



多智能体系统模型及框架

本章目录

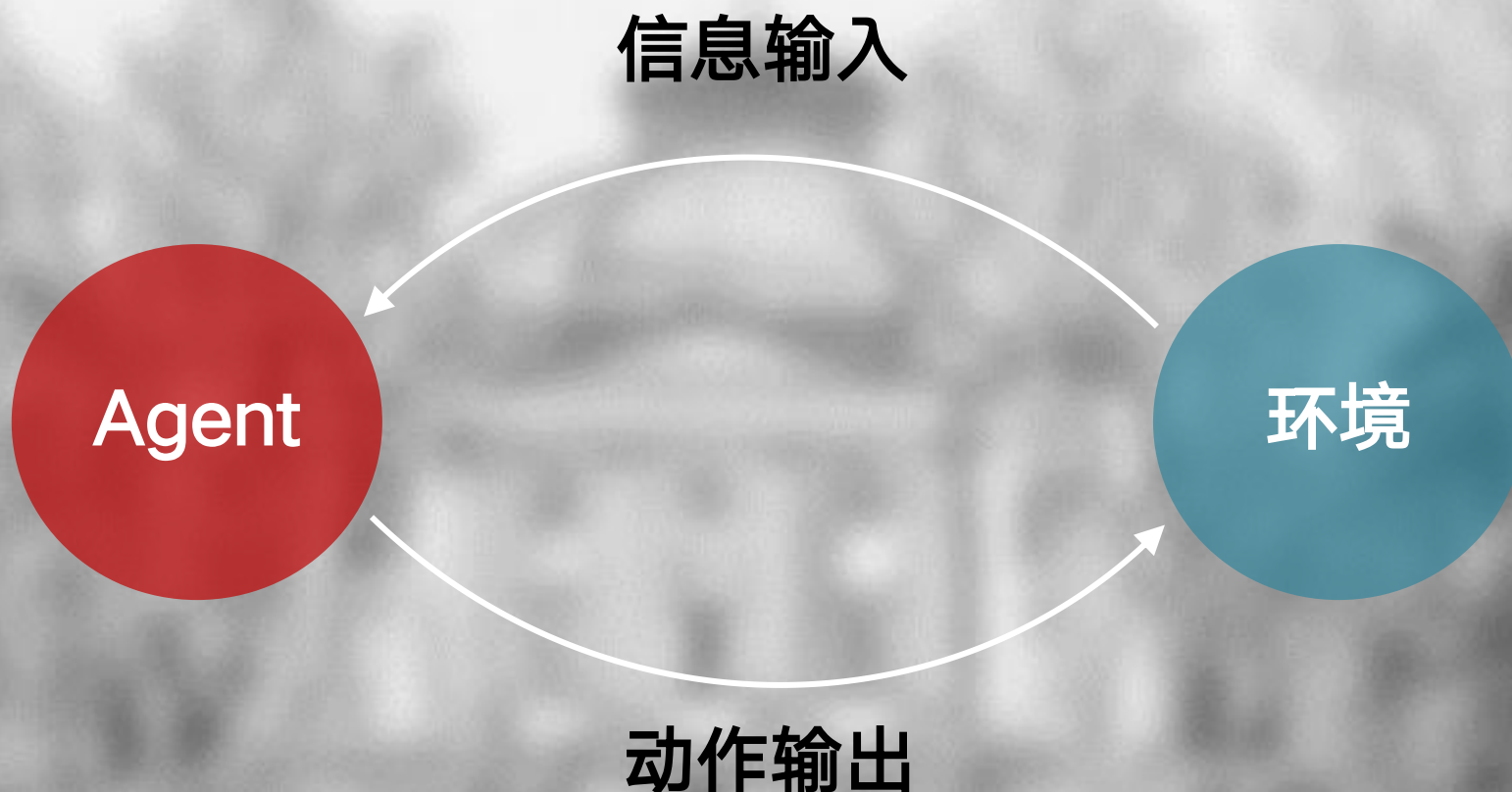
- 1.智能体与多智能体系统的基本模型与形式化定义
- 2.多智能体系统问题求解基本框架
- 3.复杂性维度
- 4.原型应用
5. Agent组成结构和构建
- 6.分层控制

1



智能体与多智能体系统的基本模型与形式化定义

Agent的抽象模型



- 1) Agent从环境获取信息输入（环境感知、信息交互）
- 2) Agent经过决策作出动作作用于环境

Agent的抽象模型——与环境的交互



Agent依赖于以下几个方面：

- 1) Agent 及其环境的先验知识。
- 2) 与环境的交互历史，其中包括：
 - 当前环境的观察值。
 - 先前经验和观察值，或从学习中获得到的其他数据。
- 3) 试图达到的目标或整个世界状态的偏好。
- 4) 它能够执行的最原始动作，即能力。

Agent的基本形式化定义

Agent

***Agent:* $a = \langle R, Loc, Obj, S \rangle$**

资源: $R = \{R_1, R_2, \dots R_j\};$

局部信息: $Loc = \{Loc_1, Loc_2, \dots Loc_k\};$

目标: $Obj = \{obj_1, obj_2, \dots obj_l\};$

行为策略: $S = \{s_1, s_2, \dots s_m\};$

Agent基本形式化定义实例分析

Agent

Agent: $a = \langle R, Loc, Obj, S \rangle$

资源: $R = ?$

局部信息: $Loc = ?$

目标: $Obj = ?$

行为策略: $S = ?$



多Agent系统的基本形式化定义

Multi
agent
system

MAS: mas = $\langle A, E, OBJ, Rule \rangle$

Agent集合: $A = \{a_1, a_2, \dots, a_j\};$

Agent交互连接: $E = \{(a_1, a_2), \dots, (a_i, a_j)\};$

目标: $OBJ = \{O_1, O_2, \dots, O_k\};$

系统规则: $Rule = \{rul_1, rul_2, \dots, rul_l\};$

多Agent系统形式化定义的实例分析

Multi
agent
system

MAS: mas = $\langle A, E, OBJ, Rule \rangle$

Agent集合: $A=?$

Agent交互连接: $E=?$

目标: $OBJ=?$

系统规则: $Rule=?$

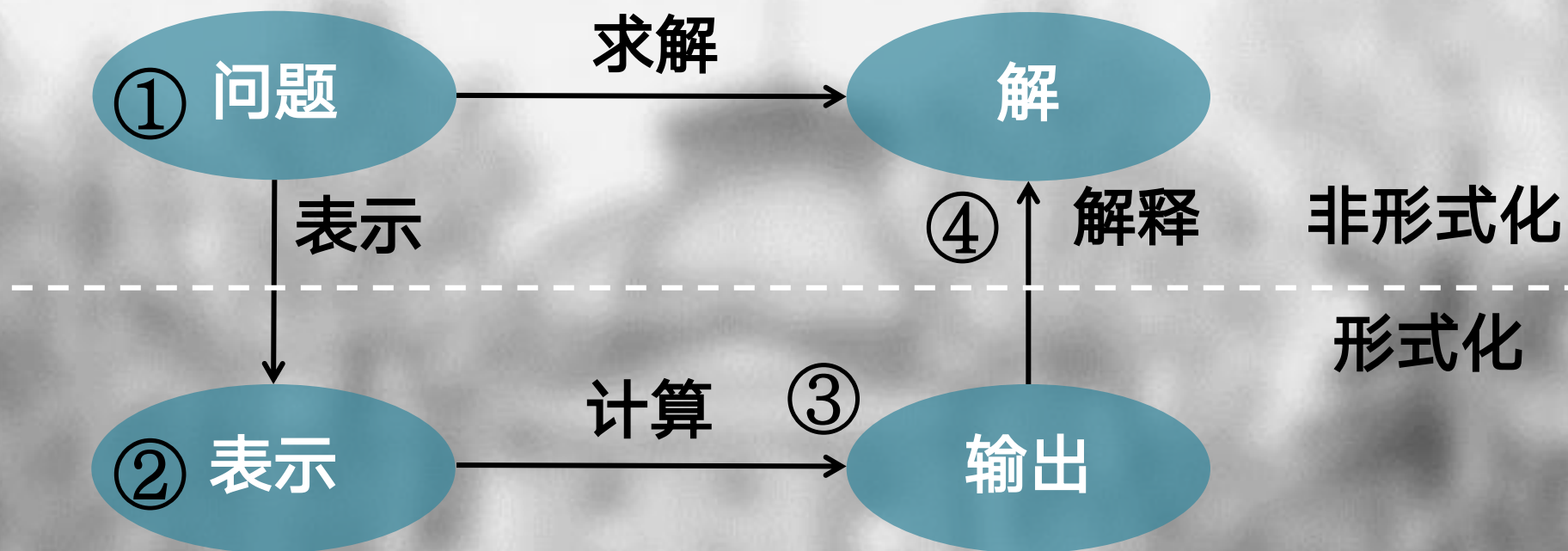


2



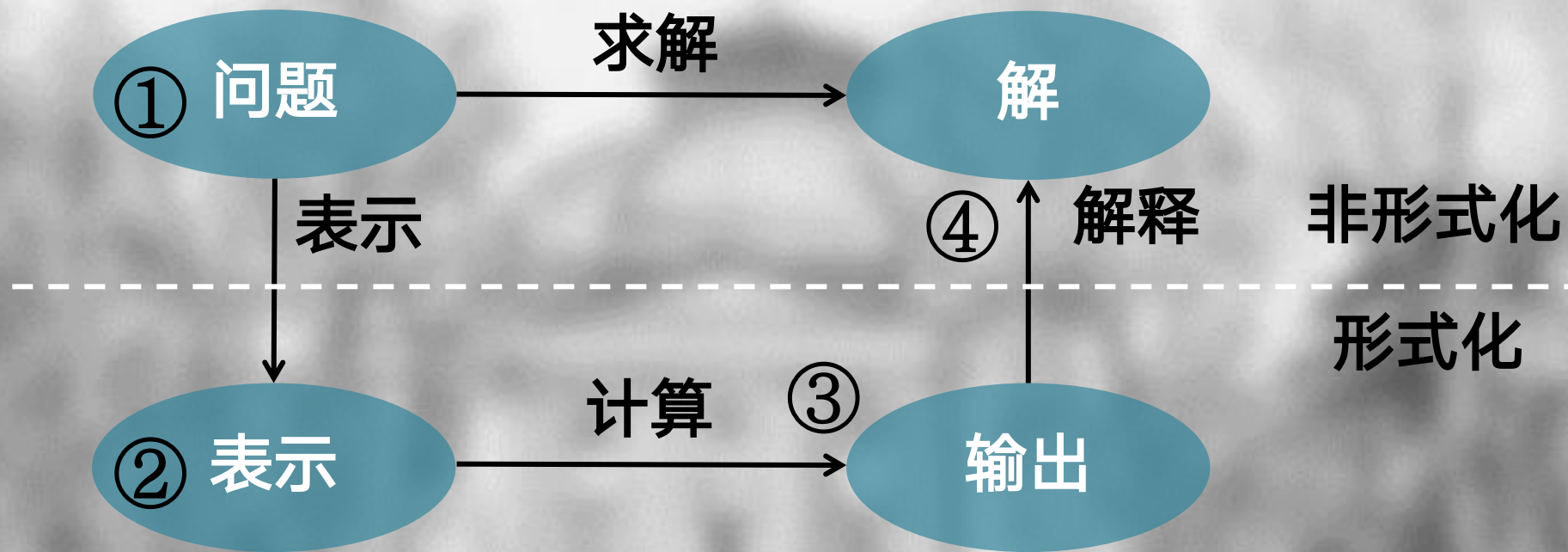
多智能体系统问题求解基本框架

多智能体系统问题求解基本框架



- ① 具体化任务，并制定解决方案。
- ② 用特定的语言表达问题,以便计算机进行推理。
- ③ 用计算机计算出相应结果进行输出，可以给用户呈现一个答案或是在环境中需要执行的一系列行为。
- ④ 解释作为问题的解决方案的输出结果。

多智能体系统问题求解基本框架



知识是可以用来解决本领域问题的领域信息。
表示方案是 Agent 中使用的知识形式。
知识库则是 Agent 中存储的所有知识的表达

① 确定问题

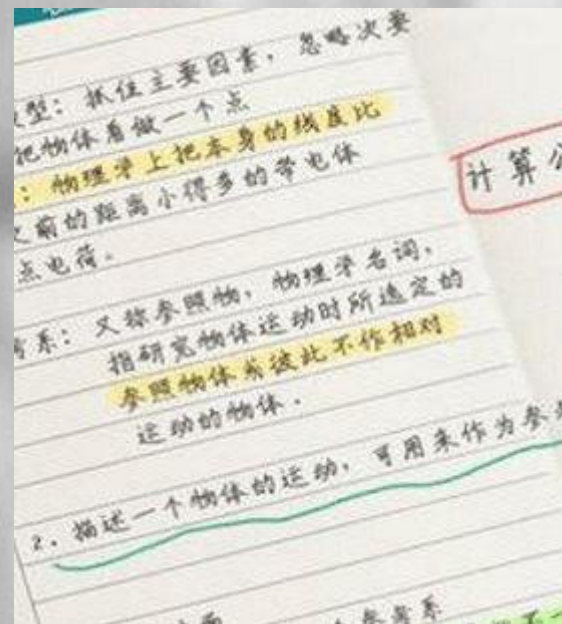
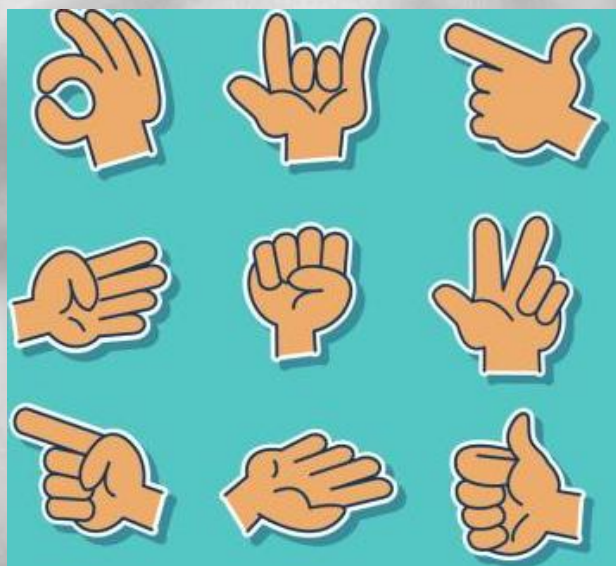
确定问题解的构成，对问题的描述进行提炼



② 表示——对问题、环境、系统进行建模

符号和符号系统

表示：表示问题，便于计算机来解决它。



符号是符号系统中被操作的基本单元

Agent可以看做通过操作符号来产生动作。

② 表示——对问题、环境、系统进行建模

抽象和抽象层次

Agent可以用物理符号系统对世界进行建模。
世界模型是世界中真实存在的详情和世界的动态行为的表达。
所有的模型都是**抽象**出来的。

抽象的层次是指按抽象程度排列的一个偏序。
低层抽象比高层抽象拥有更多的细节。

模型，不是看它是否正确，而是看它是否**有用**。

② 表示——对问题、环境、系统进行建模

高层抽象？
and
低层抽象？



以传送机器人为例

② 表示——对问题、环境、系统进行建模

两个共有的层次

- 知识层：是将Agent所知道的、所相信的以及它的目标抽象出来。知识层考虑的是Agent知道什么，而不是它如何进行推理。
- 符号层：是对Agent的推理进行描述。为了实现知识层，Agent需要通过操作符号来产生答案。

知识层说明的是对于外部世界，Agent相信什么，它的目标是什么；

符号层是对Agent内部关于外部世界推理的描述。

③ 推理与行为（计算并输出）

符号的操作产生行为被称为推理

人工智能表达的主要是需要计算什么（what），而不是怎样（how）进行计算。

③ 推理与行为（计算并输出）

在决定 Agent要做什么时，有三个方面的计算必须区别开来：

- **设计时刻推理**：是在设计Agent时进行的推理。它由Agent 的设计者而不是 Agent本身来完成。
- **离线计算**：在观察世界、需要行动之前，Agent能做的计算。Agent离线获取背景知识和数据，并将其汇编成可用的形式，称之为知识库。背景知识可以在设计时或离线状态时给出。
- **在线计算**：是在观察环境及在环境中进行某些行为时由Agent完成的计算。在线状态获取的信息称为观察。Agent必须使用知识库和观察来决定要做什么。

③ 推理与行为（计算并输出）

构造agent的两大一般策略

第一个策略：简化环境并为这些环境建立复杂的推理系统。

第二个策略：在复杂环境中建立简单的agent。

④ 定义解

解的类型

最优解：问题的最优解是指在所有的可能解中最符合求解质量度量的最好的解。这个度量用序数来描述。但是在一些情况下，例如综合多种标准或不确定推理时，我们需要一种要考虑相对大小的基数度量。



序数度量



基数度量

④ 定义解

解的类型

可满足解：Agent经常并不需要最优解，只需一些满足条件的解就可以。

近似最优解：近似最优解是指其质量度量接近理论上存在的最优解。例如，机器人为了送出垃圾可能并不需要达到最优的传输距离，可能只需要在最优距离10%范围以内就可以。

可能解：一个有希望的解。

练习

- 从图书馆到地铁口最近距离下校车的承载人次。
- 自动检测课堂活跃程度的智能体

3



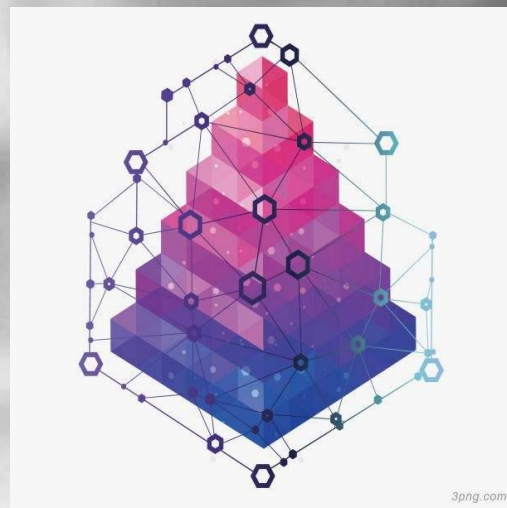
智能体设计的复杂性维度

复杂性维度简介

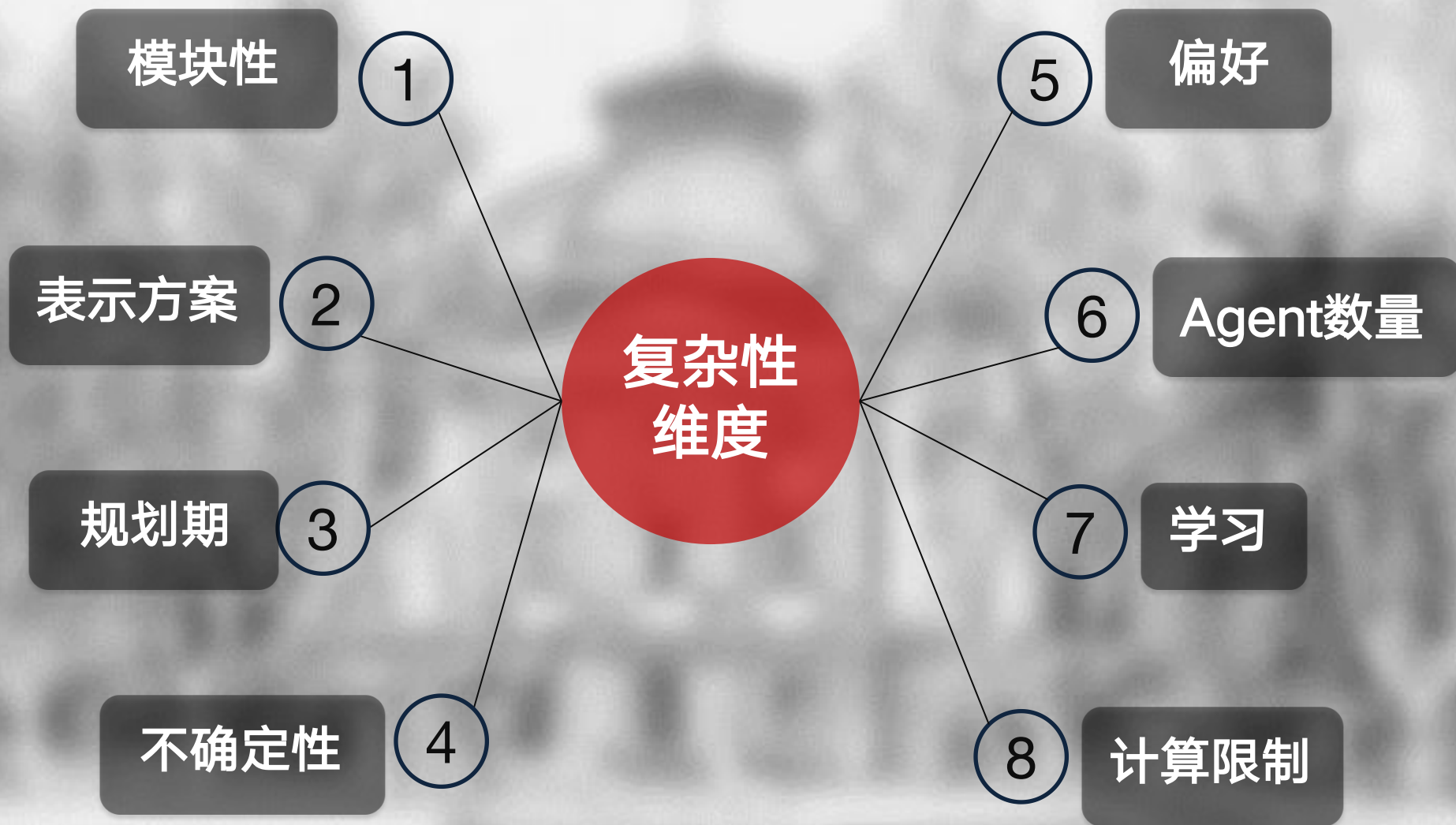
什么是智能体的复杂行维度？

从自动调温器到在竞争环境中有多种目标的企业，Agent在环境中行为的复杂性各不相同。在Agent设计过程中，存在着多个**复杂性维度**。这些维度可以分开考虑，但构建智能Agent时必须组合起来。

复杂性维度定义了人工智能的**设计空间**，空间里的不同点可以通过改变维度值来获得。



复杂性维度分类





模块性维度

模块性维度中Agent的结构

扁平的：系统中没有组织结构

模块化的：系统可以被分解成独立的可理解的交互模块。

分层的：系统是模块化的，模块本身分解成了交互模块，它们中的每一个又都是一个分层系统，由此循环下去，直到分解成最简单的组件。

在**扁平结构**或**模块化的结构**中，Agent主要进行**单层**抽象上的推理，而在**分层结构**中，Agent主要是在**多层**抽象上进行推理。低层结构主要涉及低层抽象的推理。

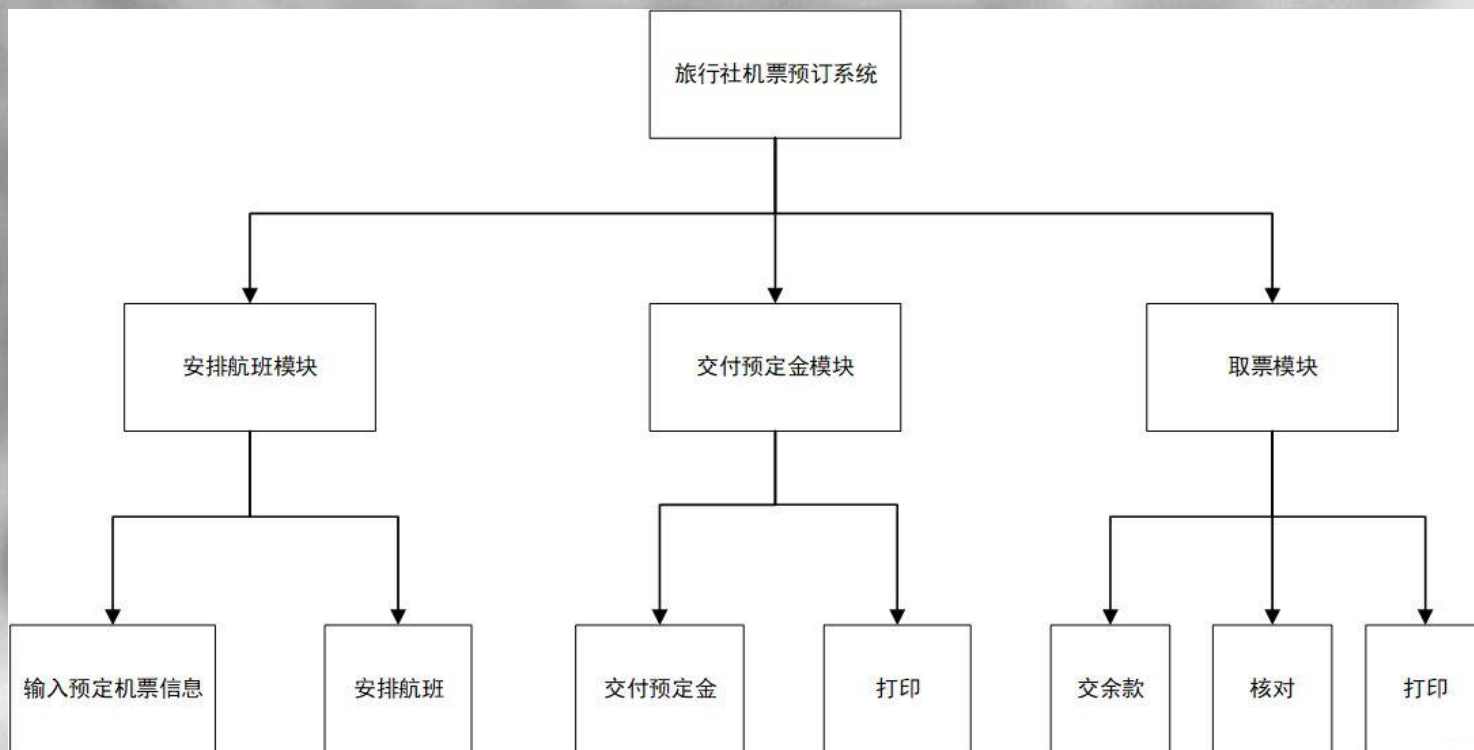
模块化维度场景示例

扁平结构： Agent会将所有需要完成的工作按照某一顺序完成。

模块化： Agent会将任务分解为若干子任务，再考虑子任务中的细节，例如：订票、到达出发地机场、到达目的地机场、到达目的地。到达出发地机场，到达目的地机场，到达度假的地点)。

分层结构： Agent 将用分层的方式解决这些子任务，直至将问题分解成一些简单的问题。

练习



2



表示方案维度

表示方案维度简介

表示方案维度主要讨论如何描述世界。

在表示方案维度中，Agent主要通过以下几个方面进行推理：



表示方案维度Agent的推理途径

1.状态

在最简单的层次上，Agent可以通过一系列**独立的确**
定状态来进行推理。**状态**分Agent的**内部状态**（信念
状态）和**环境状态**两部分。

举例：**自动恒温器**的两个信念状态：**关闭**和**加热**。三
个环境状态：**冷**、**舒适**和**热**。利用信念状态和环境状
态的组合可以描述现实中的情况。

环境状态	信念状态
环境较冷	加热
环境较热	关闭
环境舒适	关闭

表示方案维度Agent的推理途径

2.特征

用**状态特征**或**状态真假**的命题进行推理，比列举一系列状态进行推理更加容易。**状态**可以用一系列**特征**描述，其中每个特征在每个状态都有一个值。

举例：某个观察房屋内灯泡是否出现故障的Agent。可能有每个开关位置、状态（工作是否良好、是否短路、是否被破坏）和每个灯牌是否能正常工作的特征。**描述开关的状态**时可以使用**up**和**down**两种值。通过这种方式描述照明设备的状态。



布尔型命题有**真**、**假**两种情况。利用十个命题就可以描述 2^{10} 种状态，更加**简单**。此外用状态的**紧凑表示法**更容易理解，表明Agent已经掌握一些**规律**。

表示方案维度Agent的推理途径

3.个体与关系

特征可以依赖于关系与个体。单个个体上的关系是一种属性，个体之间的每一种可能关系上都存在特征。用个体及它们之间的关系进行相关性描述，比对特征或命题直接处理更加简洁。

举例：看护房屋照明系统的Agent可以将灯泡与开关看做个体，并观察它们之间的位置关系position与连接关系connect-to。利用位置关系表示开关s1开启的特征：`position(s1,up)`。

补充：Agent有时需要对无限个体的集合进行处理，例如数的集合。此时无法用状态或特征描述，只能在关系层面上进行推理，例如用函数关系推理。

3



规划期维度

规划期维度简介

规划期维度用来说明Agent规划的**向前时间的程度**。当Agent决定做出某个动作时，能够观察到**未来的远近**，被称为**规划期**。规划期是Agent认为它的**动作结果所能影响的向前程度**。

Agent做规划时所考虑的时间点被称为**阶段**。

规划期维度中Agent的分类



练习

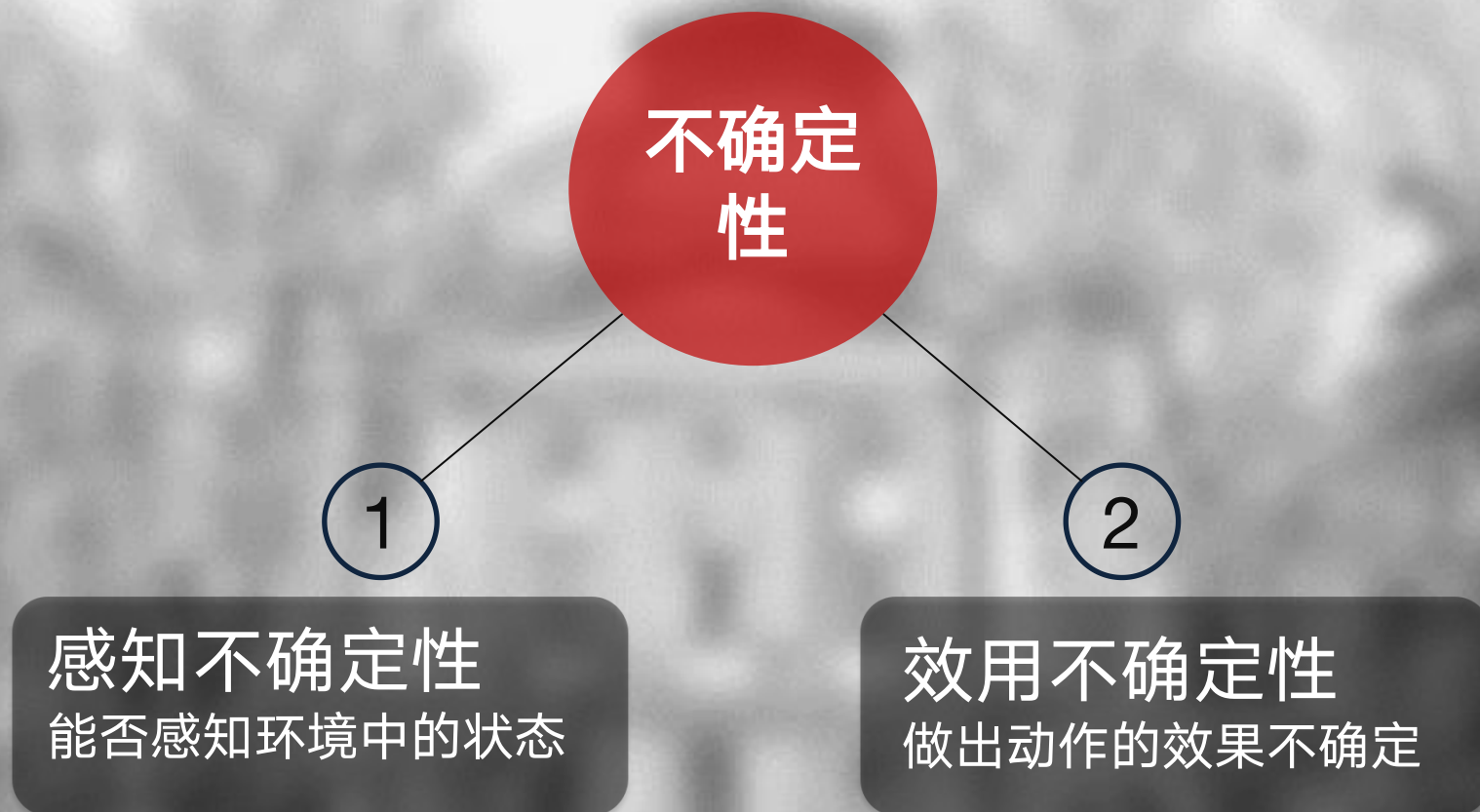
- 张三突然来学校找李四。
- 张三计划在今年暑假时去云南旅游。
- 张三每周一、周五都去体育馆锻炼。

4



不确定性维度

不确定性维度简介



感知不确定性

感知不确定性 维度

1

完全可观察

Agent直接观察世界状态

2

部分可观察

Agent不能直接观察状态或相同的观察结果导致很多状态。

效用不确定性

```
graph TD; A((效用不确定性)) --- B((1)); A --- C((2)); B --- D[确定性的]; C --- E[随机的]
```

效用不
确定性

1

确定性的

动作所导致的状态由动作及之前的状态决定。


2

随机的

对于结果状态，只能给出概率分布。

练习

1				2	
---	---	--	--	---	--

		4
3		



偏好维度

偏好维度简介

Agent会为自身获取更优的结果，做出某一动作优于另一动作选择的唯一原因是其偏好动作会导致更理想的结果。

举例：医师一般会考虑痛苦、预期寿命、生命质量、金钱成本（对病人、医生和社会）以及其他必要的东西。当这些条件发生冲突时，医生需要给出折中的考虑，即**偏好**。



偏好维度简介

偏好维度是看Agent是否具有**目标**和**复杂偏好**。

目标：在某一最终状态下要达到的**完成目标**，或是在所有已访问过的状态中**必须被保持的目标**。

举例：机器人的目标可能是拿取两杯咖啡，并保持不对他人造成伤害。

复杂偏好：在不同时期**权衡**各种期望的结果。**序数偏好**表示只注重偏好的排序。**基数偏好**涉及有关值的大小。

举例：某人喜欢黑咖啡的程度超过茶，这是序数偏好。而基数偏好给定了饮料味道和等待时间的权衡，如果味道好，某人能够容忍的等待时间就比较长。

6

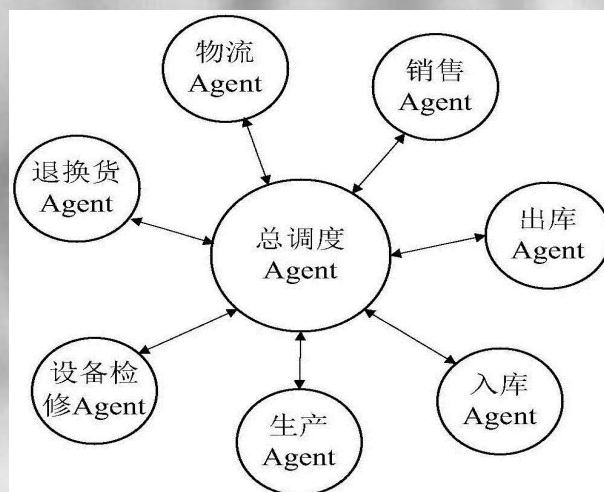


Agent数量维度

单Agent推理、多Agent推理

单Agent推理： Agent会假设其他Agent为环境的一部分。如果没有其他Agent，或者其他Agent的动作不会对自身造成影响，则假设是合理的。

多Agent推理： Agent会将其他Agent的推理考虑进来。当其他Agent的目标或偏好部分依赖于此Agent的行为，或Agent必须与其他Agent通信时，会有这种情况。如果Agent同时进行动作，或环境只是部分可观察的，那么Agent一起进行的推理将会更加困难。





学习维度

学习维度简介

通常情况下，Agent需要一些先验知识或其他资源来帮助它进行决策。

学习维度由以下两个方面决定：

- 1.已有的知识
- 2.学到的知识（从数据或经验获取）

学习一般指的是找到与数据相符的最好模型。

8



计算限制维度

计算限制维度简介

有时Agent可以计算出最好的行为，但是受限于**计算资源**和**计算时间**的限制，不能及时给出计算结果。在有计算限制的情况下，Agent给出的计算结果可能不是最优的。



计算限制维度中Agent的性质



任意时间算法是解的质量随时间的推移而提高的一种算法。它指的是可以在任意时间内产生**当前最佳解**，但如果给定**更多时间**，可能会产生**更好的解决方案**。

复杂性维度总结与整理

维度	值
模块性	
表示方案	
规划期	
感知不确定性	
效用不确定性	
偏好	
学习	
Agent 数量	
计算限制	

复杂性维度总结与整理

维度	值
模块性	扁平的、模块化的、分层的
表示方案	状态、特征、关系
规划期	无规划的、有限阶段、不确定阶段、无限阶段
感知不确定性	完全可观察、部分可观察
效用不确定性	确定性的、随机的
偏好	目标、复杂偏好
学习	已知的知识、学到的知识
Agent 数量	单个 Agent 、多 Agent
计算限制	完全理性、有限理性

9

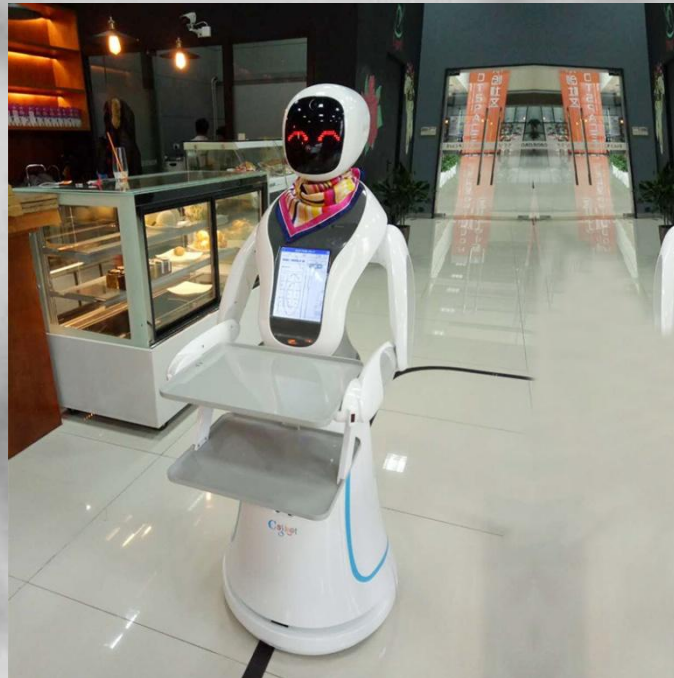


多维交互

多维交互

表示方案维度与模块性维度交互：一些层次上的简单模块可以用一系列有限状态集进行推理，而其他抽象层次可能要对个体及关系进行推理。

举例：送餐机器人。保持平衡的模块相对简单；判断包裹、人与房间关系的模块较为复杂。



多维交互

规划期维度与模块性维度交互举例：在高层次时，小狗跑过来并得到治疗可以得到即时奖励。当决定将它的爪子放在哪里时，需要很长时间来得到奖励。

效用不确定性与模块性维度交互：在分层结构的某个层次上，某个动作可能是确定性的，但在另外的层次则可能是随机的。

偏好模型与不确定性交互：Agent必须权衡满足由一定概率的主要目标，还是满足具有更高概率的策略目标。

多维交互

模块性中可以使用**多Agent**。设计单个Agent的一种方式**是构造多个拥有共同目标的交互Agent**，这样能够使较高层次的Agent智能地执行动作。

学习可以通过**特征**来进行描述，决定哪个特征值能够最好地预测其他特征的值。**学习**也可以通过**个体及关系**来进行。

维度中**模块性**和**有限理性**能使推理更加**有效**。虽然形式化体系变得复杂，但却能够通过将系统**分解**成更小的组件并提供所需的近似值，使Agent能够在**有限时间和有限内存中及时执行动作**，有利于构建复杂系统。

4

原型应用



本节介绍

人工智能领域的应用广泛而且非常多样化，本节我们介绍两个应用领域,来研究智能推理及动作背后的原理。

1.自主传输机器人

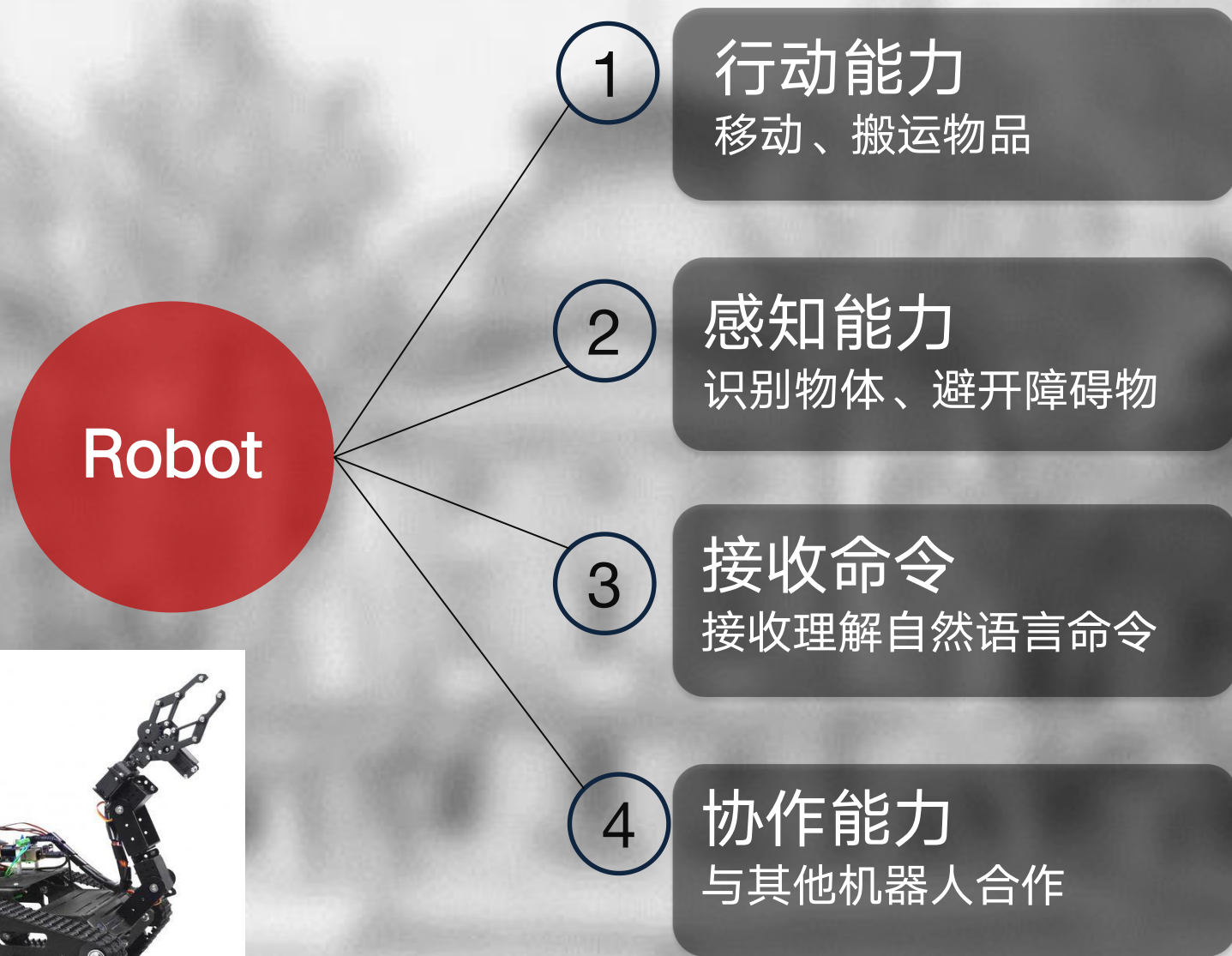
2.交易Agent

1

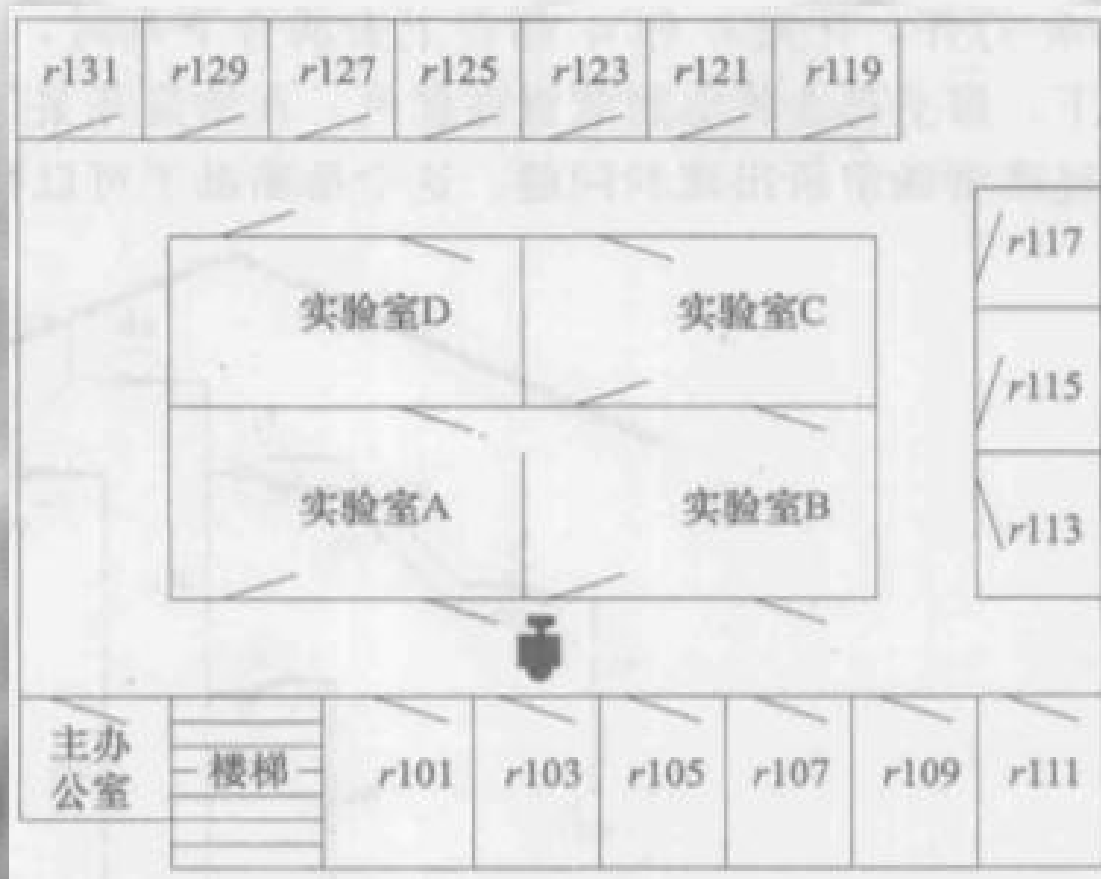


自主传输机器人

自主传输机器人功能

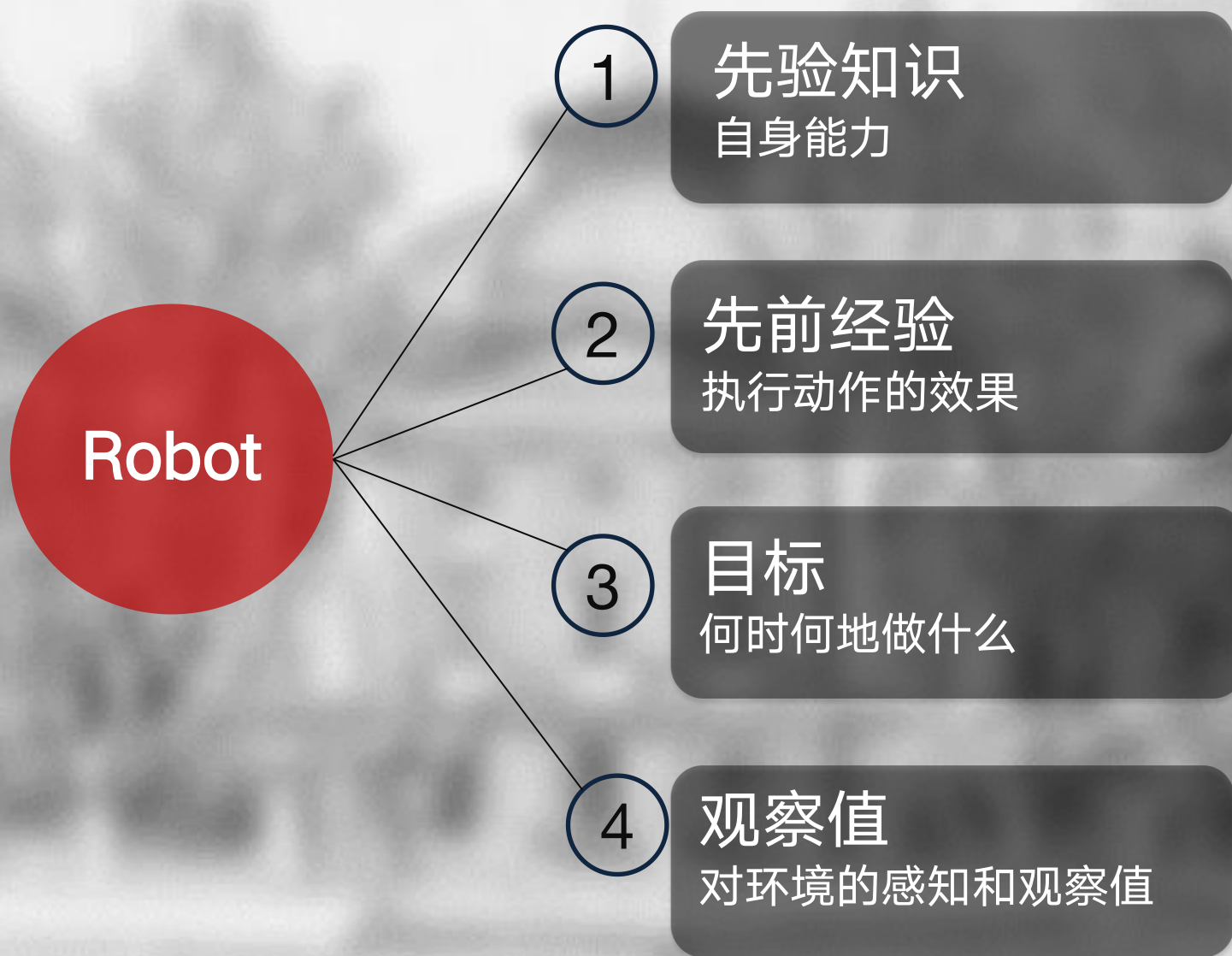


工作环境及任务示例



工作环境由多个房间、过道和楼梯组成。机器人需要在房间之间传递包裹、饮料或菜肴。

自主传输机器人的输入



复杂性维度

推理任务中，每一个维度上都能增加概念复杂度

层次分解

效用不确定

规划期

团队协作

感知不确定

学习



2



交易Agent

交易Agent的输入

交易Agent在信息环境进行交互，为用户提供获取商品的服务



交易Agent的复杂性维度

层次分解

问题域复杂，需要进行层次分解。为某些服务提供专门的模块以简化问题。
例如：航班优化转乘模块。

不确定性

旅行Agent不知道航班何时会被取消或延迟。
因此需要为意料之外的事情做计划。

多Agent

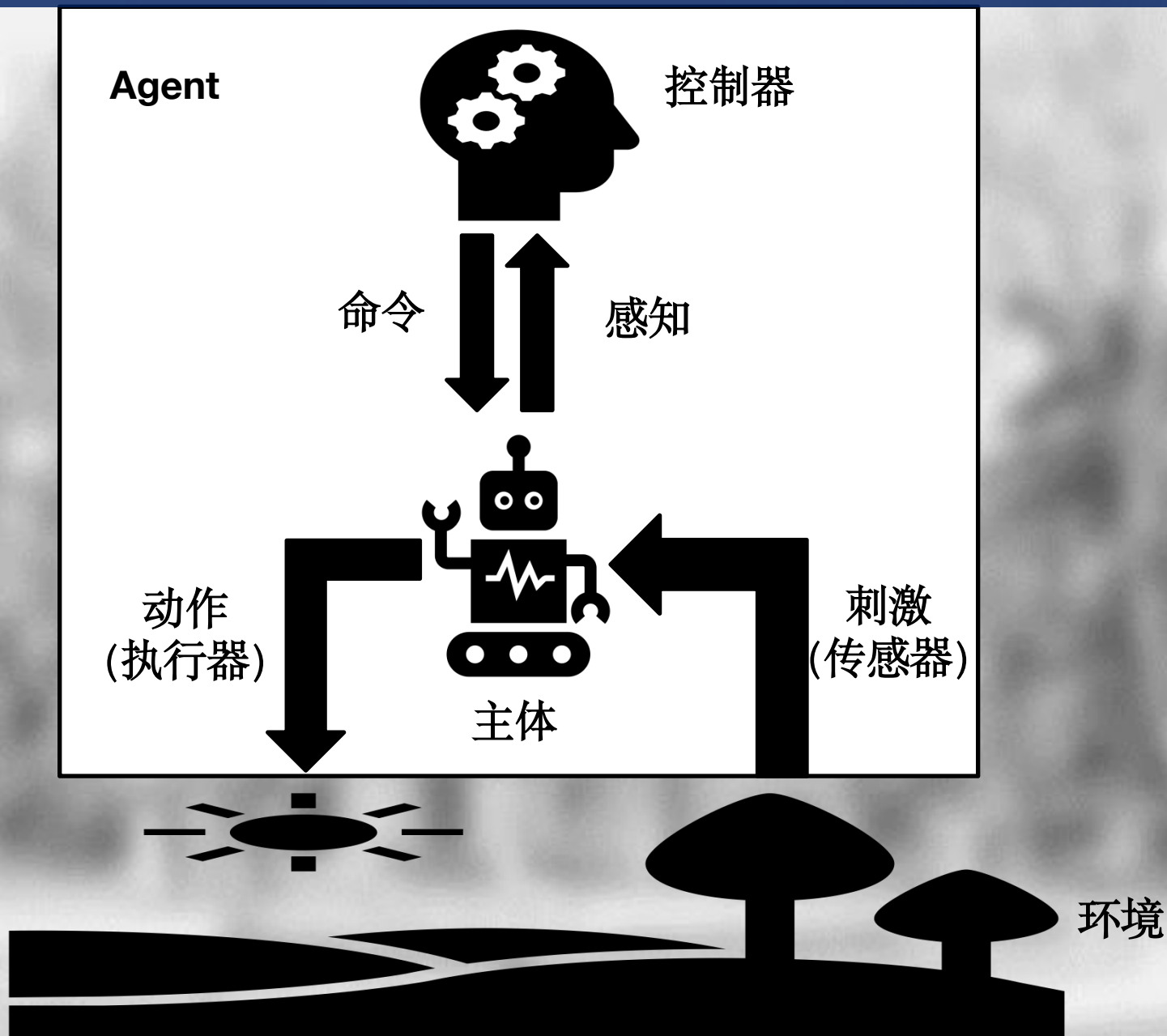
对其他Agent的行为进行推理。
例如，在拍卖系统中，可能同时存在多个竞争性Agent，此时需要对其推理。

5



Agent组成结构和构建

Agent系统组成



Agent控制器的构建

以下我们来考虑构建一个家居交易Agent,它监控着多种家居用品价格(如它监视着某些特定交易,并记录卫生纸的涨价情况)及家中相应存量。它必须决定是否购进某物及在何种价格时购进。感知对象为当前商品价格及家中存量。命令为Agent决定购买的各类商品的数量(若不购进则数目为0)。



Agent控制器的构建

Agent在构建时需要给出以下定义：

离散时间和稠密时间： 设 T 为时间点集合，若任意两个时间点间仅存在有限个时间点，则称 T 是**离散**的。但如果 T 中任意两个时间点间都存在另一个时间点，则认为 T 是**稠密**的。

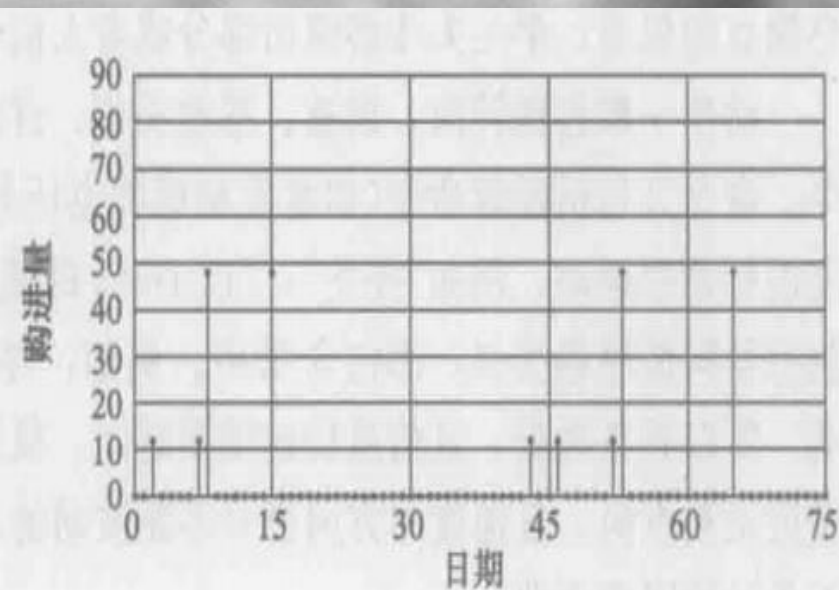
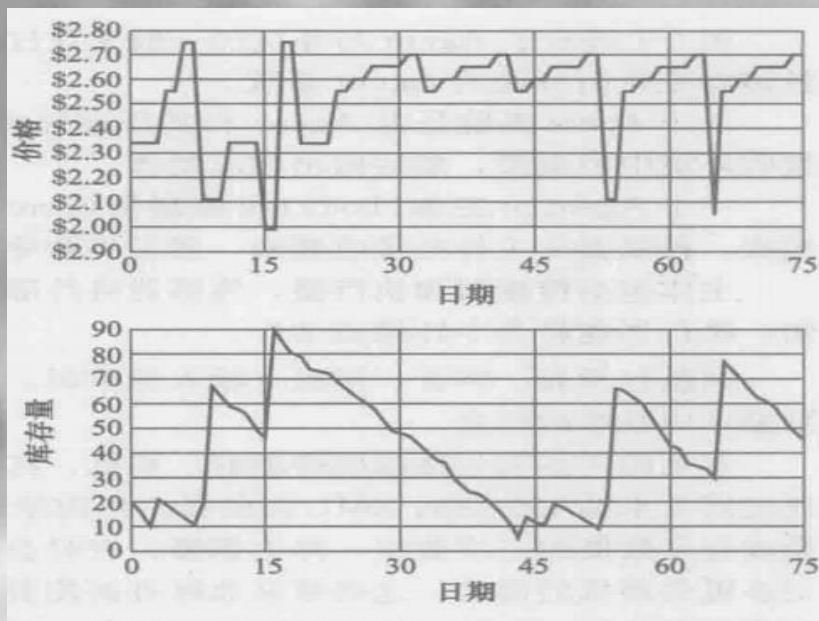
感知轨迹： 假定 P 为在 T 时间内所有可能感知对象的集合，从 T 到 P 的函数为**感知轨迹函数**，描述了每一时间点所观察到的事物。

命令轨迹： 假定 C 是所有命令的集合，一个为从 T 到 C 的函数为**命令轨迹函数**，其代表在每个时间点的命令。

Agent控制器的构建

从感知轨迹映射到命令轨迹的转换过程是有因果联系的，如果对于所有时刻 t ，在 t 时间的命令都仅由 t 和其之前的感知信息决定。因果限制是必需的，因为Agent处于时间流中，所以 t 时刻的命令不可能依靠 t 之后的感知信息。

控制器需要完成因果转换的具体实现。



感知轨迹描述了每个时间点(如每天)的商品价格和库存数量

命令轨迹描述每个时间点Agent购进的商品数量

Agent的信念状态和因果转换

因果转换： 在上面的例子中，因果转换是对于任意时刻，Agent需要购买多少日用品取决于历史价格、历史库存量(包括时下价格和现有库存)及过往购买历史。

信念状态： 因为Agent不能直接获得它们的全部历史信息，它只能获得当前的感知信息和它仍能记住的信息。因此一个 Agent 在时间 t 时的**信念状态**是其所能记住的所有以前时间的信息的总和。**信念状态**可以包含任何信息，仅受制于其存储器大小和处理能力限制。

Agent的因果转换过程

控制器必须保存Agent的信念状态并决定每个时刻发出何种命令(**指令函数**)。当它做这些时，它应该获得的信息需包括自身信念状态和当前感知。

离散时间下的**信念状态转换函数**可表示为：

$$\text{remember: } S \times P \rightarrow S$$

其中，S是信念状态集，P是可能认知的集合；表示状态S是在信念状态S之后观察到p得出的信念状态。

指令函数可表示为：

$$\text{do: } S \times P \rightarrow C$$

其中，S是信念状态集，P是可能认知的集合，C是可能指令的集合；C表示当前的信念状态为S观察到P时控制器需要发出的指令C。

Agent的控制器实现

为了实现上例中的因果转换，控制器必须跟踪过去20天内的价格，信念状态便包含20个数据(ave):

$$\text{ave} = \text{ave} + (\text{new} - \text{old})/20$$

为了使控制器更加简单，其可以不需记忆过去20天的历史来获取平均值，而改为仅存储平均值，并使用平均值来替代最早价格，信念状态便仅包含一个数据(ave):

$$\text{ave} := \text{ave} + (\text{new} - \text{ave})/20$$

讨论

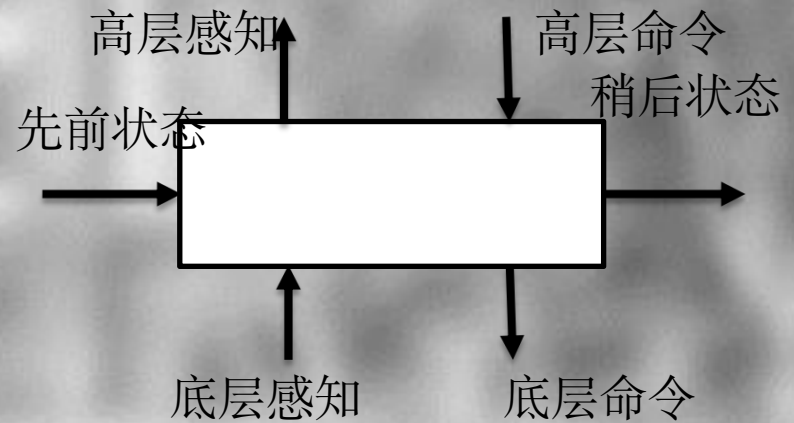
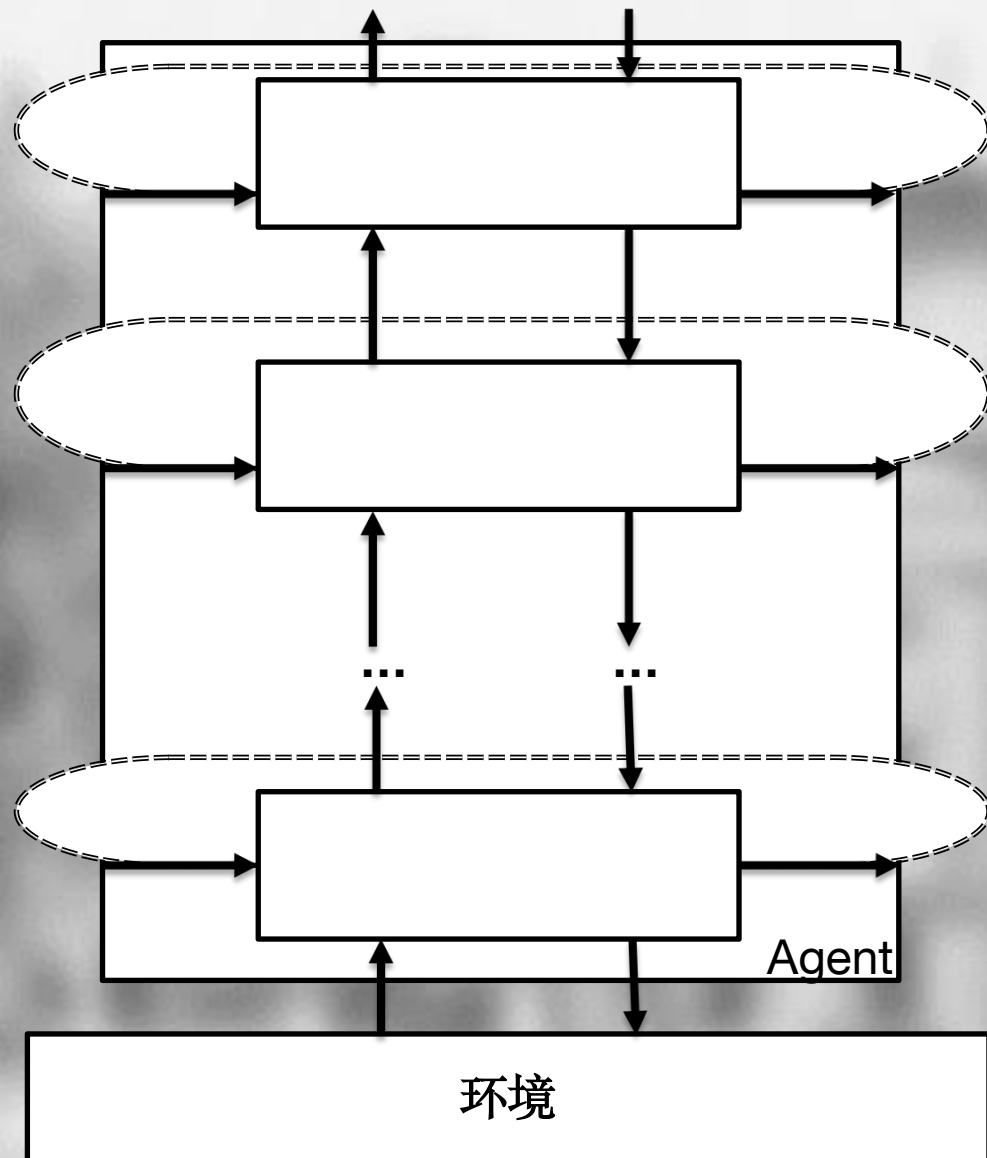
- 学生平时分Agent
- 信念状态
- 信念状态转换函数
- 指令函数

6



分层控制

分层agent系统

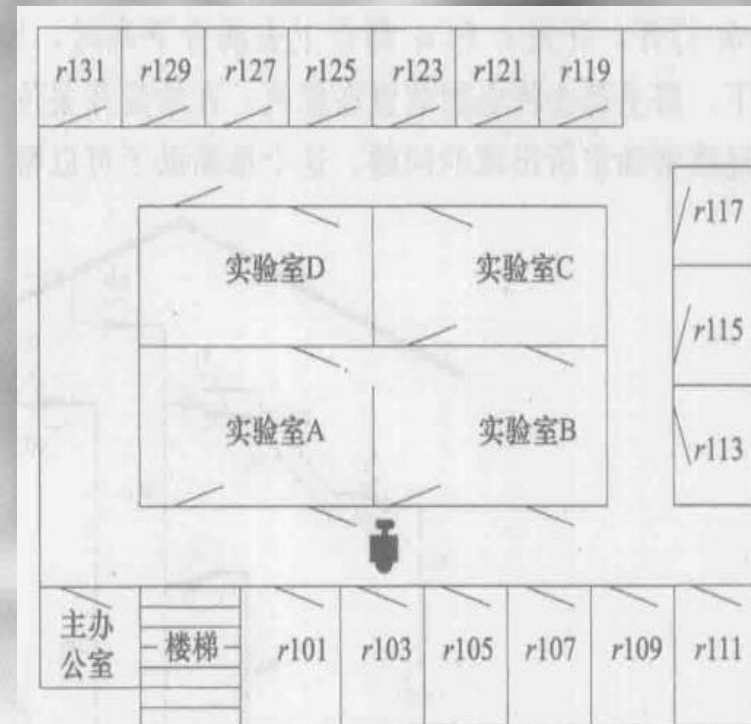


一个理想的分层Agent系统结构。未标注的矩形表示层，箭头表示信息流。虚线说明某时刻的输出作为下一时刻的输入

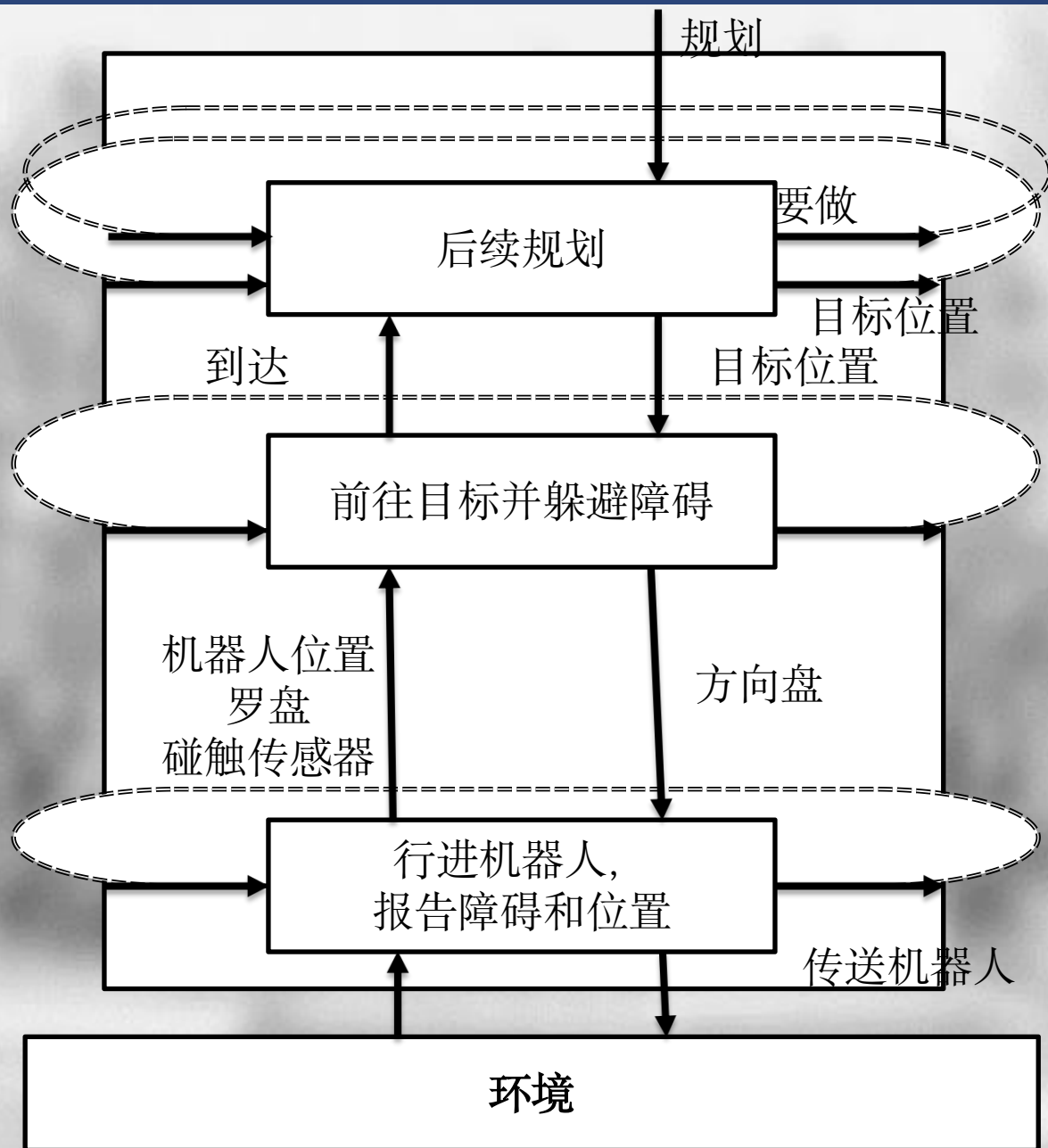
分层Agent系统

给定一个传送机器人能在躲避障碍物的同时执行高级的导航任务。假定传送机器人需要在避开可能的障碍物的同时按照顺序访问中下图环境的一系列位置，那么分层控制器如何设计？并给定以下假定：

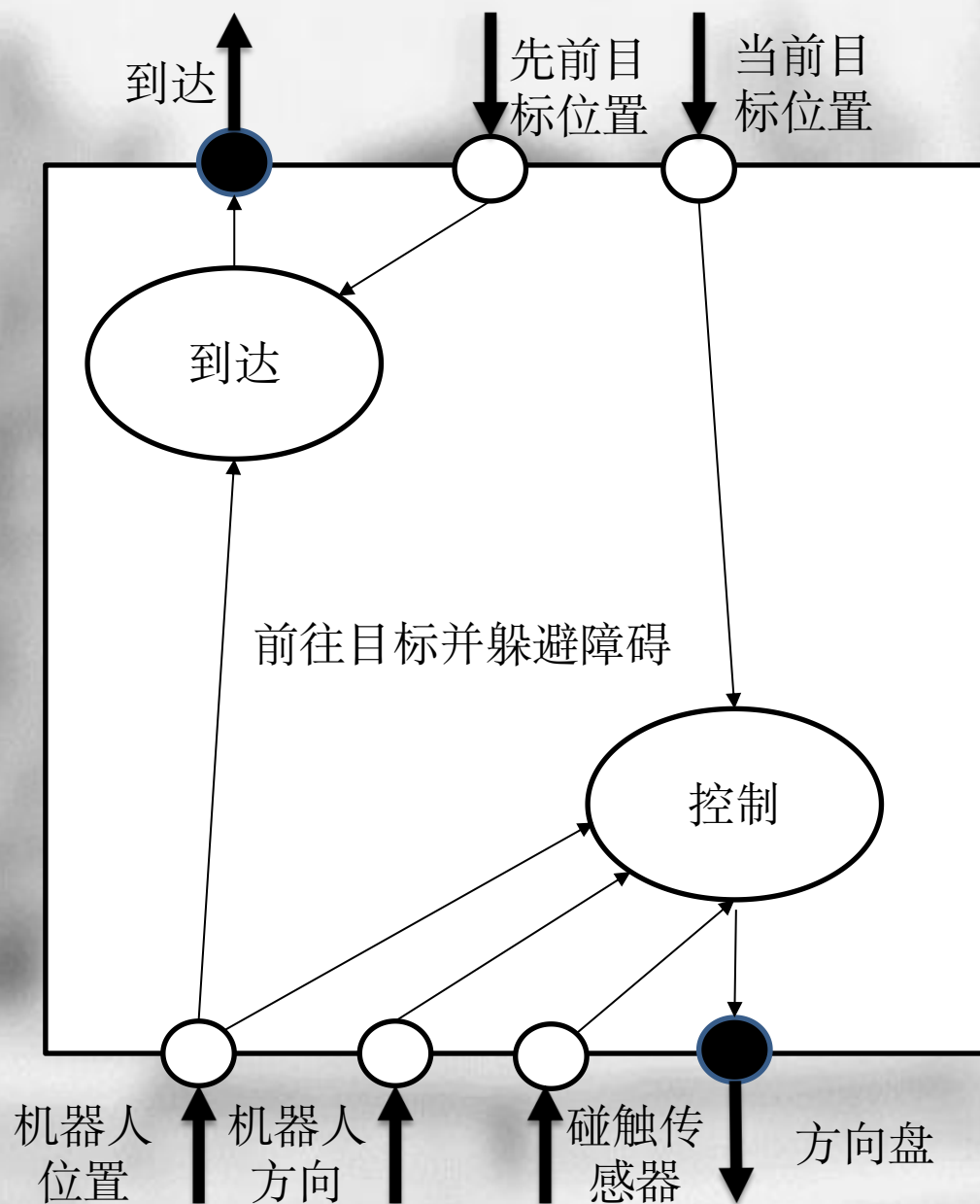
1. 机器人只能向正前方运动或按照一个固定的半径绕圈。
2. 机器人有一个位置传感器，可以给自己提供当前坐标和方向。还有唯一一个伸向正前稍偏右的触觉传感器，用于探测是否碰触到障碍。
3. 传感器指向前方偏右30度的方向。机器人不自带地图，且环境可以改变。(如:障碍可以移动)。



传送机器人的分层分解



传送机器人控制器的中间层



传送机器人控制器的顶层

