

# ЛЕКЦИЯ 3

## УПРАВЛЕНИЕ ПЕРЕБОРОМ. ОТСЕЧЕНИЕ

---

# Статическое отсечение

$A_0: - A_1, A_2, \dots, A_k, !, A_{k+1}, \dots, A_n.$

?  $- A_0.$

- откат для подцелей  $A_1, A_2, \dots, A_k$  не производится;
- для подцелей  $A_{k+1}, \dots, A_n$  откат возможен.

цифра (0) .

цифра (1) : - ! .

цифра (2) .

# Отсечение используется

- для предотвращения ненужных вычислений;
- для моделирования ветвления "Q: если A, то B, иначе C":

Q: - A, !, B.

Q: - C.

- для выражения отрицания. Например, правило P: - not (A) . равносильно совокупности правил:

P: - A, !, fail.

P.

# Динамическое отсечение

```
run() :-  
    male(X),  
        B = programControl::getBackTrack(),  
        parent(X, _),  
        programControl::cutBackTrack(B),  
        write(X), nl,  
    fail;  
_ = readLine().
```

# Динамическое отсечение.

## Замена статическим

```
class predicates
    isParent: (string) determ.
clauses
    isParent (X) :-
        parent (X, _),
        !.

run () :-
    male (X),
        isParent (X),
        write (X), nl,
    fail;
    _ = readLine () .
```

# Предикат `findall` и конструкция `[... || ...]`

- Найти всех родителей:

```
findall(X, parent(X, _), List);  
List = [X || parent(X, _)].
```

- Найти всех персон:

```
findall(X, (male(X); female(X)), List);  
List = [X || male(X); female(X)].
```

- Декартово произведение множества мужчин и множества женщин:

```
List = [tuple(X, Y) || male(X), female(Y)].
```

# Режимы детерминизма предикатов

	> 1 решения	$\leq 1$ решения	0 решений
м. б. ложь	nondeterm	determ	failure
всегда истина	multi	procedure	erroneous

По умолчанию используется режим procedure, если предикат объявляется в разделе (class) predicates, и режим nondeterm, если предикат объявляется в разделе (class) facts.

# Потоки параметров

( $\dot{\phantom{i}}$ ) – входной аргумент

( $\circ$ ) – выходной аргумент

Произвольный поток параметров - `anyflow`.



# Пример «Кино»

Аня, Боря, Витя, Гриша и Даша решают, пойти ли им в кино.

- если пойдет Аня, то пойдет и Боря;
- пойдет либо Витя, либо Гриша, возможно, и оба пойдут;
- точно пойдет либо Боря, либо Даша, но не оба вместе;
- Витя с Дашей либо пойдут вместе, либо вместе не пойдут;
- если пойдет Гриша, то пойдут также Аня и Витя.

# Пример «Кино»

```
class predicates
    proposition: (integer, integer, integer) nondeterm.
    indicator: (integer Индикатор [out]) multi.
    solution: (integer, integer, integer, integer, integer, integer)
        nondeterm (o,o,o,o,o,o).
clauses
    indicator(0).          % не пойдет
    indicator(1).          % пойдет

    % если А пойдет, то и В пойдет
    proposition(1, А, В):- А = 1, В = 1; А = 0.
    % хотя бы кто-то из В и Г пойдет
    proposition(2, В, Г):- В = 1; Г = 1.
    % пойдет либо В, либо Д, но не оба вместе
    proposition(3, В, Д):- В = 1, Д = 0; В = 0, Д = 1.
    % В и Д либо оба пойдут, либо оба не пойдут
    proposition(4, В, Д):- В = 1, Д = 1; В = 0, Д = 0.
```

# Пример «Кино»

```
solution(A, Б, В, Г, Д):-
```

```
    indicator(A), indicator(Б),  
    proposition(1, А, Б),  
    indicator(В), indicator(Г),  
    proposition(2, В, Г),  
    indicator(Д),  
    proposition(3, Б, Д),  
    proposition(4, В, Д),  
    proposition(1, Г, А),  
    proposition(1, Г, В).
```

```
run() :-
```

```
    solution(A, Б, В, Г, Д),  
        write("Аня=", А, ", Боря=", Б, ", Витя=", В, ",  
Гриша=", Г,  
            ", Даша=", Д), nl,  
    fail;  
    _ = readLine().
```

# Пример «Шкатулки»

На золотой шкатулке:

- "Портрет не здесь";
- "Портрет в серебряной шкатулке".

На серебряной шкатулке:

- "Портрет не в золотой";
- "Портрет в свинцовой".

На свинцовой шкатулке:

- "Портрет не здесь";
- "Портрет в золотой".

На одной из шкатулок оба высказывания истинны, на другой оба ложны, на третьей одно истинно, другое ложно. В какой шкатулке находится портрет?

# Пример «Шкатулки»

```
class predicates
    box: (symbol Color) multi (o).
    proposition: (symbol, integer, symbol)
determ.
    statement: (symbol, integer, symbol,
integer Истинность)
        nondeterm (i,i,i,o) determ.
    solution: (symbol ЦветШкатулкиСПортретом)
nondeterm (o).
clauses
    box ("золото") .
    box ("серебро") .
    box ("свинец") .
```

# Пример «Шкатулки»

```
proposition("золото", 1, PortraitBoxColor):-
```

```
    PortraitBoxColor <> "золото".
```

```
proposition("золото", 2, "серебро").
```

```
proposition("серебро", 1, PortraitBoxColor):-
```

```
    PortraitBoxColor <> "золото".
```

```
proposition("серебро", 2, "свинец").
```

```
proposition("свинец", 1, PortraitBoxColor):-
```

```
    PortraitBoxColor <> "свинец".
```

```
proposition("свинец", 2, "золото").
```

```
statement(Box, Number, PortraitBoxColor, 1):-
```

```
    proposition(Box, Number, PortraitBoxColor).
```

```
statement(Box, Number, PortraitBoxColor, 0):-
```

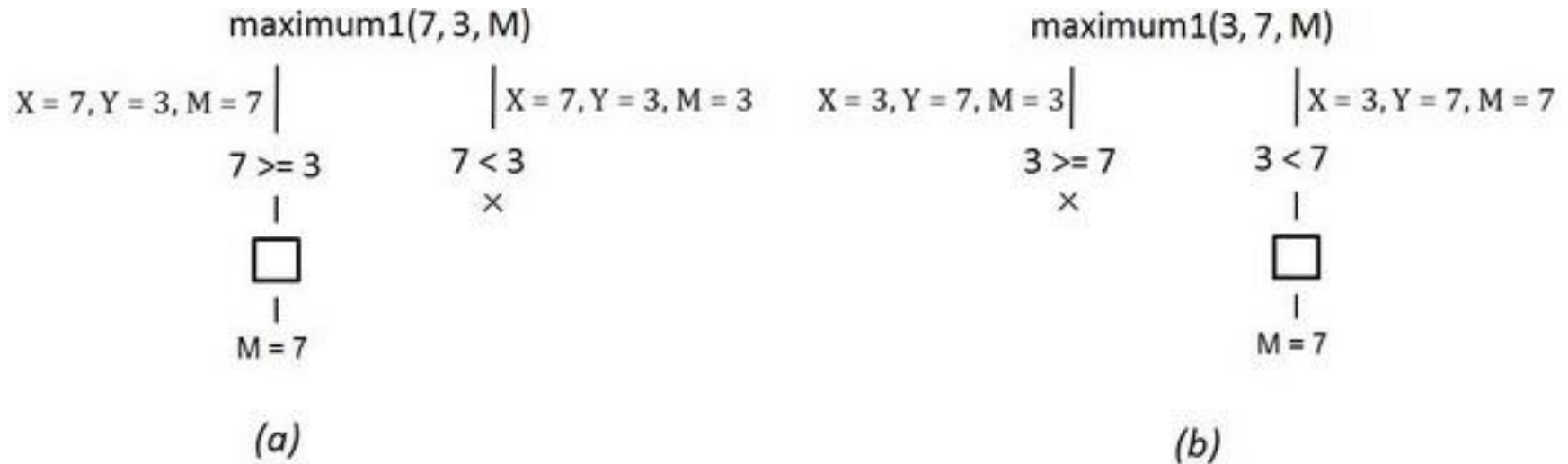
```
    not(proposition(Box, Number, PortraitBoxColor)).
```

# Пример «Шкатулки»

```
solution(PortraitBoxColor):-
    box(PortraitBoxColor),
    box(Color1),
    statement(Color1, 1, PortraitBoxColor, 1),
    statement(Color1, 2, PortraitBoxColor, 1),
    box(Color2), Color2 <> Color1,
    statement(Color2, 1, PortraitBoxColor, 0),
    statement(Color2, 2, PortraitBoxColor, 0),
    box(Color3), Color3 <> Color1, Color3 <> Color2,
    statement(Color3, 1, PortraitBoxColor, X),
    statement(Color3, 2, PortraitBoxColor, 1 - X).

run() :-
    solution(PortraitBoxColor),
    write("Портрет в шкатулке цвета: ",
PortraitBoxColor),
    fail;
    _ = readLine().
```

# Пример. Определение максимума





# Пример. Определение максимума

maximum2(7, 3, M)

X = 3, Y = 7, M = 7 |  
7 >= 3  
|  
☐  
|  
M = 7

(a)

maximum(3, 7, M)

X = 3, Y = 7, M = 3 |  
3 >= 7  
×

X = 3, Y = 7, M = 7  
☐  
|  
M = 7

(b)

# Пример. Определение максимума

```
class predicates
    maximum1: (integer, integer, integer [out]) nondeterm.
    maximum2: (integer, integer, integer [out]) determ.
    maximum: (integer, integer, integer [out]).
clauses
    maximum1(X, Y, X):- X >= Y. % вариант без отсечений
    maximum1(X, Y, Y):- X < Y.

    maximum2(X, Y, X):- X >= Y, !. % зеленое отсечение
    maximum2(X, Y, Y):- X < Y.

    maximum(X, Y, X):- X >= Y, !. % красное отсечение
    maximum(_, Y, Y).
```

# Пример. Определение максимума

```
run() :-  
    maximum1(3, 7, M),    % maximum1(7, 3, M),  
        write(M), nl,  
    fail;  
maximum2(3, 7, M),  
write(M), nl,  
fail;  
maximum(3, 7, M),  
write(M),  
_ = readLine().
```

# Пример «Квадратное уравнение»

```
class predicates
    solution: (real A, real B, real C, real* [out],
integer Mark [out]).
    roots: (real A, real B, real D, real* Решение [out]).
    print: (integer, real* Решение).
clauses
    solution(0, 0, 0, [], -1):- !.
    solution(0, 0, _, [], 0):- !.
    solution(0, B, C, [-C/B], 1):- !.
    solution(A, B, C, L, 2):-
        roots(A, B, B^2 - 4 * A * C, L).

    roots(A, B, 0, [-B/(2 * A)]):- !.
    roots(_, _, D, []):- D < 0, !.
    roots(A, B, D, [(-B + Q)/(2 * A), (-B - Q)/(2 * A)]):-
        Q = math::sqrt(D * 1).
```

# Пример «Квадратное уравнение»

```
print(-1, _):- !, write("Решение - любое число").
print(0, _):- !, write("Уравнение линейное, решений нет.").
print(1, [X]):- !, write("X = ", X).
print(2, [X]):- !, write("X1 = X2 = ", X).
print(2, [X1, X2]):- !, write("X1 = ", X1, ", X2 = ", X2).
print(_, _):- write("Решений нет").
```

```
run() :-
```

```
    write("Введите коэффициенты уравнения "
          "A*X^2 + B*X + C = 0\nA = "),
    A = read(), clearInput(),
    write("B = "),
    B = read(), clearInput(),
    write("C = "),
    C = read(), clearInput(),
    solution(A, B, C, L, Mark),
    print(Mark, L),
    _ = readLine().
```

# Пример «Генерация двузначных чисел»

```
class facts
    digit: (integer).
class predicates
    twoDigitNumber: (integer) nondeterm (o).
clauses
    digit(0).      digit(1).      digit(2).      digit(3).      digit(4).
    digit(5).      digit(6).      digit(7).      digit(8).      digit(9).

    twoDigitNumber(10 * A + B):-
        digit(A), A > 0, digit(B).
run():-
    twoDigitNumber(X),
        write(X), nl,
    fail;
    write("Список четных двузначных чисел:\n"),
    L = [10 * A + B || digit(A), A > 0, digit(B), B mod 2 = 0],
    write(L),
    _ = readLine().
```