)^ARMEMPTY PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ONTABLE(D)

ONTABLE(D)^CLEAR(D)^ARMEMPTY PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(D) True pushed off Stack

----------- ONTABLE(D)

ONTABLE(D)^CLEAR(D)^ARMEMPTY PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ONTABLE(D) True pushed off Stack

----------- ONTABLE(D)^CLEAR(D)^ARMEMPTY PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ONTABLE(D)^CLEAR(D)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(D) ONTABLE(D) PICKUP(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ONTABLE(D) PICKUP(D)

HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(D) True pushed off Stack

----------- ONTABLE(D) PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ONTABLE(D) True pushed off Stack

----------- PICKUP(D) HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

Action Performed :::::PICKUP(D)

----------- HOLDING(D)^CLEAR(C) STACK(D,C)

ON(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(D)^CLEAR(C) Compound Split Before Re-entry

----------- CLEAR(C) HOLDING(D) STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(C) True pushed off Stack

----------- HOLDING(D) STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(D) True pushed off Stack

----------- STACK(D,C) ON(C,B)

ON(B,A) ONTABLE(A)

-----------

Action Performed :::::STACK(D,C)

----------- ON(C,B)

ON(B,A) ONTABLE(A)

-----------

ON(C,B) False Replaced with action

----------- CLEAR(B) HOLDING(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(B) True pushed off Stack

----------- HOLDING(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(C) False Replaced with action

----------- ARMEMPTY CLEAR(C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(C) False Replaced with action

----------- ARMEMPTY CLEAR(D) ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,C)

ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ON(D,C) True pushed off Stack

----------- ON(D,C)^CLEAR(D)^ARMEMPTY UNSTACK(D,C)

ONTABLE(C) ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ON(D,C)^CLEAR(D)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(D) ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY True pushed off Stack

----------- CLEAR(D) ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(D) True pushed off Stack

----------- ON(D,C) UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ON(D,C) True pushed off Stack

----------- UNSTACK(D,C) ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

Action Performed :::::UNSTACK(D,C)

----------- ONTABLE(C)

ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ONTABLE(C) True pushed off Stack

----------- ONTABLE(C)^CLEAR(C)^ARMEMPTY PICKUP(C) HOLDING(C)^CLEAR(B)

STACK(C,B) ON(B,A) ONTABLE(A)

-----------

ONTABLE(C)^CLEAR(C)^ARMEMPTY Compound Split Before Re-entry

----------- ARMEMPTY CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ARMEMPTY False Replaced with action

----------- HOLDING(D) HOLDING(D) PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(D) True pushed off Stack

----------- HOLDING(D) PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(D) True pushed off Stack

----------- PUTDOWN(D) CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

Action Performed :::::PUTDOWN(D)

----------- CLEAR(C) ONTABLE(C) PICKUP(C)

HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

CLEAR(C) True pushed off Stack

----------- ONTABLE(C) PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

ONTABLE(C) True pushed off Stack

----------- PICKUP(C) HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

Action Performed :::::PICKUP(C)

----------- HOLDING(C)^CLEAR(B) STACK(C,B)

ON(B,A) ONTABLE(A)

-----------

HOLDING(C)^CLEAR(B) Compound Split Before Re-entry

----------- CLEAR(B) HOLDING(C) STACK(C,B) ON(B,A) ONTABLE(A)

-----------

CLEAR(B) True pushed off Stack

----------- HOLDING(C) STACK(C,B) ON(B,A) ONTABLE(A)

-----------

HOLDING(C) True pushed off Stack

----------- STACK(C,B) ON(B,A) ONTABLE(A)

-----------

Action Performed :::::STACK(C,B)

----------- ON(B,A) ONTABLE(A)

-----------

ON(B,A) True pushed off Stack

----------- ONTABLE(A)

-----------

ONTABLE(A) True pushed off Stack

--Steps to perfrom-- UNSTACK(A,C) PUTDOWN(A) UNSTACK(D,B) STACK(D,C) UNSTACK(D,C) PUTDOWN(D)

PICKUP(C) STACK(C,B) UNSTACK(C,B) PUTDOWN(C) PICKUP(B) STACK(B,A) PICKUP(D) STACK(D,C) UNSTACK(D,C) PUTDOWN(D) PICKUP(C) STACK(C,B)

## Conclusion

#### The specifics of AI planning are observed, implemented through the block problem.