

课后作业2-1

- 举3个你所知道的Agent的例子，尽可能准确地定义以下问题：
 - （1） Agent所处的环境（物理环境、软件环境等），环境中的状态，环境的种类（是否完全可观察、是否有不确定性、是否是多Agent、是否是序贯、是否是连续）。
 - （2） Agent可执行的动作库，以及执行这些动作的前提条件。
 - （3） Agent的设计目标，即要实现什么。

参考答案2-1

扫地机器人

- 物理环境：需要打扫的房间
- 环境中的状态：机器人所处的位置、机器人所处位置是否有垃圾、机器人的当前朝向
- 环境的种类：完全可观察的、确定性的、单Agent、片段式的、离散的
- 动作库：
 - 向前移动（向前移动不会越出边界）
 - 清理垃圾（当前位置处有垃圾）
 - 沿顺时针方向转动 90° （无条件）
- 设计目标：清理房间内的垃圾

参考答案2-1

恒温器

- 物理环境：需要保持恒定温度范围的房间
- 环境中的状态：环境温度、与外界的热交换程度
- 环境的种类：部分可观察的、确定性的、单Agent、片段式的、连续的
- 动作库：
 - 制冷（温度高于恒定温度）
 - 制热（温度低于恒定温度）
 - 停止（无条件）
- 设计目标：维持房间温度在恒定范围内

参考答案2-1

王者荣耀游戏AI

- 软件环境：王者荣耀游戏场景
- 环境中的状态：系统信息（游戏时间，野怪信息，防御塔信息）、玩家信息（属性信息，位置，技能冷却时间）
- 环境的种类：部分可观察的、随机性、多Agent、序贯的、连续的
- 动作库：
 - 移动到指定地点（无条件）
 - 进行普通攻击（无条件） - 释放技能攻击（技能不再冷却时间内）
 - 购买装备（拥有金币高于所要购买的装备价格）
- 设计目标：获得游戏的胜利

课后作业2-2

- 证明下面的问题：

（1）对于每一个纯反应式Agent，存在一个行为等价的标准Agent。

（2）存在标准Agent，没有与之行为等价的纯反应式Agent。

参考答案2-2

(1)

形式上，纯反应式 **Agent** 的行为可以表示成函数：

$$Ag: E \rightarrow Ac$$

标准**Agent**的行为可以表示成函数：

$$Ag: \mathcal{R}^E \rightarrow Ac$$

已知 \mathcal{R}^E 是以状态结束的序列所组成的 \mathcal{R} 的子集，所以
 $E \subseteq \mathcal{R}^E$ 。

得证对于每一个纯反应式**Agent**，存在一个行为等价的标准**Agent**。

参考答案2-2

(2)

标准Agent的行为可以表示成函数：

$$Ag: \mathcal{R}^E \rightarrow \mathcal{A}c$$

设存在标准Agent记为 Ag_i ，其行为可表示为：

$$Ag_i: E_{t-1}, \mathcal{A}c_{t-1}, E_t \rightarrow \mathcal{A}c_t$$

而没有纯反应式Agent可表示 Ag_i ，

得证存在标准Agent，没有与之行为等价的纯反应式Agent。

课后作业2-3

- 有两种方法通过效用函数定义任务：通过效用与状态的关系（ $u: E \rightarrow \mathbb{R}$ ）或者通过效用与运行的关系（ $u: \mathcal{R} \rightarrow \mathbb{R}$ ）。严格来说，第二种效用函数比第一种效用函数有更强的表达能力。给出一个关于运行的效用函数的例子，这个效用函数不能通过与状态有关的效用来定义。

参考答案2-3

统计水果忍者游戏内，成功切割的水果数占出现的总水果数的比例。
即：

$$u(r) \triangleq \frac{\text{在一回合游戏中成功切割的水果数}}{\text{在一回合游戏中总共出现的水果数}}$$

因为多个连续状态中的成功切割水果数和出现水果数存在联系，所以我们无法从单个状态中统计成功切割水果数和出现水果数。故只能使用关于运行的效用函数，而不能使用关于状态的效用函数。



课后作业2-4

- 考虑环境 $Env_1 = \langle E, e_0, \tau \rangle$ ，具体定义如下：

$$E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6\}$$

$$\tau(e_0 \xrightarrow{\alpha_0}) = \{e_1, e_2, e_3\}$$

$$\tau(e_0 \xrightarrow{\alpha_1}) = \{e_4, e_5, e_6\}$$

这个环境中有两个可能的Agent:

$$Ag_1(e_0) = \alpha_0$$

$$Ag_2(e_0) = \alpha_1$$

不同运行的概率：

$$P(e_0 \xrightarrow{\alpha_0} e_1 \mid Ag_1, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_0} e_2 \mid Ag_1, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_0} e_3 \mid Ag_1, Env_1) = 0.6$$

$$P(e_0 \xrightarrow{\alpha_1} e_4 \mid Ag_2, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_1} e_5 \mid Ag_2, Env_1) = 0.3$$

$$P(e_0 \xrightarrow{\alpha_1} e_6 \mid Ag_2, Env_1) = 0.5$$

假设效用函数 u_1 的定义如下：

$$u_1(e_0 \xrightarrow{\alpha_0} e_1) = 8$$

$$u_1(e_0 \xrightarrow{\alpha_0} e_2) = 7$$

$$u_1(e_0 \xrightarrow{\alpha_0} e_3) = 4$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_4) = 8$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_5) = 2$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_6) = 5$$

给定这些定义，求 Ag_1 和 Ag_2 关于 Env_1 和 u_1 的期望效用，并解释哪个Agent是关于 Env_1 和 u_1 的最优Agent。

参考答案2-4

Ag0的期望效用

$$A_{g0} = (0.2 \times 8) + (0.2 \times 7) + (0.6 \times 4) = 1.6 + 1.4 + 2.4 = 5.4$$

Ag1的期望效用

$$A_{g1} = (0.2 \times 8) + (0.3 \times 2) + (0.5 \times 5) = 1.6 + 0.6 + 2.5 = 4.7$$

因此Agent 0是关于Env1和u1的最优Agent

课后作业2-6

- 在真空吸尘器世界的例子中，函数new给出了把谓词加入Agent的数据库中的定义，给出函数new的完整定义（如果需要可以使用伪代码）。

参考答案2-6

function *database* (*DB*, *Per*) returns an database *DB*

begin

 initialize the *DB* to an empty set

 if $\alpha == forward$ then

 update the predicate *In*(*x*, *y*) and add to *DB*

 if *Per* == *dirt* then

 add *Dirt*(*x*, *y*) to *DB*

 if $\alpha == turn$ then

 update the predicate *Facing*(*d*) and add to *DB*

 return *DB*

end function *database*

课后作业2-7

- 通过给出真空吸尘器世界例子中的默认规则，完成这个例子。你认为解决方案是否直观？是否优美？是否紧凑？

参考答案2-7

解决方案直观，因为每条规则表意清晰，简单易懂；
不优美，因为规则数目众多，且内容与结构相似；不
紧凑，因为存在更简洁的表示方法，默认规则没有考
虑部分规则间的共性加以总结提炼。

课后作业2-8

- 把真空吸尘器世界的规模扩大到 10×10 个方格的规模。使用上面给出的方法，大致需要多少条规则对这个放大的例子进行编码？对规则进行一般化处理，实现一个更一般的决策机制。

参考答案2-8

有 $1 + 2 * (9 * 10 + 9 * 1) + 4 * 18 + 2 = 273$ 条规则。

更一般的决策机制如下：

① 清理垃圾

$$\text{In}(x, y) \wedge \text{Dirt}(x, y) \rightarrow \text{Do}(\text{suck})(1)$$

② 从南到北走，如面朝北的 $(0,0) \rightarrow$ 面朝南的 $(1,9)$

$$\text{In}(x, y) \wedge \text{Facing}(\text{north}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{forward})(x \% 2 = 0, y < 9) \quad (2)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{north}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 0, y = 9) \quad (3)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{east}) \rightarrow \text{Do}(\text{forward})(x \% 2 = 0, y = 9) \quad (4)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{east}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 1, y = 9) \quad (5)$$

参考答案2-8

③ 从北到南走，如面朝南的(1,9) → 面朝北的(2,0)

$$\text{In}(x, y) \wedge \text{Facing}(\text{south}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{forward})(x \% 2 = 1, y > 0) \quad (6)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{south}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 1, x < 9, y = 0) \quad (7)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{west}) \rightarrow \text{Do}(\text{turn})(x \% 2 = 1, x < 9, y = 0) \quad (8)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{north}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 1, x < 9, y = 0) \quad (9)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{east}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{forward})(x \% 2 = 1, x < 9, y = 0) \quad (10)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{east}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 0, x < 9, y = 0) \quad (11)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{south}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 0, x < 9, y = 0) \quad (12)$$

$$\text{In}(x, y) \wedge \text{Facing}(\text{west}) \wedge \neg \text{Dirt}(x, y) \rightarrow \text{Do}(\text{turn})(x \% 2 = 0, x < 9, y = 0) \quad (13)$$

达到(9,0)后，按照原路返回至(0,0)，逻辑与上述规则一致，不再枚举。

课后作业2-9

- 考虑下图中的并发MetateM程序，解释Agent在这个系统中的行为。

提示：有5个Agent:

- ❑ SnowWhite是资源的提供者。可以把她想象成白雪公主，手中有糖
- ❑ eager, greedy, courteous, shy是资源的消费者。可以把他们想象成不同类型的小矮人

```
SnowWhite(ask)[give]:  
    ●ask(x) ⇒ ◇give(x)  
    give(x) ∧ give(y) ⇒ (x = y)  
eager(give)[ask]:  
    start ⇒ ask(eager)  
    ●give(eager) ⇒ ask(eager)  
greedy(give)[ask]:  
    start ⇒ □ask(greedy)  
courteous(give)[ask]:  
    ((¬ask(courteous) S give(eager)) ∧  
    (¬ask(courteous) S give(greedy))) ⇒ ask(courteous)  
shy(give)[ask]:  
    start ⇒ ◇ask(shy)  
    ●ask(x) ⇒ ¬ask(shy)  
    ●give(shy) ⇒ ◇ask(shy)
```

参考答案2-9

Snow White

- **Snow White** 有一些资源，她要将这些资源分发给**Dwarves**。
- 她一次只给一个**dwarf**资源
- 她会一直讲资源分配给任意一个请求资源的**dwarf**

下面是**Snow White**的并发**Metate**程序

Snow-White(ask)[give]:

⊙ ask(x) \Rightarrow \diamond give(x)

give(x) \wedge give(y) \Rightarrow (x = y)

Eager

- **Eager**初始时会请求资源，然后每当他收到资源，就再次请求资源。

eager(give)[ask]:

start \Rightarrow ask(eager)

⊙ give(eager) \Rightarrow ask(eager)

Greedy

- **Greedy**一直请求资源

greedy(give)[ask]:
start $\Rightarrow \square \text{ask}(\text{greedy})$

Courteous

- **Courteous**比较有礼貌，他只会在 ‘eager’ 和 ‘greedy’ 得到资源后请求资源。

courteous(give)[ask]:
 $((\neg \text{ask}(\text{courteous}) \mathcal{S} \text{give}(\text{eager})) \wedge$
 $(\neg \text{ask}(\text{courteous}) \mathcal{S} \text{give}(\text{greedy}))) \Rightarrow$
 $\text{ask}(\text{courteous})$

Shy

- **Shy**只会在其他人没有请求时请求资源。

shy(give)[ask]:

$start \Rightarrow \diamond \text{ask}(\text{shy})$

$\odot \text{ask}(x) \Rightarrow \neg \text{ask}(\text{shy})$

$\odot \text{give}(\text{shy}) \Rightarrow \diamond \text{ask}(\text{shy})$

课后作业2-10

- 回忆在2.3节中讨论的真空吸尘器的例子，使用STRIPS的符号形式化表示这个Agent可提供的操作。

参考答案2-10

	Suck	Turn
Pre	$\text{Dirt}(x,y) \wedge \text{In}(x,y)$	Facing(north) (以north 为例, 其他方向类似)
Del	$\text{Dirt}(x,y)$	Facing(north)
Add	$\neg \text{Dirt}(x,y)$	Facing(east)

	Forward	Forward
Pre	$\text{In}(x,y) \wedge \text{Facing}(\text{north}) \wedge \neg \text{In}(x,2)$	$\text{In}(x,y) \wedge \text{Facing}(\text{east}) \wedge \neg \text{In}(2,y)$
Del	$\text{In}(x,y)$	$\text{In}(x,y)$
Add	$\text{In}(x,y+1)$	$\text{In}(x+1,y)$

参考答案2-10

	Forward	Forward
Pre	$\text{In}(x,y) \wedge \text{Facing}(\text{south}) \wedge \neg \text{In}(x,0)$	$\text{In}(x,y) \wedge \text{Facing}(\text{west}) \wedge \neg \text{In}(0,y)$
Del	$\text{In}(x,y)$	$\text{In}(x,y)$
Add	$\text{In}(x,y-1)$	$\text{In}(x-1,y)$

课后作业2-11

- 考虑用如下谓词描述的积木世界：

谓词	含义
$\text{On}(x, y)$	物体 x 在物体 y 之上
$\text{OnTable}(x)$	物体 x 在桌面上
$\text{Clear}(x)$	物体 x 上没有东西
$\text{Holding}(x)$	机械臂拿着物体 x
ArmEmpty	机械臂为空

Agent关于积木 A 、 B 、 C 的初始信念 B_0 和意图 i 为：

$Beliefs\ B_0$	$Intention\ i$
$\text{Clear}(B)$	$\text{Clear}(A)$
$\text{Clear}(C)$	$\text{Clear}(B)$
$\text{On}(C, A)$	$\text{On}(B, C)$
$\text{OnTable}(A)$	$\text{OnTable}(A)$
$\text{OnTable}(B)$	$\text{OnTable}(C)$
ArmEmpty	ArmEmpty

Agent有一个动作集合：

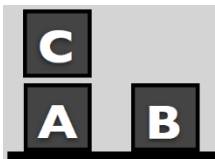
$$Ac = \{\text{Stack}, \text{Unstack}, \text{Pickup}, \text{PutDown}\}$$

$Stack(x, y)$	
pre	$Clear(y) \ \& \ Holding(x)$
del	$Clear(y) \ \& \ Holding(x)$
add	$ArmEmpty \ \& \ On(x, y)$
$UnStack(x, y)$	
pre	$On(x, y) \ \& \ Clear(x) \ \& \ ArmEmpty$
del	$On(x, y) \ \& \ ArmEmpty$
add	$Holding(x) \ \& \ Clear(y)$
$Pickup(x)$	
pre	$Clear(x) \ \& \ OnTable(x) \ \& \ ArmEmpty$
del	$OnTable(x) \ \& \ ArmEmpty$
add	$Holding(x)$
$PutDown(x)$	
pre	$Holding(x)$
del	$Holding(x)$
add	$OnTable(x) \ \& \ ArmEmpty \ \& \ Clear(x)$

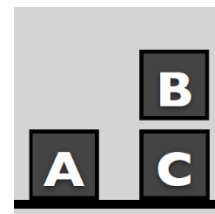
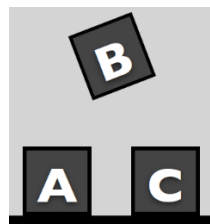
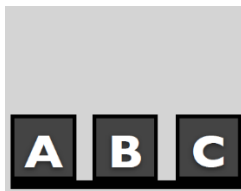
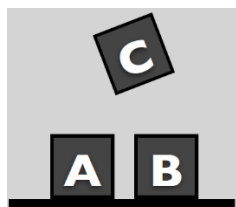
给定初始信念 B_0 和意图 i ，计算一个规划 π 。画出该规划开始时的环境，以及每次执行了动作后的环境。

参考答案2-11

- 规划 $\pi = (Unstack(C, A), PutDown(C), Pickup(B), Stack(B, C))$
- 规划开始时的环境为：



- 依次执行规划 π 中每个动作后的环境为：



课后作业2-12

- 以下的伪代码为实用推理（BDI）Agent定义了一个控制回路：

```
1.
2.   $B := B_0$ ;
3.   $I := I_0$ ;
4.  while true do
5.    get next percept  $\rho$ ;
6.     $B := brf(B, \rho)$ ;
7.     $D := options(B, I)$ ;
8.     $I := filter(B, D, I)$ ;
9.     $\pi := plan(B, I)$ ;
10.   while not ( $empty(\pi)$  or  $succeeded(I, B)$  or  $impossible(I, B)$ ) do
11.      $\alpha := hd(\pi)$ ;
12.      $execute(\alpha)$ ;
13.      $\pi := tail(\pi)$ ;
14.     get next percept  $\rho$ ;
15.      $B := brf(B, \rho)$ ;
16.     if  $reconsider(I, B)$  then
17.        $D := options(B, I)$ ;
18.        $I := filter(B, D, I)$ ;
19.     end-if
20.     if not  $sound(\pi, I, B)$  then
21.        $\pi := plan(B, I)$ 
22.     end-if
23.   end-while
24. end-while
```

参照此伪代码，解释以下组件的用途/作用：

- a) 变量 B 、 D 和 I
- b) 感知 ρ
- c) $brf(...)$ 函数
- d) $options(...)$ 函数
- e) $filter(...)$ 函数
- f) $plan(...)$ 函数
- g) $sound(...)$ 函数
- h) $succeeded(...)$ 函数和 $impossible(...)$ 函数
- i) $reconsider(...)$ 函数——在回答这部分问题的时候，你应该要明确说明此函数应具有的属性，以及可以假定其正常运行的情况。

参考答案2-12

- a) B: 信念, Agent对当前环境的理解; D: 愿望, Agent希望发生的最终结果;
I: 意图, 根据信念与愿望和上一时刻意图筛选得到。
- b) p: 对环境的感知
- c) brf: 信念修正函数, 根据过去的信念和新的感知, 对信念集合进行修正
- d) options: 选择生成函数, 根据信念与意图, 生成可选的愿望集合。
- e) filter: 过滤函数, 从多个选项中选择出最佳选择
- f) plan: 规划函数, 根据当前信念与意图生成规划
- g) sound: 判断给定B时, π 是否是实现I的正确规划
- h) succeeded: 判断当前意图是否已经实现, impossible:判断当前意图是否不再可能实现
- i) reconsider: 决定是否需要重新考虑意图

课后作业2-13

- 用Brooks的归类式结构的方法求解第2.3节描述的真空吸尘器的例子。

参考答案2-13

按照优先级从高到低定义归类式体系结构的“情景，动作”规则：

1.if 所在区域存在灰尘

then 吸尘

2. if 前方为边界

then 转向

3. 向前移动

课后作业2-14

- 用第2.3节描述的基于逻辑的方法求解探测火星的例子。

参考答案2-14

火星探测Agent

目标：携带尽量多的样本送到母舰上

内部状态（知识库DB）：

Obstacle():附近有障碍物

Scrap():附近有碎屑

Base():在基地

Sample():携带样本

动作集合：

Turn:改变方向

Pickup:捡起样本

Pickdown: 放下样本

Randomwalk: 随机移动

Moveup: 沿着梯度上升的方向走

Movedown: 沿着梯度下降的方向走

Throw: 扔下两个碎屑

Collect: 收集一个碎屑

感知函数see:

Agent附近是否有障碍物

Agent附近是否有碎屑

Agent是否在基地

Agent附近是否有样本

状态转移函数next:

$next(DB,p) = (DB \setminus old(DB)) \cup new(DB,p)$

参考答案2-14

演绎规则:

$$Obstacle() \rightarrow Do(Turn)$$

$$\neg Obstacle() \wedge Sample() \wedge Base() \rightarrow Do(Pickdown)$$

$$\neg Obstacle() \wedge Sample() \wedge \neg Base() \rightarrow Do(Throw) \wedge Do(Moveup)$$

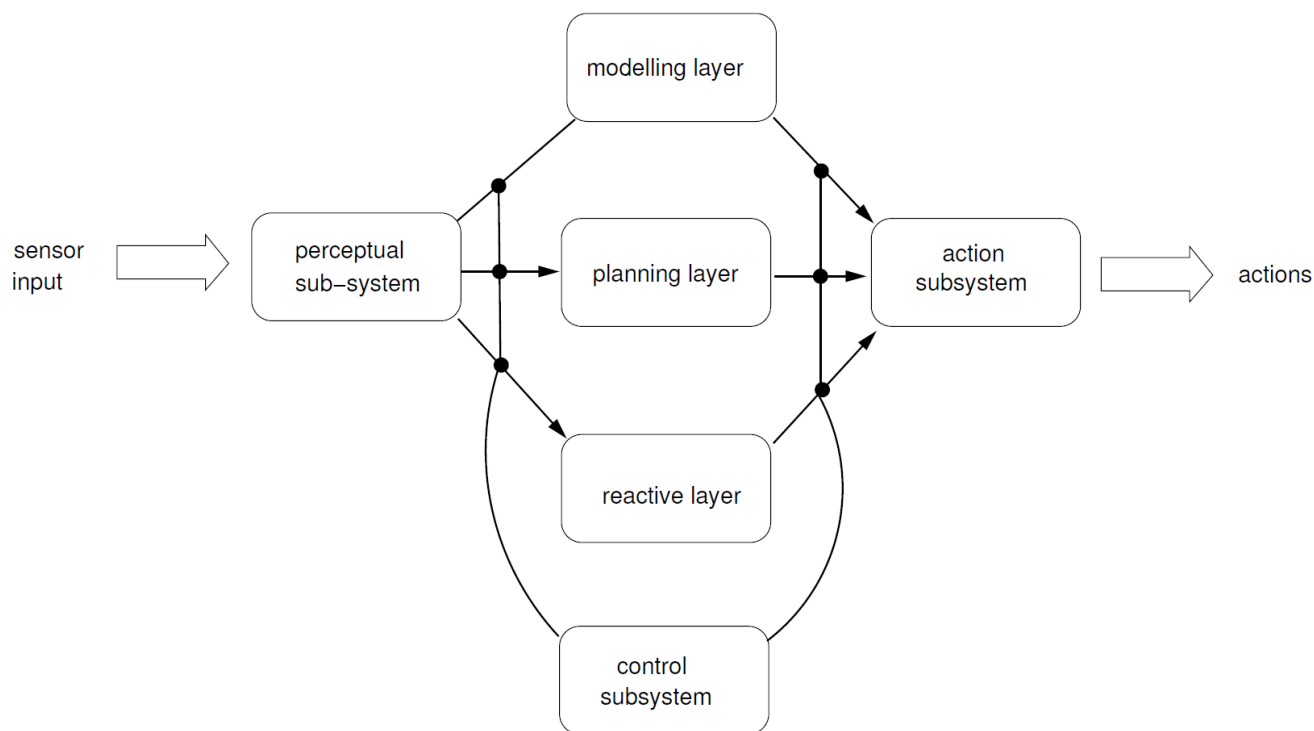
$$\neg Obstacle() \wedge Sample() \rightarrow Do(Pickup)$$

$$\neg Obstacle() \wedge Scrap() \rightarrow Do(Collect) \wedge Do(Movedown)$$

$$\neg Obstacle() \wedge \neg Sample() \wedge \neg Base() \wedge \neg Scrap() \rightarrow Do(Randomwalk)$$

课后作业2-15

- 下列示意图阐明了Touring机体系统结构的关键子系统：



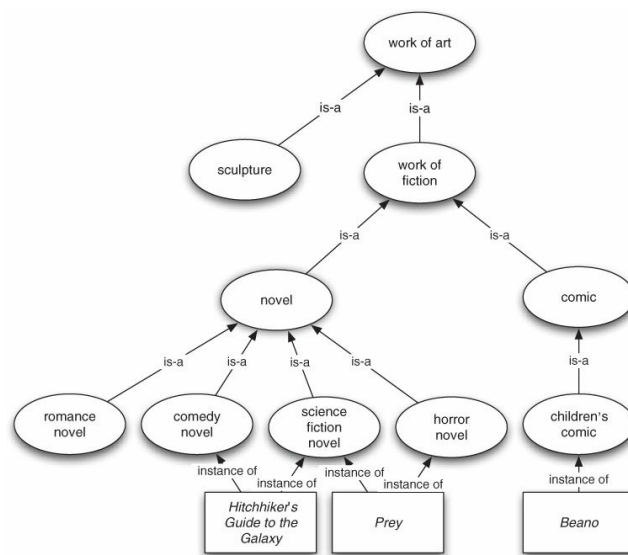
描述这个系统结构的各个部分，包括解释其中三个决策层如何实现产生反应式行为和预动行为的目标。

参考答案2-15

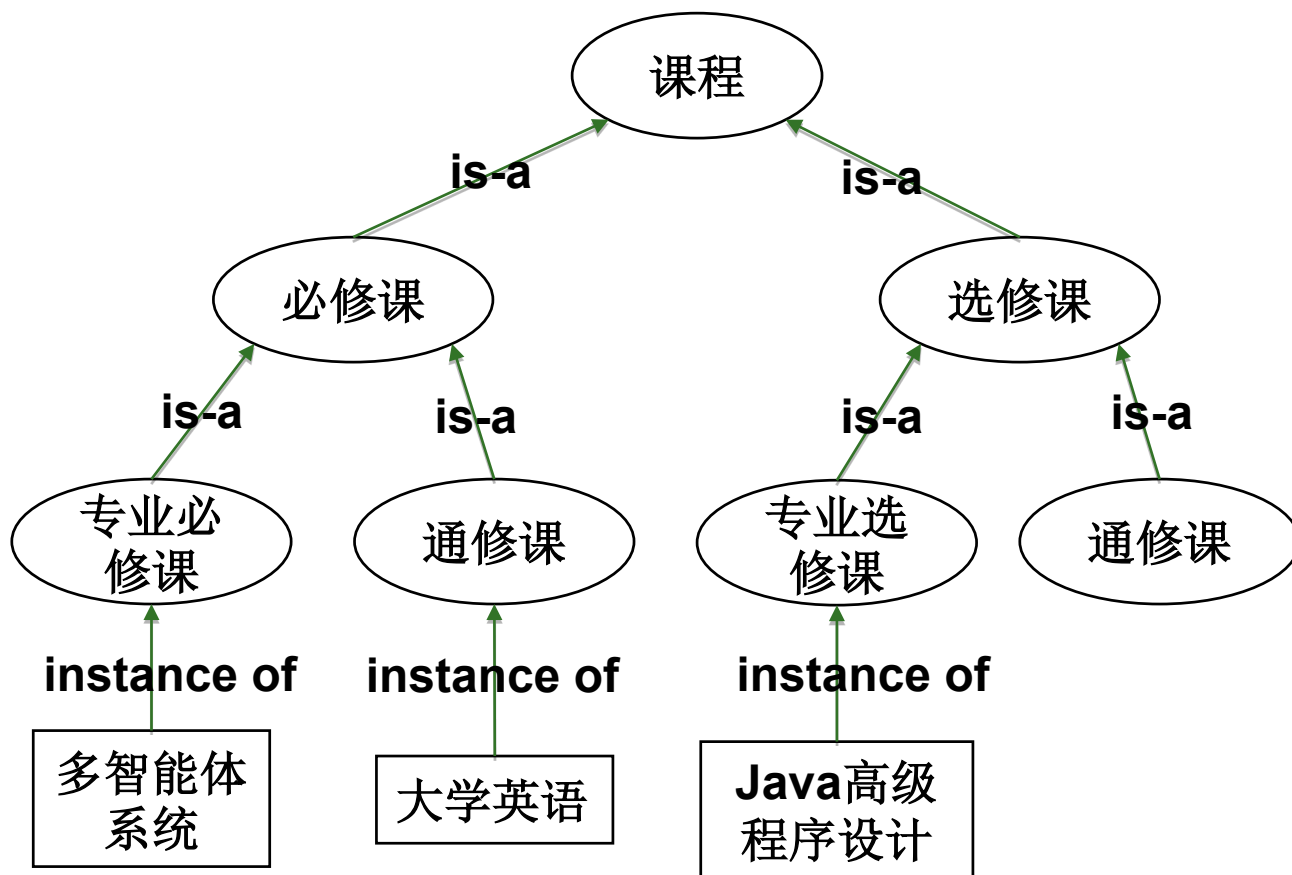
- Turing机是一个混合架构的例子，它将反应性行为与慎思行为相结合。它由三层组成，每个层并行运行，每一层都可以访问感知子系统，该子系统负责将从传输器输入获得的感知转换为可用于推理的谓词。此外，每一层都可以生成随后可以执行的操作。控制子系统负责监控感知子系统中接收到的感知信号，然后确定哪些动作（如果有的话）应该从不同的层次执行。即决定哪一层控制Agent。特别地，控制子系统可以抑制传感器信息到特定的层，或者它可以审查不同层产生的动作。
- 反应层负责响应环境的变化。它定义了一组situation-action规则，如果这些规则映射到传感器输入，则触发这些规则。例如，如果Agent控制一辆自动驾驶车辆，并且它意外地检测到前面的路缘，它可以停止（或减速）并转向避开路缘。
- 规划层负责确定实现Agent目标所需的操作，在正常操作下，这一层决定Agent应该做什么，这是通过一套计划实现的。
- 建模层表示世界上的各种实体。它负责模拟世界，包括其他Agent，以及用于确定Agent目标，或者如果检测到与其他Agent的任何冲突，则用于解决这些冲突的计划目标。每当生成一个目标时，它就会被传递到规划层，由规划层决定最后的行动。

课后作业3-1

- 使用课上所学的本体设计方法构建一个本体。使用类似课件中的框图表示类别层次和实例关系，并描述相关的属性和至少2个可能的属性约束（可用自然语言表达）。应满足的需求有：
 - 课程，可以分为必修课与选修课，其中必修课分为专业必修课和通修课，选修课分为专业选修课和公选课。课程的属性包括课程编号、授课老师、选课人数等等。
 - 《多智能体系统》是专业必修课
 - 《大学英语》是通修课
 - 《Java高级程序设计》是专业选修课
- 内容可根据上述需求进行补充。属性约束请自行设计，约束符合常理即可。



参考答案3-1



属性:

课程类具有课程编号、授课教师、选课人数三个属性。

属性约束:

如, 每个课程只能有一个课程编号, 每个课程的选课人数必须是非负整数

属性约束符合常理, 且属于课件中属性约束的任何一类(势约束、类别约束、范围约束、领域约束)即可。

课后作业3-2

- 利用课后作业3-1中设计的本体，使用KQML语言描述一段对话：
 - A向B查询（evaluate）《多智能体系统》课程的选课人数情况。
 - B告知A《多智能体系统》选课人数为30人。
- 需要描述清楚每条消息的语用词和参数，你可以使用自然语言表达消息内容。

参考答案3-2

```
(evaluate  
  :sender A  
  :receiver B  
  :ontology NJUCourse  
  :reply-with q1  
  :content 多智能体系统 选课人数  
)
```

```
(reply  
  :sender B  
  :receiver A  
  :ontology NJUCourse  
  :in-reply-to q1  
  :content 多智能体课程 选课人数 30  
)
```

消息内容可用自然语言描述，符合题意即可。用其他语言描述，含义正确也可。

课后作业3-3

- 利用课后作业3-1中设计的本体，使用KQML语言描述一段对话：
 - A向B查询《机器学习导论》课程的所有信息。
 - B告知A《机器学习导论》的课程编号为30000150。
 - B告知A《机器学习导论》的授课老师为周志华老师。
 - B告知A《机器学习导论》选课人数为300人。
 - B告知A当前通信流结束。
- 需要描述清楚每条消息的语用词和参数，你可以使用自然语言表达消息内容。

参考答案3-3

```
(stream-about
  :sender A
  :receiver B
  :ontology NJUCourse
  :reply-with q2
  :content 机器学习导论 所有信息
)

(tell
  :sender B
  :receiver A
  :ontology NJUCourse
  :in-reply-to q2
  :content 机器学习导论 课程编号 30000150
)

(tell
  :sender B
  :receiver A
  :ontology NJUCourse
  :in-reply-to q2
```

```
  :content 机器学习导论 授课老师 周志华
)

(tell
  :sender B
  :receiver A
  :ontology NJUCourse
  :in-reply-to q2
  :content 机器学习导论 选课人数 300
)

(eos
  :sender B
  :receiver A
  :in-reply-to q2
)
```

消息内容可用自然语言描述，符合题意即可。用其他语言描述，含义正确也可。

课后作业3-4

- 使用自然语言，描述下面两段KQML消息的含义

```
(a) (ask-if
      :sender      A
      :receiver    B
      :language    OWL
      :ontology    pizza
      :reply-with  q1
      :content ( (margherita isa Pizza)
                  (margherita hasTopping mozzarella) )
    )
```

```
(b) (tell
      :sender      A
      :receiver    B
      :language    OWL
      :ontology    pizza
      :reply-with  q1
      :content (not (hawaiian isa ItalianPizza))
    )
```

参考答案3-4

(a) A向B询问Margherita是否是一种比萨，以及Margherita是否使用Mozzarella制作。

(b) A告诉B：Hawaiian不是一种比萨。

消息内容描述大致正确即可。

课后作业3-5

- 使用KQML语言，描述如何实现合同网协议。

参考答案3-5

合同网包含五个阶段：识别、通知、竞标、授予、实现，其中识别和实现阶段无法使用KQML语言描述，因此仅描述合同网的通知、竞标、授予阶段。

假设有三个Agent：A，B，C，其中A通知B，C，B，C向A竞标，A最终选择向B授予任务。

通知：

```
(tell
  :sender A
  :receiver B
  :ontology cnet
  :reply-with q1
  :content 任务内容
)
(tell
  :sender A
  :receiver C
  :ontology cnet
  :reply-with q2
  :content 任务内容
)
```


参考答案3-5

竞标:

(tell

:sender B

:receiver A

:ontology cnet

:in-reply-to q1

:reply-with q3

:content 投标

)

(tell

:sender C

:receiver A

:ontology cnet

:in-reply-to q2

:reply-with q4

:content 投标

)

授予:

(reply

:sender A

:receiver B

:ontology cnet

:in-reply-to q3

:content 竞标成功, 授予任务

)

(reply

:sender A

:receiver C

:ontology cnet

:in-reply-to q4

:content 竞标失败

)