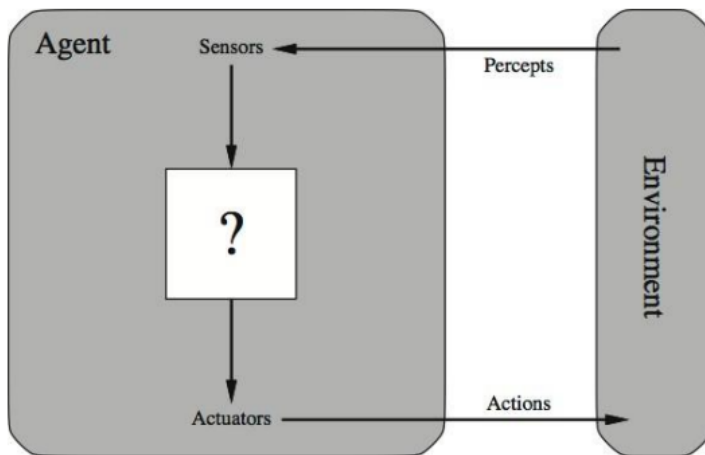


Lecture2 智能 Agent

1. Agent 和环境

Agent 的定义



一个 Agent 是任何可以被视为

- 通过传感器感知环境
- 通过驱动器作用于环境

一个 Agent 程序是下面三个类型的循环

- 感知 perceive
- 思考 think (中间的 ?)
- 行动 action

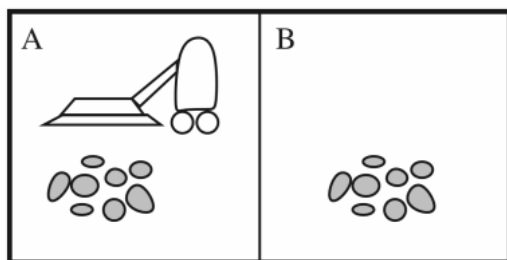
一个 Agent 的结构为

- **Agent = Architecture + Program**
- 协同且兼容

Agent 的示例

	人类 Agent	机器 Agent
传感器 Sensors	眼睛, 耳朵和其它器官	摄像机和红外测距仪
驱动器 Actuators	头、推、嘴、和其它器官	各种马达

真空吸尘器



Percept	Action
[A, clean]	Right
[A, dirty]	Suck
[B, clean]	Left
[B, dirty]	Suck

- 感知：位置和内容，例如环境脏的程度
- 思考：Agent Function，从感知到行动的映射
- 行动：左右移动，吸尘，等待

环境的定义与种类

- **完全可观测 / 部分可观测**：一个 agent 的传感器让它在每个时间点访问环境的完整状态
- **确定性 / 随机**：环境的下一个状态是否完全由当前状态和 agent 执行的操作决定，如果除了其他代理的行动环境是确定性的，那么环境是战略性的
- **情景 / 顺序**：agent 的经验被划分为原子性的“情节”，每一情节由行动者感知并执行单一动作组成，而每一情节中的动作选择只取决于情节本身
- **静态 / 动态**：当一个 agent 在决策时，环境是不变的，如果环境本身不随时间而变化，但代理的性能分数会变化，那么环境就是半动态的
- **离散 / 连续**：有限数量的不同的，明确定义的感知和行动
- **单一 agent / 多 agent**：一个 agent 在一个环境中操控的只有自己
- **知晓 / 不知晓**：agent 的设计者可能知道也可能不知道环境构成，如果环境是未知的，agent 将需要知道它是如何工作的，以便作出决定

Environment	Observable	Agents	Deterministic	Static	Discrete
8-puzzle	Fully	Single	Deterministic	Static	Discrete
Chess	Fully	Multi	Deterministic	(Semi)Static	Discrete
Poker	Partially	Multi	Stochastic	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Static	Discrete
Car	Partially	Multi	Stochastic	Dynamic	Continuous
Cleaner	partially	Single	Stochastic	Dynamic	Continuous

2. Agent 的合理性

合理的 Agent 的定义

For each possible percept sequence, a **rational agent** should select an action that is expected to **maximize its performance measure**, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

对于每一个可能的感知序列，合理的 agent 都应该选择一个行动，考虑到感知序列提供的证据和 agent 拥有的任何内在知识，它应该选择一个有望最大化其性能衡量的行动

合理性的评判

合理性是相对于性能来衡量的

判断合理性基于

- 定义成功标准的性能度量
 - 根据实际环境中希望得到的结果来设计性能度量
- agent 对环境的先验知识
- agent 可以执行的可能行动
- agent 到目前为止的感知序列

PEAS

当我们定义一个合理的 agent 时，我们将这些属性分组到任务环境的问题说明（PEAS）中

我们想要为这个任务环境设计的合理 agent 就是解决方案

PEAS 代表

- 性能 Performance
- 环境 Environment
- 驱动器 Actuators
- 传感器 Sensors

PEAS 示例 —— 自动驾驶汽车

- P: 安全、舒适、时间、合法驾驶
- E: 道路、其他车辆、行人、路标
- A: 方向盘，油门，刹车，信号，喇叭
- S: 摄像机，声纳，GPS，速度计，里程表，加速计，引擎传感器，键盘

PEAS 示例 —— 真空吸尘器

- P: 清洁、效率、行驶距离、电池寿命长、安全
- E: 房间，桌子，木地板，地毯，不同的障碍物
- A: 轮子，不同的刷子，吸尘器
- S: 摄像头，污垢检测传感器，悬崖传感器，碰撞传感器，红外墙传感器

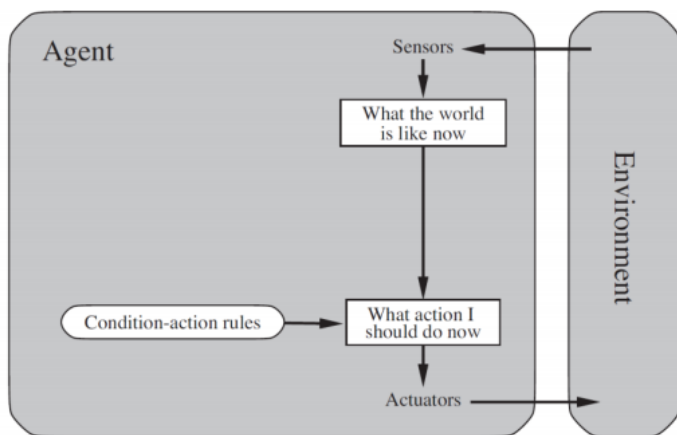
3. Agent 的种类

Agent 的种类介绍

四种基本类型 + 一种高级类型，以智能的程度为顺序

- 简单的反射 agent
- 基于模型的反射 agent
- 基于目标的 agent
- 基于效能的 agent
- 上述都可以推广到**基于学习 agent**（机器学习等），从而提高它们的性能并产生更好的行动

简单反射 agent

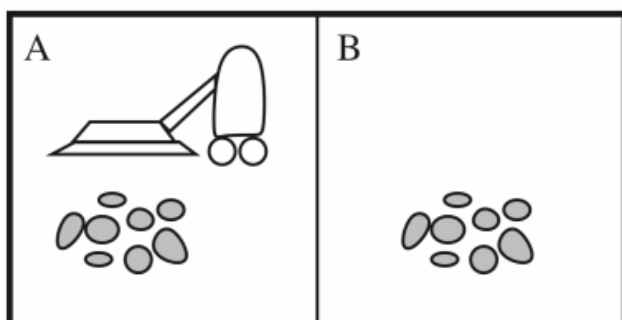


Action = LookUp(State)

简单反射代理**根据当前状态**选择动作，只忽略感知历史

简单但有限，只有在**完全可观测**的环境下才能起作用，也就是说正确的行动只基于当前的感知（而没有感知历史）

基于简单反射的真空吸尘器 agent

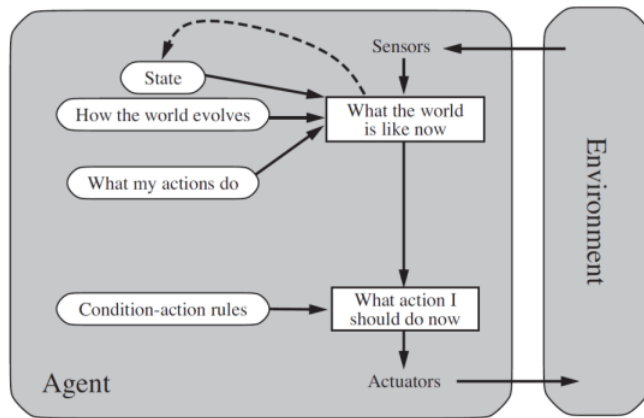


Percept	Action
[A, clean]	Right
[A, dirty]	Suck
[B, clean]	Left
[B, dirty]	Suck

```
1 Function Vacuum_agent(location, content) return action
2     if content = Dirty then return Suck
3     else if location = A then return Right
4     else return Left
```

- 如果吸尘器没有位置传感器会怎么样
 - 我们可以改写代码，让吸尘器随机向左 / 向右走

基于模型的反射 agent



State = Update(State); Action = LookUp(State)

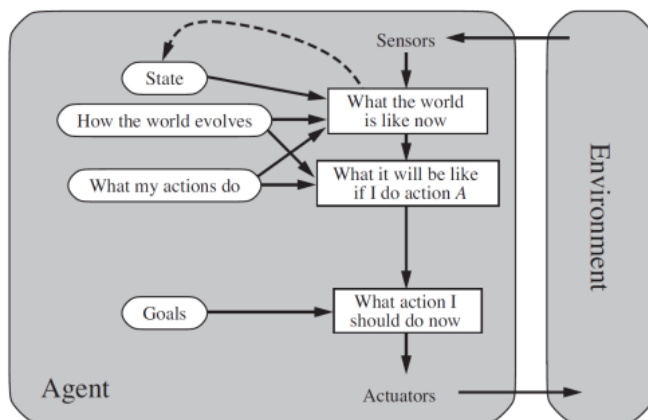
通过跟踪现在看不到的部分环境来处理部分可观测性

内部状态取决于感知历史（以最佳情况去猜测环境的情况），可能不精确

建立环境模型

- 环境是如何独立于 agent 而发展的
- agent 的行为如何影响环境

基于目标的 agent



State = Update(State); Action = Goal(LookUp(State))

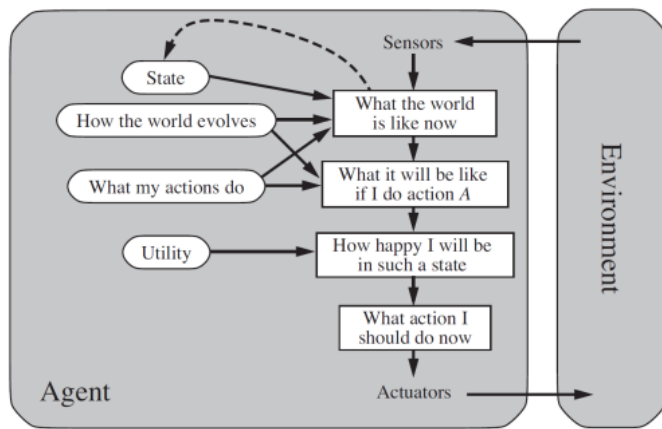
了解环境的当前状态是不够的，agent 需要一些目标信息

agent 程序将目标信息与环境模型相结合，选择实现该目标的行动

考虑未来的情况“如果我做了 A 这样的事情，会发生什么？”

灵活，因为支持决策的知识是明确表示的，并且可以修改

基于效能的 agent



State = Update(State); Action = Utility(Goal(LookUp(State)))

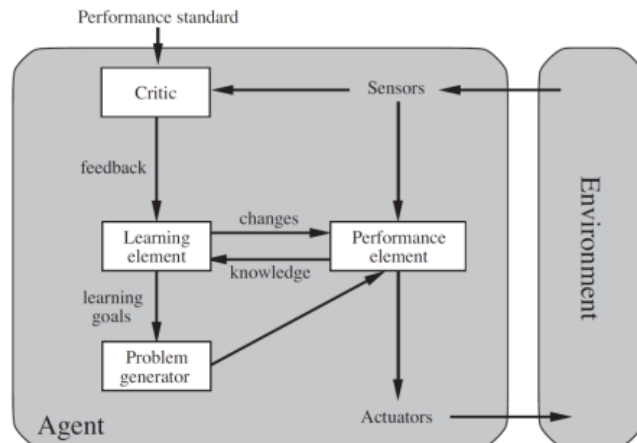
有时候，达到预期的目标是不够的，为了到达目的地，我们可能会寻找更快、更安全、更便宜的旅行方式

agent 的幸福也应该考虑在内，我们称之为效用 utility

效用函数是 agent 的性能度量

由于世界的不确定性，效用 agent 选择了使期望效用最大化的行为

基于学习的 agent



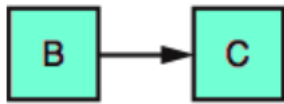
手工编写代理程序是非常乏味的，似乎需要一些更快捷的方法

基于学习的 agent 有四个组件

- **学习元素**：负责改进，提高 agent 的能力
- **性能元素**：负责选择外部行动，到目前为止，我们认为代理是这样的
- **评价元素**：agent 在一个固定的性能标准下做得有多好
- **问题生成器**：允许 agent 去探索，局部最优比不上全局最优，希望探索全局最优解

4. Agent 状态的表示

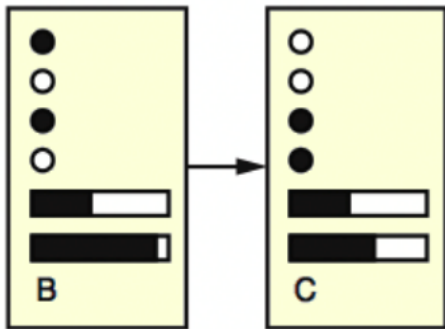
原子化表示 Atomic Representation



环境的每个状态都是一个没有内部结构的黑盒子

- 找到一条行驶路线，每个状态都是一个城市
- 人工智能算法：搜索，博弈，马尔可夫决策过程，隐藏马尔可夫模型等

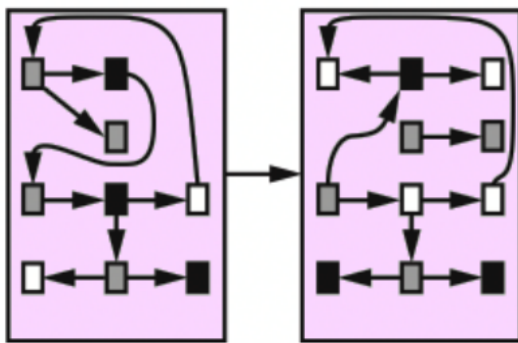
属性化表示 Factored Representation



每个状态都有一些属性值特性

- GPS 定位，油箱里的汽油量
- 人工智能算法：约束满足 CSP 和贝叶斯网络等

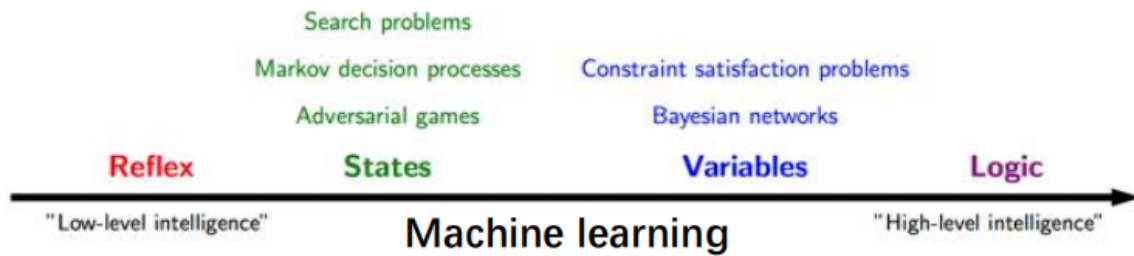
结构化表示 Structured Representation



对象状态之间的关系可以显式表示，属性之间具有关系

- 人工智能算法：一阶逻辑、基于知识学习、自然语言处理

5. 智能体 Intelligent Agent 回顾



- 智能体的概念是人工智能的核心
- 人工智能的目标是设计有用的、反应的、自主的、甚至是社会的积极的智能体
- 一个 agent 通过感知感知它的环境，并通过驱动器行动
- 性能衡量方法评估 agent 的行为
- 一个行动以最大限度地提高其预期性能的 agent 被称为合理的 agent
- PEAS：任务环境规范
 - 性能
 - 环境
 - 驱动器
 - 传感器
- Agent = Architecture + Program
- Agent 的四个类型
 - 简单的反射 agent
 - 基于模型的反射 agent
 - 基于目标的 agent
 - 基于效能的 agent
- 代理可以通过学习提高自己的表现
- 通过三种状态表示来提高表达能力
 - 原子的
 - 属性的
 - 结构的