

- 题目
  - 逻辑模式
  - 分析
    - 命题
    - 知识
    - 任务目标
    - 思维过程
    - 结论
  - ASP 表示
  - AF 表示
  - 总结
- 

## 题目

古代一位国王和他的张、王、李、赵、钱五位将军，一同出外打猎，各人的箭上都刻有自己的姓氏。打猎中，一只鹿中箭倒下，但不知是何人所射。

张说：“或者是我射中的，或者是李将军射中的。”

王说：“不是钱将军射中的。”

李说：“如果不是赵将军射中的，那么一定是王将军射中的。”

赵说：“既不是我射中的，也不是王将军射中的。”

钱说：“既不是李将军射中的，也不是张将军射中的。”

国王让人把射中鹿的箭拿来，看了看，说：“你们五位将军的猜测，只有两个人的话是真的。”

请根据国王的话，判定以下哪项是真的？

(A) 张将军射中此鹿。 (B) 王将军射中此鹿。 (C) 李将军射中此鹿。 (D) 赵将军射中此鹿。 (E) 钱将军射中此鹿。

## 逻辑模式

溯因推理

## 分析

命题

- p1: 鹿是张将军或李将军射中的。
- p2: 鹿不是钱将军射中的。
- p3: 如果不是赵将军射中的，那么一定是王将军射中的。
- p4: 既不是赵将军射中的，也不是王将军射中的。
- p5: 既不是李将军射中的，也不是张将军射中的。

知识

- 将军：张将军、王将军、李将军、赵将军、钱将军。
- 有一个人射中了鹿。
- 只有两个命题为真。

## 任务目标

目标：哪位将军射中了鹿？

## 思维过程

- 若p1正确，鹿是张将军或李将军射中的。
  - p5错误，其与p1相互否定。
  - p3错误，其与p1相互否定。
  - p4正确，其符合p1。
  - p2正确，其符合p1。
  - 根据知识“只有两个命题为真。”，此次推论不成立，即p1正确不成立。
- 若p1错误，鹿不是张将军或李将军射中的。
  - p5正确，其与p1相互否定。
  - 若p3正确，如果不是赵将军射中的，那么一定是王将军射中的。
    - p4错误，其与p3相互否定。
    - p2正确，其符合p3。
    - 根据知识“只有两个命题为真。”，此次推论不成立，即p3正确不成立。
  - 若p3不正确，既不是赵将军射中的，也不是王将军射中的。
    - p4正确，其与p3相互否定。
    - 若p2正确，鹿不是钱将军射中的。
      - 根据知识“只有两个命题为真。”，此次推论不成立，即p2正确不成立。
    - 若p2错误，鹿是钱将军射中的。
      - 符合条件，因此结果为鹿是钱将军射中的。此时，p4、p5正确，p1、p2、p3错误。

## 结论

赵将军和钱将军的话是对的。鹿是钱将军射中的。

## ASP 表示

字面含义转化,易于阅读和理解,难以机械转换

```
person(z).
person(w).
person(l).
person(zhao).
person(q).
```

```
p(z) :- shoot(z), not shoot(l).
p(z) :- not shoot(z), shoot(l).
p(w) :- not shoot(q).
p(l) :- shoot(zhao).
p(l) :- not shoot(zhao), shoot(w).
p(zhao) :- not shoot(zhao), not shoot(w).
```

```
p(q) :- not shoot(l), not shoot(z).
```

```
1{shoot(X) : person(X)}1.
```

```
2{p(X) : person(X)}2.
```

```
#show p/1.
```

```
#show shoot/1.
```

添加一些基本规则,统一命题谓词,相对易于转化.

```
rule(eitherOr).
```

```
rule(neitherNor).
```

```
rule(ifThen).
```

```
rule(not_).
```

```
rule(bothAnd).
```

```
rule(nStatementsTrue).
```

```
rule(allStatementsTrue).
```

```
rule(noStatementTrue).
```

```
reason(X, Y, eitherOr) :-      p(X), not p(Y), X!=Y, r(X), r(Y).
```

```
reason(X, Y, eitherOr) :- not p(X),      p(Y), X!=Y, r(X), r(Y).
```

```
reason(X, Y, ifThen)   :-      p(X),      p(Y), X!=Y, r(X), r(Y).
```

```
reason(X, Y, ifThen)   :- not p(X),      X!=Y, r(X), r(Y).
```

```
reason(X, Y, bothAnd)  :-      p(X),      p(Y), X!=Y, r(X), r(Y).
```

```
% generated
```

```
r(X) :- person(X).
```

```
r(1..8).
```

```
r(shootz;shootl;shootw;shootq;shootzhao).
```

```
person(z).
```

```
person(w).
```

```
person(l).
```

```
person(zhao).
```

```
person(q).
```

```
p(z) :- reason(1, 2, eitherOr).
```

```
p(1) :- p(shootz).
```

```
p(2) :- p(shootl).
```

```
p(w) :- not p(shootq).
```

```
p(1) :- reason(3, 4, ifThen).
```

```
p(3) :- not p(shootzhao).
```

```
p(4) :- p(shootw).
```

```
p(zhao) :- reason(5, 6, bothAnd).
```

```
p(5) :- not p(shootzhao).
```

```
p(6) :- not p(shootw).
```

```

p(q) :- reason(7, 8, bothAnd).
p(7) :- not p(shootl).
p(8) :- not p(shootz).

p(king) :- 2{p(z);p(w);p(l);p(zhao);p(q)}2.
1{p(shootz);p(shootl);p(shootw);p(shootq);p(shootzhao)}1.
:- not p(king).

answer(X) :- p(X), r(X) = r(shootz;shootl;shootw;shootq;shootzhao).

#show p/1.

```

运行结果:

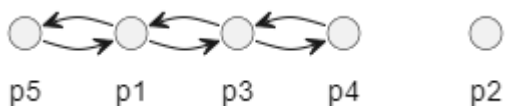
```

clingo version 5.4.0
Reading from ...gn\cases\case-analysis\asp-analysis.lp
Solving...
Answer: 1
p(shootq) p(3) p(5) p(6) p(7) p(8) p(zhao) p(q) p(king)
SATISFIABLE

Models      : 1
Calls       : 1
Time        : 0.010s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time    : 0.000s

```

## AF 表示



## 总结

- 可以统一谓词来表示逻辑类型的命题. 在实现中转化为了 `p(X) :- reason(a, b, type)` 格式；
- 没有归纳出“有n个命题成立”的推理规则，`X{p}X`；
- 要将原子命题转化为 `p(n) :- condition`, 对于复杂多变的问题，难以表达 `condition`。在本例中 `p(n) :- condition` 可以表示为 `p(n) :- p(shootz)`. 或 `p(n) :- shoot(zhao)`，但对于其他问题，都需要重新定义谓词。例如有的问题需要定义某人在某时刻吃了什么，则 `condition` 需要表达为 `p(n) :- eat(person, food, time)`。