Part1 文档

选用的语言：prolog。

一，过河问题。

1. 如何运行，

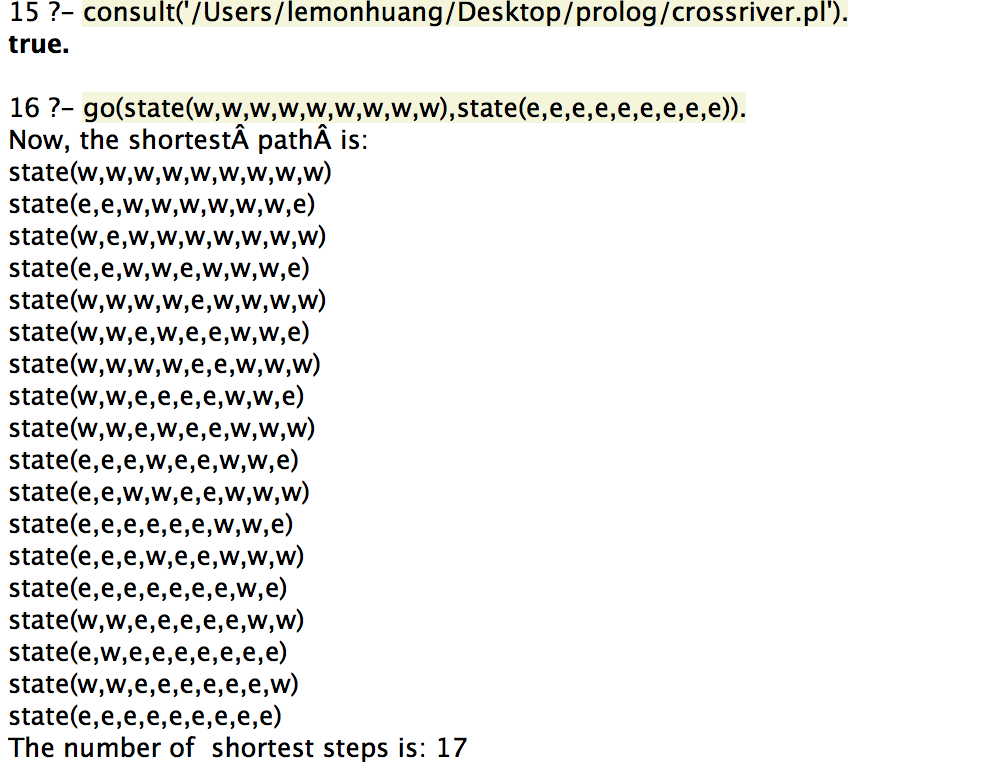
首先，在swi-prolog中加载文件‘crossriver.pl’，然后键入go(state(w,w,w,w,w,w,w,w,w),state(e,e,e,e,e,e,e,e,e)).运行即可。

图1，第一题运行截图

1. 如何解决该问题

对于过河问题，老师上课给出了一个农夫与狼、羊、大白菜一起过的例子，在我们的这次作业中，我也参考了这个例子。但是这里的过河问题比老师给的例子要复杂。

首先，这里有8个角色要过河，会划船的人变成了3个，在老师给的例子中只设置了四个状态，农夫、狼、羊和大白菜这四者的状态，这种设置方法在该题中是没有问题的，因为船的状态永远跟农夫的状态保持一致，所以船的状态是可以省略的。然而在本题中，由于会划船的人变成了3个，那么穿的状态就无法跟其中的某一个保持一致，所以要多设置一个船的状态，船只有在西岸时才能划向东岸，只有在东岸时才能划向西岸。

其次，unsafe的状态也会比例子中复杂。这里需要保证罪犯不会伤害家人，父亲不会惩罚女儿，母亲不会惩罚儿子。不仅要考虑两岸的情况，船上的情况也要考虑，父亲和母亲不能带着罪犯划船，父亲不能带着女儿划船，母亲不能带着儿子划船。

对于如何选择最短路径，这里我是通过list\_length([],K)方法得到每个可行解所要经过的路径步数，然后比较每次得到的步数，如果比之前的步数少，就将该次的路径打印出来，如果不比之前的步数少，就忽略这个结果。那么最后打印出来的就是最佳路径。中间通过assert方法将每次比较中比较小的路径存下来。

1. 自然语言叙述该问题的一个解决方案

这里用(Police,Criminal,Father,Mother,SonA,SonB,DaughterA,DaughterB,Boat)以此表示警察、罪犯、父亲、母亲、儿子1，儿子2，女儿1，女儿2，船的状态，下面的步骤将找到该问题的一个解（同时也是最优解，因为在此题中

1. 警察带罪犯过去。状态（e,e,w,w,w,w,w,w,e）
2. 警察回来 ,状态 (w,e,w,w,w,w,w,w,w)
3. 警察带第一个儿子过去，(e,e,w,w,e,w,w,w,e)
4. 警察带罪犯回来 (w,w,w,w,e,w,w,w,w)
5. 父亲带第二个儿子过去 (w,w,e,w,e,e,w,w,e)
6. 父亲自己回去 (w,w,w,w,e,e,w,w,w)
7. 父亲带母亲过去 (w,w,e,e,e,e,w,w,e)
8. 母亲回去 (w,w,e,w,e,e,w,w,w)
9. 警察带罪犯过去 (e,e,e,w,e,e,w,w,e)
10. 父亲回去 (e,e,w,w,e,e,w,w,w)
11. 父亲和母亲一起过去(e,e,e,e,e,e,w,w,e)
12. 母亲回去 (e,e,e,w,e,e,w,w,w)
13. 母亲和第一个女儿一起过去(e,e,e,e,e,e,e,w,e)
14. 警察和罪犯一起回去(w,w,e,e,e,e,e,w,w)
15. 警察带女儿过去 (e,w,e,e,e,e,e,e,e)
16. 警察回去 (w,w,e,e,e,e,e,e,w)
17. 警察带罪犯一起过去(e,e,e,e,e,e,e,e,e)

二，爱因斯坦的谜题

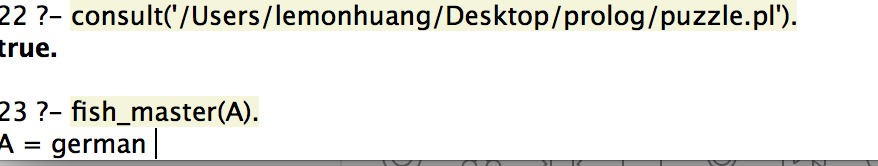
1. 如何运行：加载‘puzzle.pl’ 文件，然后键入fish\_master(M).方法，该方法将返回养鱼的人是谁。

图2 ，第二题运行截图

1. 关于如何实现

该问题比较简单，根据爱因斯坦讲述的故事，通过规则next *to来限定两个房子是相邻的，通过规则is*\_left来限定两个房子的左右关系，类似的，通过first和center规则来限定某个房子的位置。

之后将爱因斯坦讲说的故事一条条加进名为A的一个list中，程序将会一条条回溯可能的情况直至找到唯一的结果。最终，A中的结果即为所有的房子以及其相关属性的集合。通过查询其中的fish的所有者即可或者问题的答案。

1. 自然语言叙述该问题的解决方案，根据爱因斯坦的叙述，我们可以将其中的信息整合最后根据问题：养鱼的人是谁，我们可以知道是有一个人养鱼的，那么最终就可以将所有的房屋中的主人、主人的爱好、主人养的宠物完成匹配。最后得到的结果也是German养fish，与程序运行结果是一致的。

三，

1. 如何运行：

加载文件‘owncar.pl’，键入all\_owner(All).得到运行结果如下图：

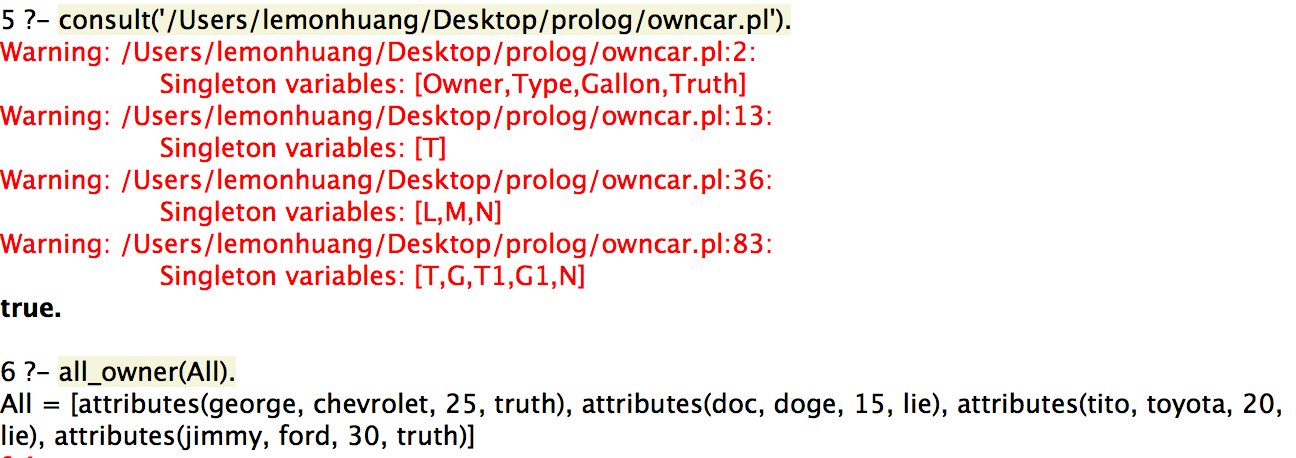


图3，第三题运行截图

1. 如何解决该问题：

类似于第二题，这个问题的解决也是要将许多的事实与汽车相关的一组属性进行匹配。在这里，与一辆汽车对应的有其所有者、汽车类型、汽车的gas mileage。然而不同的是这个问题中有些人可能是说谎的，所以并不是所有人说的话都能构成事实的一部分。由于gas mileage比较小的两位车主在谈到他们车的gas mileage的时候总是会说谎，那么他们谈到gas mileage的话不能成为事实。这是我们可以在原来的属性组中增加一条Truth的属性来判断车主是可信的或者说谎的，当其可信的时候他说的所有话都能构成事实，当他们说谎的时候，他们关于gas mileage的说法不能成为事实。

将各位车主所说的话罗列出来，然后将其中涉及到汽车所有者的话直接判定为事实，将涉及到gas mileage的话根据其是否可信来决定是不是能够成为事实。

将事实一条条进行匹配，并将结果放在名为All的一个list中，最终，通过程序的不断回溯，All中的结果即为所有的属性集合。

1. 自然语言描述该问题的解。

根据他们的对话，Doc说他的gas mileage是20，那么假设Doc是诚实的，那么他说他的gas mileage是20就是正确的，然而如果他的gas mileage是20，那么他总是会说谎的，所以两者矛盾，那么Doc一定不是诚实的而且他的gas mileage只能是15.

Tito说Doc的gas mileage是20，那么Tito是说谎的，所以Tito自己的gas mileage就是20了。

Tito是说谎的，那么他说George的gas mileage比Jimmy的高，则George的gas mileage比Jimmy的低，George的是25，Jimmy的是30.

Jimmy和George是诚实的，那么Jimmy说Tito's gas mileage is higher than the guy who drives the Dodge. 则Doc 的车型是Dodge。

根据George said: The guy who owns the Ford is getting 30 miles per gallon. The guy who gets 20 miles per gallon doesn't own a Chevrolet. 可知，Jimmy的是Ford，Tito的不是Chevrolet那么就是Toyota。

所以最终结果是：George，Chevrolet，25； Doc Dodge 15；

Tito，Toyota，20；Jimmy，Ford，30

与程序运行结果是一致的。