

人工智能导论

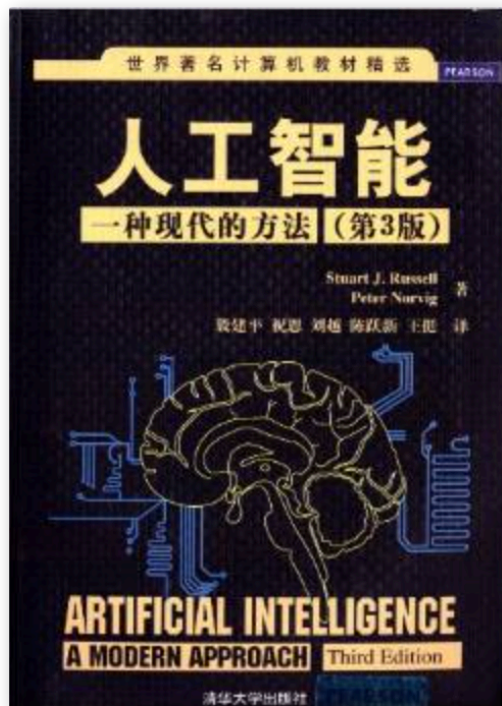
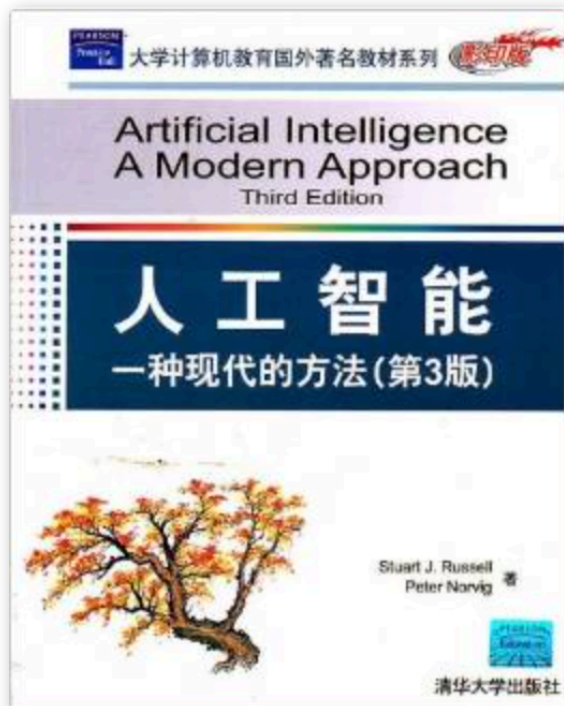
<https://www.lamda.nju.edu.cn/introAI>

课程讨论QQ群： 698776029

教材

课程名称：人工智能

教材：AIMA



<http://aima.cs.berkeley.edu/>

课程主页



时间： 周三10:10-12:00 逸A-117

课程主页：

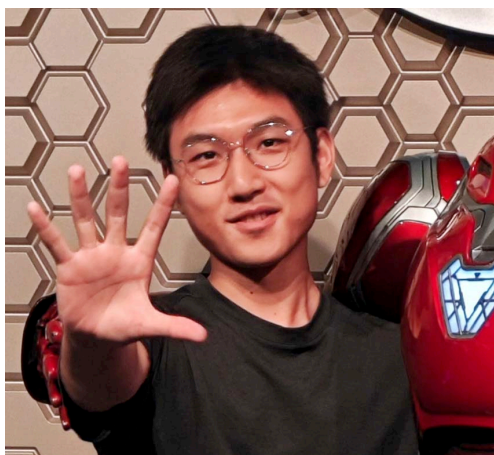
<http://www.lamda.nju.edu.cn/IntroAI>

助教



王鹏远

<http://www.lamda.nju.edu.cn/wangpy>



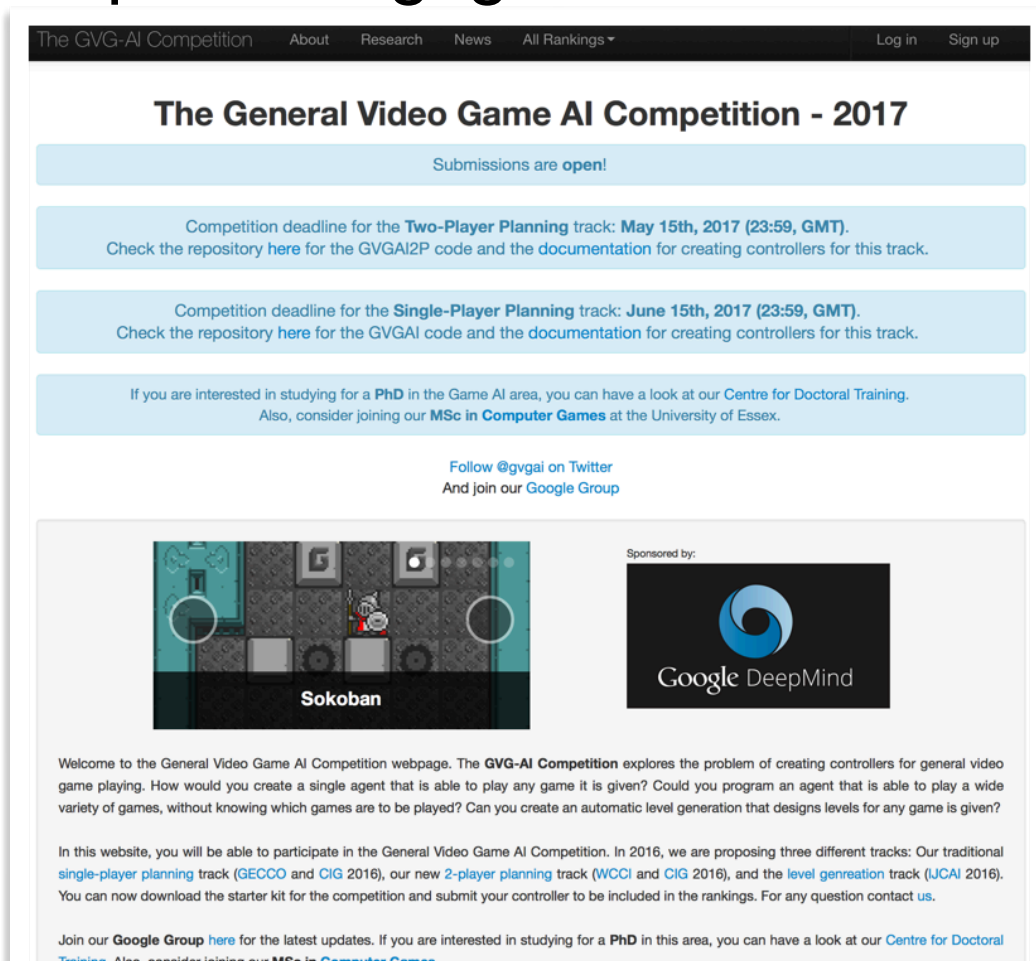
李立和

<http://www.lamda.nju.edu.cn/lilh>

本次课程有五次作业--让计算机自己玩游戏

将基于GVGAI框架，请开始熟悉该框架：

<http://www.gvgai.net>



The screenshot shows the homepage of the GVGAI 2017 competition. The header includes navigation links: 'The GVG-AI Competition', 'About', 'Research', 'News', 'All Rankings', 'Log in', and 'Sign up'. The main heading is 'The General Video Game AI Competition - 2017'. Below this, a blue box states 'Submissions are open!'. Two more blue boxes provide deadlines: 'May 15th, 2017 (23:59, GMT)' for the Two-Player Planning track and 'June 15th, 2017 (23:59, GMT)' for the Single-Player Planning track. A light blue box mentions PhD and MSc opportunities at the University of Essex. Social media links for Twitter and Google Group are provided. A game preview for 'Sokoban' is shown with a screenshot of the game and the Google DeepMind logo. The footer contains a welcome message and information about the competition tracks and starter kit.

The GVG-AI Competition About Research News All Rankings Log in Sign up

The General Video Game AI Competition - 2017


Submissions are open!

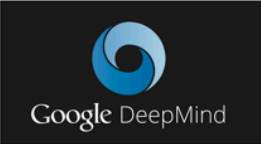
Competition deadline for the **Two-Player Planning** track: **May 15th, 2017 (23:59, GMT)**.
Check the repository [here](#) for the GVGAI2P code and the documentation for creating controllers for this track.

Competition deadline for the **Single-Player Planning** track: **June 15th, 2017 (23:59, GMT)**.
Check the repository [here](#) for the GVGAI code and the documentation for creating controllers for this track.

If you are interested in studying for a **PhD** in the Game AI area, you can have a look at our [Centre for Doctoral Training](#).
Also, consider joining our **MSc in Computer Games** at the University of Essex.

[Follow @gvgai on Twitter](#)
And join our [Google Group](#)

 Sokoban

Sponsored by:


Welcome to the General Video Game AI Competition webpage. The **GVG-AI Competition** explores the problem of creating controllers for general video game playing. How would you create a single agent that is able to play any game it is given? Could you program an agent that is able to play a wide variety of games, without knowing which games are to be played? Can you create an automatic level generation that designs levels for any game is given?

In this website, you will be able to participate in the General Video Game AI Competition. In 2016, we are proposing three different tracks: Our traditional **single-player planning** track (GECCO and CIG 2016), our new **2-player planning** track (WCCI and CIG 2016), and the **level generation** track (JCAI 2016). You can now download the starter kit for the competition and submit your controller to be included in the rankings. For any question contact [us](#).

Join our [Google Group](#) [here](#) for the latest updates. If you are interested in studying for a **PhD** in this area, you can have a look at our [Centre for Doctoral Training](#). Also, consider joining our **MSc in Computer Games**.

平时作业 与 期末考核

作业1: Bait游戏 - 关于搜索

作业2: 黑白棋游戏 - 关于博弈

作业3: Aliens游戏- 关于监督学习

作业4: Freeway游戏 - 关于强化学习

作业5: Mini AlphaGo - 关于强化学习和博弈

期末论文：

1. 寻找一个人工智能应用问题
2. 调研该问题现有解决方案
3. 对问题进行抽象，提出一种不同的解决方案，描述你的解决方案采用的AI技术
4. 分析现有技术的局限，提出具体的技术挑战

考核评分

- 前4次每次作业 每次20%
- 期末论文20%
- 在完成前4次作业的基础上，
完成第五次作业获得额外10分

俞扬

www.lamda.nju.edu.cn/yuy

yuy@nju.edu.cn

章宗长

<https://ai.nju.edu.cn/zhangzongzhang/>

zzzhang@nju.edu.cn



南京大學
NANJING UNIVERSITY

Lecture 1: Introduction

What is artificial intelligence?



1956 Dartmouth meeting: “Artificial Intelligence”

John McCarthy:

“ It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.”



1927-2011

Marvin Minsky:

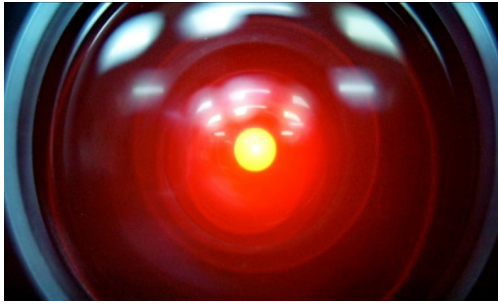
“ to make computers be capable of doing things that when done by a human, would be thought to require intelligence ”



1927-2016

we will discuss the concept and the history of AI in the last class

What we call AI in movies



2001: A Space Odyssey
1968



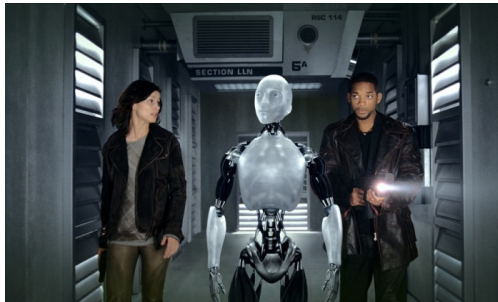
The Matrix
1999



A.I. Artificial Intelligence
2001



Wall-E
2008



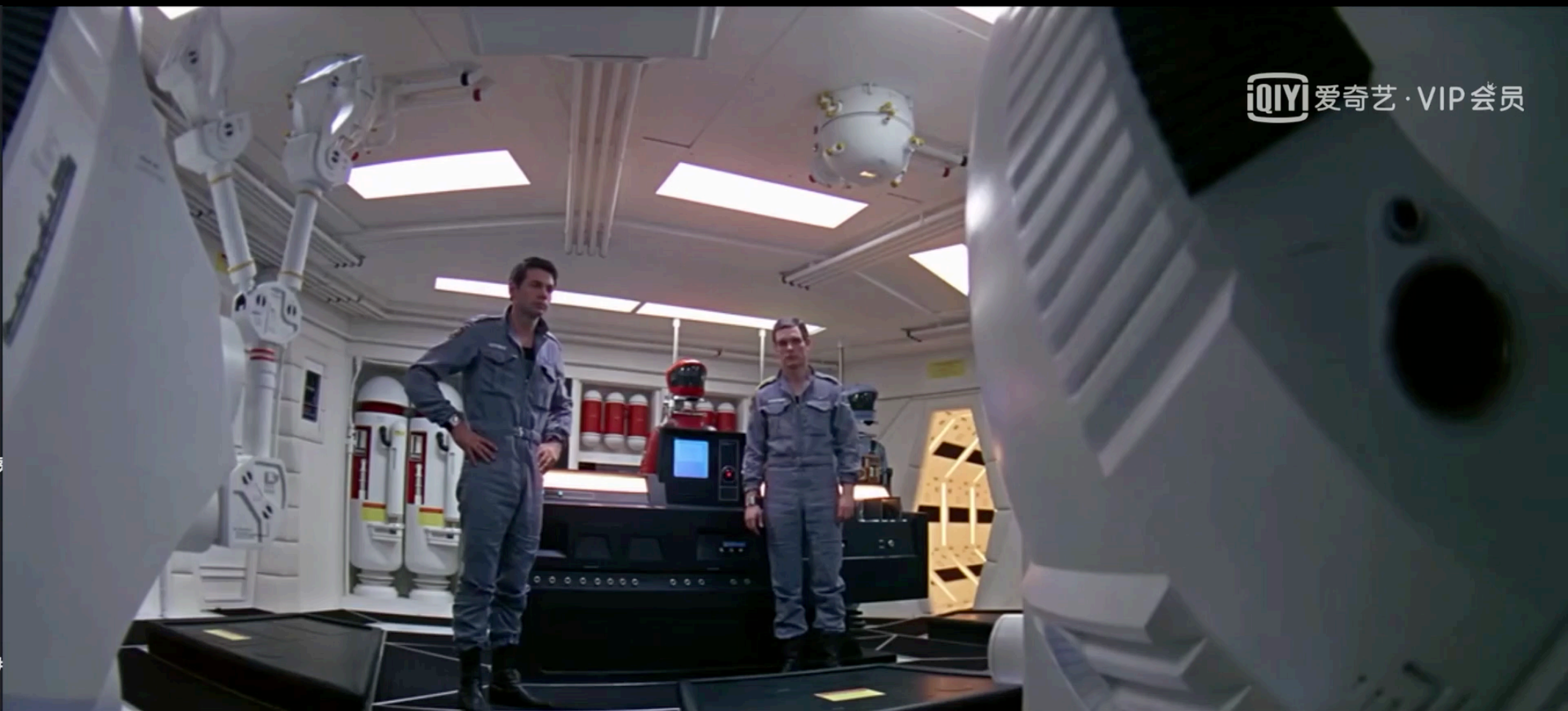
I, Robot
2004



The Terminator
1984



Interstellar
2014



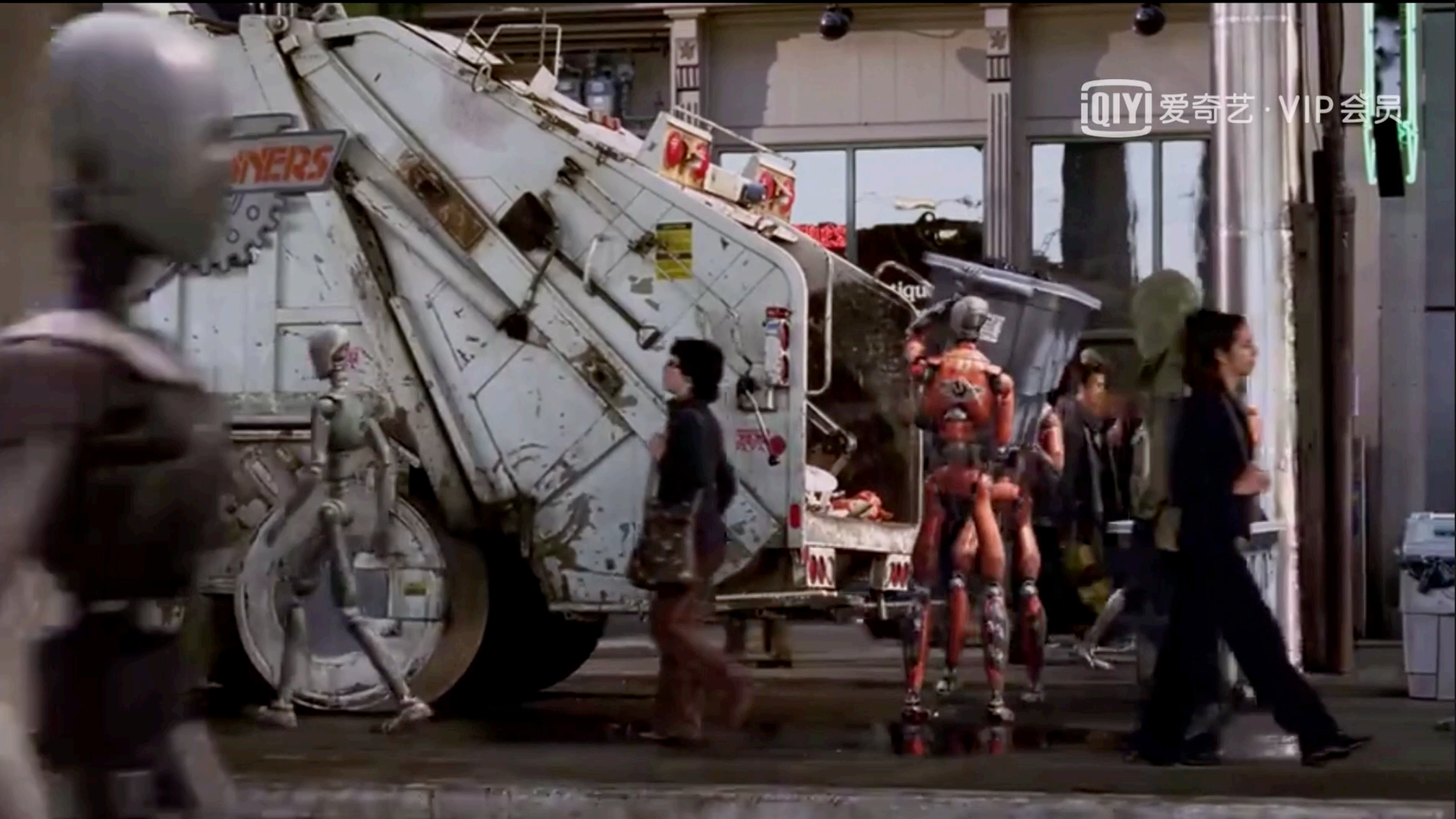
LipNet: Sentence-level lipreading

Yannis M. Assael, Brendan Shillingford
Shimon Whiteson, Nando de Freitas

We thank University of Sheffield, J. Barker, M. Cooke, S. Cunningham and X. Shao, for the GRID dataset.

电影：I, Robot (2004)

iQIYI 爱奇艺 · VIP 会员



Boston Dynamics 2018

腾讯视频



What AI we do have



人脸检测、识别



S.I.R.I.



自动驾驶



推荐系统



下棋

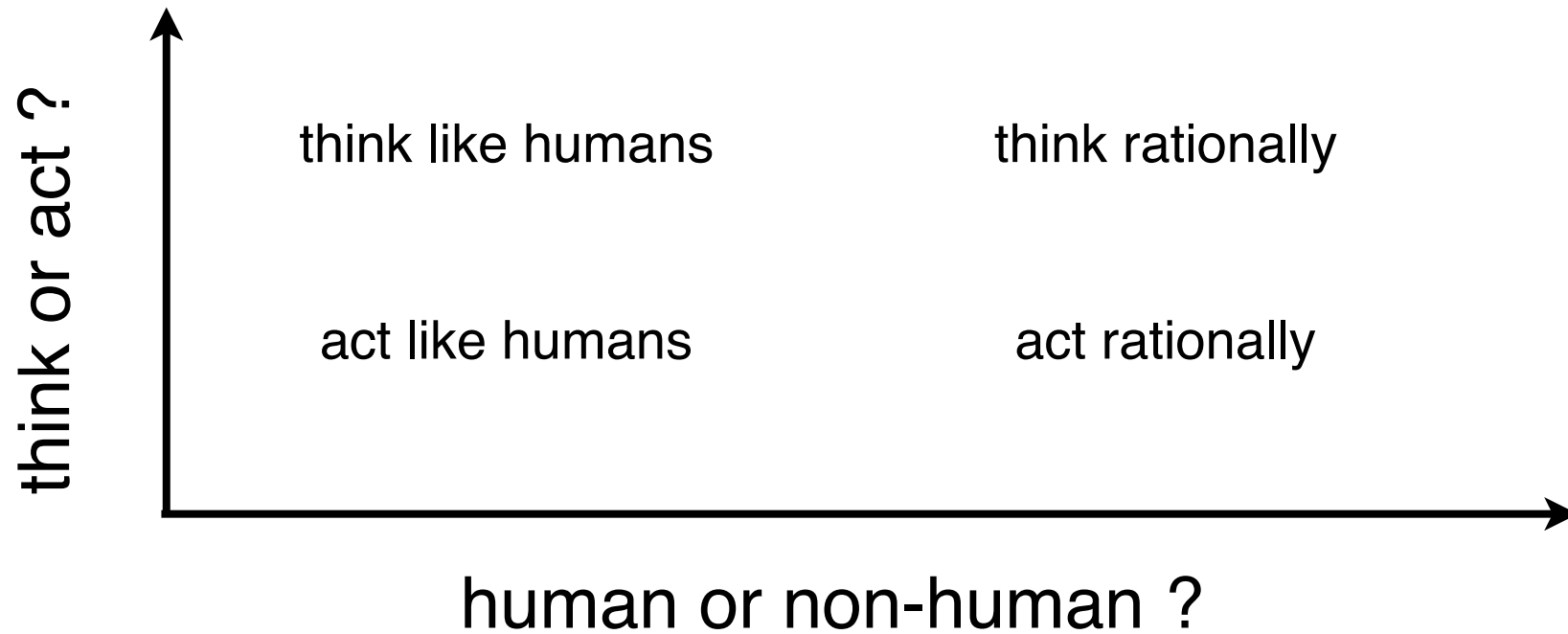


BigDog

What is AI?



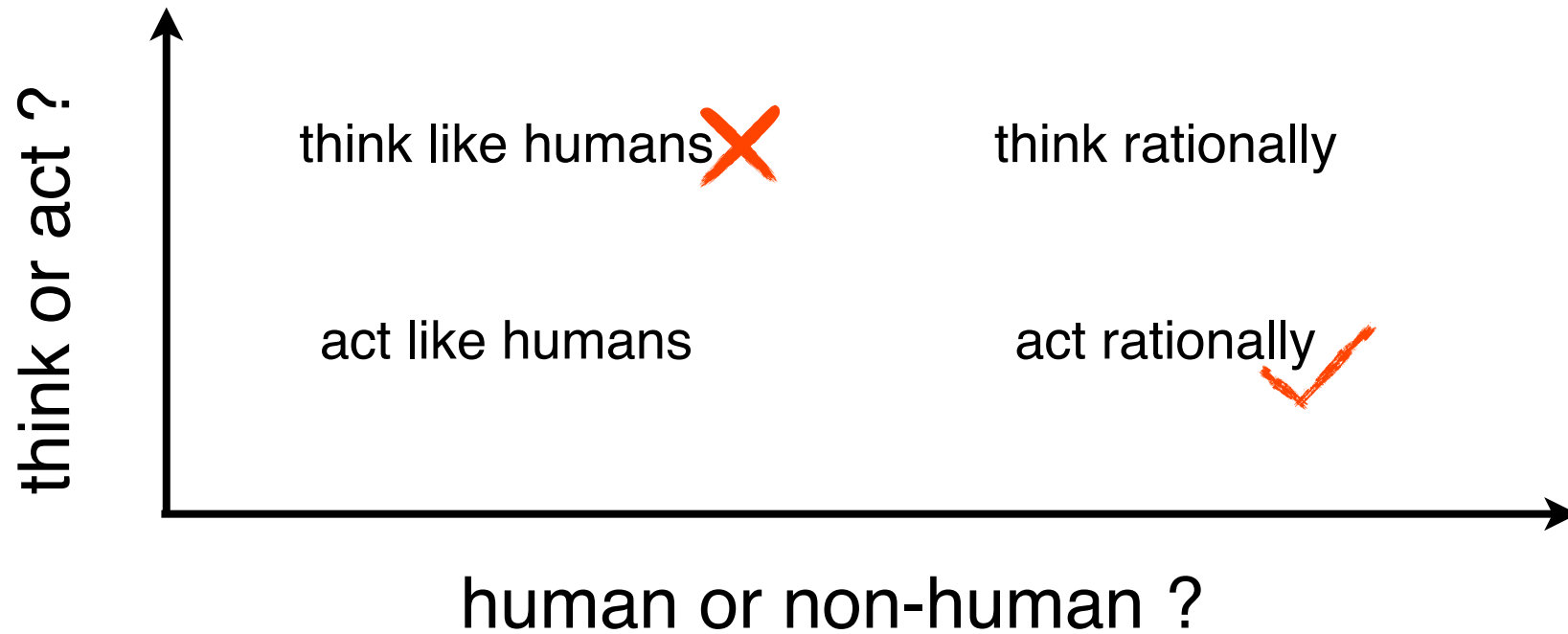
AI is a system that



What is AI?



AI is a system that



Current top AI systems



AlphaGo



2016年3月，AlphaGo 战胜韩国职业选手李世石（九段）

2017年1月初，快棋版本 Master 取得60:0战绩

Current top AI systems



DeepStack & Libratus



2017年1月左右，在一对一
无限注德州扑克上大幅赢
过职业选手

推理期

知识期

学习期



AI

60-70年
代

80年代
初期

90年代
中期



1950 1956

2006

2016

What we will learn



Search 搜索与规划

Knowledge 知识表达与处理

Uncertainty 不确定建模

Learning 机器学习

What we will do

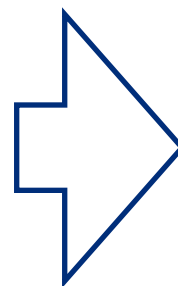


Search 搜索与规划

Knowledge 知识表达与处理

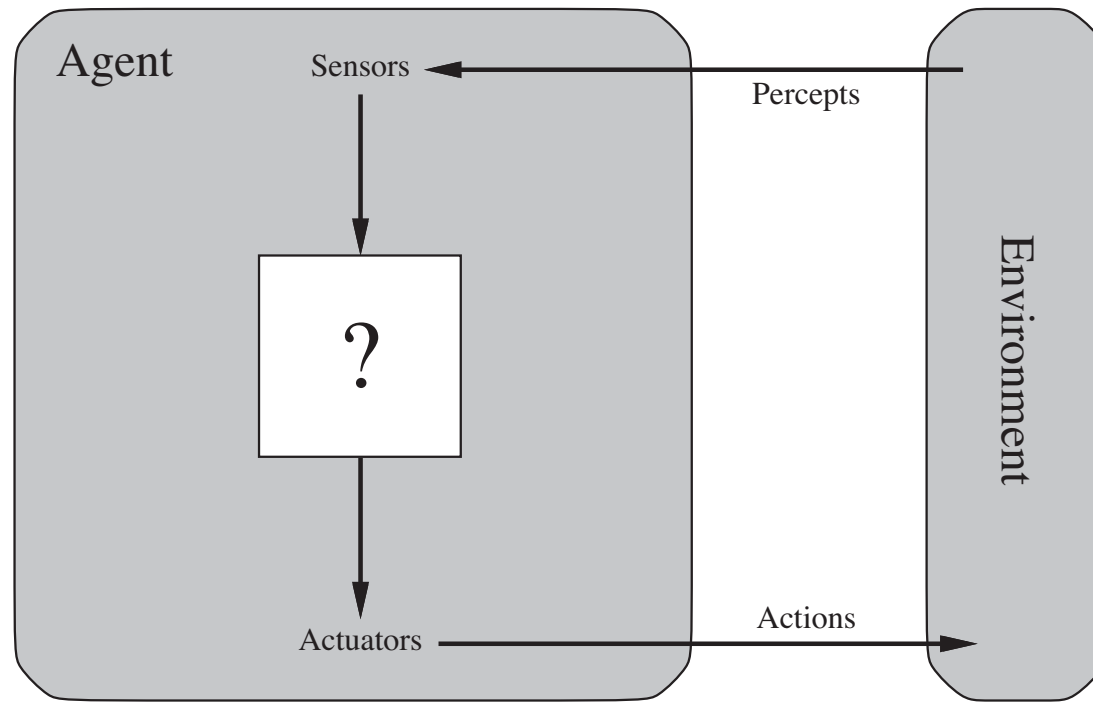
Uncertainty 不确定建模

Learning 机器学习



General
Game Player

Agent



Agents include humans, robots, softbots, thermostats, etc.

The agent function maps from percept histories to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

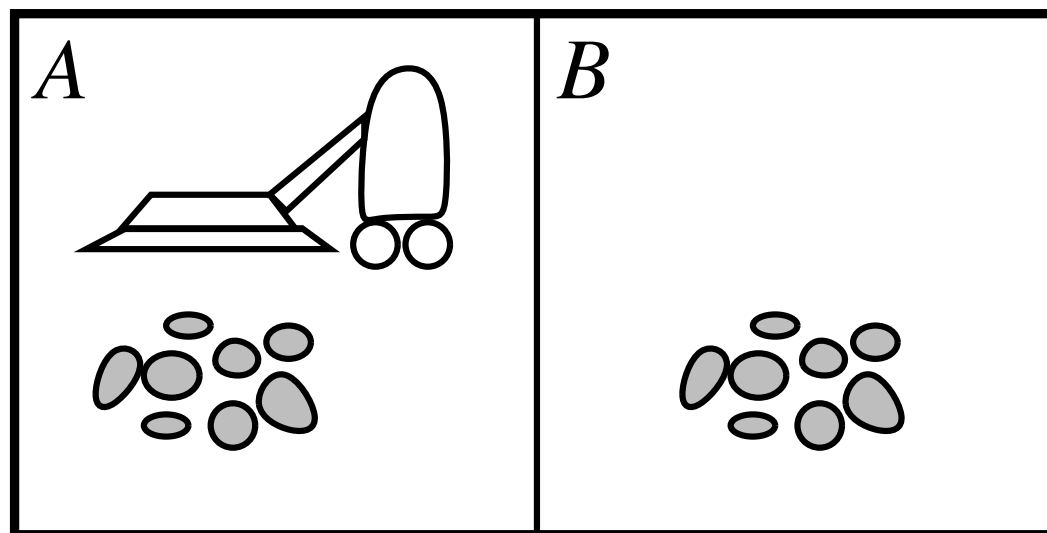
The agent program runs on the physical architecture to produce f

Example: Vacuum-cleaner world



Percepts: location and contents, e.g., $[A, \textit{Dirty}]$

Actions: *Left*, *Right*, *Suck*, *NoOp*



A vacuum-cleaner agent



Percept sequence	Action
$[A, Clean]$	<i>Right</i>
$[A, Dirty]$	<i>Suck</i>
$[B, Clean]$	<i>Left</i>
$[B, Dirty]$	<i>Suck</i>
$[A, Clean], [A, Clean]$	<i>Right</i>
$[A, Clean], [A, Dirty]$	<i>Suck</i>
\vdots	\vdots

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*

What is the **right** function?

Can it be implemented in a small agent program?



To design an agent, we need to specify
four-dimensions:

Performance measure?

Environment?

Actuators?

Sensors?

Examples of PEAS



Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Environment types



In six-dimensions:

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

Agent types

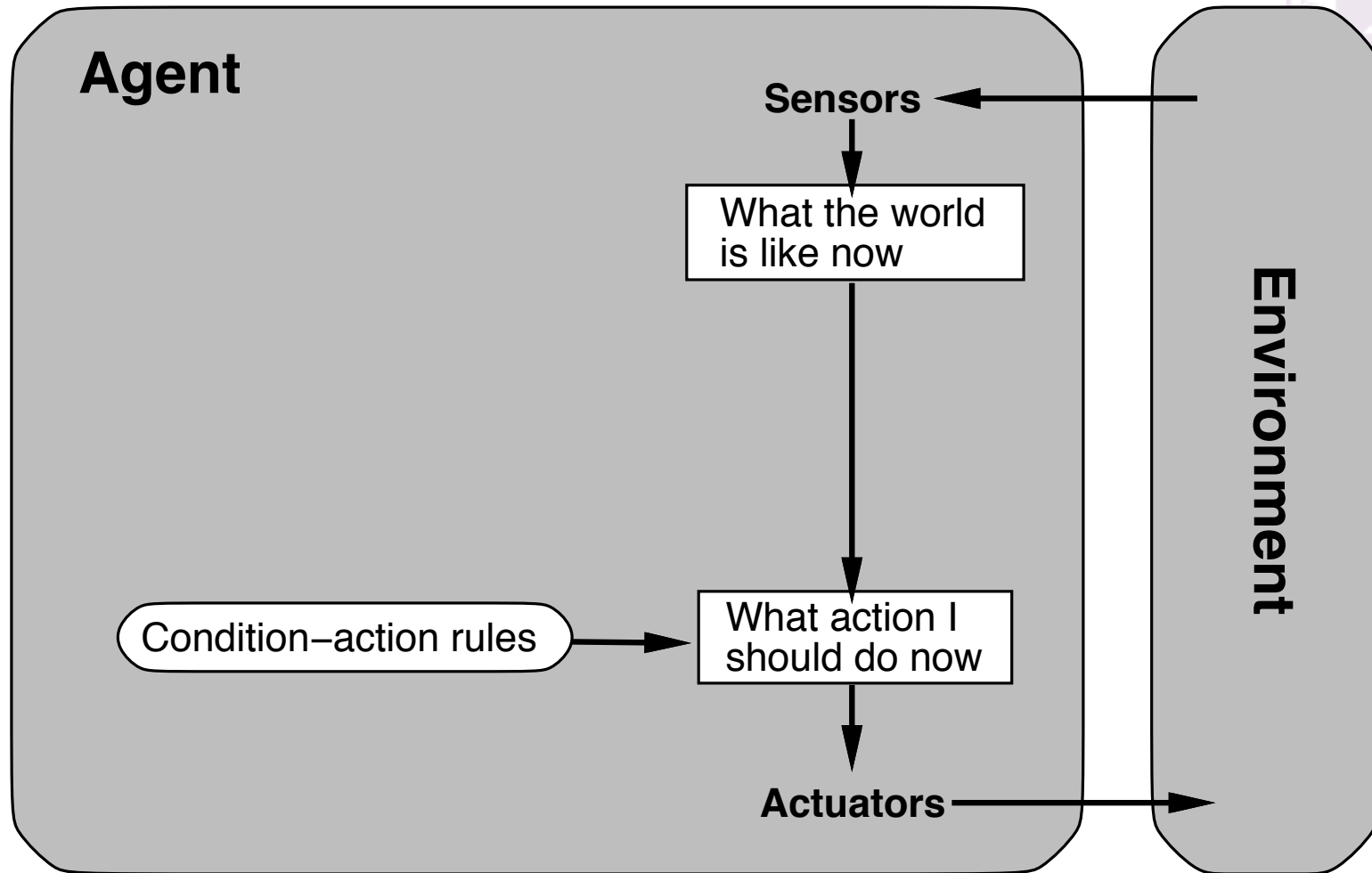
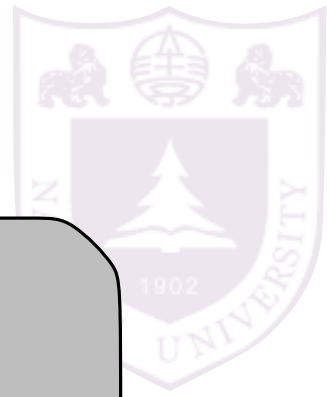


Four basic types in order of increasing generality:

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents

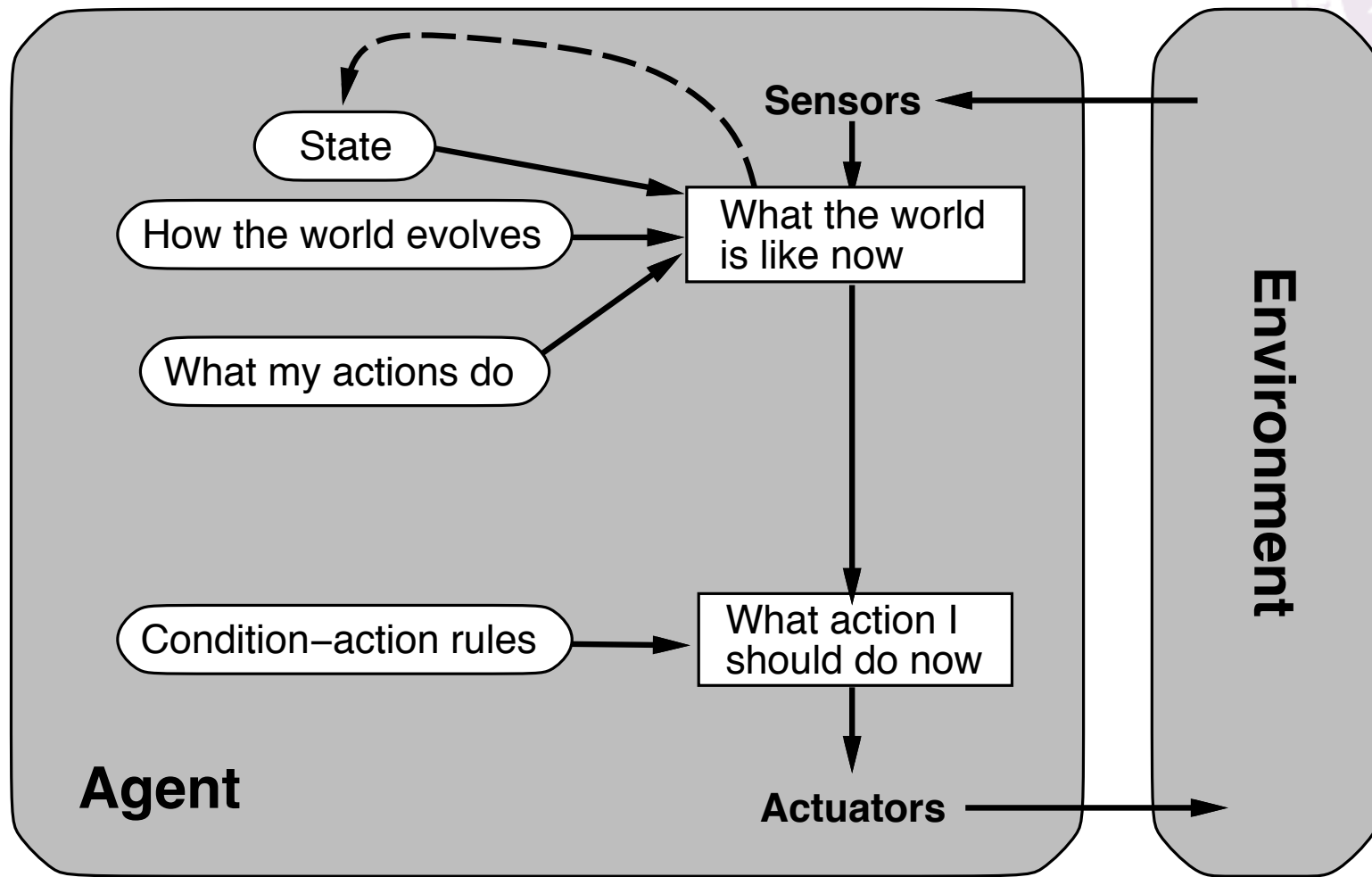
Simple reflex agents



function REFLEX-VACUUM-AGENT($[location, status]$) **returns** an action

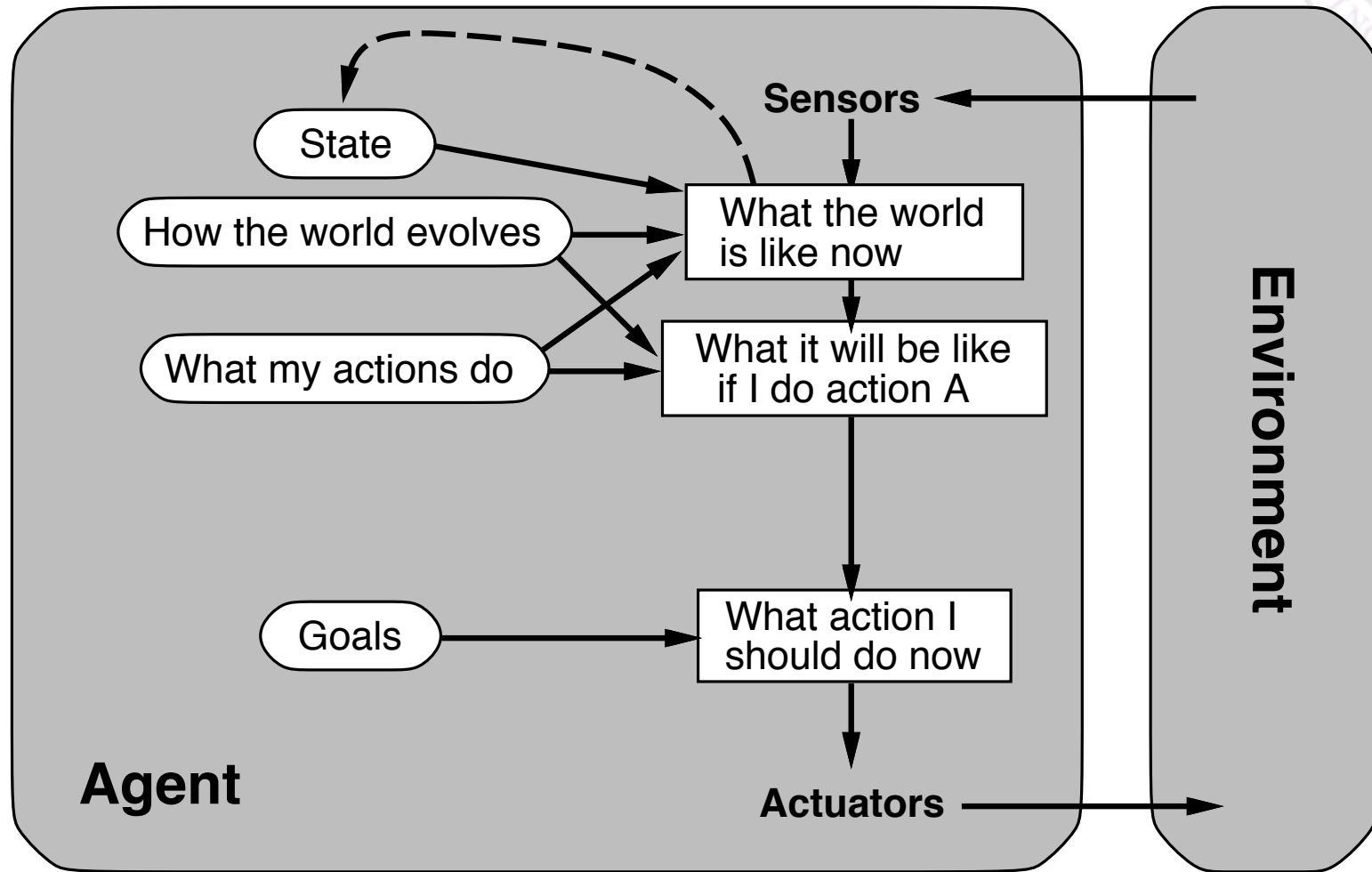
if $status = Dirty$ **then return** *Suck*
else if $location = A$ **then return** *Right*
else if $location = B$ **then return** *Left*

Reflex agents with state

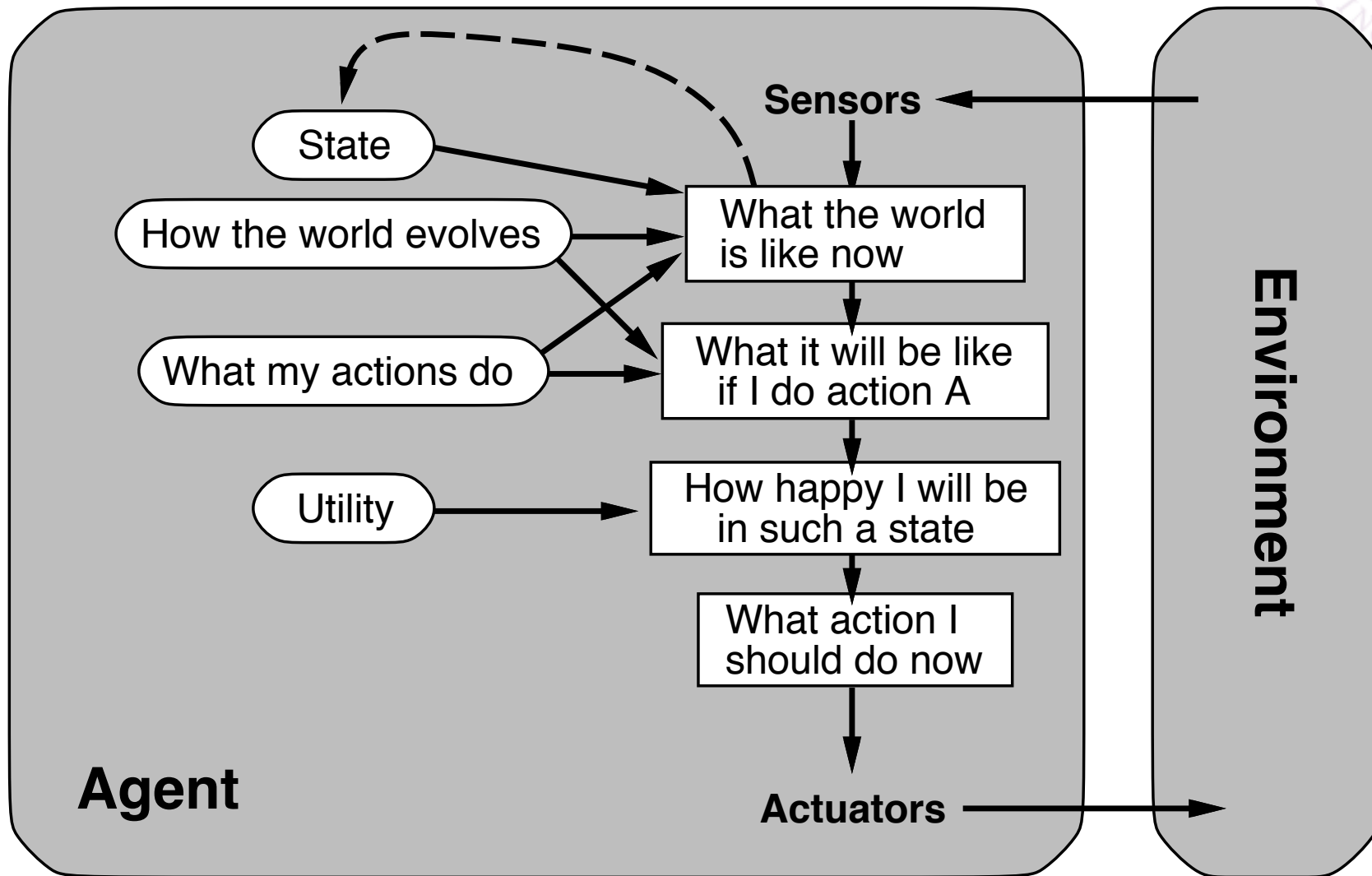


```
function REFLEX-VACUUM-AGENT( [location,status] ) returns an action
static: last_A, last_B, numbers, initially  $\infty$ 
  if status = Dirty then ...
```

Goal-based agents



Utility-based agents



Learning agents

