# Использование вероятностно-комбинаторного обобщения знаний в обучении с подкреплением

А.С. Мисник

Семинар по математической логике, 2026

# Мотивация: RL и интерпретируемость

**Обучение с подкреплением (RL):** агент выбирает действия $a \in \mathbf{A}$ в состояниях $s \in \mathbf{S}$, максимизируя дисконтированную награду

$$V^{\pi}(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \;\middle|\; S_0 = s_0,\, \pi\right].$$

**Проблема нейросетевых методов:**

- Оптимальная политика закодирована в весах сети — *нечитаема*
- Большой объём данных и вычислений для сходимости

**Альтернатива:** методы *индуктивного логического вывода*, порождающие явные правила вида

«состояние $s$ удовлетворяет шаблону $H \Rightarrow$ выбрать действие $a$».

Вероятностно-комбинаторный формальный метод (ВКФ) — один из таких подходов, объединяющий решёточную структуру гипотез с вероятностной генерацией.

# Задача индуктивного вывода: постановка

**Входные данные:**

- $G^+ \subset \{0,1\}^d$ — **примеры** (выигрышные состояния / полезные переходы)
- $G^- \subset \{0,1\}^d$ — **контрпримеры** (проигрышные состояния / бесполезные переходы)

**Гипотеза** — подмножество битовых позиций $B \subseteq \{1, \ldots, d\}$.

Говорят, что $B$ *покрывает* строку $s \in \{0,1\}^d$, если на всех позициях из $B$ у строки стоит 1:

$$\forall j \in B : \; s_j = 1.$$

**Корректная гипотеза:** $B$ покрывает хотя бы один пример из $G^+$ и **ни одного** элемента из $G^-$.

Таким образом, корректная гипотеза — это **достаточное условие** принадлежности к классу $G^+$, не опровергнутое ни одним контрпримером.

# Задача индуктивного вывода: формальная постановка

**Задача ВКФ:** найти минимальный набор корректных гипотез $\mathcal{H} = \{H_1, \ldots, H_k\}$, такой что

$$\forall g^+ \in G^+ \quad \exists H_i \in \mathcal{H} : H_i \text{ покрывает } g^+.$$

Иными словами: покрыть все примеры наименьшим числом корректных шаблонов.

## Связь с символьной логикой

Каждая гипотеза $H$ задаёт конъюнкцию литералов:

$$H = \bigwedge_{j \in B} x_j, \quad B \subseteq \{1, \ldots, d\}.$$

Задача ВКФ — построить минимальное **дизъюнктивное покрытие** $G^+$ такими конъюнктами без ложных срабатываний на $G^-$.

# Кандидаты в гипотезы: решёточная структура

Пусть $O \subseteq G^+$ — примеры, $F$ — битовые позиции (признаки), $I \subseteq O \times F$ — обучающая выборка.

**Поляры** (соответствие Галуа): $A' = \{f \in F : \forall o \in A,\ o_f = 1\}$, $B' = \{o \in O : \forall f \in B,\ o_f = 1\}$.

**Кандидат** $\langle A, B\rangle$: $B = A'$, $A = B'$. Задаёт гипотезу $B$ — множество позиций, равных 1 у всех объектов группы $A$.

## Теорема (Вилле, 1982)

Кандидаты образуют **полную решётку** $L(O, F, I)$:

$$\langle A_1, B_1\rangle \wedge \langle A_2, B_2\rangle = \langle (A_1 \cup A_2)'',\ B_1 \cap B_2\rangle,$$
$$\langle A_1, B_1\rangle \vee \langle A_2, B_2\rangle = \langle A_1 \cap A_2,\ (B_1 \cup B_2)''\rangle.$$

Задача ВКФ — найти $\wedge$-неразложимые элементы этой решётки, не покрывающие ни одного контрпримера.

# Фантомные сходства: проблема переобучения

**Фантомное сходство** — кандидат $\langle A, B \rangle$, где каждый объект из $A$ принадлежит $G^+$ по *своей* независимой причине, а признаки $B$ оказались общими случайно.

Такие кандидаты *не являются* настоящими закономерностями — они возникают как артефакт обучения, когда алгоритм «переобучается» на случайных совпадениях признаков.

- Каждый объект из $A$ имеет *свою* причину принадлежать $G^+$
- Признаки $B$ случайно совпали у всех объектов из $A$
- Кандидат $\langle A, B \rangle$ ложно «объясняет» $A$ через $B$

# Фантомные сходства: пример

Пусть $O = \{o_1{=}\text{B737MAX},\ o_2{=}\text{MC21},\ o_3{=}\text{SJ100},\ o_4{=}\text{A350}\}$ — самолёты без сертификата лётной годности, описанные проблемами $F = \{f_1{=}\text{оперение},\ f_2{=}\text{двигатель},\ f_3{=}\text{ругательство}\}$:

| $O \mid F$ | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| $o_1$ (B737MAX) | 1 | 0 | 0 |
| $o_2$ (MC21) | 1 | 0 | 1 |
| $o_3$ (SJ100) | 0 | 1 | 1 |
| $o_4$ (A350) | 0 | 1 | 0 |

Два *настоящих* сходства: $\langle\{o_1, o_2\}, \{f_1\}\rangle$ и $\langle\{o_3, o_4\}, \{f_2\}\rangle$.
*Фантомное*: $\langle\{o_2, o_3\}, \{f_3\}\rangle$ — признак $f_3$ у обоих случаен:

- $o_2$ не сертифицирован из-за $f_1$ (оперение)
- $o_3$ не сертифицирован из-за $f_2$ (двигатель)
- нацарапанное ругательство $f_3$ — случайное совпадение

# Фантомные сходства: оценка вероятности

Здесь $n$ — число сопутствующих признаков, $p$ — вероятность каждого из них у объекта, $b$ — число примеров в фантомном сходстве.

**Теорема 1 (Виноградов, 2024)**

При $p \geq \left(-\ln(1-\varepsilon)/n\right)^{1/b}$ вероятность появления фантомного сходства $b$ случайных $p$-примеров не менее $\varepsilon > 0$.

**Интерпретация:** даже при небольшой вероятности $p$ наличия случайного признака и малом числе примеров $b$ фантомное сходство возникает с ненулевой вероятностью.

Это означает, что фантомные гипотезы — *неотъемлемое* следствие самой постановки задачи индуктивного вывода, а не артефакт конкретного алгоритма.

# Фантомные сходства: роль контрпримеров

## Теорема 2 (Виноградов, 2024)

При $n \to \infty$, $p = \sqrt{a/n}$, вероятность фантомного сходства двух примеров, не устранённого $m = c\sqrt{n}$ контрпримерами, стремится к

$$1 - e^{-a} - a\, e^{-a}\left(1 - e^{-c\sqrt{a}}\right) > 0.$$

Даже при числе контрпримеров $m = c\sqrt{n}$ фантомное сходство выживает с положительной вероятностью.

**Следствие:** единственный эффективный инструмент борьбы — рост $|G^-|$, опережающий рост $n$. В задачах RL это означает необходимость собирать больше проигрышных состояний при увеличении размерности кодировки.

# Генерация гипотез: операции замыкания

**Идея:** вместо полного перебора решётки — случайное блуждание к одному кандидату.

Две операции перехода между кандидатами:

**Замыкание вниз** (добавить объект $o \in O$):

$$CbO{\downarrow}\,(\langle A, B\rangle,\, o) = \langle (A \cup \{o\})'',\; B \cap \{o\}'\rangle.$$

**Замыкание вверх** (добавить признак $f \in F$):

$$CbO{\uparrow}\,(\langle A, B\rangle,\, f) = \langle A \cap \{f\}',\; (B \cup \{f\})''\rangle.$$

**Свойство монотонности:** обе операции сохраняют порядок на кандидатах — если $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$, то неравенство выполнено и после применения той же операции к обоим.

# Генерация гипотез: алгоритм

**Спаривающая цепь Маркова** поддерживает пару $Min \leq Max$ и на каждом шаге применяет одну случайную операцию к обоим кандидатам:

1: $Min \leftarrow \langle O, O' \rangle, \quad Max \leftarrow \langle F', F \rangle$
2: **while** $Min \neq Max$ **do**
3: выбрать $r \in O \cup F$ равномерно случайно
4: **if** $r \in O$ **then** применить $CbO{\downarrow}\,(\cdot, r)$ к $Min$ и $Max$
5: **else** применить $CbO{\uparrow}\,(\cdot, r)$ к $Min$ и $Max$
6: **return** $Min$ ▷ случайный кандидат решётки

Монотонность гарантирует $Min \leq Max$ на каждом шаге, поэтому $Min$ и $Max$ неизбежно сходятся.

Многократный запуск алгоритма даёт **случайную выборку кандидатов** — набор гипотез ВКФ.

# Пример работы алгоритма: данные

Пусть имеется 4 объекта и 4 признака:

| Объект | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
|---|---|---|---|---|
| $o_1$ | 1 | 0 | 1 | 0 |
| $o_2$ | 1 | 0 | 0 | 1 |
| $o_3$ | 0 | 1 | 1 | 0 |
| $o_4$ | 0 | 1 | 0 | 1 |

Начальное состояние алгоритма:

$$Min = \langle \{o_1, o_2, o_3, o_4\},\ \emptyset \rangle \qquad \text{(наименьший кандидат)}$$
$$Max = \langle \emptyset,\ \{f_1, f_2, f_3, f_4\} \rangle \qquad \text{(наибольший кандидат)}$$

Инвариант: $Min \leq Max$ (т.е. $B_{Min} \subseteq B_{Max}$) сохраняется на каждом шаге.

# Пример работы алгоритма: трассировка

| Шаг | $r$ | $Min = \langle A_{min}, B_{min} \rangle$ | $Max = \langle A_{max}, B_{max} \rangle$ |
|---|---|---|---|
| 0 | — | $\langle \{o_1, o_2, o_3, o_4\}, \emptyset \rangle$ | $\langle \emptyset, \{f_1, f_2, f_3, f_4\} \rangle$ |
| 1 | $o_1$ | $\langle \{o_1, o_2, o_3, o_4\}, \emptyset \rangle$ | $\langle \{o_1\}, \{f_1, f_3\} \rangle$ |
| 2 | $o_2$ | $\langle \{o_1, o_2, o_3, o_4\}, \emptyset \rangle$ | $\langle \{o_1, o_2\}, \{f_1\} \rangle$ |
| 3 | $f_1$ | $\langle \{o_1, o_2\}, \{f_1\} \rangle$ | $\langle \{o_1, o_2\}, \{f_1\} \rangle$ |

**Шаг 1** ($r = o_1 \in O$): $o_1 \in A_{min}$, поэтому $Min$ не меняется; $Max$ сужается до объектов с $\{f_1, f_3\}$.

**Шаг 2** ($r = o_2 \in O$): $Max$ сужается ещё — общий признак $\{o_1, o_2\}$ только $f_1$.

**Шаг 3** ($r = f_1 \in F$): $f_1 \in B_{max}$, поэтому $Max$ не меняется; $Min$ поднимается до $\{o_1, o_2\}$.

$Min = Max = \langle \{o_1, o_2\}, \{f_1\} \rangle$ — алгоритм завершён. Гипотеза: $B = \{f_1\}$.

# Игра Ним: модель

**Правила:** 2 игрока, $K$ куч камней; за один ход берут **1–3 камня** из любой одной кучи; **проигрывает** тот, кто берёт последний камень.

**Пространство состояний:** $s = (n_1, \ldots, n_K)$, где $n_i \in \{0, \ldots, 2^N - 1\}$ — размер $i$-й кучи.

**Классификация состояний (P/N-позиции):**

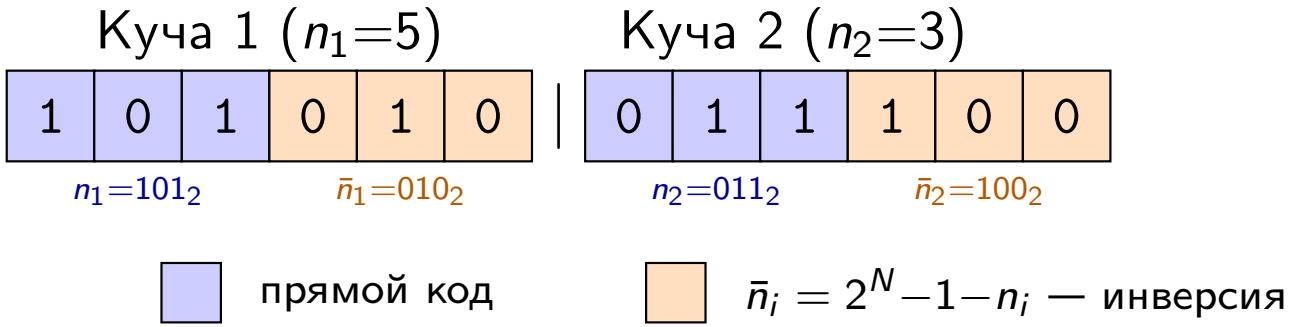
- **P-позиция (проигрышная):** все возможные ходы ведут в N-позиции
- **N-позиция (выигрышная):** существует ход в P-позицию
- База: $(0, \ldots, 0)$ — P-позиция (последний камень уже взят)

**Задача RL:** научиться всегда делать ход из N-позиции в P-позицию (оптимальная стратегия).

**Масштаб:** при $K$=2, $N$=4 (кучи до 15 камней) — уже $16^2 = 256$ состояний; при $N$=10 — свыше миллиона.

# Игра Ним: кодировка состояний

**Кодировка состояния** $s = (n_1, \ldots, n_K)$ в битовую строку: для каждой кучи $i$ конкатенируем прямой код $n_i$ и инверсию $\bar{n}_i = 2^N - 1 - n_i$.




Итого: 101 010 | 011 100 — 12 бит

# Экстраполяция стратегий: алгоритм

**Задача:** по малой выборке состояний построить полное описание оптимальной стратегии.

**Алгоритм экстраполяции:**

1. Сформировать случайную выборку состояний с известными метками; выигрышные $\to G^+$, проигрышные $\to G^-$
2. Запустить ВКФ (спаривающая цепь Маркова, повторно): получить набор корректных гипотез
3. Удалить дубликаты — гипотезы, покрывающие одни и те же примеры
4. Найти покрытие $\mathcal{H}$ минимальной мощности

Полученный набор $\mathcal{H}$ применяется как стратегия *без дополнительного обучения*: состояние $s$ считается выигрышным, если хотя бы одна $H_i \in \mathcal{H}$ покрывает $v(s)$.

**Ключевое свойство:** метод требует лишь малой доли состояний для обучения и затем экстраполирует на всё пространство.

# Экстраполяция стратегий: результаты для Ним

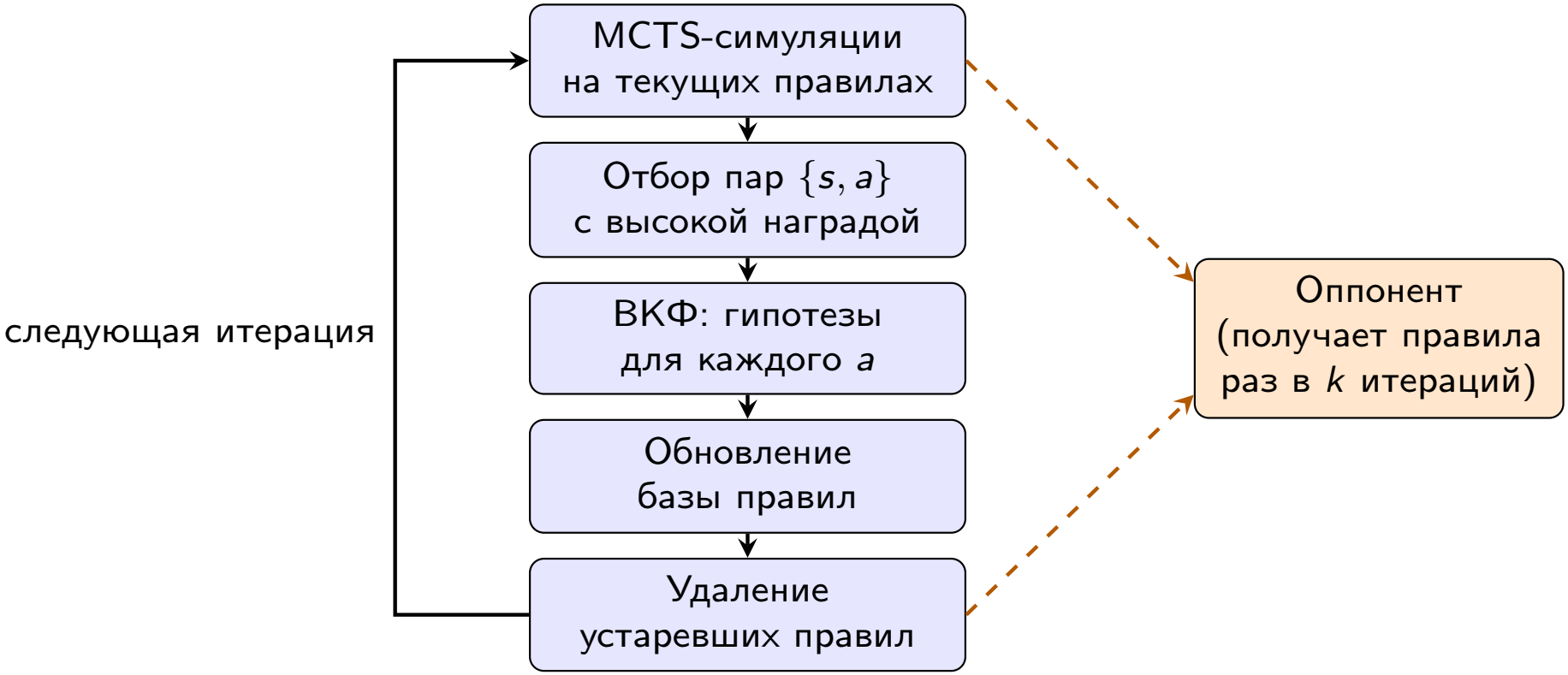
| Количество куч | Максимальное количество камней в куче | Количество состояний | Количество выигрышных состояний | Количество проигрышных состояний | Количество порожденных гипотез | Количество необъясненных состояний |
|---|---|---|---|---|---|---|
| 2 | 4 | 16 | 12 | 4 | 6 | 0 |
| 2 | 5 | 25 | 19 | 6 | 7 | 0 |
| 2 | 7 | 49 | 36 | 13 | 8 | 0 |
| 2 | 8 | 64 | 48 | 16 | 7 | 0 |
| 3 | 3 | 27 | 20 | 7 | 11 | 0 |

**Ключевые наблюдения:**

- Число гипотез **не растёт** с увеличением размера куч — нет экспоненциального взрыва
- Итоговая база содержит лишь **6–8 гипотез** независимо от размера пространства состояний

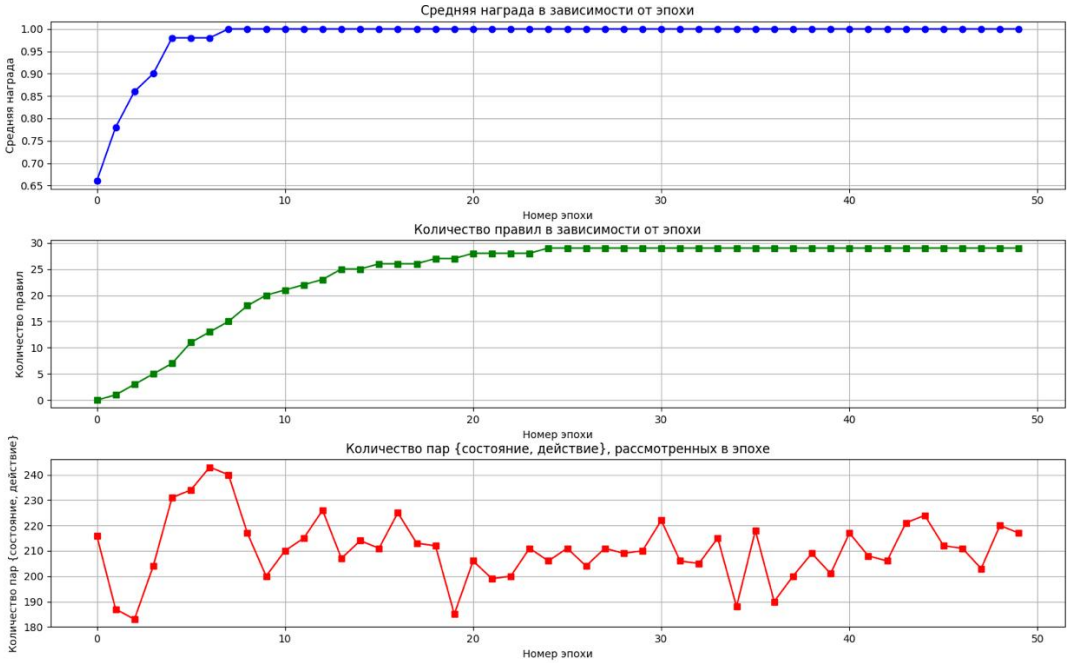
# Обучение в реальном времени: алгоритм (Ним)

**Цель:** итеративно улучшать стратегию без полного перебора состояний.



# Обучение в реальном времени: результаты (Ним)

**Параметры:** 2 кучи, до 10 камней.

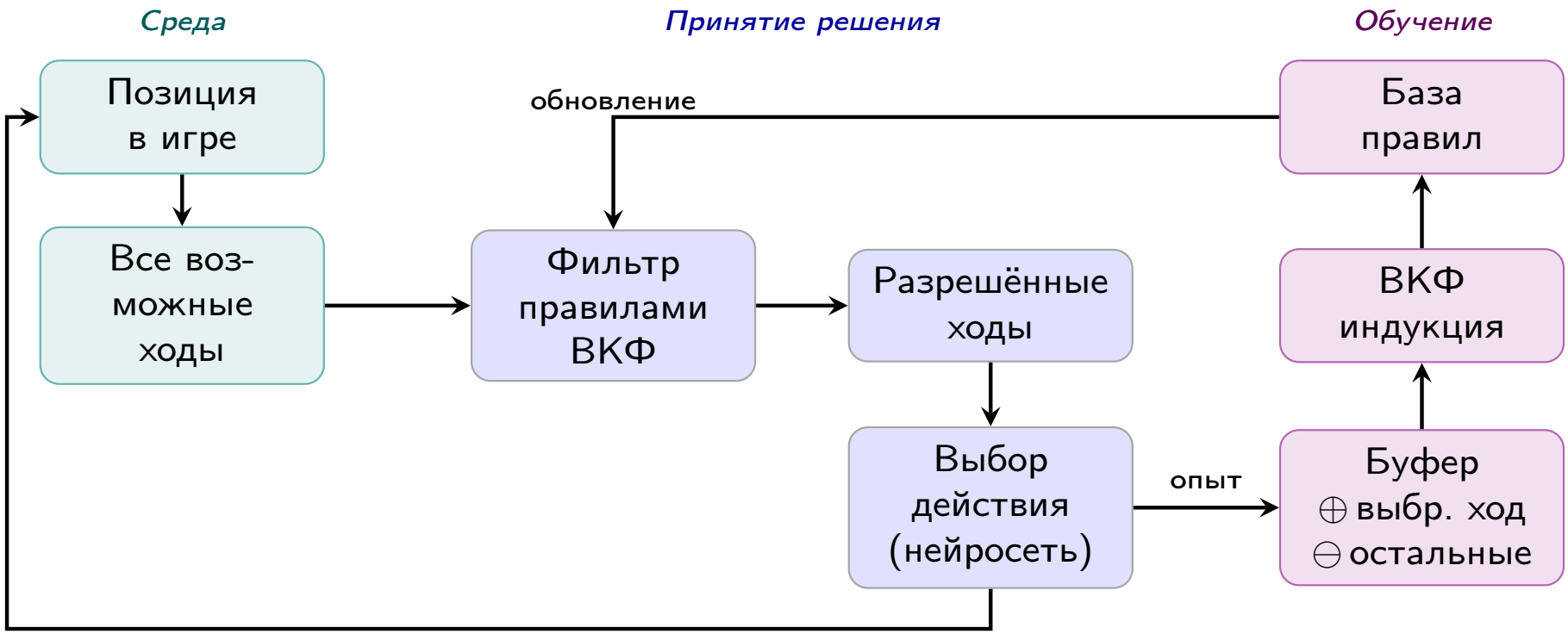



**Наблюдения:**

- Стратегия оптимальна уже к **7-й эпохе**
- Число правил **существенно меньше** числа состояний — высокая степень обобщения

# Обучение в реальном времени: алгоритм (нарды)

**Цель:** использовать ВКФ-гипотезы как *сужающий фильтр* для нейросетевого агента.



# Результаты: нарды

Сравниваемые методы при **одинаковом вычислительном бюджете**:

- **TD-Gammon** — классический алгоритм нейросетевого обучения
- **DQN** — более свежая архитектура обучения с воспроизведением опыта
- **ВКФ-оптимизатор** — нейросеть оценивает ценность, правила ВКФ сужают пространство ходов

| Метод | Побед против TD-gammon | Побед против DQN | Побед против ВКФ |
|---|---|---|---|
| TD-gammon | - | 464 | 438 |
| DQN | 536 | - | 446 |
| **ВКФ** | **562** | **554** | **-** |

Турнир: 1000 партий на каждую пару, усреднено по 10 запускам.

# Выводы

- **Индуктивный логический вывод** на битовых строках — интерпретируемая альтернатива нейросетевым методам RL для дискретных пространств состояний
- ВКФ формирует **компактный набор правил**: число гипотез не растёт экспоненциально с ростом пространства состояний
- **Вероятностный аспект** (спаривающая цепь Маркова) обеспечивает случайную выборку кандидатов; фантомные сходства устраняются ростом числа контрпримеров
- **Два режима применения:**
  - *Экстраполяция*: компактное описание оптимальной стратегии по малой выборке (Ним)
  - *Реальное время*: ВКФ + нейросеть превосходит TD-Gammon и DQN при равном бюджете (нарды)
- **Ограничение:** требуется компактная бинарная кодировка состояний

# Спасибо за внимание!

misnik.as@phystech.edu