哈爾濱工業大學

人工智能实验报告

题　　目　知识表示

专　　业　人工智能领域方向（2+x 模式）

学　　号　2022110829

姓　　名　杨明达

第一章 基础实验——实验报告写的内容

一．背景简介/问题描述

1.1 背景介绍

猴子摘香蕉问题：

一个房间里，天花板上挂有一串香蕉，有一只猴子可在房间里任意活动（到处走动，推移箱子，攀登箱子等）。设房间里还有一只可被猴子移动的箱子，且猴子登上箱子时才能摘到香蕉，问猴子在某一状态下（设猴子位置为 A，香蕉位置在 B，箱子位置为 C），如何行动可摘取到香蕉。

二．算法介绍

2.1 定义描述状态的谓词

ON(x,y): x 在 y 处；
HANG(w,y):w 悬挂在 y 处
MONBOX (z): z 站在箱子上；
HOLDS(z): z 手里拿着香蕉

2.2 变元的个体域

w 的个体域是{banana}
x 的个体域是{monkey,box}
y 的个体域是{A,B,C}
z 的个体域是{monkey}

2.3 问题的初始状态

$$\mathrm{ON}(monkey, A) \wedge HANG(banana, B) \wedge \mathrm{ON}(box, C) \wedge \neg MONBOX(monkey) \wedge \neg HOLDS(monkey)$$

2.4 问题的目标状态

$$\mathrm{ON}(monkey, \mathrm{B}) \wedge \neg HANG(banana, \mathrm{B}) \wedge \mathrm{ON}(box, \mathrm{B}) \wedge MONBOX(monkey) \wedge HOLDS(monkey)$$

2.5 描述操作的谓词

Monkey_Move(a,b):猴子从 a 处运动到 b 处
Monkey_Movebox(b,c):猴子推箱子从 b 处运动到 c 处
Monkey_Downbox(b):猴子在 b 处从箱子上下来
Monkey_Upbox(b):猴子在 b 处爬上箱子
Operation_Banana_Picking(b):猴子在 b 处摘到香蕉

2.6 各操作的条件和动作

Monkey_Move(a,b)

条件：$ON(monkey, a)$、$\neg MONBOX(monkey)$、$\neg HOLDS(monkey)$

动作：删除表：$ON(monkey,a)$

添加表：$ON(monkey,b)$

Monkey_Movebox(b,c)

条件：$ON(monkey,b)$、$ON(box,b)$、$\neg MONBOX(monkey)$、$\neg HOLDS(monkey)$

动作：删除表：$ON(monkey,b)$、$ON(box,b)$

添加表：$ON(monkey,c)$、$ON(box,c)$

Monkey_Downbox(b)

条件：$ON(monkey,b)$、$ON(box,b)$、$MONBOX(monkey)$、$\neg HOLDS(monkey)$

动作：删除表：$MONBOX(monkey)$

添加表：$\neg MONBOX(monkey)$

Monkey_Upbox(b)

条件：$ON(monkey,b)$、$ON(box,b)$、$\neg MONBOX(monkey)$、$\neg HOLDS(monkey)$

动作：删除表：$\neg MONBOX(monkey)$

添加表：$MONBOX(monkey)$

Operation_Banana_Picking(b)

条件：$ON(monkey,b)$、$ON(box,b)$、$MONBOX(monkey)$、$\neg HOLDS(monkey)$

动作：删除表：$\neg HOLDS(monkey)$

添加表：$HOLDS(monkey)$

2.7 求解过程

状态一：

$$\mathrm{ON}(monkey,A)\wedge HANG(banana,B)\wedge \mathrm{ON}(box,C)\wedge \neg MONBOX(monkey)\wedge \neg HOLDS(monkey)$$

动作一：Money_Move(A,C)
状态二：

$$\mathrm{ON}(monkey,C)\wedge HANG(banana,B)\wedge \mathrm{ON}(box,C)\wedge \neg MONBOX(monkey)\wedge \neg HOLDS(monkey)$$

动作二：Monkey_Movebox(C,B)
状态三：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge \mathrm{ON}(box,B)\wedge \neg MONBOX(monkey)\wedge \neg HOLDS(monkey)$$

动作三：Monkey_Upbox(B)
状态四：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge \mathrm{ON}(box,B)\wedge MONBOX(monkey)\wedge \neg HOLDS(monkey)$$

动作四:Operation_Banana_Picking(B)
状态五：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge \mathrm{ON}(box,B)\wedge MONBOX(monkey)\wedge HOLDS(monkey)$$

三. 算法实现

3.1 建立一个位置数组 A、B、C 分别为 1、2、3

```
1.  Position_Array={'1':'A','2':'B','3':'C'}
```

3.2 定义 Condition 类，存入各个元素的位置信息

本环节将输入的四个数分别赋值到 Condition 类中的 monkey、banana、box、monkey_onbox 四个状态值。

```
1.  class Condition:
2.      def __init__(Condition,monkey=1,banana=2,box=3,monkey
    _onbox=0):
3.          Condition.monkey=monkey
4.          Condition.banana=banana
5.          Condition.box=box
6.          Condition.monkey_onbox=monkey_onbox
```

3.3 判断输入的元素位置信息是否合法

本环节判断输入的四个数是否合乎常规，即 monkey、banana、box 三个元素不能超越其位置限制，monkey_onbox 不能超出 0\1 限制，当猴子在箱子上的时候猴子和箱子的位置要相等。若输入值合法则返回值，否则退出程序。

```
1.  def Legal_Judgment(Condition):
2.      if (Condition.monkey<1 or Condition.monkey>3)\
3.      or(Condition.banana<1 or Condition.banana>3)\
4.      or(Condition.box<1 or Condition.box>3)\
5.      or(Condition.monkey_onbox<0 or Condition.monkey_onbox
    >1)\
6.      or(Condition.monkey!=Condition.box and Condition.monk
    ey_onbox==1):
7.          print("输入元素位置信息不合法")
8.          os._exit(0)
9.      else:
10.         return Condition
```

3.4 猴子运动函数

本环节将猴子的位置信息赋值为箱子的位置信息，达到猴子运动到箱子的动作功能。

```
1.  def Monkey_Move(Condition,Moment):
2.          print(" 第 "+str(Moment)+" 步 , 猴 子 从 位 置 "+str(Position_Array[str(Condition.monkey)])+ \
3.                                          " 运 动 到 位 置 "+ str(Position_Array[str(Condition.box)]))
4.      Condition.monkey=Condition.box
5.      return Condition
```

3.5 猴子移动箱子函数

本环节将猴子的位置信息赋值为香蕉的位置信息，将箱子的位置信息赋值为香蕉的位置信息，达到猴子移动到箱子到香蕉下面的动作功能。

```
1.  def Monkey_Movebox(Condition,Moment):
2.          print(" 第 "+str(Moment)+" 步 , 猴 子 将 箱 子 从 位 置 "+str(Position_Array[str(Condition.box)])+ \
3.                                          " 移 动 到 位 置 "+str(Position_Array[str(Condition.banana)]))
4.      Condition.monkey=Condition.banana
5.      Condition.box=Condition.banana
6.      return Condition
```

3.6 猴子从箱子上下来函数

本环节将 monkey_onbox 从 1 变为 0，达到猴子从箱子上下来的动作功能。

```
1.  def Monkey_Downbox(Condition,Moment):
2.      Condition.monkey_onbox=0
3.      print("第"+str(Moment)+"步,猴子从箱子上下来")
4.      return Condition
```

3.7 猴子爬上箱子函数

本环节将 monkey_onbox 从 0 变为 1，达到猴子爬上箱子的动作功能。

```
1.  def Monkey_Upbox(Condition,Moment):
2.      Condition.monkey_onbox=1
3.      print("第"+str(Moment)+"步,猴子爬上箱子")
4.      return Condition
```

3.8 摘香蕉行动函数

本环节大量运用判断语句。整体思路是：

若猴子、箱子、香蕉位置相同且猴子在箱子上面，就摘到香蕉。

若猴子在箱子上但不和香蕉位置相同，则让猴子从箱子上下来、移动箱子到香蕉位置、爬上箱子、摘到香蕉。

若猴子不在箱子上但三者位置相同，则让猴子爬上箱子、摘到香蕉。

若猴子不在箱子上、猴子位置与箱子位置相同但与香蕉位置不同，则让猴子移动箱子到香蕉位置、爬上箱子、摘到香蕉。

若猴子不在箱子上、猴子位置与箱子位置不同、箱子位置与香蕉位置相同，则让猴子运动到箱子位置、爬上箱子、摘到香蕉。

若猴子不在箱子上、猴子、箱子、香蕉位置各不相同，则让猴子运动到箱子位置、移动箱子到香蕉位置、爬上箱子、摘到香蕉。

```
1.  def Operation_Banana_Picking(Condition):
2.      Moment=1
3.      if Condition.monkey==Condition.banana \
4.      and Condition.box==Condition.banana \
5.      and Condition.monkey_onbox==1:
6.          print("第%d 步,猴子摘到香蕉"%(Moment))
7.      else:
8.          if Condition.monkey_onbox==1:
9.              if Condition.monkey!=Condition.banana:
10.                 Condition=Monkey_Upbox(Monkey_Movebox(Mon
    key_Downbox(Condition,Moment),Moment+1),Moment+2)
11.                 Moment=Moment+3
12.         else:
13.             if Condition.monkey==Condition.box:
14.                 if Condition.box==Condition.banana:
15.                     Condition=Monkey_Upbox(Condition,Mome
    nt)
16.                     Moment=Moment+1
17.                 else:
18.                     Condition=Monkey_Upbox(Monkey_Movebox
    (Condition,Moment),Moment+1)
19.                     Moment=Moment+2
20.             else:
21.                 if Condition.box==Condition.banana:
22.                     Condition=Monkey_Upbox(Monkey_Move(Co
    ndition,Moment),Moment+1)
23.                     Moment=Moment+2
24.                 else:
```

```
                Condition=Monkey_Upbox(Monkey_Movebox(Monkey_Move(Condition,Moment),Moment+1),Moment+2)
                Moment=Moment+3
        print("第%d 步,猴子摘到香蕉"%(Moment))
```

3.9 主函数接收输入的状态变量，对其进行合法检验，最终开始摘香蕉行动

本环节读取各个元素的位置信息，即四个输入值，然后将其分别存入 Conditon 类中，调用判断输入的元素位置信息是否合法环境函数、最后调用摘香蕉行动函数。

```
if __name__=="__main__":
    Reminder=input("请输入各个元素的位置信息:\n")
    Condition_Input=Reminder.split(" ")
    Condition=Condition(int(Condition_Input[0]),int(Condition_Input[1]),int(Condition_Input[2]),int(Condition_Input[3]))
    Condition=Legal_Judgment(Condition)
    Operation_Banana_Picking(Condition)
```

四. 讨论及结论

4.1 实验验证

（1）猴子在 A 处，香蕉在 B 处，箱子在 C 处，猴子不在箱子上

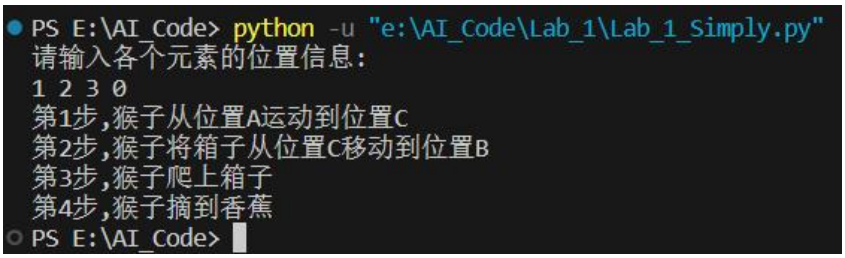

图 1 测试 1

（2）猴子在 A 处，香蕉在 C 处，箱子在 A 处，猴子在箱子上

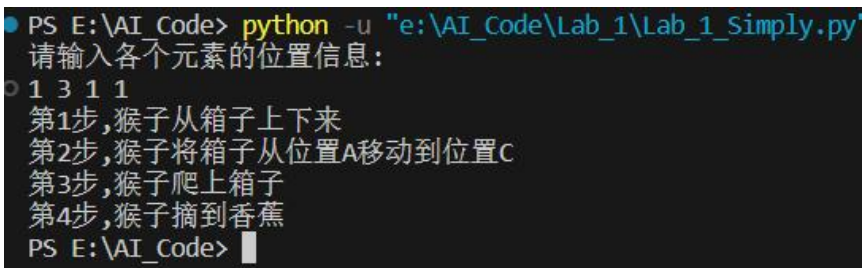

图 2 测试 2

（3）猴子在 A 处，香蕉在 A 处，箱子在 A 处，猴子不在箱子上

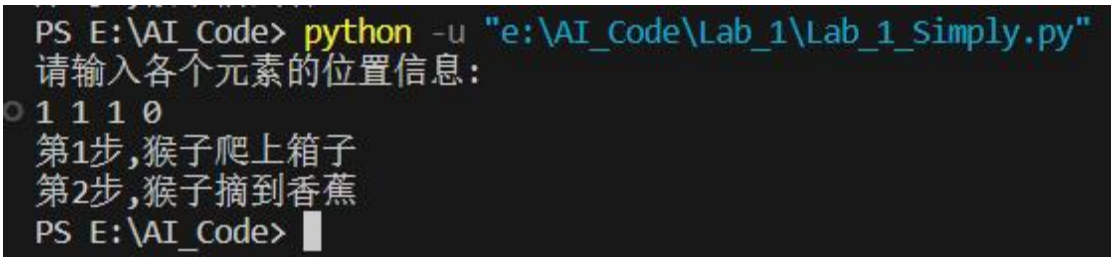

图 3 测试 3

（4）猴子在 A 处，香蕉在 A 处，箱子在 A 处，猴子在箱子上

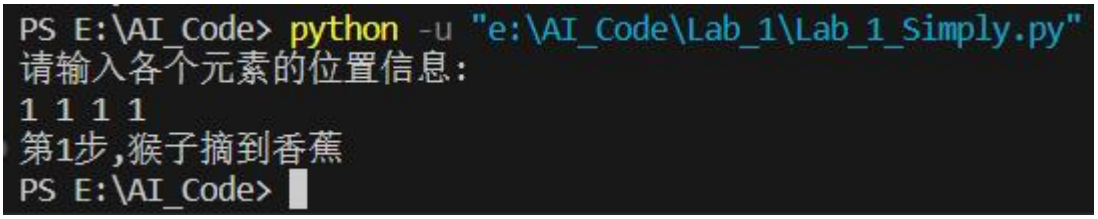

图 4 测试 4

4.2总结讨论

通过本次实验，我又熟悉了知识的表示方法，尤其是一阶谓词表示和产生式系统，加深了对该类问题的印象，提高了解决该类问题的能力。我认识到知识表示的重要性，合适的知识表示可以大大简化问题，加速算法的求解过程。通过本次实验，我也熟练了解 python 编程的过程，在程序优化中锻炼自己。

第二章 进阶实验——老师说的拓展内容

一．背景简介/问题描述

1.1 背景介绍

猴子摘香蕉问题：

一个房间里，天花板上挂有一串香蕉，一瓶毒药，有一只猴子可在房间里任意活动（到处走动，推移箱子，攀登箱子等）。设房间里还有两只可被猴子移动的箱子，且猴子登上箱子时才能摘到香蕉（躲避毒药），问猴子在某一状态下，如何行动可摘取到香蕉。（猴子可以在摘到香蕉前执行破坏毒药的任务）

假设猴子在 A，香蕉在 B，毒药在 C，1 号箱子在 D，2 号箱子在 E。

二．算法介绍

2.1 定义描述状态的谓词

ON(x,y): x 在 y 处；
HANG(w,y):w 悬挂在 y 处；
IFON(z):z 不在箱子上；
MONBOX (z,a): z 站在 a 上；
EMPTY(z):z 手中是空的；
HOLDS(z,b): z 手里拿着 b

2.2 变元的个体域

w 的个体域是{banana,poison}
x 的个体域是{monkey,box_1,box_2}
y 的个体域是{A,B,C,D,E,F}
z 的个体域是{monkey}

2.3 问题的初始状态

$$\mathrm{ON}(monkey,A)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,D)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge IFON(monkey)\wedge EMPTY(monkey)$$

2.4 问题的目标状态

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,B)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge MONBOX(monkey,box\_1)\wedge HOLDS(monkey,banana)$$

2.5 描述操作的谓词

Monkey_Move(a,b):猴子从 a 处运动到 b 处
Monkey_Movebox_1(b,c):猴子推 1 号箱子从 b 处运动到 c 处

Monkey_Movebox_2(b,c):猴子推 2 号箱子从 b 处运动到 c 处

Monkey_Downbox_1(b):猴子在 b 处从 1 号箱子上下来
Monkey_Downbox_2(b):猴子在 b 处从 2 号箱子上下来
Monkey_Upbox_1(b):猴子在 b 处爬上 1 号箱子
Monkey_Upbox_2(b):猴子在 b 处爬上 2 号箱子
Operation_Picking(b):猴子在 b 处拿到（毒药或香蕉）

2.6 各操作的条件和动作

Monkey_Move(a,b)

条件：$ON(monkey,a)$、$EMPTY(monkey)$

动作：删除表：$ON(monkey,a)$

添加表：$ON(monkey,b)$

Monkey_Movebox_1(b,c)

条件：$ON(monkey,b)$、$ON(box\_1,b)$、$IFON(monkey)$、$EMPTY(monkey)$

动作：删除表：$ON(monkey,b)$、$ON(box\_1,b)$

添加表：$ON(monkey,c)$、$ON(box\_1,c)$

Monkey_Movebox_2(b,c)

条件：$ON(monkey,b)$、$ON(box\_2,b)$、$IFON(monkey)$、$EMPTY(monkey)$

动作：删除表：$ON(monkey,b)$、$ON(box\_2,b)$

添加表：$ON(monkey,c)$、$ON(box\_2,c)$

Monkey_Downbox_1(b)

条件：$ON(monkey,b)$、$ON(box\_1,b)$

$MONBOX(monkey,box\_1)$、$EMPTY(monkey)$

动作：删除表：$MONBOX(monkey,box\_1)$

添加表：$IFON(monkey)$

Monkey_Downbox_2(b)

条件：$ON(monkey,b)$、$ON(box\_2,b)$

$MONBOX(monkey,box\_2)$、$EMPTY(monkey)$

动作：删除表：$MONBOX(monkey, box\_2)$

添加表：$IFON(monkey)$

Monkey_Upbox_1(b)

条件：$ON(monkey, b)$、$ON(box\_1, b)$、$IFON(monkey)$、$EMPTY(monkey)$

动作：删除表：$IFON(monkey)$

添加表：$MONBOX(monkey, box\_1)$

Monkey_Upbox_2(b)

条件：$ON(monkey, b)$、$ON(box\_2, b)$、$IFON(monkey)$、$EMPTY(monkey)$

动作：删除表：$IFON(monkey)$

添加表：$MONBOX(monkey, box\_2)$

Operation_Picking(b)（以在 1 号箱子摘到香蕉为例）

条件：$ON(monkey, b)$、$ON(box\_1, b)$、

$MONBOX(monkey, box\_1)$、$EMPTY(monkey)$

动作：删除表：$EMPTY(monkey)$

添加表：$HOLDS(monkey, banana)$

2.7 求解过程

状态一：

$$\mathrm{ON}(monkey, A) \wedge HANG(banana, B) \wedge HANG(poison, C) \wedge \mathrm{ON}(box\_1, D) \wedge \mathrm{ON}(box\_2, E) \wedge IFON(monkey) \wedge EMPTY(monkey)$$

动作一：Monkey_Move(A,D)

状态二：

$$\mathrm{ON}(monkey, D) \wedge HANG(banana, B) \wedge HANG(poison, C) \wedge \mathrm{ON}(box\_1, D) \wedge \mathrm{ON}(box\_2, E) \wedge IFON(monkey) \wedge EMPTY(monkey)$$

动作二：Monkey_Movebox_1(D,C)

状态三：

$$\mathrm{ON}(monkey, C) \wedge HANG(banana, B) \wedge HANG(poison, C) \wedge \mathrm{ON}(box\_1, C) \wedge \mathrm{ON}(box\_2, E) \wedge IFON(monkey) \wedge EMPTY(monkey)$$

动作三：Monkey_Upbox_1(C)
状态四：

$$\mathrm{ON}(monkey,C)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,C)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge MONBOX(monkey,box\_1)\wedge EMPTY(monkey)$$

动作四：Operation_Picking(C)
状态五：

$$\mathrm{ON}(monkey,C)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,C)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge MONBOX(monkey,box\_1)\wedge HOLDS(monkey,poison)$$

动作五：Monkey_Downbox_1(C)
状态六：

$$\mathrm{ON}(monkey,C)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,C)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge IFON(monkey)\wedge EMPTY(monkey)$$

动作六：Monkey_Movebox_1(C,B)
状态七：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,B)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge IFON(monkey)\wedge EMPTY(monkey)$$

动作七：Monkey_Upbox_1(B)

状态八：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,B)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge MONBOX(monkey,box\_1)\wedge EMPTY(monkey)$$

动作八：Operation_Picking(B)

状态九：

$$\mathrm{ON}(monkey,B)\wedge HANG(banana,B)\wedge HANG(poison,C)\wedge \mathrm{ON}(box\_1,B)$$
$$\wedge \mathrm{ON}(box\_2,E)\wedge MONBOX(monkey,box\_1)\wedge HOLDS(monkey,banana)$$

三．算法实现

3.1建立一个位置数组A、B、C、D、E、F分别为1、2、3、4、5、6

```
1.  Position_Array={'1':'A','2':'B','3':'C','4':'D','5':'E','
    6':'F'}
```

3.2定义Condition类，存入各个元素的位置信息

本环节将输入的四个数分别赋值到 Condition 类中的 monkey、banana、poison、box_1、box_2、monkey_onbox_1、monkey_onbox_2 七个状态值。

```
class Condition:
    def __init__(Condition,monkey=1,banana=2,poison=3,box_1=4,box_2=5,monkey_onbox_1=0,monkey_onbox_2=0):
        Condition.monkey=monkey
        Condition.banana=banana
        Condition.poison=poison
        Condition.box_1=box_1
        Condition.box_2=box_2
        Condition.monkey_onbox_1=monkey_onbox_1
        Condition.monkey_onbox_2=monkey_onbox_2
```

3.3判断输入的元素位置信息是否合法

本环节判断输入的四个数是否合乎常规，即 monkey、banana、poison、box_1、box_2 五个元素不能超越其位置限制，monkey_onbox_1、monkey_onbox_2 不能超出 0\1 限制且不能相等，当猴子在箱子上的时候猴子和箱子的位置要相等。若输入值合法则返回值，否则退出程序。

```
def Legal_Judgment(Condition):
    if (Condition.monkey<1 or Condition.monkey>6)\
    or(Condition.banana<1 or Condition.banana>6)\
    or(Condition.poison<1 or Condition.poison>6)\
    or(Condition.box_1<1 or Condition.box_1>6)\
    or(Condition.box_2<1 or Condition.box_2>6)\
    or(Condition.monkey_onbox_1<0 or Condition.monkey_onbox_1>1)\
    or(Condition.monkey_onbox_2<0 or Condition.monkey_onbox_2>1)\
    or(Condition.monkey!=Condition.box_1 and Condition.monkey_onbox_1==1)\
    or(Condition.monkey!=Condition.box_2 and Condition.monkey_onbox_2==1)\
    or(Condition.box_1==Condition.box_2):
        print("输入元素位置信息不合法")
        os._exit(0)
    else:
        return Condition
```

3.4猴子运动到1号箱子函数

本环节将猴子的位置信息赋值为 1 号箱子的位置信息，达到猴子运动到箱子的动作功能。

```
  def Monkey_Move_1(Condition,Moment):
            print(" 第 "+str(Moment)+" 步 , 猴 子 从 位 置 "+str(Position_Array[str(Condition.monkey)])+ \
                                        " 运 动 到 位 置 "+ str(Position_Array[str(Condition.box_1)]))
      Condition.monkey=Condition.box_1
      return Condition
```

3.5猴子运动到2号箱子函数

本环节将猴子的位置信息赋值为 2 号箱子的位置信息，达到猴子运动到箱子的动作功能。

```
  def Monkey_Move_2(Condition,Moment):
            print(" 第 "+str(Moment)+" 步 , 猴 子 从 位 置 "+str(Position_Array[str(Condition.monkey)])+ \
                                        " 运 动 到 位 置 "+ str(Position_Array[str(Condition.box_2)]))
      Condition.monkey=Condition.box_2
      return Condition
```

3.6猴子移动1号箱子到香蕉函数

本环节将猴子的位置信息赋值为香蕉的位置信息，将 1 号箱子的位置信息赋值为香蕉的位置信息，达到猴子移动到箱子到香蕉下面的动作功能。

```
  def Monkey_Movebox_1_banana(Condition,Moment):
        print(" 第 "+str(Moment)+" 步 , 猴 子 将 1 号 箱 子 从 位 置 "+str(Position_Array[str(Condition.box_1)])+ \
                                        " 移 动 到 位 置 "+str(Position_Array[str(Condition.banana)]))
      Condition.monkey=Condition.banana
      Condition.box_1=Condition.banana
      return Condition
```

3.7猴子移动2号箱子到香蕉函数

本环节将猴子的位置信息赋值为香蕉的位置信息，将 2 号箱子的位置信息赋值为香蕉的位置信息，达到猴子移动到箱子到香蕉下面的动作功能。

```
  def Monkey_Movebox_2_banana(Condition,Moment):
        print(" 第 "+str(Moment)+" 步 , 猴 子 将 2 号 箱 子 从 位 置 "+str(Position_Array[str(Condition.box_2)])+ \
                                        " 移 动 到 位 置 "+str(Position_Array[str(Condition.banana)]))
      Condition.monkey=Condition.banana
      Condition.box_2=Condition.banana
```

```
6.        return Condition
```

3.8猴子移动1号箱子到香毒药函数

本环节将猴子的位置信息赋值为毒药的位置信息，将 1 号箱子的位置信息赋值为毒药的位置信息，达到猴子移动到箱子到毒药下面的动作功能。

```
1.   def Monkey_Movebox_1_posion(Condition,Moment):
2.         print("第"+str(Moment)+"步，猴子将 1 号箱子从位置"+str(Position_Array[str(Condition.box_1)])+ \
3.                                         "移动到位置"+str(Position_Array[str(Condition.poison)]))
4.        Condition.monkey=Condition.poison
5.        Condition.box_1=Condition.poison
6.        return Condition
```

3.9猴子移动2号箱子到香毒药函数

本环节将猴子的位置信息赋值为毒药的位置信息，将 2 号箱子的位置信息赋值为毒药的位置信息，达到猴子移动到箱子到毒药下面的动作功能。

```
1.   def Monkey_Movebox_2_posion(Condition,Moment):
2.         print("第"+str(Moment)+"步，猴子将 2 号箱子从位置"+str(Position_Array[str(Condition.box_2)])+ \
3.                                         "移动到位置"+str(Position_Array[str(Condition.poison)]))
4.        Condition.monkey=Condition.poison
5.        Condition.box_2=Condition.poison
6.        return Condition
```

3.10猴子从1号箱子上下来函数

本环节将 monkey_onbox_1 从 1 变为 0，达到猴子从 1 号箱子上下来的动作功能。

```
1.   def Monkey_Downbox_1(Condition,Moment):
2.        Condition.monkey_onbox_1=0
3.        print("第"+str(Moment)+"步,猴子从 1 号箱子上下来")
4.        return Condition
```

3.11猴子爬上1号箱子函数

本环节将 monkey_onbox_1 从 0 变为 1，达到猴子爬上 1 号箱子的动作功能。

```
1.   def Monkey_Upbox_1(Condition,Moment):
2.        Condition.monkey_onbox_1=1
3.        print("第"+str(Moment)+"步,猴子爬上 1 号箱子")
4.        return Condition
```

3.12猴子从2号箱子上下来函数

本环节将 monkey_onbox_2 从 1 变为 0，达到猴子从 2 号箱子上下来的动作功能。

```
1.  def Monkey_Downbox_2(Condition,Moment):
2.      Condition.monkey_onbox_2=0
3.      print("第"+str(Moment)+"步,猴子从 2 号箱子上下来")
4.      return Condition
```

3.13猴子爬上2号箱子函数

本环节将 monkey_onbox_2 从 0 变为 1，达到猴子爬上 2 号箱子的动作功能。

```
1.  def Monkey_Upbox_2(Condition,Moment):
2.      Condition.monkey_onbox_2=1
3.      print("第"+str(Moment)+"步,猴子爬上 2 号箱子")
4.      return Condition
```

3.14 摘香蕉行动函数

本环节大量运用判断语句。整体思路是：

若猴子、1 号箱子、香蕉位置相同且猴子在 1 号箱子上面或猴子、2 号箱子、香蕉位置相同且猴子在 2 号箱子上面，就摘到香蕉。

若猴子在 1 号箱子上但不和香蕉位置相同，则让猴子从 1 号箱子上下来、移动 1 号箱子到香蕉位置、爬上 1 号箱子、摘到香蕉。若猴子在 2 号箱子上但不和香蕉位置相同，则让猴子从 2 号箱子上下来、移动 2 号箱子到香蕉位置、爬上 2 号箱子、摘到香蕉。

若猴子不在 1 号或 2 号箱子上但三者位置相同，则让猴子爬上 1 号或 2 号箱子、摘到香蕉。

若猴子不在 1 号或 2 号箱子上、猴子位置与 1 号或 2 号箱子位置相同但 1 号或 2 号箱子位置与香蕉位置不同，则让猴子移动 1 号或 2 号箱子到香蕉位置、爬上 1 号或 2 号箱子、摘到香蕉。

若猴子不在 1 号或 2 号箱子上、猴子位置与 1 号或 2 号箱子位置不同、1 号或 2 号箱子位置与香蕉位置相同，则让猴子运动到 1 号或 2 号箱子位置、爬上 1 号或 2 号箱子、摘到香蕉。

若猴子不在箱子上、猴子、箱子、香蕉位置各不相同，则根据猴子和 1 号和 2 号的远近决定让猴子选择最近可行路径的箱子，运动到箱子位置、移动箱子到

香蕉位置、爬上箱子、摘到香蕉。

```
 def Operation_Banana_Picking(Condition,Moment):
     if Condition.monkey==Condition.banana \
      and (Condition.box_1==Condition.banana and Condition.monkey_onbox_1==1) \
      or (Condition.box_2==Condition.banana and Condition.monkey_onbox_2==1):
         global End
         End=Moment
         print("第"+str(Moment)+"步,猴子摘到香蕉")
     else:
          if (Condition.monkey_onbox_1==1) or (Condition.monkey_onbox_2==1):
              if Condition.monkey_onbox_1==1:
                  if Condition.monkey!=Condition.banana:
                       Condition=Monkey_Upbox_1(Monkey_Movebox_1_banana(Monkey_Downbox_1(Condition,Moment),Moment+1),Moment+2)
                      Moment=Moment+3
              else:
                  if Condition.monkey!=Condition.banana:
                       Condition=Monkey_Upbox_2(Monkey_Movebox_2_banana(Monkey_Downbox_2(Condition,Moment),Moment+1),Moment+2)
                      Moment=Moment+3
          else:
               if (Condition.monkey==Condition.box_1) or (Condition.monkey==Condition.box_2):
                  if Condition.monkey==Condition.box_1:
                       if Condition.box_1==Condition.banana:

                           Condition=Monkey_Upbox_1(Condition,Moment)
                          Moment=Moment+1
                      else:
                           Condition=Monkey_Upbox_1(Monkey_Movebox_1_banana(Condition,Moment),Moment+1)

                          Moment=Moment+2
                  else:
                       if Condition.box_2==Condition.banana:

                           Condition=Monkey_Upbox_2(Condition,Moment)
```

```
                            Moment=Moment+1
                        else:
                            Condition=Monkey_Upbox_2(Monkey_Movebox_2_banana(Condition,Moment),Moment+1)

                            Moment=Moment+2
                else:
                    if (Condition.box_1==Condition.banana) or (Condition.box_2==Condition.banana):
                        if Condition.box_1==Condition.banana:

                            Condition=Monkey_Upbox_1(Monkey_Move_1(Condition,Moment),Moment+1)
                            Moment=Moment+2
                        else:
                            Condition=Monkey_Upbox_2(Monkey_Move_2(Condition,Moment),Moment+1)
                            Moment=Moment+2
                    else:
                        if(abs(Condition.monkey-Condition.box_1)+abs(Condition.monkey-Condition.box_1)< \
                            abs(Condition.monkey-Condition.box_2)+abs(Condition.monkey-Condition.box_2)):
                            Condition=Monkey_Upbox_1(Monkey_Movebox_1_banana(Monkey_Move_1(Condition,Moment),Moment+1),Moment+2)
                            Moment=Moment+3
                        if(abs(Condition.monkey-Condition.box_1)+abs(Condition.monkey-Condition.box_1)> \
                            abs(Condition.monkey-Condition.box_2)+abs(Condition.monkey-Condition.box_2)):
                            Condition=Monkey_Upbox_2(Monkey_Movebox_2_banana(Monkey_Move_2(Condition,Moment),Moment+1),Moment+2)
                            Moment=Moment+3
                        else:
                            if(abs(Condition.box_1-Condition.banana)<abs(Condition.box_1-Condition.banana)):
                                Condition=Monkey_Upbox_1(Monkey_Movebox_1_banana(Monkey_Move_1(Condition,Moment),Moment+1),Moment+2)
                                Moment=Moment+3
                            if(abs(Condition.box_1-Condition.banana)>abs(Condition.box_1-Condition.banana)):
```

```
                                    Condition=Monkey_Upbox_2(Monkey_Movebox_2_banana(Monkey_Move_2(Condition,Moment),Moment+1),Moment+2)
                                    Moment=Moment+3
            End=Moment+1
            print("第"+str(Moment)+"步,猴子摘到香蕉")
```

3.15 打碎毒药行动函数

本环节大量运用判断语句。整体思路是：

若猴子、1 号箱子、毒药位置相同且猴子在 1 号箱子上面或猴子、2 号箱子、毒药位置相同且猴子在 2 号箱子上面，就打碎毒药。

若猴子在 1 号箱子上但不和毒药位置相同，则让猴子从 1 号箱子上下来、移动 1 号箱子到毒药位置、爬上 1 号箱子、摘到毒药。若猴子在 2 号箱子上但不和毒药位置相同，则让猴子从 2 号箱子上下来、移动 2 号箱子到毒药位置、爬上 2 号箱子、打碎毒药。

若猴子不在 1 号或 2 号箱子上但三者位置相同，则让猴子爬上 1 号或 2 号箱子、打碎毒药。

若猴子不在 1 号或 2 号箱子上、猴子位置与 1 号或 2 号箱子位置相同但 1 号或 2 号箱子位置与毒药位置不同，则让猴子移动 1 号或 2 号箱子到毒药位置、爬上 1 号或 2 号箱子、打碎毒药。

若猴子不在 1 号或 2 号箱子上、猴子位置与 1 号或 2 号箱子位置不同、1 号或 2 号箱子位置与毒药位置相同，则让猴子运动到 1 号或 2 号箱子位置、爬上 1 号或 2 号箱子、打碎毒药。

若猴子不在箱子上、猴子、箱子、毒药位置各不相同，则根据猴子和 1 号和 2 号的远近决定让猴子选择最近可行路径的箱子，运动到箱子位置、移动箱子到毒药位置、爬上箱子、打碎毒药。

```
  def Destory_Posion(Condition,Moment):
      if Condition.monkey==Condition.poison \
       and (Condition.box_1==Condition.poison and Condition.monkey_onbox_1==1) \
       or (Condition.box_2==Condition.poison and Condition.monkey_onbox_2==1):
          global End
          End=Moment
          print("第"+str(Moment)+"步,猴子破坏毒药")
```

```
8.      else:
9.          if (Condition.monkey_onbox_1==1) or (Condition.mo
    nkey_onbox_2==1):
10.             if Condition.monkey_onbox_1==1:
11.                 if Condition.monkey!=Condition.poison:
12.                     Condition=Monkey_Upbox_1(Monkey_Moveb
    ox_1_posion(Monkey_Downbox_1(Condition,Moment),Moment+1),M
    oment+2)
13.                     Moment=Moment+3
14.             else:
15.                 if Condition.monkey!=Condition.poison:
16.                     Condition=Monkey_Upbox_2(Monkey_Moveb
    ox_2_posion(Monkey_Downbox_2(Condition,Moment),Moment+1),M
    oment+2)
17.                     Moment=Moment+3
18.         else:
19.             if (Condition.monkey==Condition.box_1) or (Co
    ndition.monkey==Condition.box_2):
20.                 if Condition.monkey==Condition.box_1:
21.                     if Condition.box_1==Condition.poison:

22.                         Condition=Monkey_Upbox_1(Conditio
    n,Moment)
23.                         Moment=Moment+1
24.                     else:
25.                         Condition=Monkey_Upbox_1(Monkey_M
    ovebox_1_posion(Condition,Moment),Moment+1)

26.                         Moment=Moment+2
27.                 else:
28.                     if Condition.box_2==Condition.poison:

29.                         Condition=Monkey_Upbox_2(Conditio
    n,Moment)
30.                         Moment=Moment+1
31.                     else:
32.                         Condition=Monkey_Upbox_2(Monkey_M
    ovebox_2_posion(Condition,Moment),Moment+1)

33.                         Moment=Moment+2
34.             else:
35.                 if (Condition.box_1==Condition.poison) or
     (Condition.box_2==Condition.poison):
```

```
                    if Condition.box_1==Condition.poison:
                        Condition=Monkey_Upbox_1(Monkey_Move_1(Condition,Moment),Moment+1)
                        Moment=Moment+2
                    else:
                        Condition=Monkey_Upbox_2(Monkey_Move_2(Condition,Moment),Moment+1)
                        Moment=Moment+2
                else:
                    if(abs(Condition.monkey-Condition.box_1)+abs(Condition.monkey-Condition.box_1)< \
                        abs(Condition.monkey-Condition.box_2)+abs(Condition.monkey-Condition.box_2)):
                        Condition=Monkey_Upbox_1(Monkey_Movebox_1_posion(Monkey_Move_1(Condition,Moment),Moment+1),Moment+2)
                        Moment=Moment+3
                    if(abs(Condition.monkey-Condition.box_1)+abs(Condition.monkey-Condition.box_1)> \
                        abs(Condition.monkey-Condition.box_2)+abs(Condition.monkey-Condition.box_2)):
                        Condition=Monkey_Upbox_2(Monkey_Movebox_2_posion(Monkey_Move_2(Condition,Moment),Moment+1),Moment+2)
                        Moment=Moment+3
                else:
                    if(abs(Condition.box_1-Condition.poison)<abs(Condition.box_1-Condition.poison)):
                        Condition=Monkey_Upbox_1(Monkey_Movebox_1_posion(Monkey_Move_1(Condition,Moment),Moment+1),Moment+2)
                        Moment=Moment+3
                    if(abs(Condition.box_1-Condition.poison)>abs(Condition.box_1-Condition.poison)):
                        Condition=Monkey_Upbox_2(Monkey_Movebox_2_posion(Monkey_Move_2(Condition,Moment),Moment+1),Moment+2)
                        Moment=Moment+3
        End=Moment
        print("第"+str(Moment)+"步,猴子破坏毒药")
```

四．讨论及结论

4.1 实验验证

（1）猴子在A处，香蕉在B处，毒药在C处、1号箱子在D处，2号箱子在E处、猴子不在箱子上，启动先破坏毒药后摘香蕉任务

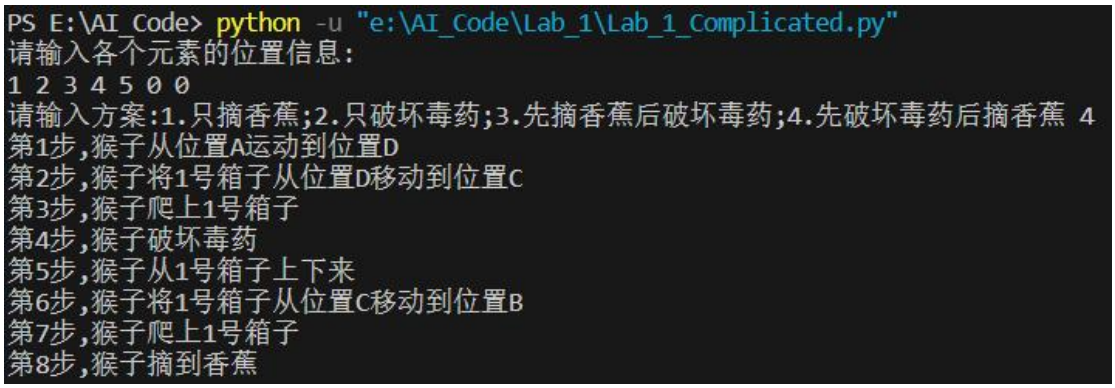

图 5 测试 5

（2）猴子在A处，香蕉在E处，毒药在C处、1号箱子在F处，2号箱子在A处、猴子不在箱子上，启动先摘香蕉后破坏毒药任务

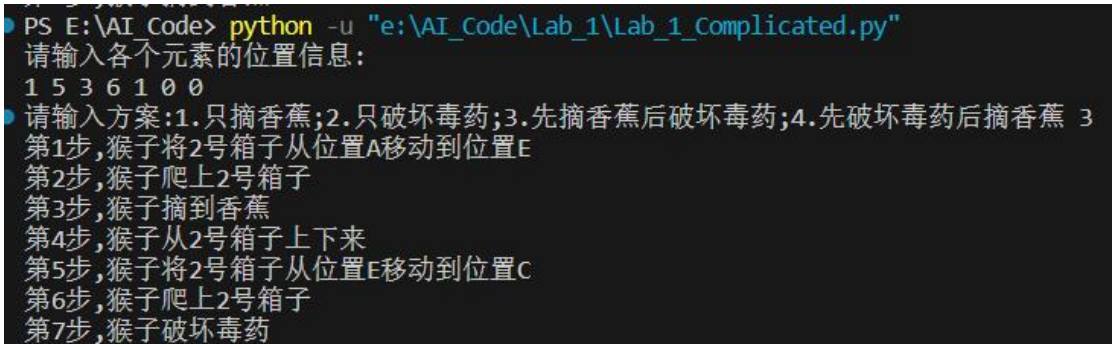

图 6 测试 6

（3）猴子在A处，香蕉在A处，毒药在B处、1号箱子在A处，2号箱子在B处、猴子在1号箱子上，启动先摘香蕉后破坏毒药任务

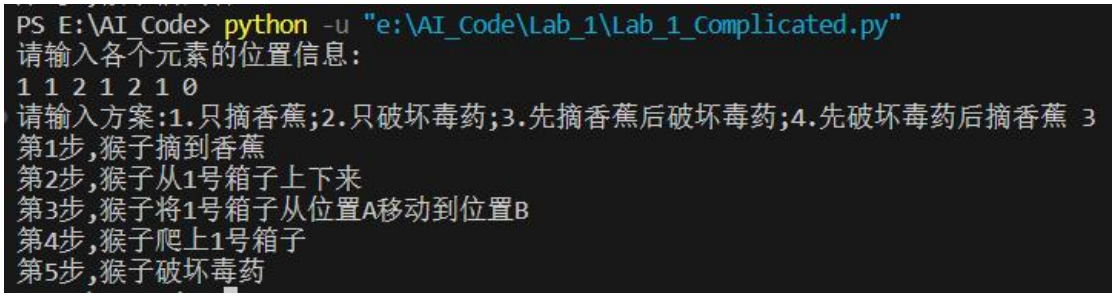

图 7 测试 7

4.2总结讨论

通过本次实验，我又熟悉了知识的表示方法，尤其是一阶谓词表示和产生式系统，加深了对该类问题的印象，提高了解决该类问题的能力。我认识到知识表示的重要性，合适的知识表示可以大大简化问题，加速算法的求解过程。通过本次实验，我也熟练了解 python 编程的过程，在程序优化中锻炼自己。除此之外，我通过进行拓展实验，也能更深刻地了解知识表示处理复杂问题时，扮演者重要角色。

参考文献

[1] https://github.com/usiege/UCAS/blob/master/

[2] 王万森.人工智能原理及其应用[M].北京：电子工业出版社，2012.09.01

[3] https://blog.csdn.net/hustlei/article/details/121688673