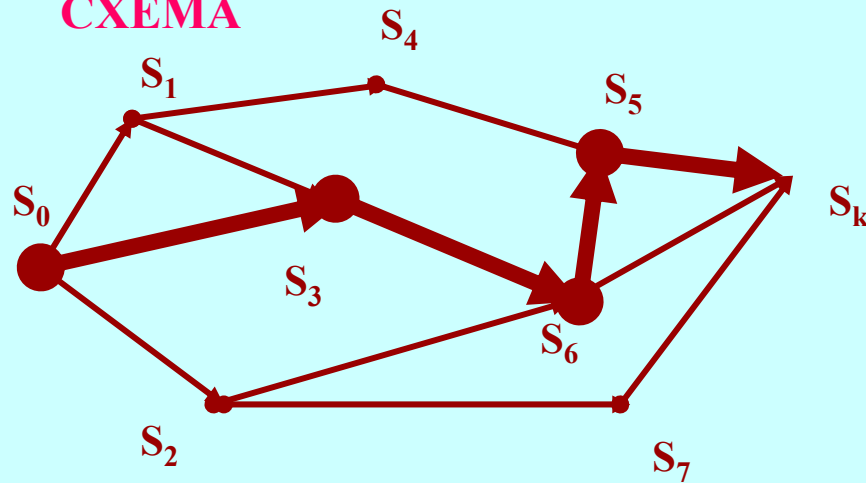


# Интеллектуальные системы и технологии

Лекция 1. Приложение  
Решение задач в классическом ИИ

# МЕТОД ПРОСТРАНСТВА СОСТОЯНИЙ

## СХЕМА

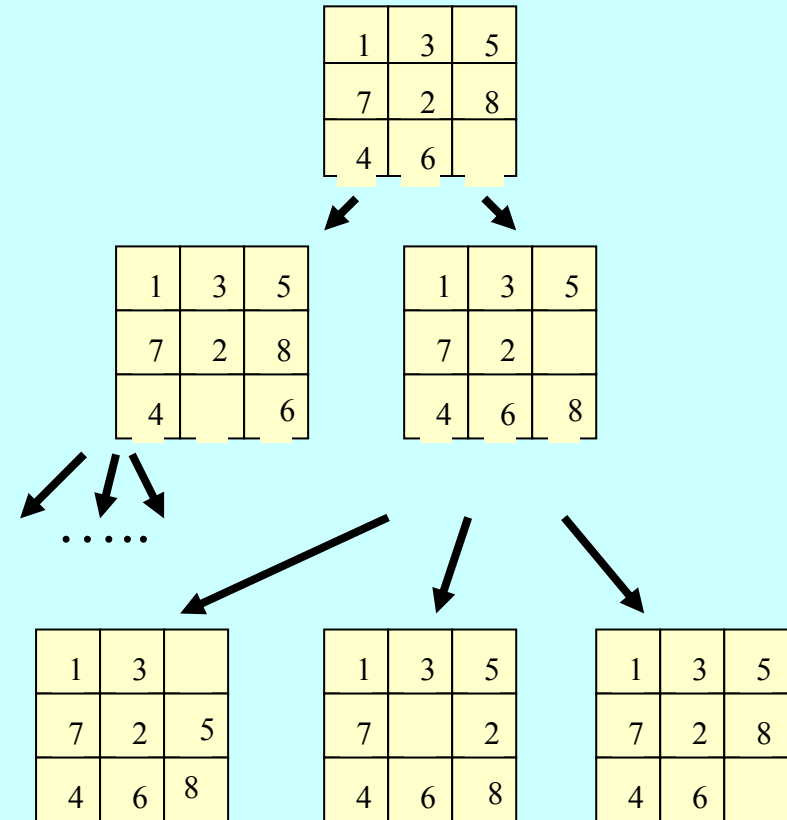


## ФОРМАЛИЗМ

Заданы:

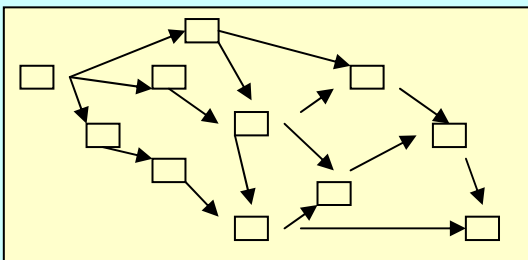
1. Множество состояний  $\{S_i\}$
  2. Исходное  $S_0$  и конечные состояния  $\{S_k\}$  или их свойства (описание)
  3. Множество переходов  $\{p_r(S_i, S_j)\}$
- Определить путь от  $S_0$  до любого  $S_k$

## ПРИМЕР (игра в 8)



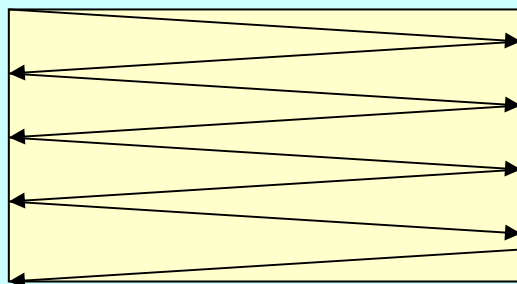
ОПЕРАТОРЫ:  $S_m = \text{вправо}(S_n)$ ,  
 $S_m = \text{влево}(S_n)$ ,  $S_m = \text{вверх}(S_n)$ ,  
 $S_m = \text{вниз}(S_n)$ ,

# СТРАТЕГИИ КОМБИНАТОРИКИ

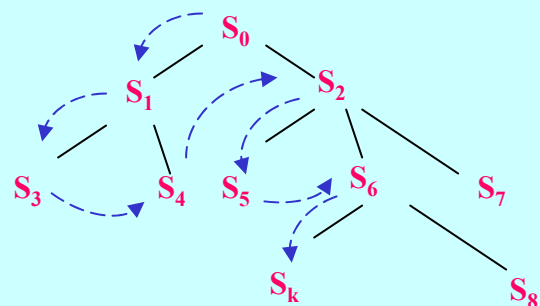
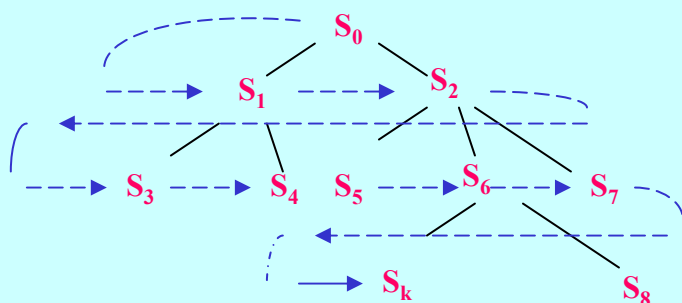
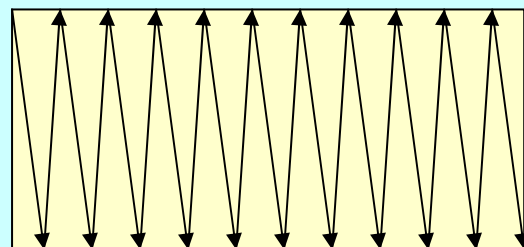


ОБЛАСТЬ  
СКАНИРОВАНИЯ

ПЕРЕБОР В «ШИРИНУ»



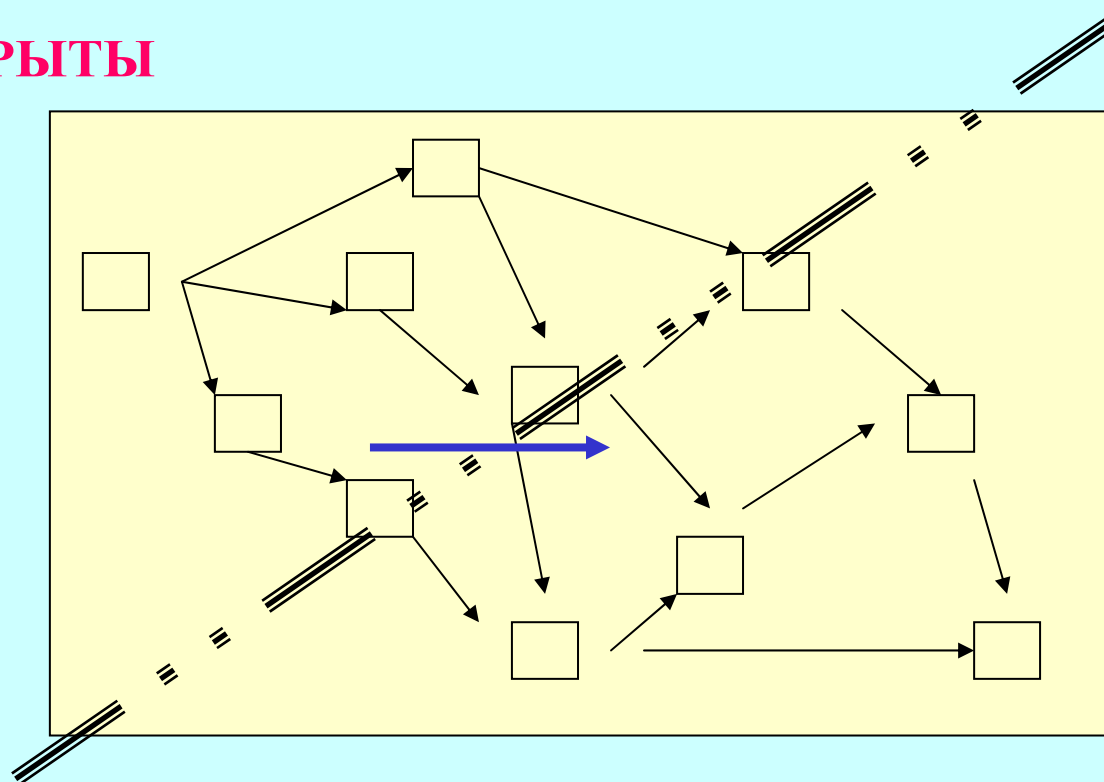
ПЕРЕБОР В «ГЛУБИНУ»



# СТРАТЕГИИ КОМБИНАТОРИКИ

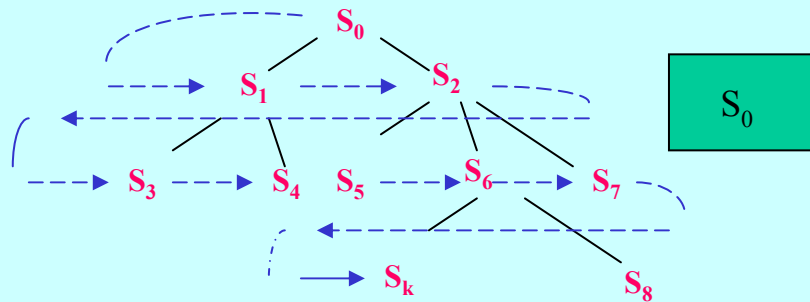
ЗАКРЫТЫ

«ФРОНТ»



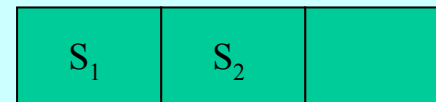
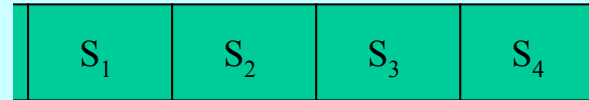
ТРИ МНОЖЕСТВА ВЕРШИН

# ПОЛНЫЙ ПЕРЕБОР В ШИРИНУ (поиск)



$S_0$

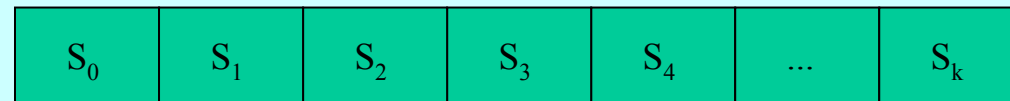
СПИСОК ОТКРЫТ



СТРОКА

$S_k$

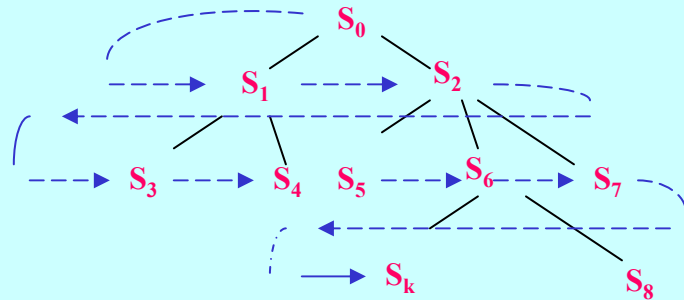
| РОД. ВЕРШ | ДОЧЕРНИЕ ВЕРШИНЫ |       |       |   |
|-----------|------------------|-------|-------|---|
|           | 1                | 2     | 3     | 4 |
| $S_0$     | $S_1$            | $S_2$ |       |   |
| $S_1$     | $S_3$            | $S_4$ |       |   |
| $S_2$     | $S_5$            | $S_6$ | $S_7$ |   |
| $S_5$     | $S_k$            | $S_8$ |       |   |
| $S_0$     |                  |       |       |   |



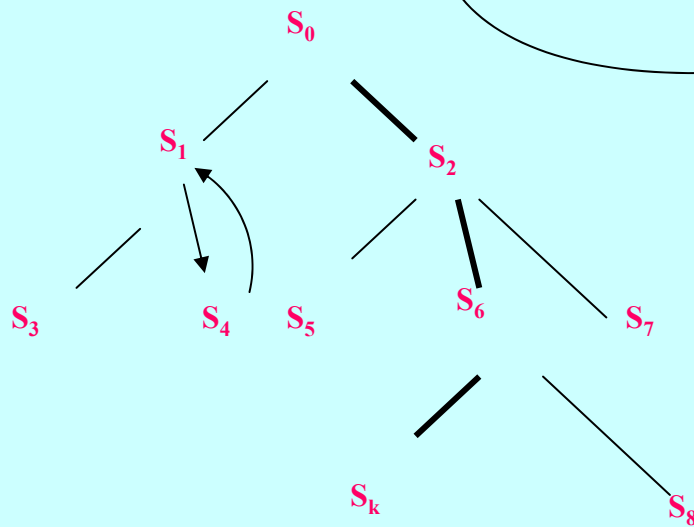
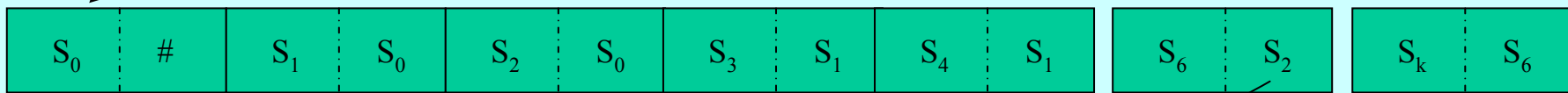
СПИСОК ЗАКРЫТ

=?

# ПОЛНЫЙ ПЕРЕБОР В ШИРИНУ (построение пути)



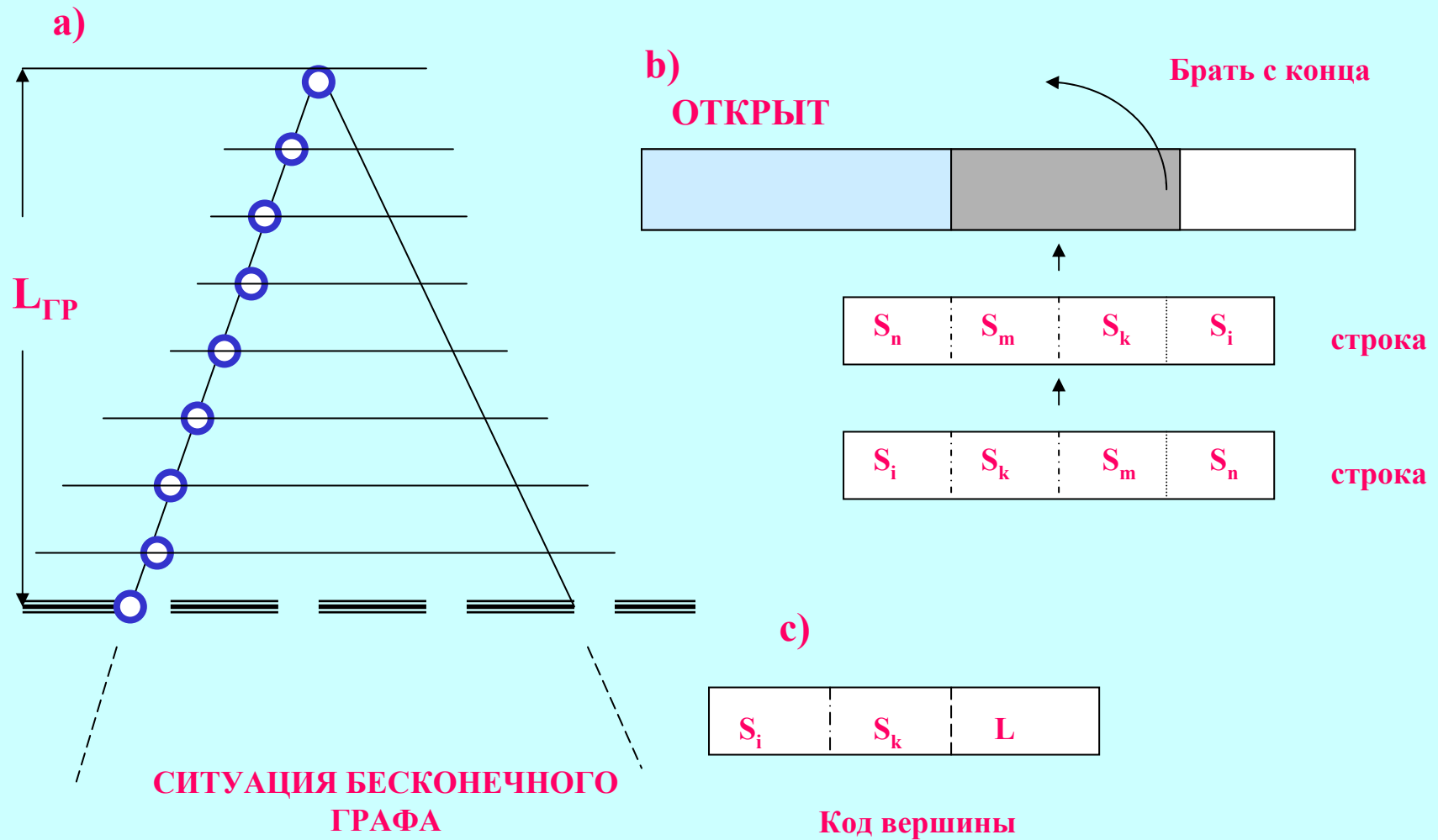
СПИСОК ЗАКРЫТ



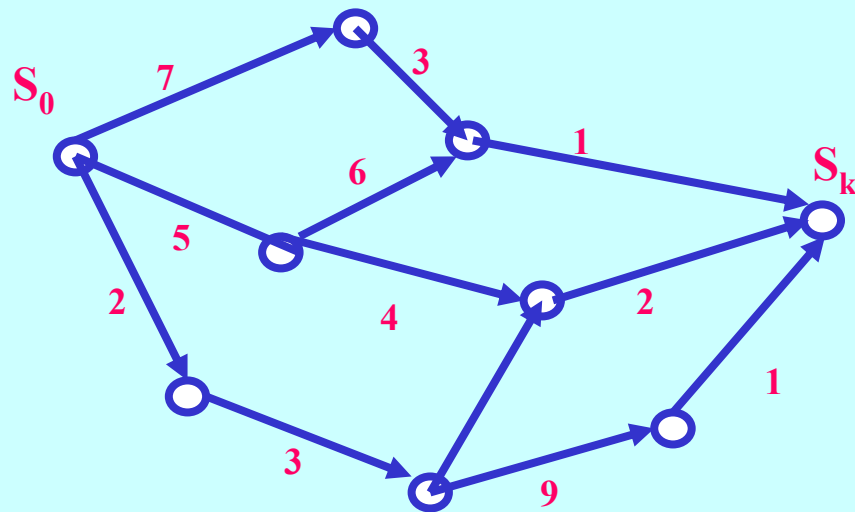
Добавляются проверки на присутствие в списке «ЗАКРЫТ»

# ПОЛНЫЙ ПЕРЕБОР В ГЛУБИНУ

## АДАПТАЦИЯ АЛГОРИТМА



## ПЕРЕБОРЫ С ЦЕНОЙ



с) Код вершины

|       |       |   |          |
|-------|-------|---|----------|
| $S_i$ | $S_k$ | L | $C_{0k}$ |
|-------|-------|---|----------|

$C_{0k} = \sum C_{ij}$  - стоимость пути от  $S_i$   $S_k$

Дополнительная процедура

## АДАПТАЦИЯ АЛГОРИТМА

на поиск для  $\text{MIN}(C_{0k})$

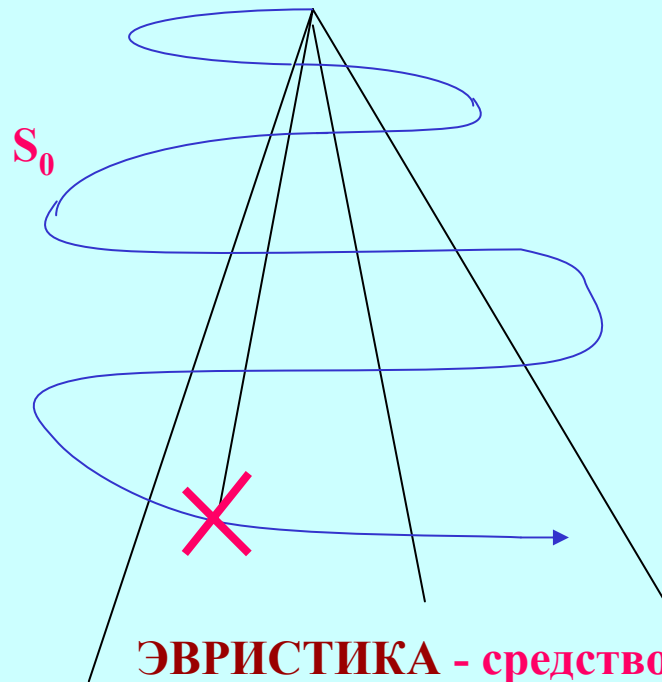
ОТКРЫТ



Ситуация «возврата» по сети - если альтернативный маршрут имеет лучшую «цену», то соответствующее ему представление вершины «замещает» её текущее значение



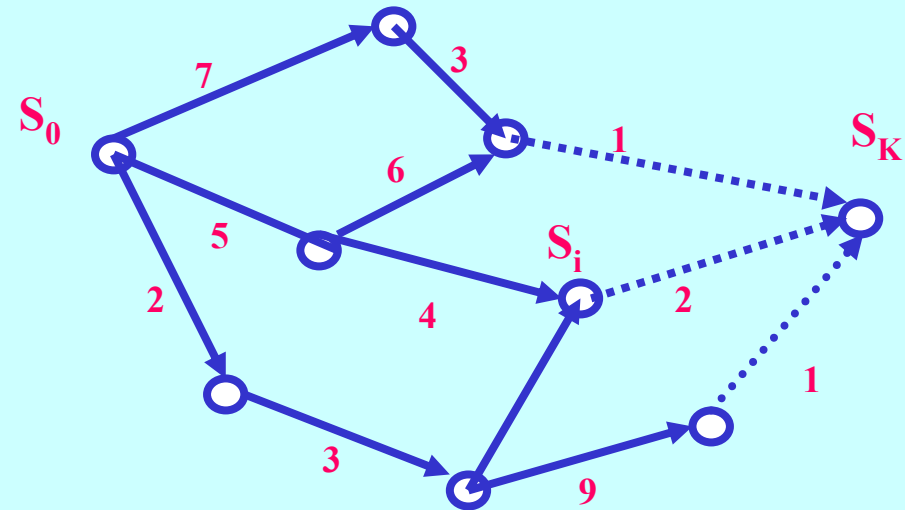
# ПЕРЕБОРЫ С ЭВРИСТИКАМИ



**ЭВРИСТИКА** - средство (идея, правило, принцип, критерий,...), рекомендуемое прекратить комбинаторику по одному или ряду «направлений» - предотвратить комбинаторный взрыв

**ЭВРИСТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ**

## КЛАССИЧЕСКАЯ ЭВРИСТИКА

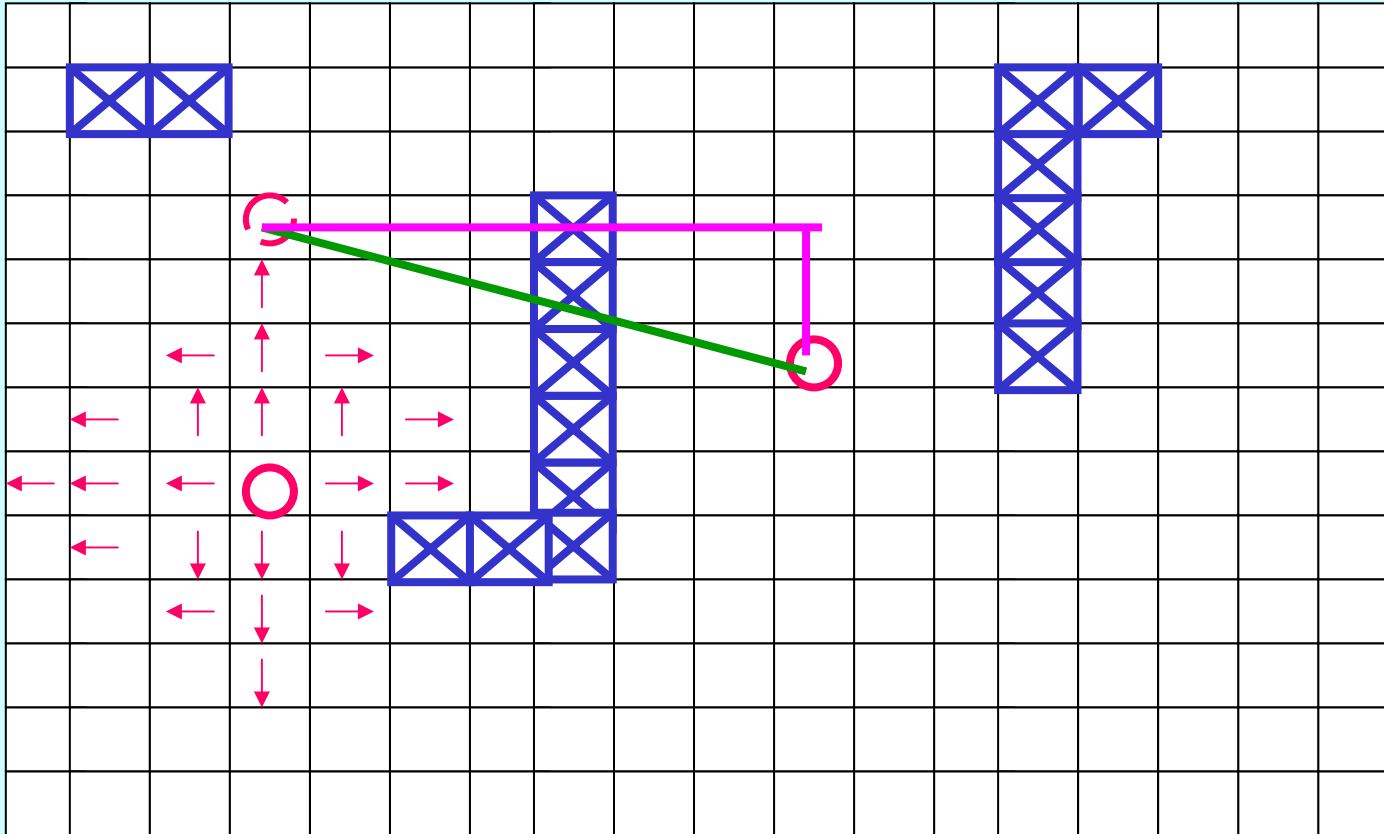


$$\hat{f}(S_i) = c(S_0, S_i) + \hat{h}(S_i, S_K)$$

оценки

$$\hat{h}(S_i, S_K) \leq h(S_i, S_K)$$

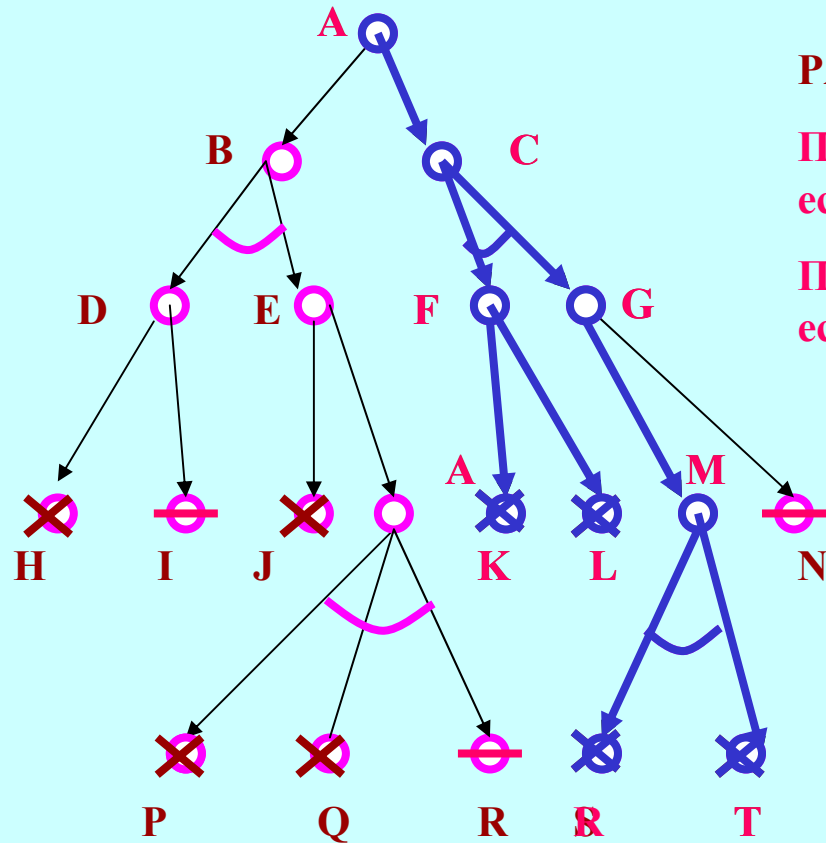
## ПЕРЕБОРЫ С ЭВРИСТИКАМИ(пример)



$$\hat{h}_1 = d_1 = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$$

$$\hat{h}_2 = d_2 = |x_i - x_k| + |y_i - y_k|$$

## 2.2. ПРЕДСТАВЛЕНИЯ НА И/ИЛИ ГРАФАХ



**РАЗМЕТКА** - подтвердить , что A - разрешима

**Правило 1:** вершина типа «И» - разрешима, если разрешимы все дочерние

**Правило 2** вершина типа «ИЛИ» - разрешима, если разрешима хотя бы одна из дочерних

**АЛГОРИТМЫ ПЕРЕБОРА** -  
в ширину, глубину, с ценой, с эвристиками

**ПОСТРОЕНИЕ РЕШЕНИЯ** собрать подграф

**X** - «РАЗРЕШИМА»

**○** - «НЕРАЗРЕШИМА»



## МИНИМАКСНАЯ СТРАТЕГИЯ

|           |           |           |
|-----------|-----------|-----------|
|           | <b>B1</b> | <b>B2</b> |
| <b>A1</b> | 4         | 2         |
| <b>A2</b> | 1         | 8         |

|           |           |           |
|-----------|-----------|-----------|
|           | <b>B1</b> | <b>B2</b> |
| <b>A1</b> | 2         | 9         |
| <b>A2</b> | -3        | 6         |

|           |           |           |
|-----------|-----------|-----------|
|           | <b>B1</b> | <b>B2</b> |
| <b>A1</b> | -2        | 2         |
| <b>A2</b> | 1         | -8        |

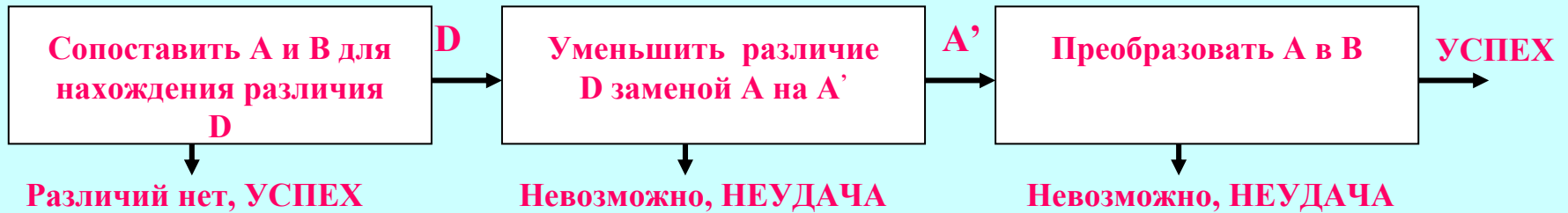
|           |           |           |
|-----------|-----------|-----------|
|           | <b>B1</b> | <b>B2</b> |
| <b>A1</b> | -4        | -2        |
| <b>A2</b> | -1        | -8        |

**Правило: выбор определяется  
максимальной из потенциально  
минимальных потерь противника**

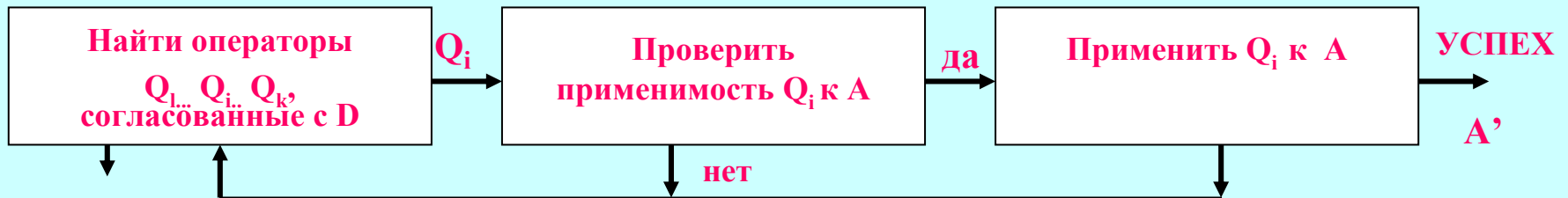
## 2.4. УНИВЕРСАЛЬНЫЙ РЕШАТЕЛЬ ПРОБЛЕМ

### СХЕМА

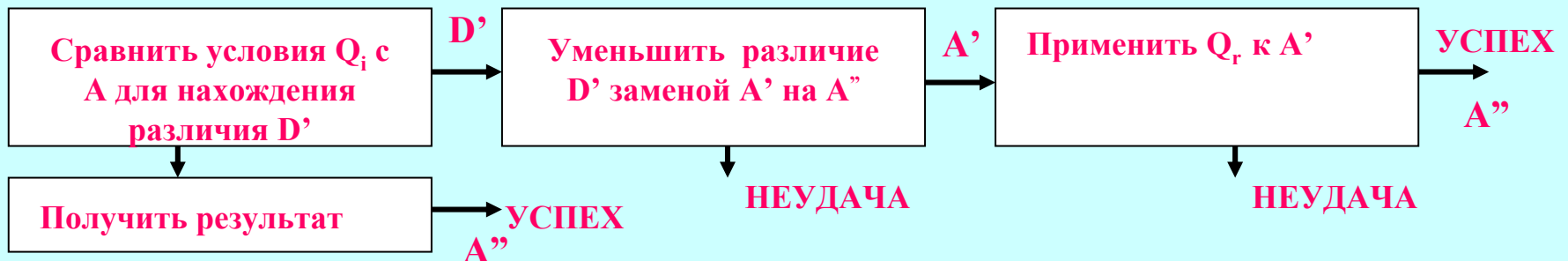
Цель 1. Преобразовать  $A$  в  $B \implies T(A, B)$



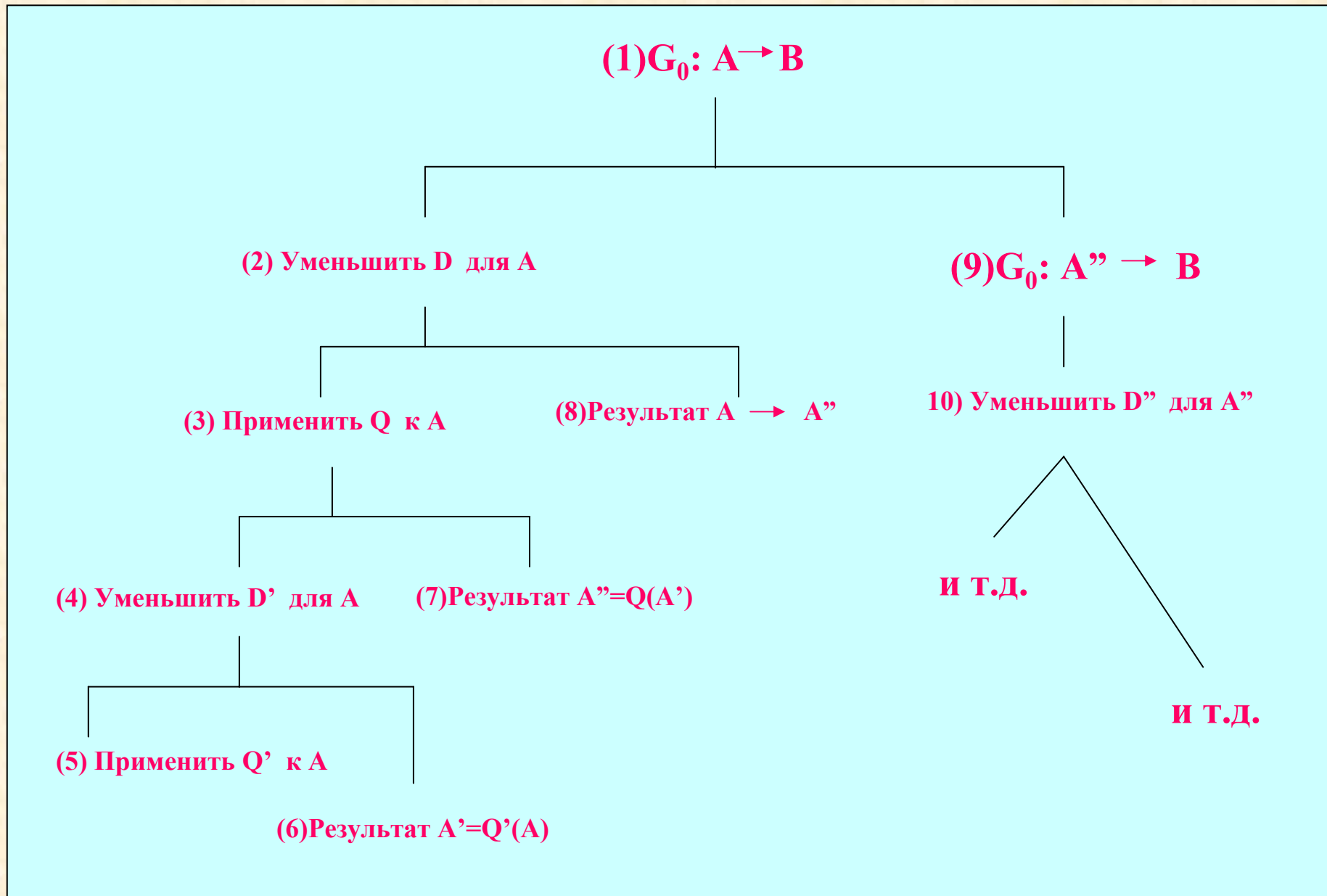
Цель 2. Уменьшить различия между  $A$  и  $B \implies C(Q, D, A, B)$



Цель 3. Применить оператор  $Q$  к  $A \implies E(Q, A)$

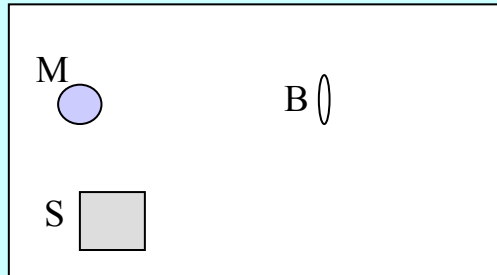


## 2.4. УНИВЕРСАЛЬНЫЙ РЕШАТЕЛЬ ПРОБЛЕМ (GPS)



## 2.4.УНИВЕРСАЛЬНЫЙ РЕШАТЕЛЬ ПРОБЛЕМ

### Задача об обезьяне и банананах



**M-** обезьяна, **S-**ящик,**B-** бананны,  
**множество мест= {p<sub>1</sub>, p<sub>2</sub>, на-S, под-B, ...}**

#### РАЗЛИЧИЯ

**D1-** Место

**D2-** Место(B)

**D3-** Содержимое (рука M)

**Порядок трудности D3>D2>D1**

#### УМЕНЬШЕНИЕ РАЗЛИЧИЙ

|    | D1 | D2 | D3 |
|----|----|----|----|
| O1 | X  |    |    |
| O2 | X  |    |    |
| O3 | X  | X  |    |
| O4 |    |    | X  |

**Операторы:**

#### O1. ПОДНЯТЬСЯ

**УСЛОВИЕ:** Место(M) = Место(S);

**ДЕЙСТВИЕ:** Место(M) = на-S

#### O2. ИДТИ

**УСЛОВИЕ:** переменная x - место;

**ДЕЙСТВИЕ:** Место(M) = x

#### O3. ПЕРЕДВИГАТЬ

**УСЛОВИЕ:** Место(M) ≠ на-S

Место(S)= Место(M)

**ДЕЙСТВИЕ:** Место(M) = x

Место(S) = x

**Операторы:**

#### O4.ХВАТАТЬ

**УСЛОВИЕ:** Место(S) = под-B

Место(M)=на-S;

**ДЕЙСТВИЕ:** Содержимое (рука M) = B



## 2.4. УНИВЕРСАЛЬНЫЙ РЕШАТЕЛЬ ПРОБЛЕМ

Цель 1: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 2: Уменьшить D3:  $T(K_1, K_2)$

Цель 1: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 3: Применить O4 к  $T_1$ :  $E(O4, T_1)$

Цель 4: Уменьшить D2  $T(K_1, K_2)$

Цель 5: Применить O3 к  $K_1$   $T(K_1, K_2)$

Цель 6: Уменьшить D1  $T(K_1, K_2)$

Цель 8: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 7: Применить O2  $T(K_1, K_2)$

Цель 9: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 10: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 12: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$

Цель 11: Преобразовать  $K_1$  в  $K_2$   $T(K_1, K_2)$