

逻辑引擎与推理

bifnudozhao@tencent.com

概要

- 背景
 - 基于知识库的代理
 - 例子
- 介词逻辑
 - 语法
 - 语义
 - 一个简单的知识库
 - 一个简单的推理过程
- 基于介词逻辑的代理

背景 - 基于知识库的代理

基于知识库的代理的核心在于**知识库 (knowledge base)**。一个知识库就是一组**句子**的集合，而每个句子，都必须符合知识库所规定的语法。当一个句子是直接给出而不是推导出来的话，则被称为**公理 (axiom)**。

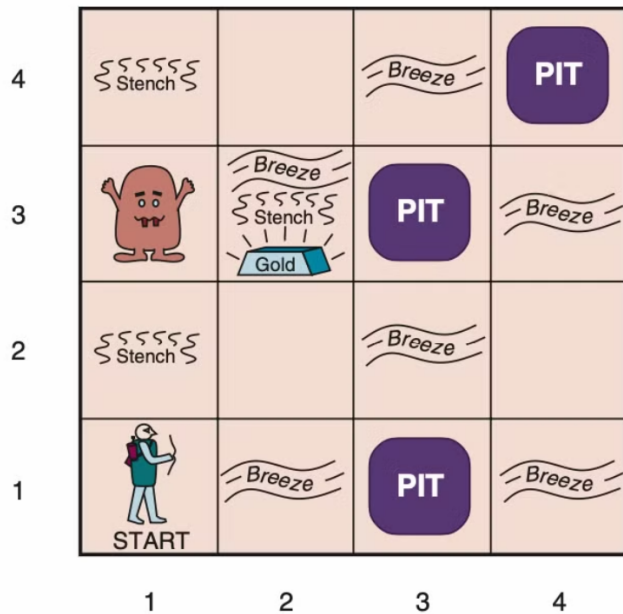
知识库里面有两个操作**TELL**以及**ASK**，两者都可以在知识库里面产生新句子，其中蕴含着**逻辑推导**的过程。

```
1  const KnowledgeBase;  
2  const timer;  
3  
4  const KnowledgeBaseAgent = (percept: Percept): Action => {  
5    TELL(KnowledgeBase, makePerceptSentence(percept, timer))  
6    const action = ASK(KnowledgeBase, makeActionQuery(timer))  
7    TELL(KnowledgeBase, makeActionSentence(action, timer))  
8    timer.tick()  
9    return action  
10 }
```

怪物世界

一个怪物世界是由一些格子组成，每个格子会有不同的属性

- **start**: 起点
- **pit**: 深渊
- **breeze**: 在深渊四周会有微风
- **monster**: 怪兽
- **stench**: 怪兽四周会有气味
- **gold**: 金块



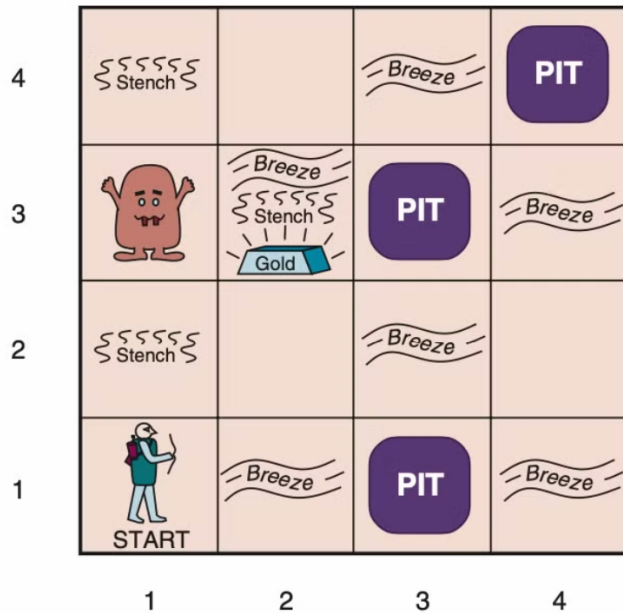
背景 - 例子

表现衡量

- 获得金块得 +1000 分
- 掉进深渊或者被怪物吃掉得 -1000 分
- 每走一步得 -1 分
- 使用了弓箭的话得 -10 分

效应器（动作）

- **Forward**: 前进一格
- **TurnLeft** | **TurnRight**: 左转, 或者右转 90 度
- **Grab**: 拿起金块
- **Shoot**: 用箭射击
- **Climb**: 爬出世界



介词逻辑 - 语法

介词逻辑的**语法**定义了合法的句子，**原子句子 (atomic sentences)** 由一个单独的 **介词符号 (prepositional symbol)** 组成。每个符号都能和一个真值所关联，而介词逻辑里，有两个具有固定意思的符号：**True** 以及 **False**。

复杂的句子可以由简单的句子组成，括号以及一些连接符，统一称为**逻辑连接符 (logical connectives)**

- \neg (not): 否定一个句子。
- \wedge (and): 一个句子如果用 \wedge 连接，则这个句子是一个 **合取**。
- \vee (or): 一个句子如果用 \vee 连接，则这个句子是一个 **析取**。
- \Rightarrow (implies): 推出，可以理解为 **规则** 或者 **if-then** 语句。
- \Leftrightarrow (if and only if): 句子 $A \Leftrightarrow B$ 是 **双向的**。

BNF (Backus-Naur Form) 语法

$$\text{Sentence} \rightarrow \text{AtomicSentence} | \text{ComplexSentence}$$
$$\text{AtomicSentence} \rightarrow \textit{True} | \textit{False} | P | Q | R | \dots$$
$$\text{ComplexSentence} \rightarrow (\text{Sentence})$$
$$| \neg \text{Sentence}$$
$$| \text{Sentence} \wedge \text{Sentence}$$
$$| \text{Sentence} \vee \text{Sentence}$$
$$| \text{Sentence} \implies \text{Sentence}$$
$$| \text{Sentence} \iff \text{Sentence}$$

Operator Precedence : $\neg, \wedge, \vee, \implies, \iff$

介词逻辑 - 语义

在介词逻辑里，**语义**实际上就是定义一个句子的真值，判断句子真值的方式如下

原子句子

- **True** 永远为真，**False** 永远为假
- 每个介词符号的真值，都必须明确指定

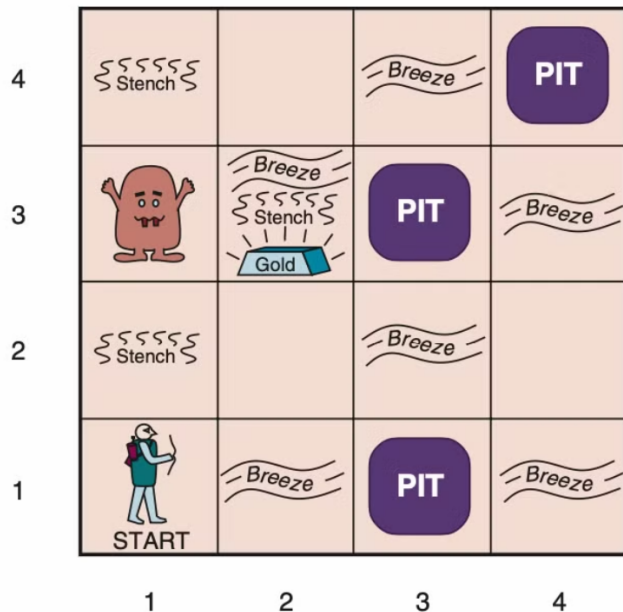
复合句子

- $\neg P$ 为真当且仅当 P 在 m 里为假。
- $P \wedge Q$ 为真当且仅当 P 和 Q 在 m 里为真。
- $P \vee Q$ 为真当且仅当 P 或者 Q 在 m 里为真。
- $P \implies Q$ 为真当且仅当在 m 里 P 为真并且 Q 为假。
- $P \iff Q$ 为真当且仅当 P 和 Q 在 m 里都为真或者都为假。

介词逻辑 - 知识库

对于怪物世界来说，我们可以针对每一个位置 $[x, y]$ 定义如下所示的符号。

- $P_{x,y}$ 为真如果在 $[x, y]$ 有一个深渊。
- $W_{x,y}$ 为真如果在 $[x, y]$ 有一只怪兽。
- $B_{x,y}$ 为真如果在 $[x, y]$ 有微风。
- $S_{x,y}$ 为真如果在 $[x, y]$ 有气味。
- $L_{x,y}$ 为真如果代理在 $[x, y]$ 。



一些例子

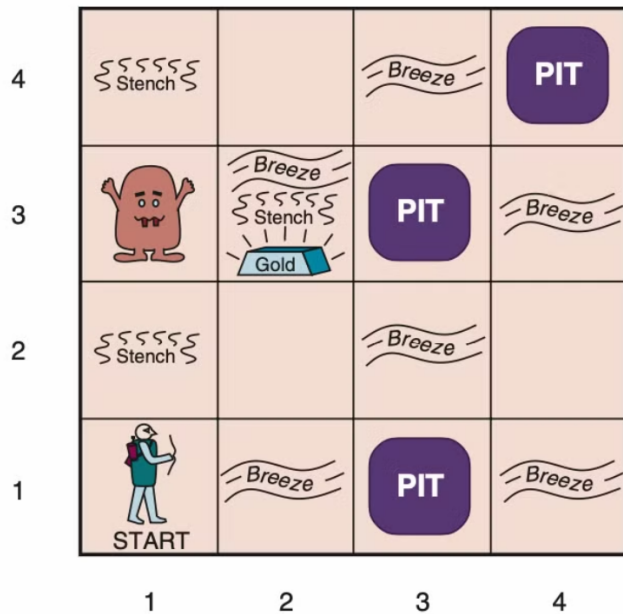
- 在 $[1, 1]$ 没有深渊。

$$R_1 : \neg P_{1,1}$$

- 一个格子有微风当且仅当它附近有深渊。

$$R_2 : B_{1,1} \iff (P_{1,2} \vee P_{2,1})$$

$$R_3 : B_{2,1} \iff (P_{1,1} \vee P_{2,2} \vee P_{3,1})$$



基于介词逻辑的代理 - 世界状态

一个知识库由两部分组成，一部分是世界的公理，另一部分是由这些公理推导出来的定理。对于怪物世界来说，我们对每个格子都知道一些基本事实：

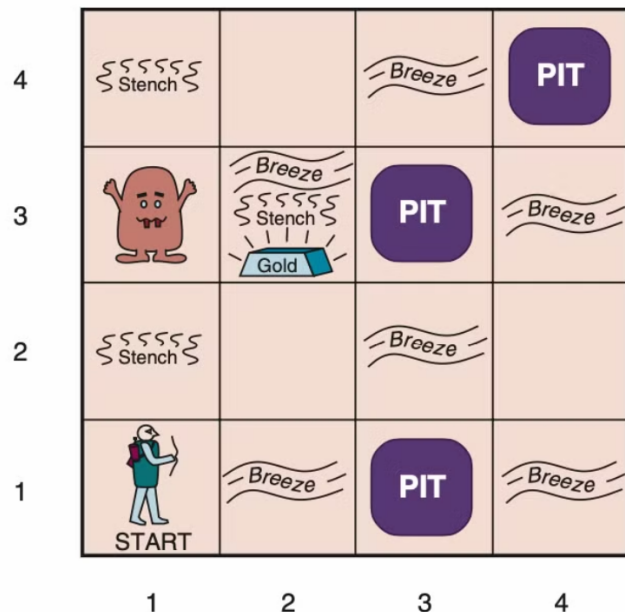
- 深渊四周的格子是有风的
- 怪物四周的格子是有气味的

用介词逻辑表示如下

$$B_{1,1} \iff (P_{1,2} \vee P_{2,1})$$

$$S_{1,1} \iff (W_{1,2} \vee W_{2,1})$$

⋮



基于介词逻辑的代理 - 世界状态

因为代理在世界探险的过程中，有很多状态会发生变化，而状态毫无疑问会跟时间挂钩，所以对于时间的描述，非常重要。我们可以在介词逻辑里引入时间的概念，只需要通过给符号增加时间标记即可

考虑在 $t=0$ 时刻，代理在左下角，那么有公理

$$L_{1,1}^0$$

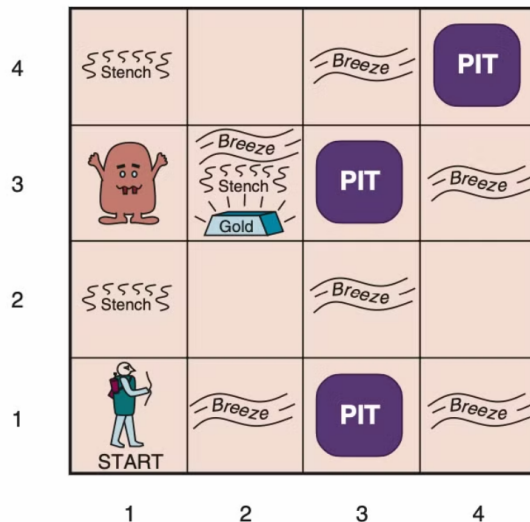
如果在 t 时刻，代理感受到微风以及气味，我们可以表示为

$$L_{x,y}^t \implies (Breeze^t \iff B_{x,y})$$

$$L_{x,y}^t \implies (Stench^t \iff S_{x,y})$$

我们甚至可以基于这样的语法，写出行动的语句

$$L_{1,1}^0 \wedge FacingEast^0 \wedge Forward^0 \implies (L_{2,1}^1 \wedge \neg L_{1,1}^1)$$



基于介词逻辑的代理 - 世界状态

我们可以基于带时间状态的句子，写出一些**效应公理 (effect axiom)**

基于介词逻辑的代理 - 流

我们把这类带有时间信息的公理称为流 (fluent)，对于一个流 F ，我们都可以写出一条公理来表达 F^{t+1} 的真值，下面是一个状态变化公理的形式

$$F^{t+1} \iff ActionCausesF^t \vee (F^t \wedge \neg ActionCausesNotF^t)$$

对于代理的位置，可以用下面的公理来了解状态变化公理的形式

而我们最关心的一个问题是，哪里可以走哪里不可以走，走哪里比较安全。由此我们可以定义一个 **OK 流** 来表述

$$OK_{x,y}^t \iff \neg P_{x,y} \wedge \neg(W_{x,y} \wedge WumpusAlive^t)$$

所以代理可以移动到任何满足 $ASK(KB, OK_{x,y}^t) = true$ 的格子。

Q & A
