

人工智能与机器学习

叶琦

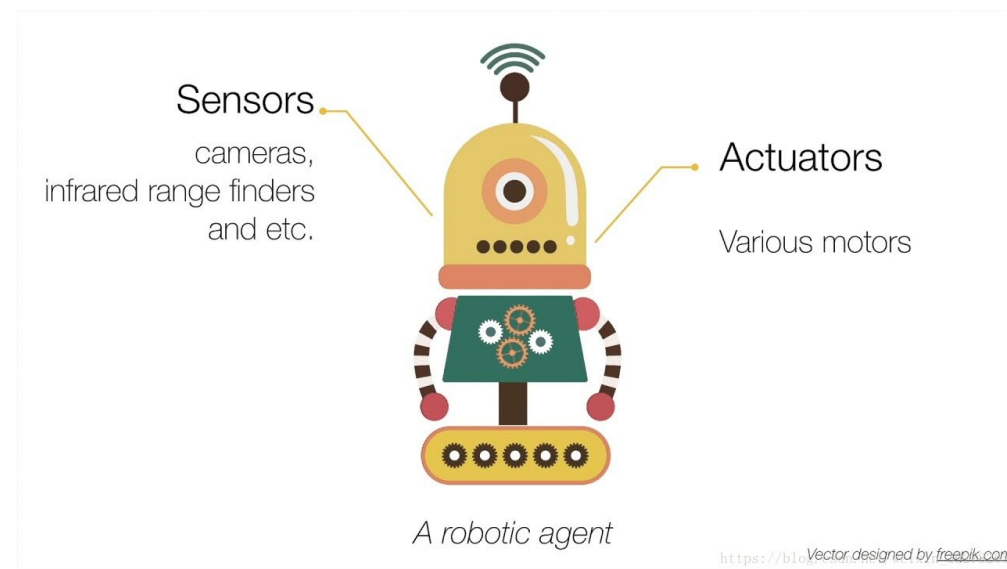
工业控制研究所

杭州 · 浙江大学 · 2022



第二章 智能AGENT（智能体）

- Agent基本定义（Agent的本质）
- 评价Agent行为（Agent是否完美）
- 任务环境（环境的多样性）
- Agent的结构（Agent分类）



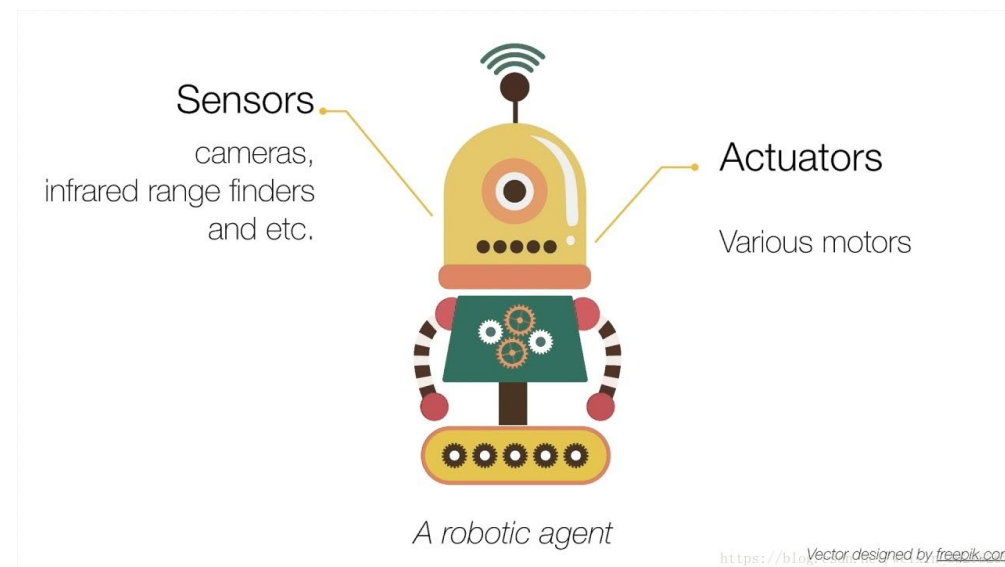
第二章 智能AGENT（智能体）

•Agent基本定义（Agent的本质）

•评价Agent行为（Agent是否完美）

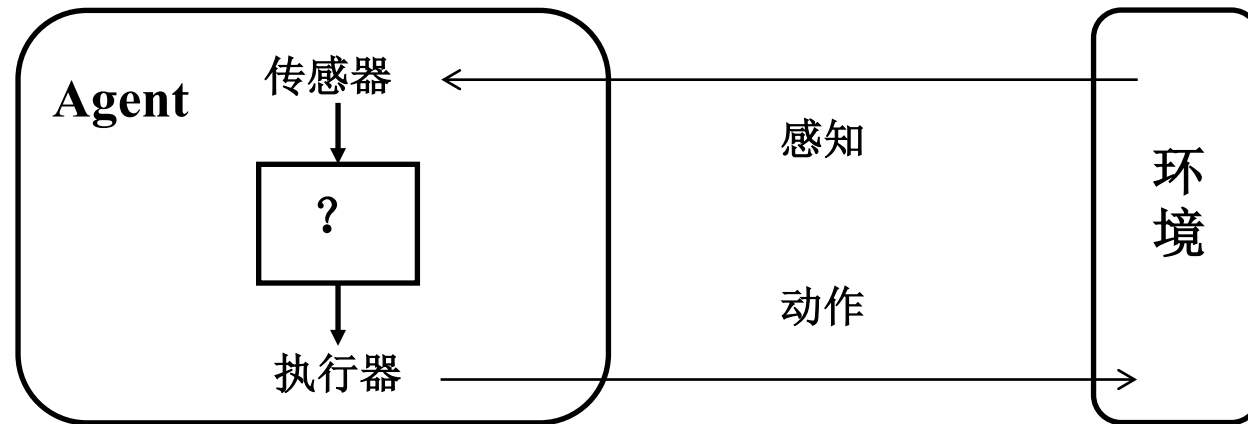
•任务环境（环境的多样性）

•Agent的结构（Agent分类）



智能体 (Agent) 基本定义

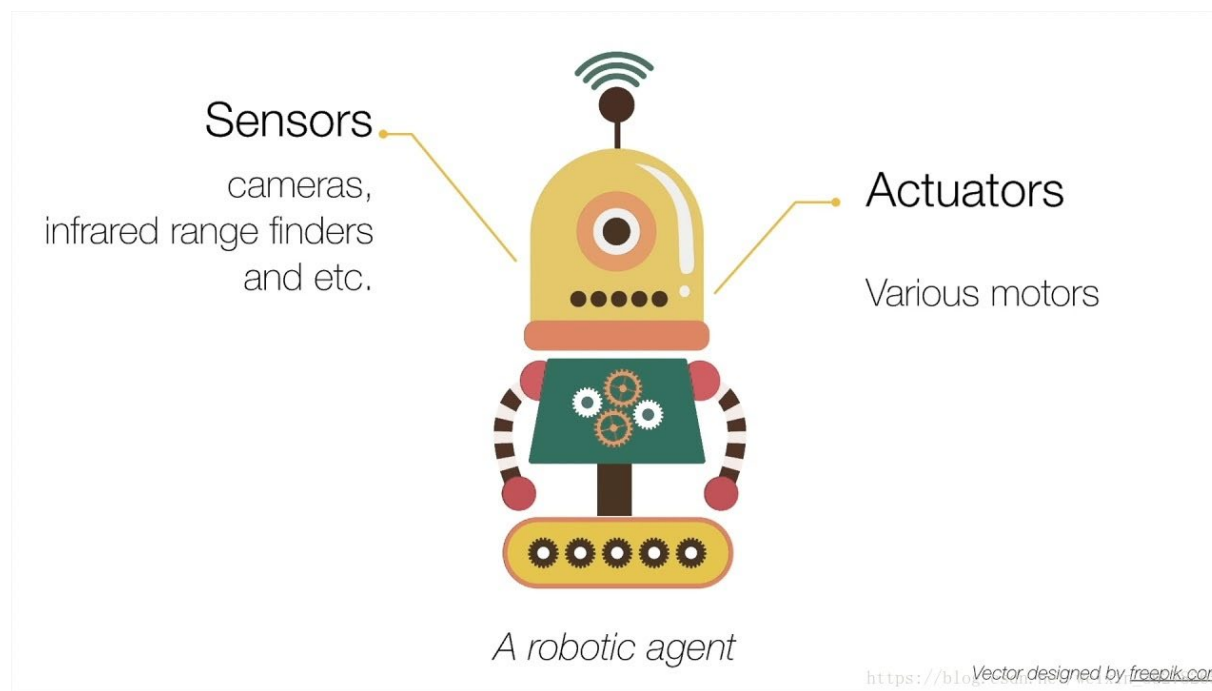
- 智能体 (Agent) : 能够感知和动作的实体
(任何独立的能够思想并可以同环境交互的实体都可以抽象为智能体)



智能体 (Agent) 基本定义

- 智能体 (Agent) : 能够**感知**和**动作**的实体
(任何独立的能够思想并可以同环境交互的实体都可以抽象为智能体)

- 人类智能体
- 机器人智能体
- 软件智能体
- VR/AR眼镜
-



- VR/AR眼镜

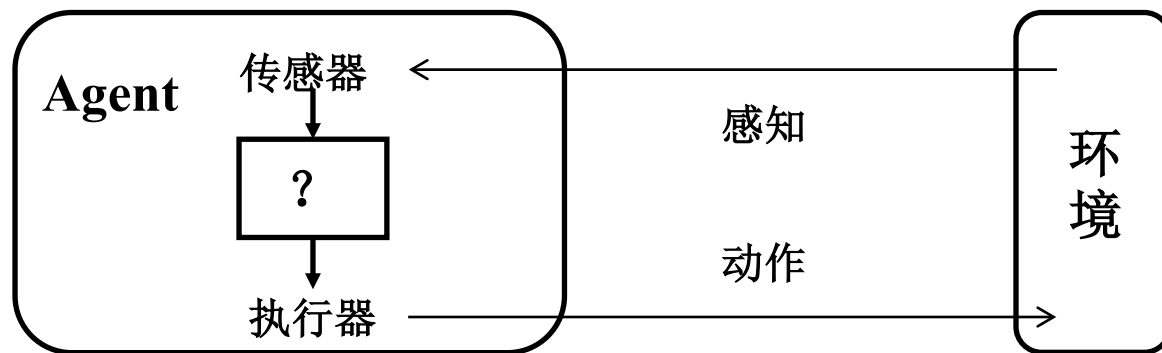


| Display | |
|---------------------|---|
| Optics | See-through holographic lenses (waveguides) |
| Resolution | 2k 3:2 light engines |
| Holographic density | >2.5k radiants (light points per radian) |
| Eye-based rendering | Display optimization for 3D eye position |
| | |
| Sensors | |
| Head tracking | 4 visible light cameras |
| Eye tracking | 2 IR cameras |
| Depth | 1-MP time-of-flight (ToF) depth sensor |
| IMU | Accelerometer, gyroscope, magnetometer |
| Camera | 8-MP stills, 1080p30 video |

| Audio and speech | |
|---------------------------|--|
| Microphone array | 5 channels |
| Speakers | Built-in spatial sound |
| | |
| Human understanding | |
| Hand tracking | Two-handed fully articulated model, direct manipulation |
| Eye tracking | Real-time tracking |
| Voice | Command and control on-device; natural language with internet connectivity |
| Windows Hello | Enterprise-grade security with iris recognition |
| | |
| Environment understanding | |
| 6DoF tracking | World-scale positional tracking |
| Spatial Mapping | Real-time environment mesh |
| Mixed Reality Capture | Mixed hologram and physical environment photos and videos |



智能体和环境



Agent 通过传感器感知环境，并通过执行器对所处环境产生影响。问号（？）表示从感知到动作的映射函数

简单说，一个智能体就是从**感知序列**到**动作**的一个**函数**： $f : P^* \rightarrow A$

感知信息：表示任何时刻Agent的感知输入。

感知序列：Agent收到的所有输入数据的完整历史。

Agent在任何时刻的行动选择，取决于到该时刻为止的整个感知序列。

Agent函数：将任意给定感知序列映射到Agent的动作。可以描述Agent的行为。



智能体函数 Agent function

可以采用制表（列表）的方式表示Agent函数。

该表可能很大，甚至无限大。除非要考虑的感知序列长度有限。

原则上，可以通过实验找出感知序列、并记录Agent的行动，来构造该表。

| Percept sequence (感知序列) | Action (行动) |
|-------------------------------|----------------|
| | |
| | |
| | |



智能体程序 Agent program

```
function Reflex-Vacuum-Agent([location, status])  
  returns an action  
    if status = Dirty then return Suck  
    else if location = A then return Right  
    else if location = B then return Left
```

Agent函数: 从外部反映了Agent的特性

Agent程序: 从Agent内部来看, Agent函数是通过Agent程序实现的

Agent函数: 是一个抽象的数学表示

Agent程序: 是Agent函数在某种物理体系上的具体实现



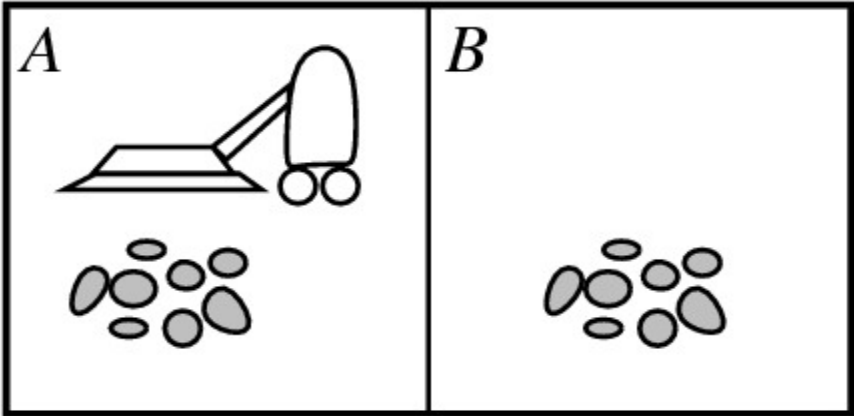
真空吸尘器世界

世界：方格A和B

感知：可以感知所处的方格（位置）、是否有灰尘（本地的状态） 如：[A, Dirty]

动作：向左移动、向右移动、吸取灰尘、什么也不做
Left, Right, Suck, No-Op

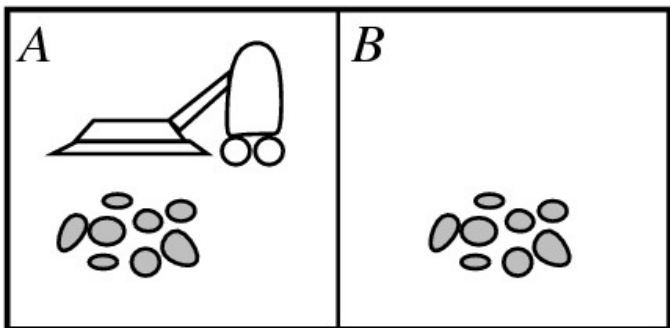
Agent函数（感知-动作映射函数）：如果当前地点有灰尘，则吸取。否则移动到另一地点。



| 感知序列 | 行动 |
|------------------------|-------|
| [A, Clean] | Right |
| [A, Dirty] | Suck |
| [B, Clean] | Left |
| [B, Dirty] | Suck |
| [A, Clean], [A, Clean] | Right |
| [A, Clean], [A, Dirty] | Suck |
| . | . |
| . | . |
| . | . |

真空吸尘器世界的一个简单Agent函数的部分列表





- 修改表格的右边就可以定义不同的真空吸尘器世界Agent。

问题:

- 怎样填表才是最好的算法?
- 如何判断Agent是 好的、坏的、智能的、愚蠢的?

Agent 函数 (部分列表)

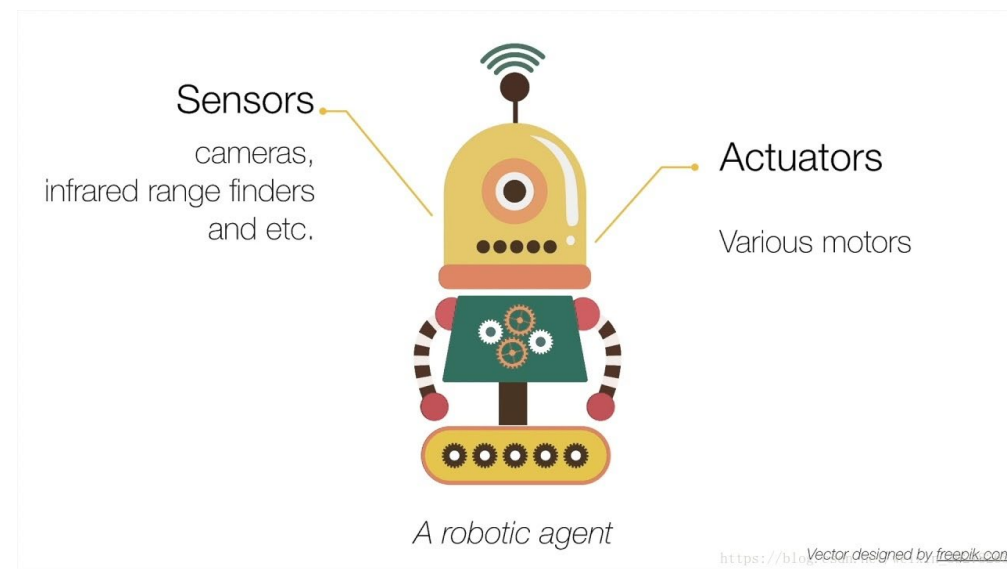
| 感知序列 | 行动 |
|------------------------|-------|
| [A, Clean] | Right |
| [A, Dirty] | Suck |
| [B, Clean] | Left |
| [B, Dirty] | Suck |
| [A, Clean], [A, Clean] | Right |
| [A, Clean], [A, Dirty] | Suck |
| ... | ... |

Agent程序

```
function Reflex-Vacuum-Agent([location, status])
returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

第二章 智能AGENT（智能体）

- Agent基本定义（Agent的本质）
- 评价Agent行为（Agent是否完美）
- 任务环境（环境的多样性）
- Agent的结构（Agent分类）



理性智能体 (Rational Agent)

| 感知序列 | 行动 |
|------------------------|-------|
| [A, Clean] | Right |
| [A, Dirty] | Suck |
| [B, Clean] | Left |
| [B, Dirty] | Suck |
| [A, Clean], [A, Clean] | Right |
| [A, Clean], [A, Dirty] | Suck |
| ... | ... |

- 理性Agent是**做事正确**的智能体。
 - 即Agent函数表格的右边都填写正确。
- 什么是“正确的事”？
 - 需要某种方法度量Agent成功与否，连同对环境、传感器、执行器的描述，为Agent 面临的任务提供一个完整的规范说明。

- 修改表格的右边就可以定义不同的真空吸尘器世界Agent。

问题：

- 怎样填表才是最好的算法？
- 如何判断Agent是 好的、坏的、智能的、愚蠢的？



性能度量：Agent成功程度的标准

性能度量：通常由理性Agent的**设计者**给出，根据实际在所处的环境中**希望得到的结果**来设计度量，而不是根据Agent**表现出的行为**。

当把Agent置于一个环境中后，它将针对收到的感知信息产生动作序列。

该动作序列引起环境历经一个状态序列。

如果该**环境状态**序列是想要的，则Agent的性能良好。

- 需要客观的性能度量（通常由理性Agent的设计者给出）
- 没固定的适合所有智能体的度量标准
- 理性Agent应该选择期望能使其性能度量最大化的行动



理性 (Rationality) 的判断

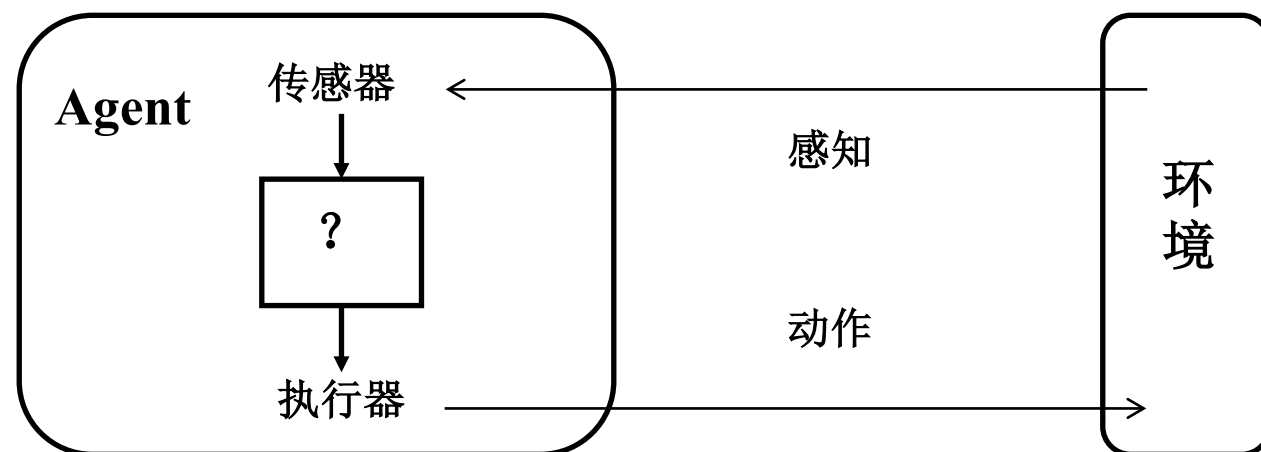
理性的判断

- 定义成功标准的性能度量

- Agent对环境的先验知识

- Agent可以执行的动作

- Agent的感知序列



理性 (Rationality) 的判断



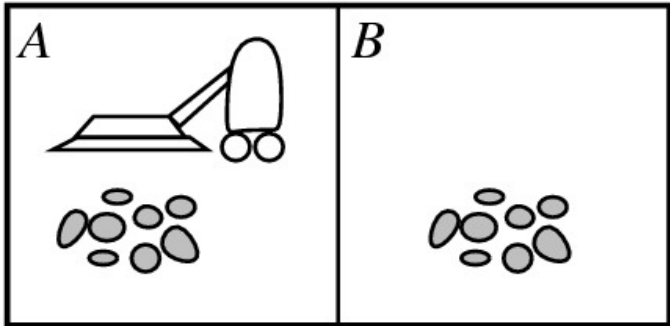
理性Agent

对于每一个可能的感知序列，根据已知的感知序列和内建的先验知识，理性Agent选择能使性能指标的期望值最大化的动作。

(这是一个重要、且比较严谨的定义)



自动真空吸尘器智能体是理性的吗？



| 感知序列 | 动作 |
|-------------------|--------------|
| <i>[A, Clean]</i> | <i>Right</i> |
| <i>[A, Dirty]</i> | <i>Suck</i> |
| <i>[B, Clean]</i> | <i>Left</i> |
| <i>[B, Dirty]</i> | <i>Suck</i> |

- 理性的判断包括：
- (1) 定义成功标准的性能度量
 - (2) Agent对环境的先验知识
 - (3) Agent可以执行的动作
 - (4) Agent的感知序列

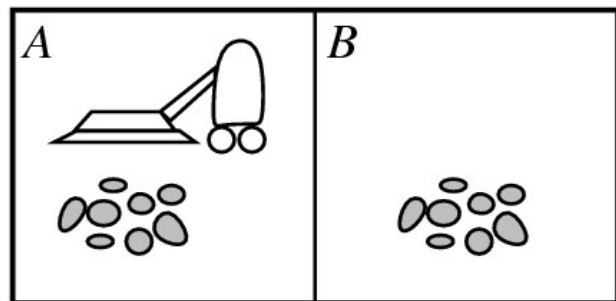
很难说！

需说明性能度量、环境知识、感知与行动

理性会随智能体所处环境变化而变化！



自动真空吸尘器智能体的理性判断



理性的判断包括：

- (1) 定义成功标准的性能度量
- (2) Agent对环境的先验知识
- (3) Agent可以执行的动作
- (4) Agent的感知序列

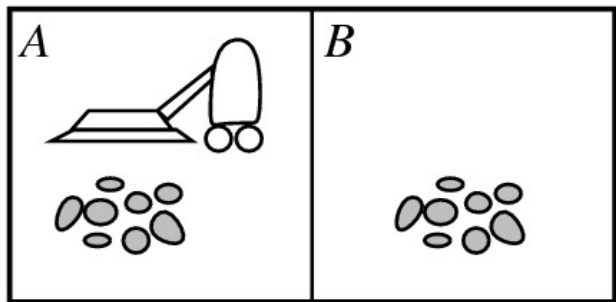
可以假设

- (1) 性能度量是在每个单位时间内对每块清洁的区域奖励1分。时间跨度为相当于1000个单位时间。
- (2) 环境的地理是已知的，这是一种先验知识。但灰尘的分布和Agent的初始位置未知。干净的方格必须保持清洁，而吸尘的动作清洁当前的方格。Left（左）和Right（右）的动作使Agent向左、向右移动。除非使Agent移出环境，此时Agent保持原位。
- (3) 可能的动作包括：Left（左）、Right（右）、Suck（吸尘）、NoOp（无动作）。
- (4) Agent可正确感知自己的位置、所在地是否有灰尘。

在这些条件下，该Agent是理性的。



自动真空吸尘器智能体的理性判断



理性的判断包括：

- (1) 定义成功标准的性能度量
- (2) Agent对环境的先验知识
- (3) Agent可以执行的动作
- (4) Agent的感知序列

同样的Agent在不同的环境下会变成非理性

考察各种真空吸尘器Agent函数的理性（习题2.2）：

- 在时间T内每打扫一格加一分？
- 在一个时间段内，每打扫一格加一分，每移动一格减一分？
- 多于k格不干净就进行惩罚？
-



理性智能体 ≠ 全知者

理性 vs. 全知

- 一个全知的Agent知道它的动作产生的实际效果，并且做出相应的动作。但一个全知者在现实中是不可能的。
- 对理性的定义并不要求全知。因为理性的选择只取决于到当时为止的感知序列。要确保的是：Agent不能漫不经心进行非常愚蠢的行动。
- 理性Agent应该经常进行观察（**观察也是一种动作Action**），因为观察有助于最大化期望性能。
- 为了修改未来的感知信息而采取动作，也称为信息收集，是理性的重要部分。例如：真空吸尘器Agent在初始未知的环境中必须进行探索。

理性 vs. 完美

- 理性是使**期望的**性能最大化。而完美是使**实际的**性能最大化。
- 完美对一个Agent而言是不合理的要求。



理性智能体要会“学习”

对理性的定义还要求Agent从所感知的东西中**学习**。

Agent最初的设定可能反映了环境的一些先验知识。但随着Agent获得经验，这些知识会改变、或增加。

有一些极端的情况下环境完全当成先验知识。在这样的情况下，Agent不需要感知和学习，只要正确地动作就可以。当然，这样的Agent非常脆弱。（参见书中的两个例子）

成功的Agent体把函数计算分成三个阶段：

- (1) **设计**阶段，设计者会完成一些计算。
 - (2) **决策**阶段，思考下一步动作的时候，Agent会做更多计算。
 - (3) **学习**阶段，当从经验中学习的时候，Agent要进行更多的计算以决定如何修正自己的行为。
- (注意设计者和Agent的分工与合作)



理性智能体要有“自主性”

- 如果Agent依赖于设计者的先验知识，而不是它自身的**感知信息**，则该Agent缺乏自主性。
- **理性Agent应该是自主的**，它应该尽可能地**学习**，以弥补不全面、不正确的先验知识。
- 为Agent提供一些初始知识和学习能力是合理的。当得到关于环境的充分经验后，理性Agent的行为才会**有效地独立于**它的先验知识。
- **与学习相结合**使得设计在很多环境下都能成功的理性Agent成为可能（学习是Agent自适应性的基础）。



对“理性”的思考



理性 \neq 先知

先验知识, 感知序列



理性 \neq 成功

动作集合

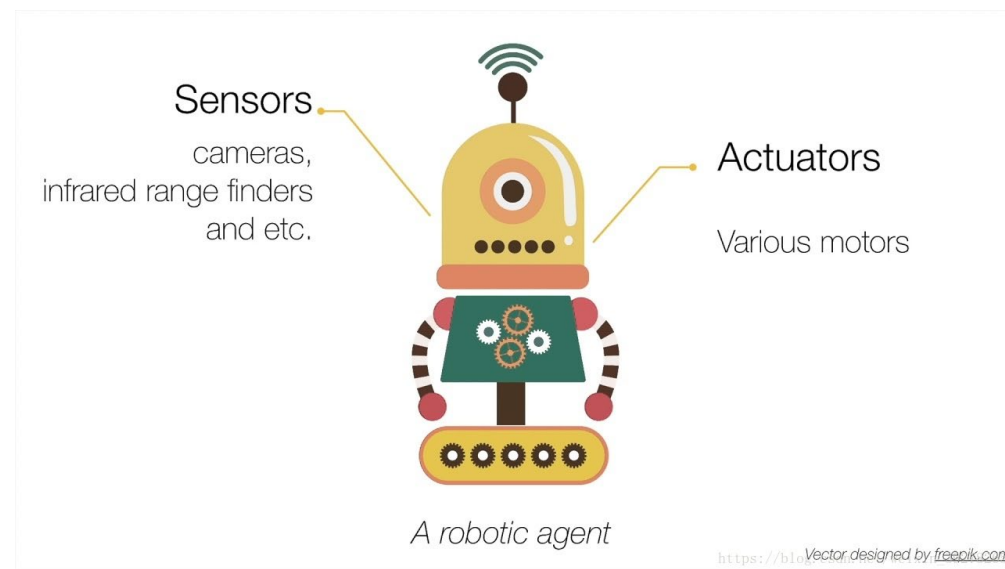


理性 \Rightarrow 探索, 学习



第二章 智能AGENT（智能体）

- Agent基本定义（Agent的本质）
- 评价Agent行为（Agent是否完美）
- **任务环境（环境的多样性）**
- Agent的结构（Agent分类）



任务环境

设计理性智能体的第一步：尽可能完整地详细说明必须指定**任务环境（PEAS）**



例如，考虑设计一个**自动驾驶出租车的任务**：

Performance性能

安全，到达目的地，利润，合法，舒适， ...

Enviroment 环境

街道/高速，汽车，行人/自行车，天气， ...

Actuators 执行器

方向盘，油门，刹车，喇叭，对讲机/显示器， ...

Sensors 传感器

摄像头，声纳，速度表，发动机传感器，键盘，GPS， ...



设计一个AR眼镜的任务

Performance性能

Enviroment 环境

Actuators 执行器

Sensors 传感器



Photo from <https://docs.microsoft.com/en-us/windows/mixed-reality>



任务环境的性质

- **完全可观察与部分可观察**

- 完全可观察：智能体能获取环境的完整状态，智能体不需要内部维护来记录世界的状况
- 部分可观察：噪声、不精确的传感器、丢失了部分状态数据.....

- **确定性与随机**

- 确定性：环境的下一个状态完全取决于当前状态和智能体的行动
- 如果环境是确定的，除非有其它智能体活动的影响，那么我们称环境是确定的。
- 随机
- 复杂的环境、难以实现预告的事件（出租车和灰尘）

- **静态与动态**

- 环境在智能体思考的时候是否会变化
- 出租车驾驶是动态的；纵横字谜游戏是静态的
- **半动态**：环境本身不随时间的流逝而变化，但智能体的性能评价随时间变化；例如计时棋赛。



任务环境的性质

- **离散与连续：**
 - 环境的状态，时间的处理方式，智能体的感知信息和行动
 - 国际象棋、出租车
- **片段式与延续式：**
 - 行动的选择是否只取决于当前片段，当前行动是否影响到未来所有的决策
 - 国际象棋、出租车、残次品检测
- **单智能体与多智能体：**
 - 哪些智能体必须被视为智能体？
 - Agent B的行为是否寻求让依赖于Agent A 的行为性能度量最大化
 - 竞争性的多Agent
 - 合作的多Agent



任务环境的性质

- **完全可观察与部分可观察**
 - 完全可观察：智能体能获取环境的完整状态，智能体不需要内部维护来记录世界的状况
- **确定性与随机**
 - 确定性：环境的下一个状态完全取决于当前状态和智能体的行动
 - 如果环境是确定的，除非有其它智能体活动的影响，那么我们称环境是确定的。
- **片段式与延续式**：行动的选择是否取决于当前片段
- **静态与动态**
 - 环境在智能体思考的时候是否会变化
 - 出租车驾驶是动态的；纵横字谜游戏是静态的
 - **半动态**：环境本身不随时间的流逝而变化，但智能体的性能评价随时间变化；例如计时棋赛。
- **离散与连续**：环境的状态，时间的处理方式，智能体的感知信息和行动
- **单智能体与多智能体**：哪些智能体必须被视为智能体？





任务环境的例子

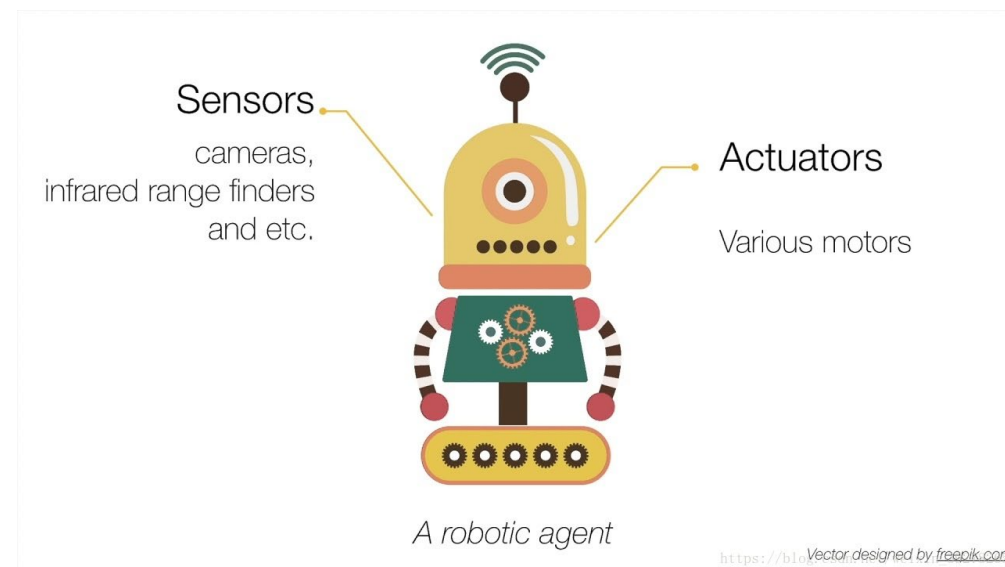
| Task Environment | Observable | Agents | Deterministic | Episodic | Static | Discrete |
|----------------------------|------------|--------|---------------|------------|---------|------------|
| Crossword puzzle | Fully | Single | Deterministic | Sequential | Static | Discrete |
| Chess with a clock | Fully | Multi | Deterministic | Sequential | Semi | Discrete |
| Poker | Partially | Multi | Stochastic | Sequential | Static | Discrete |
| Backgammon | Fully | Multi | Stochastic | Sequential | Static | Discrete |
| Taxi driving | Partially | Multi | Stochastic. | Sequential | Dynamic | Continuous |
| Medical diagnosis | Partially | Single | Stochastic | Sequential | Dynamic | Continuous |
| Image analysis | Fully | Single | Deterministic | Episodic | Semi | Continuous |
| Part-picking robot | Partially | Single | Stochastic | Episodic | Dynamic | Continuous |
| Refinery controller | Partially | Single | Stochastic | Sequential | Dynamic | Continuous |
| Interactive. English tutor | Partially | Multi | Stochastic | Sequential | Dynamic | Discrete |

任务环境的类型很大程度上决定了智能体的设计
真实的世界是部分可观，随机的，延续式的，动态的，连续的，多智能体的。



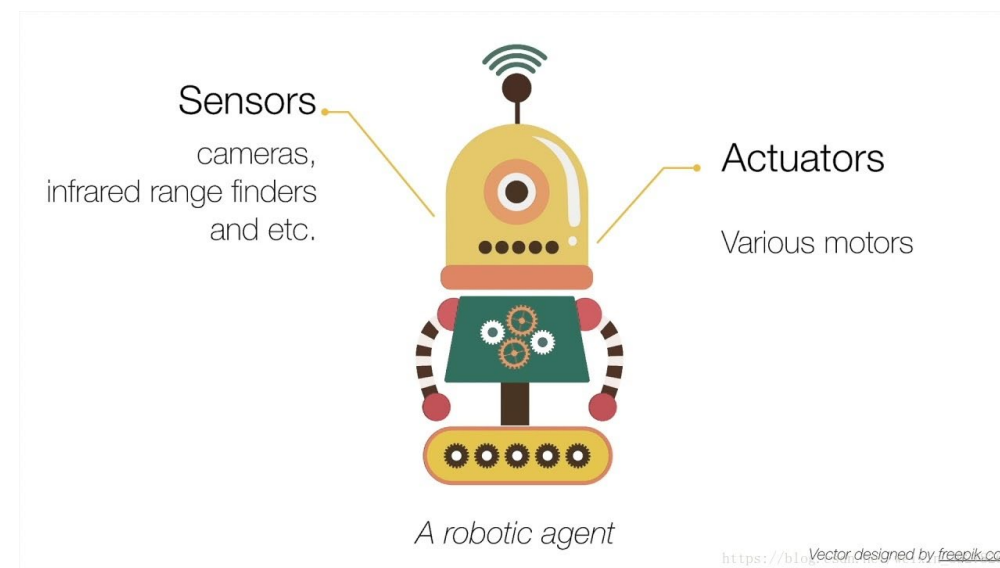
第二章 智能AGENT（智能体）

- Agent基本定义（Agent的本质）
- 评价Agent行为（Agent是否完美）
- 任务环境（环境的多样性）
- **Agent的结构（Agent分类）**





- AI的任务是设计智能体**程序**，实现把感知信息映射到行动的智能体**函数** f
- **体系结构**为**程序**提供：
 - 来自**传感器**的感知信息
 - 运行程序
 - 把程序产生的行动送到**执行器**
- 所选择的**程序**必须适合**体系结构**





- AI的任务是设计智能体**程序**，实现把感知信息映射到行动的智能体**函数** f

```
function Reflex-Vacuum-Agent([location, status])  
  returns an action  
    if status = Dirty then return Suck  
    else if location = A then return Right  
    else if location = B then return Left
```

- 必须包含针对每个可能感知序列的适当行动的函数表
- 在现实世界中是否可能?



- 必须包含针对每个可能感知序列的适当行动的函数表
- *在现实世界中是否可能?*

- **This is AI for and This is the challenge of AI!**

- 在可能的范围内，用少量代码代替庞大的表来生成理性行为。



四种基本的智能体(结构) 类型

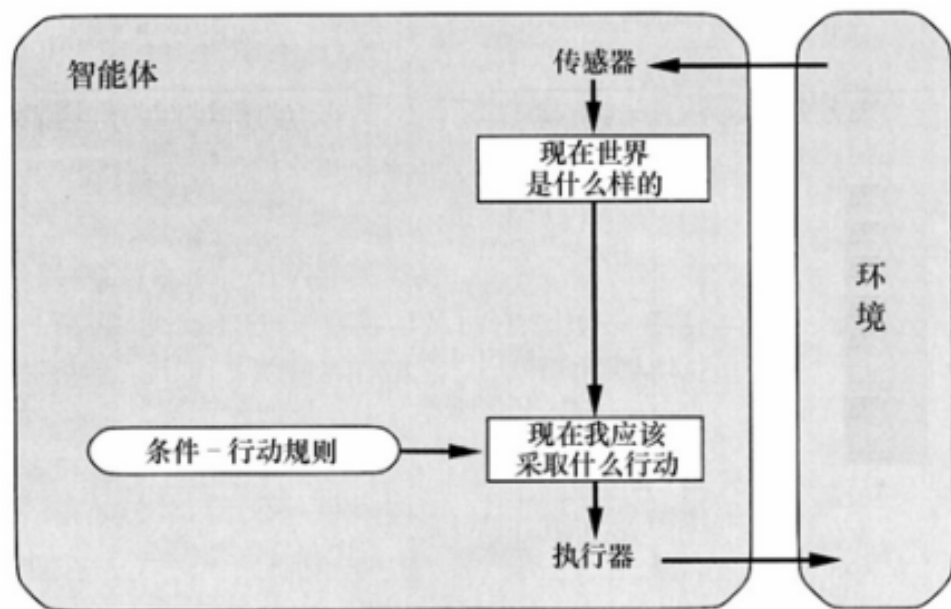


- 每种Agent程序都以**特定的方式**结合了**特定的成分**来生成**行动**
- 所有这些智能体都可以转变为**学习智能体**，能够提高性能以便生成更好的行动
- Agent程序的各组件是如何工作的



简单反射智能体

基于当前感知选择行动，忽略感知历史



- IF.....THEN..... (条件-行为规则)

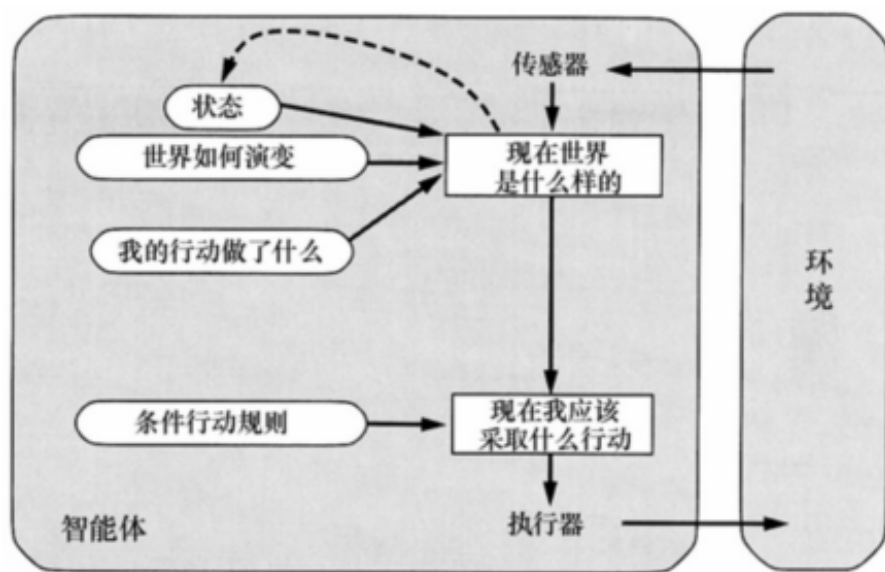
```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  static: rules, a set of condition-action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```

- 结构简单，智能有限
- 只有在可以仅根据当前感知信息来完成当前决策的情况下才能工作，即环境完全可观。
- 失败的情形？



基于模型的反射智能体



■ 世界的模型=

- 世界如何独立于智能体发展的信息(世界如何演变、环境模型)
- 智能体的行动如何影响世界的信息(我的行动做了什么、智能体自身模型)
- 其他Agent刹车、超车的判断
- 当前Agent转弯或者加速的影响
- 使用内部模型记录世界的当前状态，按反射型智能体方式选择行动

function REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action

static: *state*, a description of the current world state

rules, a set of condition-action rules

action, the most recent action, initially none

state ← UPDATE-STATE(*state*, *action*, *percept*)

rule ← RULE-MATCH(*state*, *rules*)

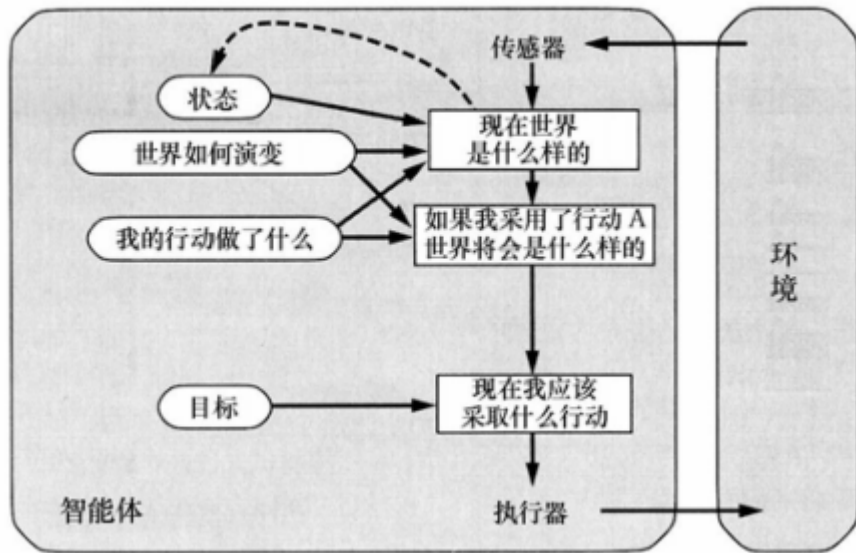
action ← RULE-ACTION[*rule*]

return *action*



基于目标的智能体

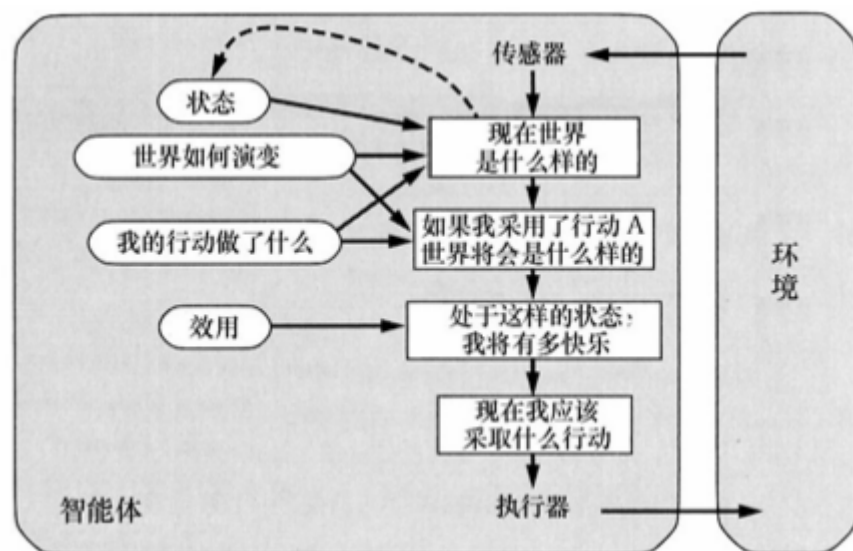
智能体追踪记录世界的状态、要达到的目标，并选择导致达成目标的行动



- 智能体不仅需要当前状态的描述，还需要**目标信息**来描述要达到的状况
- 与反射Agent相比，考虑**未来**，而不是直接把感知映射到行动
 - 智能体把目标信息和可能行动的结果信息结合起来，以选择达到目标的行动（决策）
 - 搜索和规划能帮助智能体找到达到目标的行动序列
 - 既追踪记录世界的状态，又记录要达到的目标集，并选择能达到目标的行动
- **效率降低，灵活度增加，功能增强（为什么？）**



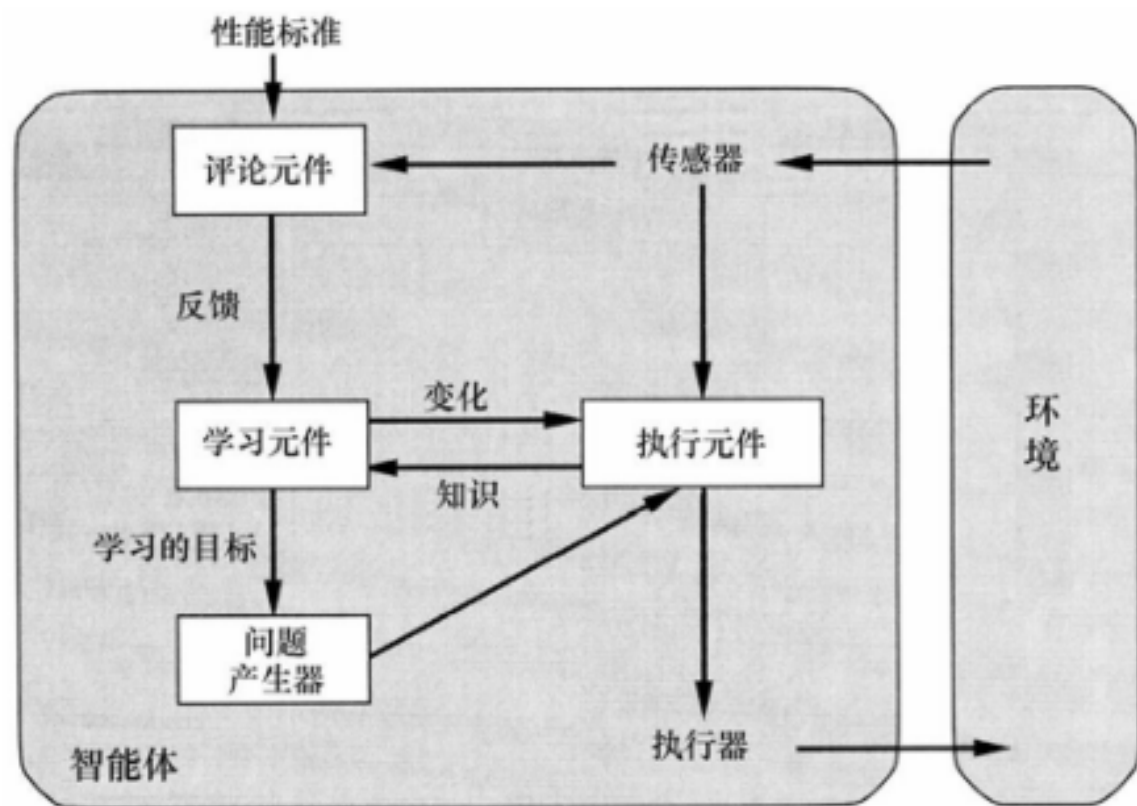
基于效用 (utility) 的智能体



- 单靠目标不足以获得高品质的行为
- 更普适的度量：效用函数
- 效用函数把状态（或状态序列）映射到实数，以描述智能体与状态相关的高兴程度
- 效用函数可以辅助进行决策：
 - 有多个相互冲突的目标可达到时实现折中
 - 多个目标都不能有把握达到时选择一个目标
- 最佳期望效用是通过计算所有可能结果状态的加权平均得到，其权值由结果的概率确定



学习智能体(Learning Agent)



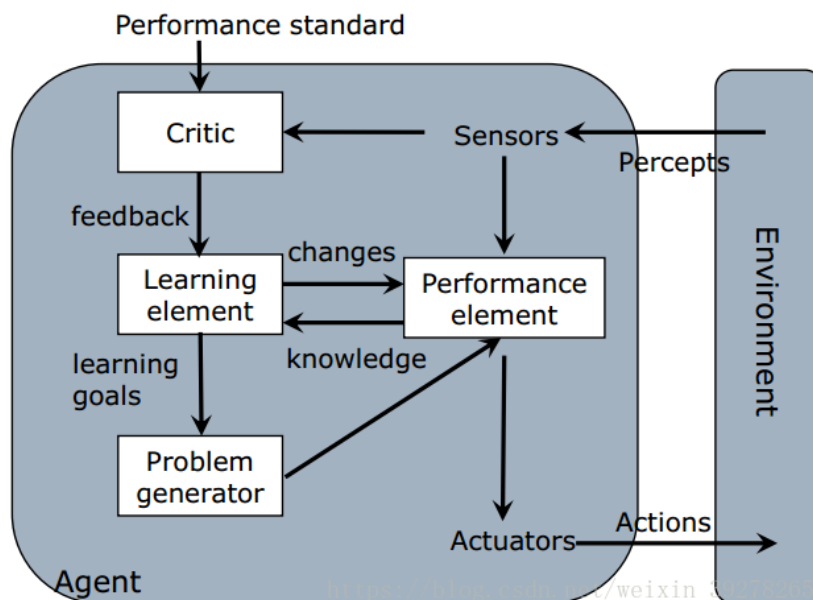
学习智能体组件

- **执行元件：** 负责选择外部动作，接受感知信息并进行行动决策
- **学习元件：** 负责改进，利用评论元件的反馈来评价智能体，并决定如何修改执行元件以将来做得更好
- **评论元件：** 根据固定的性能标准告诉学习元件智能体的运转情况如何
- **问题产生器：** 负责提议可以导致新的和有信息价值的经验的行动

智能体的学习可以看做为智能体的每个组件的**修改**过程，修改使得各组件与能得到的反馈信息更加一致，从而改进智能体的总体性能

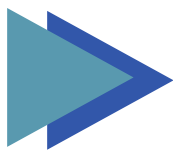
Agent程序的部件如何运转

由于前面讨论的四类都可以归为learning agent，所以这里讨论learning agent中的部件如何运转。



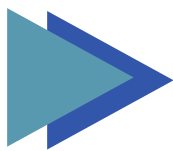
- 执行部件即前面考虑的整体Agent，接受感知，选择动作。
- 学习部件根据评判部件的反馈评价Agent做得如何，从而确定如何修改执行部件。
- 评判部件根据性能标准告知学习部件Agent的运行情况。当“将军”发生了，评判部件告知学习部件：好事情发生了（性能标准是固定的，Agent不应该修改性能标准来适应自己的行为）
- 问题生成器可向执行部件建议探索性行动，短期内可能是次优的行动，但长远而言可能是更好的行动





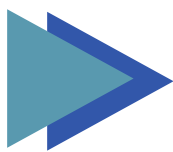
执行部件

使用执行部件在公路上行驶。执行部件包含选择驾驶行动的全部知识和过程集合。



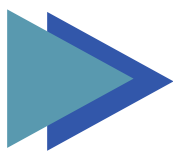
评判部件

观察评价世界，告知学习部件。不打转向灯变道，听到后车喇叭声。



学习部件

制定修改规则。如果是不好的行动，如何修改执行部件。



问题生成器

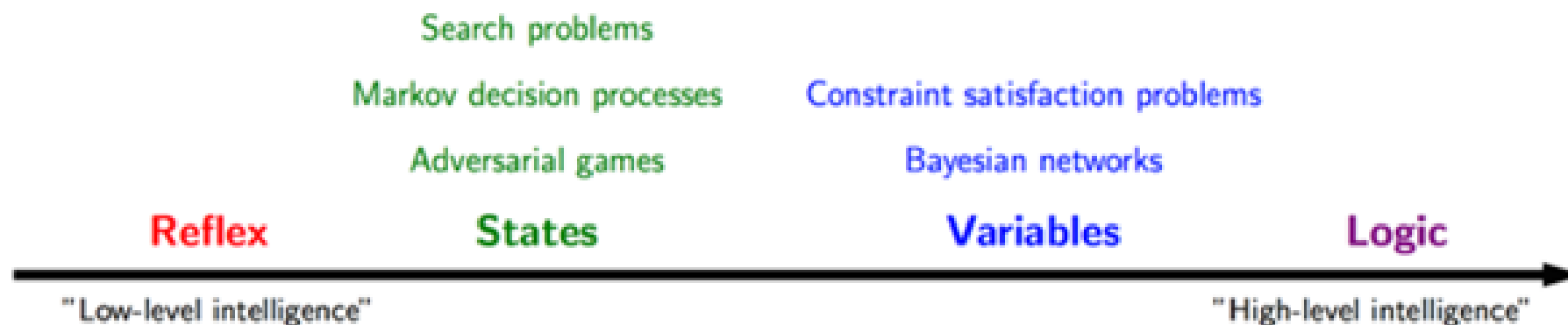
例如提议在不同路面试验一下刹车效果。



Agent各部件如何工作

Agent程序包含回答以下问题的部件：1) 当前状态；2) 当前应该采取的行动；3) 行动后果。

**As the agents get complex, so does their internal structure.
The way in which they store the internal state changes.
By its nature, a simple reflex agent does not need to store a state, but other types do.
The image below provides a high level representation of agent states, in order of increasing expressiveness power(left to right).**



如何表示状态及其转换？

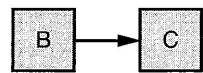
沿着复杂度和表达能力增长的轴线，有三种表示：原子表示、要素化表示、结构化表示。





01

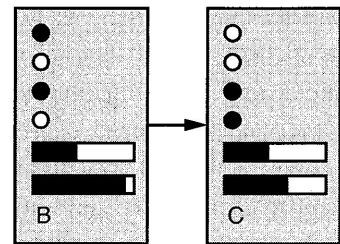
原子表示 (atomic representation) : 没有内部结构的表示。
相关内容: 搜索、博弈论、隐马尔可夫模型、马尔可夫决策过程。



(a) Atomic

02

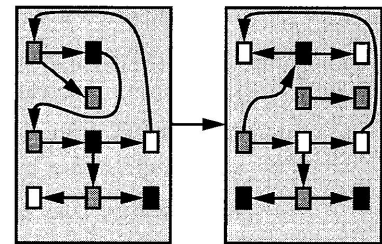
要素化表示 (factored representation) : 一个状态中包含多个要素 (原子), 即多个变量和特征的集合。
相关内容: 约束满足算法、命题逻辑、规划、Bayesian网、机器学习算法。



(b) Factored

03

结构化表示 (structured representation) : 一个状态包含对象、每个对象可能有自身的特征值, 以及与其他对象的关系。
相关内容: 关系数据库、一阶逻辑、一阶概率模型、基于知识的学习、自然语言理解。



(b) Structured



智能体特性

- **自主性**：亦称自治性，即能够在没有人或别的Agent的干预下，主动地自发地控制自身的行为和内部状态，并且还有自己的目标或意图。
- **反应性**：即能够感知环境，并通过行为改变环境。
- **适应性**：即能根据目标、环境等的要求和制约作出行动计划，并根据环境的变化，修改自己的目标和计划。
- **社会性**：即一个Agent一般不能在环境中单独存在，而要与其他Agent在同一环境中协同工作。而协作就要协商，要协商就要进行信息交流，信息交流的方式是相互通信。



智能体典型实例

- Microsoft的Office助手
- 计算机病毒（破坏**智能体**）
- 计算机游戏或模拟中的智能角色
- 贸易和谈判**智能体**（如Ebay的拍卖**智能体**）
- 网络蜘蛛Web Spider（搜索引擎中的数据搜集和索引**智能体**，如Google）
-



小结

- **定义：**智能体是可以感知环境并在环境中行动的某种东西。
智能体函数指定智能体响应任何感知序列所采取的行动。
- **性能度量**评价智能体在环境中的行为表现。理性智能体的行动使其性能度量期望值最大化。
- **任务环境**包括性能度量、外部环境、执行器和传感器。设计智能体的第一步总是把任务空间定义得尽可能完全。
- **任务环境的性质：**变化的，完全可观察的？确定性的？片段式的？静态的？离散的？单智能体的？



小结（续）

- **智能体程序**是智能体函数的实现
- **简单反射型智能体**直接对感知信息作出反应
- **基于模型的反射智能体**保持内部状态，追踪记录当前感知信息中不明显的世界信息。
- **基于目标的智能体**的行动为了达到目标，
- **基于效用的智能体**试图最大化自己期望的“快乐”
- 所有智能体都可以通过**学习**来改进其性能



作业

2.3 这道习题要探讨的是智能体函数与智能体程序的区别：

- a. 是否有不止一个智能体程序可以实现给定的智能体函数？请举例，或者说明为什么不可能。
- b. 有没有无法用任何智能体程序实现的智能体函数？
- c. 给定一个机器体系结构，能使每个智能体程序刚好实现一个智能体函数吗？
- d. 给定一个存储量为 n 比特的体系结构，可以有多少种可能的不同智能体程序？

2.5 对于下列智能体，分别给出任务环境的 PEAS 描述：

- a. 机器人足球运动员；
- b. 因特网购书智能体；
- c. 自主的火星漫游者；
- d. 数学家的定理证明助手。

■ 习题2.3

■ 习题2.5任选一小题

