

# 简单人工智能算法与离散数学

- 国外非常好的AI导论课: <https://cs50.harvard.edu/ai/2020/>, 非常基础的入门课, 课程项目有趣
- 与离散数学命题逻辑相关的两个简单项目:
  - Knight
  - Minesweeper

## Model Checking 模型检测算法

- Model 模型  
model是对每个原子命题的真值赋值。关于世界的知识体现在这些命题的真值中。model是提供世界信息的真值赋值。
- knowledge-based agents 基于知识的代理  
代理通过对内部的知识表征进行操作来推理得出结论。
- Knowledge Base(KB) 知识库  
KB是KBA所知的一系列命题。

- Entailment ( $\models$ ) 蕴含  
如果  $\alpha \models \beta$ , 那么在任意  $\alpha$  为真的世界中,  $\beta$  也为真。  
 $P$ : 今天是星期四,  $Q$ : 今天下雨,  $R$ : 去散步,  $KB: (P \wedge \neg Q) \rightarrow R, P, \neg Q$ , 结论:  $R$
- Inference 推理演算  
推理演算是从旧知识推导出新知识的过程。

- Model Checking 模型检测算法
  - to determine if  $KB \models \alpha$  (KB能否得出结论 $\alpha$ )
    - Enumerate all possible models 枚举所有可能的模型
    - if in every model where KB is true,  $\alpha$  is true, then  $KB \models \alpha$  如果所有的模型中KB为真值时 $\alpha$ 也为真值, 则KB蕴含 $\alpha$

$A \Rightarrow B$  的充要条件是  $A \rightarrow B = 1$  (即永真式)

```
class Symbol(Sentence): # 命题符号类
    def __init__(self, name):
        self.name = name
    def evaluate(self, model): # 计算一种model下的原子命题的值
        return bool(model[self.name])
    .....
```

```

class And(Symbol): # 逻辑乘
    def __init__(self, *conjunctions):
        self.conjunctions = list(conjunctions)
    def evaluate(self, model): # 计算一种model下的表达式的值
        return all(conjunct.evaluate(model) for conjunct in self.conjunctions)
    def symbols(self): # 返回复合命题中的所有原子命题
        return set.union(*[conjunct.symbols() for conjunct in self.conjunctions])
    .....

```

```

def model_check(knowledge, query): # evaluate knowledge ⊨ query ?
    def check_all(knowledge, query, symbols, model):
        '''递归实现模型检测 看不懂没关系 请把它当成黑盒'''
        if not symbols:
            if knowledge.evaluate(model):
                return query.evaluate(model)
            return True
        else:
            remaining = symbols.copy()
            p = remaining.pop()
            model_true = model.copy()
            model_true[p] = True
            model_false = model.copy()
            model_false[p] = False
            return (check_all(knowledge, query, remaining, model_true) and
                    check_all(knowledge, query, remaining, model_false))
    symbols = set.union(knowledge.symbols(), query.symbols())
    return check_all(knowledge, query, symbols, dict())

```

## 简单解释

- `def check_all(knowledge, query, symbols, model)`
- 假设你有原子命题 `p, q`
- 那么最终你会获得 `symbols = []` # 为空

```

model = {p:0, q:0}
model = {p:0, q:1}
model = {p:1, q:0}
model = {p:1, q:1}

```

- 然后将得到的所有model带入知识库和你需要的结论进行检测。如果当所有模型中满足KB=1时并且 query=1, 那么恭喜结论是正确的。

## Knight And Knave

- 每个角色要么是Knight, 要么是Knave。Knight永远说真话, 而Knave总是说谎

通过人物们的对话，推理出他们的身份

- Make a demonstration

## Mineswpeer 扫雷

A	B	C
D	1	E
F	G	H

`Or(A, B, C, D, E, F, G, H)`

```
Or(  
  And(A, Not(B), Not(C), Not(D), Not(E), Not(F), Not(G), Not(H)),  
  And(Not(A), B, Not(C), Not(D), Not(E), Not(F), Not(G), Not(H)),  
  .....  
)
```

`{A, B, C, D, E, F, G, H} = 1 !`

1	1	1
A	B	C
D	2	E

`{A, B, C, D, E} = 2, {A, B, C} = 1 => {D, E} = 1`

```
safes = [] # 安全位置
mines = [] # 地雷位置
move_made = [] # 已经点开的位置
knowledge = [] # 知识库
mark_safe(sentence, safe) # 标记安全位置
mark_mine(sentence, mine) # 标记地雷位置
inference(sentence1, sentence2) # 推理函数
```

何时我们知道安全位置和地雷位置? `{A, B, C} = 0, {A, B} = 2`

---

## Make a demonstration

---

## Get Code (Start Code)

---

- Start Code:
  - Knight: `https://cs50.harvard.edu/ai/2020/projects/1/knights/`
  - Minesweeper: `https://cs50.harvard.edu/ai/2020/projects/1/minesweeper`
- 
- My solution:
  - For git `git clone https://github.com/EIPsyCongroo/AI-with-Discrete-Mathematics.git`
  - Knight: `git checkout knight`
  - Minesweeper: `git checkout minesweeper`
  - 访问 `https://github.com/EIPsyCongroo/AI-with-Discrete-Mathematics.git` 选择分支下载 codezip