

Knowledge 知识表达与处理

1. 基于知识的Agent

1. 核心部件是Knowledge Base, KB是一个语句集合, 用知识表示语言表达, 可以通过TELL将新语句添加到知识库, ASK查询当前已经知道的内容。当ASK一个问题时, 答案应该遵循已有的知识。

2. Agent程序: 在初始化时给定一些背景知识, 调用时需要做三件事

1. TELL知识库感知到的内容
2. ASK应该执行的行动
3. TELL知识库agent所选择的行动

- **function** **KB-AGENT**(*percept*) **returns** an *action*
static: *KB*, a knowledge base
t, a counter, initially 0, indicating time

```
TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))  
action ← ASK(KB, MAKE-ACTION-QUERY(t))  
TELL(KB, MAKE-ACTION-SENTENCE(action, t))  
t ← t + 1  
return action
```

3. Logics are formal languages for representing information such that conclusions can be drawn

4. Syntax defines the sentences in the language

5. Semantics define the “meaning” of sentences;

6. 对数理逻辑中命题逻辑和一阶逻辑基础知识的回顾

7. 简单的推理过程, 判断命题逻辑蕴含的一个通用算法

- 枚举算法, 是co-NP-complete, 是可靠的也是完备的
-

```

function TT-ENTAILS?(KB,  $\alpha$ ) returns true or false
  inputs: KB, the knowledge base, a sentence in propositional logic
            $\alpha$ , the query, a sentence in propositional logic
  symbols  $\leftarrow$  a list of the proposition symbols in KB and  $\alpha$ 
  return TT-CHECK-ALL(KB,  $\alpha$ , symbols, [])

function TT-CHECK-ALL(KB,  $\alpha$ , symbols, model) returns true or false
  if EMPTY?(symbols) then
    if PL-TRUE?(KB, model) then return PL-TRUE?( $\alpha$ , model)
    else return true
  else do
    P  $\leftarrow$  FIRST(symbols); rest  $\leftarrow$  REST(symbols)
    return TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND(P, true, model)) and
           TT-CHECK-ALL(KB,  $\alpha$ , rest, EXTEND(P, false, model))

```

- 代码解释：第二个函数中，*symbols*是还没有赋值的语句，*model*是已经赋值的语句，先判断是否已经对所有变量赋值，如果已经全部赋值，则根据KB判断*model*是否为true，如果没有，则将*P*等于*symbols*中的第一个变量，然后赋值为true和false判断是否都为真

8. 判断蕴含的两种证明方法

- 模型检验：枚举所有模型，验证语句在所有模型中为真，枚举时可以使用最小冲突或爬山法（之前回溯法中的一些启发式优化）
- 定理证明，如果模型庞大而证明很短，则十分有效

- 搜索算法是完备的，有解一定可以找到；但如果推理规则不够充分，一些目标是不可达的，但如果使用推理规则：归结，当它与任何一个完备的搜索算法结合时，可以得到完备的推理算法。归结证明：为了证明 $KB \models \alpha$ 需要证明 $(KB \wedge \neg \alpha)$ 不可满足。

- 归结：TODO，课件上只是提及，但是是一个很有用的方法

- 限定子句：恰好只含一个正文字的析取式。

- Horn子句，至多只含一个正文字的析取式。使用Horn子句的推理可以使用前向链接和反向链接的算法，判定时间与之KB呈线性。

- 前向链接和后向链接中，弧线表示合取，没有弧线是析取。这里可以在书上手推一下，很简单。

2. First Order Logic FOL

- 命题逻辑中只有facts，一阶逻辑中有facts, objects, relations
- 具体知识和数理逻辑类似，不做整理