

# Механизмы искусственного интеллекта

Е. А. Черкашин

ИДСТУ им. В. М. Матросова СО РАН

10 февраля 2025 г., Иркутск

# К определению понятия «Искусственный интеллект»

К ИИ относятся программы, реализующие задачи, решение которых потребовало бы от человека использования его *когнитивных способностей*, выполнения *творческих функций*<sup>1</sup>.

Определение из книги АИМА<sup>2</sup>:

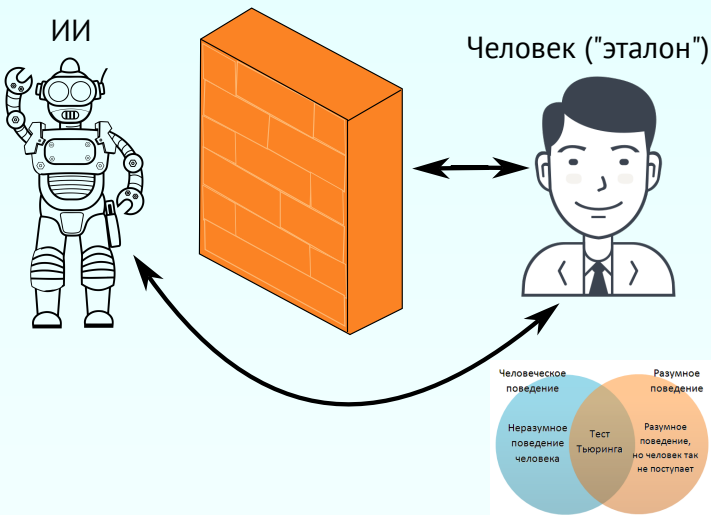
	Рационально	Как человек
Рассуждать	<input type="radio"/>	<input type="radio"/>
Вести себя	<input type="radio"/>	<input type="radio"/>

---

<sup>1</sup>Справочник ИИ, Т.1, 1990

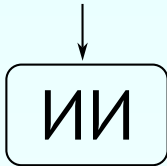
<sup>2</sup>P.Norvig, S.Russell. Artificial Intelligence Modern Approach

# Тест Тьюринга



Если Человек в процессе общения с ИИ будет думать, что общается с Человеком, то ИИ обладает требуемым свойством.

Моделирование рассуждений → моделирование «математики»



Моделирование мозга → моделирование  
самоорганизующихся систем

# Свойства и классы задач ИИ

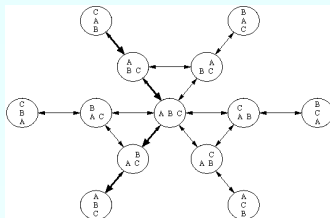
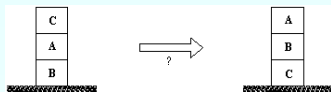
## Свойства задач

- ❑ Не существует алгоритма решения
- ❑ Обработка символьной информации
- ❑ Автоматизация принятия решения
- ❑ Решение – комбинация «возможностей»
- ❑ Обработка неполной и противоречивой информации

## Классы задач

1. Игры
2. Автоматическое доказательство теорем
3. Восприятие и распознавание образов
4. Планирование действий (Problem Solving)
5. Понимание естественного языка, перевод
6. Логическое программирование
7. Экспертные системы
8. Интеллектуальные и аналитические информационные системы
9. Восприятие и усвоение знаний (Machine learning)
10. Интеллектуальное управление [производством] ...
11. Робототехника (Robotics)
12. Системы поддержки принятия решений

# Планирование действий (Problem Solving)



## Формализация ( $SSG$ , State Space Graph)

- Структура данных для представления **состояния**
- Формализация **правил перехода** из состояния в состояние  $G$
- Программирование **распознавателя целевого состояния**  $R(v)$
- Задание **исходного состояния**  $I$

$$SSG = \langle G, R(v), I \rangle, G = \langle V, E \rangle, v \in V.$$

# Представление знаний

Естественно определить данные как некоторые сведения об отдельных объектах, а знания – о мире в целом.

**Данные** представляют информацию о существовании объектов, представляемых значениями признаков, а **знания** – информацию о существующих в мире закономерных связях между признаками и запрещающих некоторые другие сочетания свойств у объектов.

**Данные** – это информация о существовании объектов с некоторым набором свойств, **знания** – информация о **несуществовании** объектов с некоторым набором свойств.

Пусть  $H(x) \Leftrightarrow$  « $x$  является человеком», а  $M(x)$  – « $x$  – смертен».

«сократ,  $s$ , является человеком»:

$$A_1 = H(s),$$

«не существует бессмертных людей»:

$$A_2 = \neg \exists x (H(x) \& \neg M(x)),$$

преобразуем во «все люди смертны»:

$$A_3 = \forall x (H(x) \rightarrow M(x)).$$

...все объекты  $x$ , обладающие свойством  $H$ , будут обладать свойством  $M$ .

# Обработка информации «в математике»

Добавим информатики. Утверждение «Сократ – человек» представим как  $H(s)$ , где  $s$  – это Сократ. Теперь из **знания**  $A_3$  и **исходного данного**  $H(s)$  получим новую информацию о Сократе:

$$B = \left( H(s) \& \overbrace{\forall x (H(x) \rightarrow M(x))}^{A_3} \right) \rightarrow M(s).$$

Из того, что Сократ – человек и что все люди смертны, следует, что Сократ тоже смертен  $M(s)$ . Доказательство «от противного»:

1. Требуется **опровергнуть высказывание**  $\neg B = \text{True}$
2.  $B = \text{False} \vdash H(s) \& A_3 = \text{True}, M(s) = \text{False}$
3.  $H(s) \& A_3 = \text{True} \vdash H(s) = \text{True}, A_3 = \text{True}$
4. Подставим  $s$  вместо  $x$  в  $A_3$ , получим  $H(s) \rightarrow M(s)$
5.  $H(s) = \text{True}, M(s) = \text{False} \vdash H(s) \rightarrow M(s) = \text{False}$
6.  $H(s) \rightarrow M(s) = \text{False} \vdash A_3 = \text{False}$
7. Из п. 3 следует  $A_3 = \text{True}$ , а из п. 6 –  $A_3 = \text{False}$
8. В формальной (математической) логике в п. 7 получено **противоречие**
9. Следовательно,  $\neg B \neq \text{False}, B = \text{True}$
10. Из п. 9 следует  $M(s) = \text{True} \square$  (**ЧТД, QED<sup>3</sup>**)

<sup>3</sup>*Quod Erat Demonstrandum*, «что и требовалось показать», “which was to be demonstrated”



# Логическое программирование

## Сократим доказательство

Будем выписывать только **истинные высказывания**, т.е.

$M(s) \Leftrightarrow M(s) = \text{True}$  («Сократ смертен»),

$\neg M(s) \Leftrightarrow M(s) = \text{False}$  («Сократ бессмертен»)

$$B = \left( H(s) \& \overbrace{\forall x (H(x) \rightarrow M(x))}^{A_3} \right) \rightarrow M(s).$$

1. **Опроверяем**  $\neg B$

2.  $B \vdash H(s) \& A_3, M(s)$

3.  $H(s) \& A_3 \vdash H(s), A_3$

4.  $A_3\{s/x\} = H(s) \rightarrow M(s)$

5.  $H(s), \neg M(s) \vdash \neg(H(s) \rightarrow M(s))$

6.  $\neg(H(s) \rightarrow M(s)) \vdash \neg A_3$

7. Из п. 3 следует  $A_3$ , а из п. 6 –  $\neg A_3$

8. В п. 7 получено **противоречие**

9. Следовательно,  $B$

10. Из п. 9  $M(s) \square$

## Программа (теория)

```
h(s).  
m(X) :- h(X).    % h(x) -> m(x).
```

## Консультация

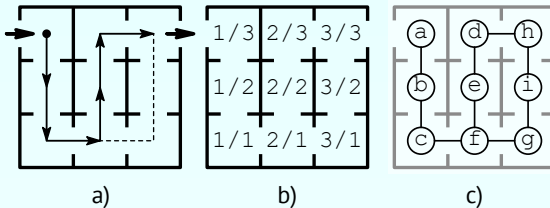
```
?- [socrates].    % Загрузка теории из файла  
true.
```

```
?- m(s).          % Запрос  
true.
```

## Объяснение вывода

```
?- trace.  
true.  
  
[trace] ?- m(s).  
    Call: (10) m(s) ?      % Запрос пользователя  
    Call: (11) h(s) ?      % Применение m(X):-h(X)  
                           % при {s/X}  
    Exit: (11) h(s) ?      % Найден факт h(s).  
    Exit: (10) m(s) ?      % следовательно m(s).  
true.
```

# Планирование действий



## Теория лабиринта

% "Данные"

```
e(a,b). e(b,c). e(c,f).  
e(f,e). e(e,d). e(d,h).  
e(f,g). e(g,i). e(i,h).
```

% "Знания"

% Что такое "путь".

```
path(A,B) :- e(A,B).  
path(A,B) :- e(A,C), path(C,B).
```

## Запрос

```
?- path(a,h).  
true ;  
true ;  
false.
```

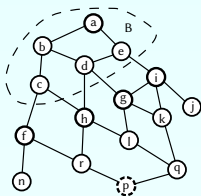
## Конструктивное решение

```
path(A,B, [A-B]) :- e(A,B).  
path(A,B, [A-C|T]) :- e(A,C), path(C,B,T).
```

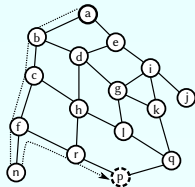
## Запрос

```
?- path(a,h,L).  
L = [a-b, b-c, c-f, f-e, e-d, d-h] ;  
L = [a-b, b-c, c-f, f-g, g-i, i-h] ;  
false.
```

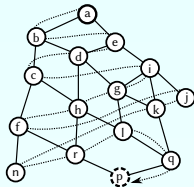
# Стратегии поиск в SSG



a)



b)



c)

## Поиск в глубину

```
dfs(V,[ ]):- r(V).  
dfs(V,[V-N|T]):- \+ r(V), after(V,N), dfs(N,T).  
r(h).  
after(X,Y):- e(X,Y); e(Y,X).  
?- dfs(a, S).  
S = [a-b, b-c, c-f, f-e, e-d, d-h] ;  
S = [a-b, b-c, c-f, f-e, e-d, d-e, e-d, d-h] ;  
S = [a-b, b-c, c-f, f-e, e-d, d-e, e-d, d-e, ... ]
```

## Поиск в ширину

```
bfs([[X|T]|_],[X|T]):-r(X),!. % (1)  
bfs([[X|T]|Ways], S):-  
    findall([Y,X|T], % (2)  
        (after(X,Y), \+ member(Y,[X|T])),  
        L),  
    append(Ways, L, N Ways), % (3)  
    bfs(N Ways, S).  
?- bfs([[a]],S).  
S = [h, d, e, f, c, b, a].
```

# Использование дополнительной информации

Зададим функцию  $f: V \rightarrow R$  следующего вида:

$$f(x) = g(x) + r(x)$$

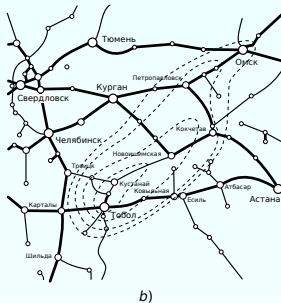
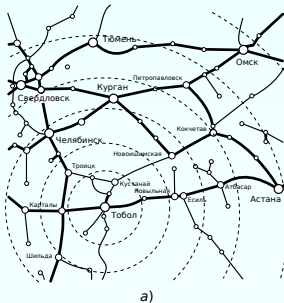
– стоимость кратчайшего пути (КП) через  $x$ ,  $g(x)$  – до  $x$ ,  $r(x)$  – КП до целевой вершины (не известен).

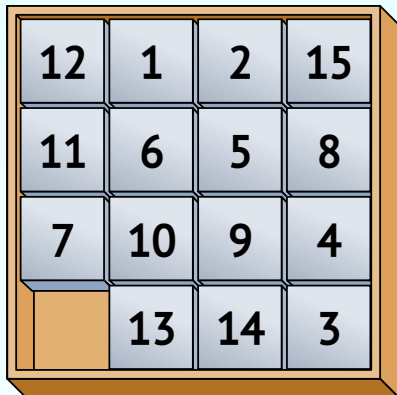
Оценка  $r(x)$  снизу –  $h(x)$ ,  $h(x) \leq r(x)$ .

Теперь

$$f(x) \leq g(x) + h(x).$$

```
bf1([_s(G,[Target|T])|_],%(1)
    Target=_s(G,[Target|T])):-!.
bf1([_s(G,[X|T])|Ways],Target=GPS,S):-
    Target\=X,
    findall(F1=_s(G1,[Y,X|T]),
        after([X|T],G,GPS,Y,G1,F1),L),
    append(L,Ways,NWays),keysort(NWays,SNWays),
    bf1(SNWays,Target=GPS,S).
after([S|R],SG,GPS,T,TG,F):-
    transdist(S,T,D),% \+ member(T,[S|R]),
    TG is SG + D,
    geodist(T,GPS,GDist),%(2)
    F is TG + GDist.
bf(Start,Target,Sol):-
    geocode(Target,Lon,Lat,_),
    bf1([_s(0,[Start])],
        Target=ll(Lon,Lat),Sol).%(3)
```





## Поиск решения с использованием эвристики

```
center% ./15-solve 100 20 1
Environment:
USE_HEURISTIC=1
Puzzle 15 solving program
State(00,00[00])
<---solution--->
0 Step -----
  5  1  3  4
  9  2  6  8
13 10 15  0
14 12  7 11
x=2, y=3
State(00,16[16])
. . . . .
17 Step -----
  1  2  3  4
  5  6  7  8
  9 10 11 12
13 14 15  0
x=3, y=3
State(17,00[17])
Tested 111 states.
```

## Без эвристики

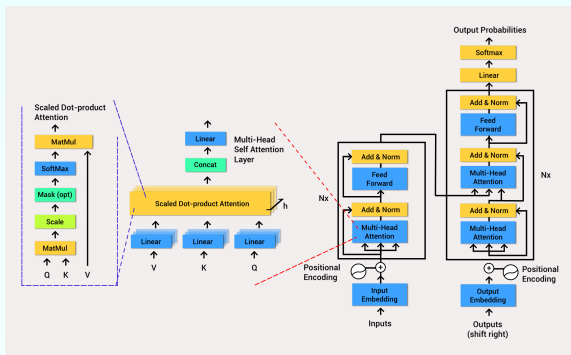
```
. . . . .
17 Step -----
  1  2  3  4
  5  6  7  8
  9 10 11 12
13 14 15  0
x=3, y=3
State(17,00[17])
Tested 921299 states.
```

# Chat GPT 4

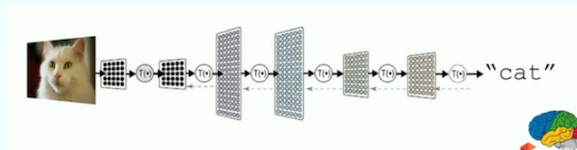
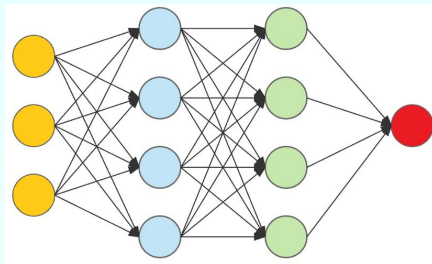
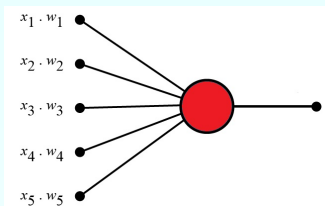
ChatGPT (Generative Pretrained Transformer, Порождающий [пред]тренированный преобразователь) – чат-бот OpenAI, запущенный в ноябре 2022.

...настроен при помощи обучения «с учителем» и «с подкреплением». Именно обучение с подкреплением делает ChatGPT **уникальным**.

...предназначен для реагирования на входные данные, представленные на естественном языке.



# Нейронные сети



# Анализ входного текста

GPT-3 Codex

GPT-3 (Generative Pre-trained Transformer 3) uses a process called tokenization to break down text. Many words map to single tokens, though longer or more complex words often break down into multiple tokens. On average, tokens are roughly 4 characters long.

Clear

Show example

Tokens

56

Characters

258

GPT-3 (Generative Pre-trained Transformer 3) uses a process called tokenization to break down text. Many words map to single tokens, though longer or more complex words often break down into multiple tokens. On average, tokens are roughly 4 characters long.

TEXT

TOKEN IDS



# Воспитание нейронной сети

## Step 1

**Collect demonstration data and train a supervised policy.**

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



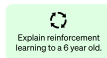
This data is used to fine-tune GPT-3.5 with supervised learning.



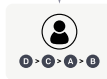
## Step 2

**Collect comparison data and train a reward model.**

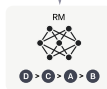
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



## Step 3

**Optimize a policy against the reward model using the PPO reinforcement learning algorithm.**

A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



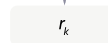
The policy generates an output.

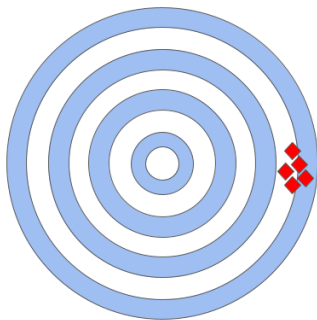


The reward model calculates a reward for the output.

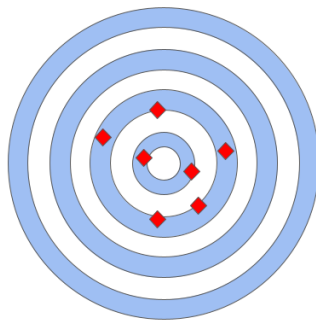


The reward is used to update the policy using PPO.





High capability  
Low alignment



Low capability  
High alignment