

1.用一阶谓词公式表示以下语句：

(1) $\neg \text{InRoom}(\text{robot}, 2)$

(2) $\text{Likes}(\text{me}, \text{music}) \wedge \text{Likes}(\text{me}, \text{painting})$

(3) $\forall x (\text{ManufacturedBy}(\text{Jones}, x) \wedge \text{Malfunctioning}(x) \rightarrow (\text{RepairAtNight}(\text{Jones}) \vee \text{HandOverToEngineer}(\text{Jones}, x)))$

(4) $\forall x \exists y (\text{Employee}(x) \rightarrow \text{Manager}(y, x))$

(5) $\exists y \forall x (\text{Employee}(x) \rightarrow \text{Manager}(y, x))$

2.谓词逻辑表达中的个体可以是常量、变元或函数：

个体是常量的例子：John, Mary, Apple 等。常量是特定的、具体的个体。

个体是变元的例子：x, y, z 等。变元是代表任意个体的符号，可以用于表示一般情况。

个体是函数的例子：FatherOf(John), AgeOf(Mary)等。函数接受一个或多个参数，并返回一个特定的个体或值。

函数与谓词的区别在于：

谓词用于描述关系或性质，返回真或假的逻辑值。例如，Likes(x, music)表示个体 x 是否喜欢音乐。

函数用于计算或映射，返回一个具体的个体或值。例如，FatherOf(x)表示个体 x 的父亲是谁。

3.产生式系统的组成部分包括：

规则 (Rule)：定义了产生式系统的行为和推理规则。

工作存储 (Working Memory)：存储当前系统状态和已知事实的知识库。

目标 (Goal)：指定系统的目标或问题。

推理引擎 (Inference Engine)：执行推理过程，根据规则和工作存储进行匹配和推导。

执行器 (Executor)：根据推理结果执行相应的动作或操作。

产生式表示法与一阶谓词逻辑表示的区别在于：

产生式表示法更注重规则、模式匹配和操作执行，更适用于推理引擎和规则系统。

一阶谓词逻辑表示更注重逻辑关系、量化和断言，更适用于逻辑推理和知识表示。

4.用产生式表示：如果一个人发烧、呕吐和出现黄疸，那么得肝炎的可能性有 70%。

IF Fever(person) \wedge Vomiting(person) \wedge Jaundice(person) THEN HepatitisPossibility(person, 0.7)

5.将以下一则消息用框架表示："某年某月某日，某地发生 6.0 级地震，若以膨胀注水孕震模式为标准，则三项地震前兆中的波速比为 0.45，水氡含量为 0.43，地形改变为 0.60。"

EarthquakeEvent:

- Date: [某年某月某日]

- Location: [某地]

- Magnitude: 6.0

- Precursors:

- P-Wave Velocity Ratio: 0.45

- Radon Content: 0.43

- Terrain Change: 0.