

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 18

Формирование эффективных программ на Prolog

Дисциплина Функциональное и логическое программирование

Студент Сиденко А.Г.

Группа ИУ7-63Б

Преподаватель Толпинская Н.Б., Строганов Ю.В.

Задание

Используя хвостовую рекурсию, разработать программу, позволяющую найти

- 1. n!
- 2. п-е число Фибоначчи

Убедиться в правильности результатов.

Для одного из вариантов вопроса и каждого задания составить таблицу, отражающую конкретный порядок работы системы

Программа

```
predicates
1
2
     factorial (integer, integer).
3
     fibonachi (integer, integer).
   clauses
4
     factorial(0, 1):- !.
5
     factorial(Next, FactorialNext):-
6
       Cur = Next - 1,
7
       factorial (Cur, Factorial Cur),
8
9
       FactorialNext = FactorialCur * Next.
10
     fibonachi (0, 0):-!.
11
     fibonachi (1, 1):-!.
12
     fibonachi (Next, Fibonachi Next):-
13
       Cur = Next - 1,
14
       Prev = Cur - 1,
15
       fibonachi (Cur, Fibonachi Cur),
16
       fibonachi (Prev, Fibonachi Prev),
17
       FibonachiNext = FibonachiCur + FibonachiPrev.
18
```

Приведем таблицу для задания 1.

```
1 goal
2 factorial (5, Rez).
```

$N_{\overline{0}}$	Состояние резольвен-	Сравниваемые термы; ре-	Дальнейшие действия:
ша-	- ТЫ	зультат; подстановка, если	прямой ход или откат
га		есть	
1	factorial(5, Rez)	По factorial(5, Rez) ищется	Определение отношения
		системой определение от-	найдено, заносится в
		ношения (по имени преди-	стек factorial(5, Rez),
		ката и списку (числу) аргу-	прямой ход
		ментов)	

2	Cur = 5 - factorial(Cur, FactorialCur), FactorialNext FactorialCur * 5	1,	Начинает «раскрываться» правило, т.е. доказывается каждое целевое утверждение в теле правила последовательно слева направо Cur = Next - 1, factorial(Cur, FactorialCur), FactorialNext = FactorialCur * Next	Прямой ход
3	Cur = 5 - factorial(Cur, FactorialCur), FactorialNext FactorialCur * 5	1,	Cur = 5 - 1	Значение утверждения true, Cur = 4, переход к следующему определению
4	factorial(4, FactorialCur), FactorialNext FactorialCur * 5		По factorial(4, FactorialCur) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	Определение отношения найдено, заносится в стек factorial(4, FactorialCur), прямой ход
5	Cur = 4 - factorial(Cur, FactorialCur), FactorialNext FactorialCur * FactorialNext FactorialCur * 5	1, = 4, =	Cur = 4 - 1	Значение утверждения true, Cur = 3, переход к следующему определению
6	factorial(3, FactorialCur), FactorialNext FactorialCur FactorialNext FactorialCur * 5	= 4, =	По factorial(3, FactorialCur) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	Определение отношения найдено, заносится в стек factorial(3, FactorialCur), прямой ход
7	Cur = 3 - factorial(Cur, FactorialCur), FactorialNext FactorialCur * FactorialCur * FactorialCur * FactorialCur * FactorialCur * FactorialNext FactorialNext FactorialCur * 5	1, = 3, = 4, =	Cur = 4 - 1	Значение утверждения true, Cur = 2, переход к следующему определению

8	factorial(2,			Πο factorial(2, FactorialCur)	Определение отноше-
	FactorialCur),			ищется системой определе-	ния найдено, заносит-
	FactorialNext		=	ние отношения (по имени	ся в стек factorial(2,
	FactorialCur	*	3,	предиката и списку (числу)	FactorialCur), прямой
	FactorialNext		=	аргументов)	, , ,
	FactorialCur	*	4,	аргументов)	ход
	FactorialNext		4 ,		
	FactorialCur * 5	5	_		
9	Cur = 2	_	1,	Cur = 4 - 1	Значение утверждения
	factorial(Cur,		Τ,		true, Cur = 1, переход к
	FactorialCur),				следующему определе-
	FactorialNext		=		нию
	FactorialCur	*	2,		IIIIIO
	FactorialNext		<u></u>		
	FactorialCur	*	3,		
	FactorialNext		=		
	FactorialCur	*	4,		
	FactorialNext		- ,		
	FactorialCur * 5	5			
10	factorial(1,			По factorial(1, FactorialCur)	Определение отноше-
	FactorialCur),			ищется системой определе-	ния найдено, заносит-
	FactorialNext		_	ние отношения (по имени	ся в стек factorial(1,
	FactorialCur	*	2,	предиката и списку (числу)	FactorialCur), прямой
	FactorialNext		<i>□</i> , =	аргументов)	ход
	FactorialCur	*	3,	api y Meniiob)	лод
	FactorialNext		=		
	FactorialCur	*	4,		
	FactorialNext		=		
	FactorialCur * 5	á			
11	Cur = 1		1,	Cur = 4 - 1	Значение утверждения
	factorial(Cur,		Τ,		true, Cur = 0, переход к
	FactorialCur),				следующему определе-
	FactorialNext		=		нию
	FactorialCur	*	1,		IIIIIO
	FactorialNext		=		
	FactorialCur	*	2,		
	FactorialNext		=		
	FactorialCur	*	3,		
	FactorialNext		=		
	FactorialCur	*	4,		
	FactorialNext		- ,		
	FactorialCur * 5	á			
	1 acountaioui c	,			

	0 1/0		
12	factorial(0,	Π o factorial $(0, FactorialCur)$	Определение отноше-
	FactorialCur),	ищется системой определе-	ния найдено, заносит-
	FactorialNext =	ние отношения (по имени	ся в стек factorial(0,
	FactorialCur * 1,	предиката и списку (числу)	FactorialCur), прямой
	FactorialNext =	аргументов)	ход
	FactorialCur * 2,		
	FactorialNext =		
	FactorialCur * 3,		
	FactorialNext =		
	FactorialCur * 4,		
	FactorialNext =		
	FactorialCur * 5		
13	FactorialNext = 1	Унификация factorial(0,	FactorialCur = 1, извле-
	* 1, FactorialNext	FactorialCur) и factorial(0,	чение из стека factorial $(0,$
	= FactorialCur *	1)	FactorialCur)
	2, FactorialNext		
	= FactorialCur *		
	3, FactorialNext		
	= FactorialCur *		
	4, FactorialNext =		
	FactorialCur * 5		
14	FactorialNext = 1	FactorialNext = 1	Извлечение из стека
	* 2, FactorialNext		factorial(1, FactorialCur)
	= FactorialCur *		
	3, FactorialNext		
	= FactorialCur *		
	4, FactorialNext =		
	FactorialCur * 5		
15	FactorialNext = 2	FactorialNext = 2	Извлечение из стека
	* 3, FactorialNext		factorial(2, FactorialCur)
	= FactorialCur *		,
	4, FactorialNext =		
	FactorialCur * 5		
16	FactorialNext = 6 *	FactorialNext = 6	Извлечение из стека
	4, FactorialNext =		factorial(3, FactorialCur)
	FactorialCur * 5		(3, 2 3333141 (31)
17	FactorialNext = $24 * 5$	FactorialNext = 24	Извлечение из стека
		2 3 5 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7 5 7	factorial(4, FactorialCur)
18		FactorialNext = 120	Резольвента пуста, вы-
		1 4000110111070 — 120	вод результата
19			Стек пуст, завершение
13			программы
1		I	HOOL DAMINDI

Приведем таблицу для задания 2.

1 goal 2 fibonachi (4, Rez).

$N_{\overline{0}}$	Состояние резольвен-	Сравниваемые термы; ре-	Дальнейшие действия:
ша-	• ТЫ	зультат; подстановка, если	прямой ход или откат
га		есть	*
1	fibonachi(4, Rez)	По fibonachi(4, Rez) ищется	Определение отношения
	, , ,	системой определение от-	найдено, заносится в
		ношения (по имени преди-	стек fibonachi(4, Rez),
		ката и списку (числу) аргу-	прямой ход
		ментов)	
2	Cur = 4 - 1, Prev =	Начинает «раскрываться»	Прямой ход
	Cur - 1, fibonachi(Cur,	правило, т.е. доказывается	
	FibonachiCur),	каждое целевое утвер-	
	fibonachi(Prev,	ждение в теле правила	
	FibonachiPrev),	последовательно слева на-	
	FibonachiNext =	право Cur = Next - 1, Prev	
	FibonachiCur +	= Cur - 1, fibonachi(Cur,	
	FibonachiPrev	FibonachiCur),	
		fibonachi(Prev,	
		FibonachiPrev),	
		FibonachiNext =	
		FibonachiCur +	
0	D 0	FibonachiPrev	n
3	$\begin{array}{ccc} \text{Prev} & = & 3 & - \\ & & & & 1 & 1 & 2 \\ \end{array}$	Cur = 4 - 1	Значение утверждения
	$1, \text{fibonachi}(3, \dots)$		true, $Cur = 3$, переход к
	FibonachiCur),		следующему определе-
	fibonachi(Prev,		НИЮ
	FibonachiPrev),		
	FibonachiNext = FibonachiCur +		
	FibonachiCur + FibonachiPrev		
4	fibonachi(3,	Prev = 3 - 1	Значение утвержде-
4	FibonachiCur),	1 1 CV — 3 - 1	Значение утверждения true, Prev = 2,
	fibonachi(2,		переход к следующему
	FibonachiPrev),		определению
	FibonachiNext =		определению
	FibonachiCur +		
	FibonachiPrev		
	1 100Hachii 1cv		

5	fibonachi(3,	По fibonachi(3,	Определение отноше-
	FibonachiCur),	FibonachiCur) ищется	ния найдено, заносится
	fibonachi(2,	системой определение	в стек fibonachi(3,
	FibonachiPrev),	отношения (по имени пре-	FibonachiCur), прямой
	FibonachiNext =	диката и списку (числу)	ход
	FibonachiCur +	аргументов)	
	FibonachiPrev	,	
6	Cur = 3 - 1, Prev =	Начинает «раскрываться»	Прямой ход
	Cur - 1, fibonachi(Cur,	правило, т.е. доказывается	
	FibonachiCur),	каждое целевое утвер-	
	fibonachi(Prev,	ждение в теле правила	
	FibonachiPrev),	последовательно слева на-	
	FibonachiNext =	право $Cur = Next - 1$, $Prev$	
	FibonachiCur +	= Cur - 1, fibonachi(Cur,	
	FibonachiPrev,	FibonachiCur),	
	fibonachi(2,	fibonachi(Prev,	
	FibonachiPrev),	FibonachiPrev),	
	FibonachiNext =	FibonachiNext =	
	FibonachiCur +	FibonachiCur +	
	FibonachiPrev	FibonachiPrev	
7	Prev = 2 -	$\operatorname{Cur} = 3$ - 1	Значение утверждения
	1, fibonachi $(2,$		true, Cur = 2, переход к
	FibonachiCur),		следующему определе-
	fibonachi(Prev,		нию
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		

8	fibonachi(2,	Prev = 2 - 1	Значение утвержде-
	FibonachiCur),	1 1 CV = 2 - 1	$\frac{\text{утвержде}}{\text{ния true, Prev}} = 1,$
	fibonachi(1,		переход к следующему
	FibonachiPrev),		определению
	FibonachiNext =		определению
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
9	fibonachi(2,	Πο fibonachi(2,	Определение отноше-
	FibonachiCur),	FibonachiCur) ищется	ния найдено, заносится
	fibonachi(1,	системой определение	в стек fibonachi(2,
	FibonachiPrev),	отношения (по имени пре-	FibonachiCur), прямой
	FibonachiNext =	диката и списку (числу)	ход
	FibonachiCur +	аргументов)	ХОД
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
10	Cur = 2 - 1, Prev =	Начинает «раскрываться»	Прямой ход
	Cur - 1, fibonachi(Cur,	правило, т.е. доказывается	•
	FibonachiCur),	каждое целевое утвер-	
	fibonachi(Prev,	ждение в теле правила	
	FibonachiPrev),	последовательно слева на-	
	FibonachiNext =	право Cur = Next - 1, Prev	
	FibonachiCur +	= Cur - 1, fibonachi(Cur,	
	FibonachiPrev,	FibonachiCur),	
	fibonachi(1,	fibonachi(Prev,	
	FibonachiPrev),	FibonachiPrev),	
	FibonachiNext =	FibonachiNext =	
	FibonachiCur +	FibonachiCur +	
	FibonachiPrev,	FibonachiPrev	
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		

11	D	0 0 1	n
11	$\begin{array}{cccc} \operatorname{Prev} & = & 1 & - & - & - & - & - & - & - & - & -$	Cur = 2 - 1	Значение утверждения
	1, fibonachi(1,		true, Cur = 1, переход к
	FibonachiCur),		следующему определе-
	fibonachi(Prev,		нию
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(1,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
12	fibonachi(1,	Prev = 1 - 1	Значение утвержде-
	FibonachiCur),		ния $true$, $Prev = 0$,
	fibonachi(0,		переход к следующему
	FibonachiPrev),		определению
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(1,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		

19	fhonochi(1	Πο fborochi/1	Ombo no novivo
13	fibonachi(1,	Πο fibonachi(1,	Определение отноше-
	FibonachiCur),	FibonachiCur) ищется	ния найдено, заносится
	fibonachi(0,	системой определение	в стек fibonachi(1,
	FibonachiPrev),	отношения (по имени пре-	FibonachiCur), прямой
	FibonachiNext =	диката и списку (числу)	ход
	FibonachiCur +	аргументов)	
	FibonachiPrev,		
	fibonachi(1,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
14	fibonachi(0,	Унификация fibonachi(1,	FibonachiCur =
	FibonachiPrev),	FibonachiCur) и	1, извлекается из
	FibonachiNext =	fibonachi(1, 1)	стека fibonachi(1,
	FibonachiCur +		FibonachiCur), прямой
	FibonachiPrev,		ход
	fibonachi(1,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
	<u> </u>		

1 5	C1 1:/0	П С1 1:/0	
15	fibonachi(0,	Π o fibonachi $(0, \frac{1}{2})$	Определение отноше-
	FibonachiPrev),	FibonachiCur) ищется	ния найдено, заносится
	FibonachiNext = 1	системой определение	B ctek fibonachi $(0, B)$
	+ FibonachiPrev,	отношения (по имени пре-	FibonachiCur), прямой
	fibonachi(1,	диката и списку (числу)	ход
	FibonachiPrev),	аргументов)	
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		
16	FibonachiNext = 1	Унификация fibonachi(0,	FibonachiPrev =
	+ FibonachiPrev,	FibonachiCur) и	0, извлекается из
	fibonachi(1,	fibonachi(0, 0)	стека fibonachi(0,
	FibonachiPrev),	(*, *)	FibonachiCur), прямой
	FibonachiNext =		ход
	FibonachiCur +		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
17	FibonachiPrev FibonachiNext =	FibonachiNext = $1 + 0$	FibonachiCur = 1, пря-
11	1 + 0, fibonachi(1,		мой ход
	FibonachiPrev),		мои ход
	1.37		
	'		
	FibonachiPrev,		
	fibonachi(2,		
	FibonachiPrev),		
	FibonachiNext =		
	FibonachiCur +		
	FibonachiPrev		

18	fibonachi(1, FibonachiPrev), FibonachiNext = 1 + FibonachiPrev, fibonachi(2, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev	По fibonachi(1, FibonachiPrev) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	Определение отношения найдено, заносится в стек fibonachi(1, FibonachiPrev), прямой ход
19	FibonachiNext = 1 + FibonachiPrev, fibonachi(2, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev	Унификация fibonachi(1, FibonachiPrev) и fibonachi(1, 1)	1, извлекается из стека fibonachi(1, FibonachiCur), прямой ход
20	FibonachiNext = 1 + 1, fibonachi(2, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev	${ m FibonachiNext}=1+1$	FibonachiCur = 2, прямой ход
21	fibonachi(2, FibonachiPrev), FibonachiNext = 2 + FibonachiPrev	По fibonachi(2, FibonachiPrev) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	ния найдено, заносится в стек fibonachi(2,
22	Cur = 2 - 1, Prev = Cur - 1, fibonachi(Cur, Fibonachi(Prev, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	Начинает «раскрываться» правило, т.е. доказывается каждое целевое утверждение в теле правила последовательно слева направо Cur = Next - 1, Prev = Cur - 1, fibonachi(Cur, FibonachiCur), fibonachi(Prev, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev	Прямой ход

23	Prev = 1 - 1, fibonachi(1, FibonachiCur), fibonachi(Prev, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	Cur = 2 - 1	Значение утверждения true, Cur = 1, переход к следующему определению
24	fibonachi(1, FibonachiCur), fibonachi(0, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	Prev = 1 - 1	Значение утверждения true, Prev = 0, переход к следующему определению
25	fibonachi(1, FibonachiCur), fibonachi(0, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	По fibonachi(1, FibonachiCur) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	Определение отношения найдено, заносится в стек fibonachi(1, FibonachiCur), прямой ход
26	fibonachi(0, FibonachiPrev), FibonachiNext = FibonachiCur + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	Унификация fibonachi(1, FibonachiCur) и fibonachi(1, 1)	FibonachiCur = 1, извлекается из стека fibonachi(1, FibonachiCur), прямой ход
27	fibonachi(0, FibonachiPrev), FibonachiNext = 1 + FibonachiPrev, FibonachiNext = 2 + FibonachiPrev	По fibonachi(0, FibonachiCur) ищется системой определение отношения (по имени предиката и списку (числу) аргументов)	Определение отношения найдено, заносится в стек fibonachi(0, FibonachiCur), прямой ход

28	FibonachiNext = 1	Унификация fibonachi(0,	FibonachiPrev =
	+ FibonachiPrev,	FibonachiCur) и	0, извлекается из
	FibonachiNext = 2 +	fibonachi(0, 0)	стека fibonachi(0,
	FibonachiPrev		FibonachiCur), прямой
			ход
29	FibonachiNext = 1 +	FibonachiNext = 1 + 0	FibonachiPrev =
	0, FibonachiNext = $2 +$		1, извлекается из
	FibonachiPrev		стека fibonachi(3,
			FibonachiCur), прямой
			ход
30		FibonachiNext = 2 + 1	FibonachiNext =
			3, извлекается из
			стека fibonachi(4,
			FibonachiCur), Pe-
			зольвента пуста, вывод
			результата
31			Стек пуст, завершение
			программы

Вывод

Эффективный способ организации рекурсии – хвостовая рекурсия. Эффективность рекурсивной процедуры повышается благодаря отсечению неперспективных путей поиска решения. Используя «!» – отсечение. Которое сократит количество выполняемых унификаций для достижения максимальной эффективности работы системы.

Ответы на вопросы

1. Что такое рекурсия? Как организуется хвостовая рекурсия в Prolog? Как организовать выход из рекурсии в Prolog?

Рекурсия позволяет использовать в процессе определения предиката его самого.

Хвостовая рекурсия: Для ее осуществления рекурсивный вызов определяемого предиката должен быть последней подцелью в теле рекурсивного правила и к моменту рекурсивного вызова не должно остаться точек возврата (непроверенных альтернатив).

Параметры должны изменяться на каждом шаге так, чтобы в итоге либо сработал базис рекурсии, либо условие выхода из рекурсии, размещенное в самом правиле.

2. Какое первое состояние резольвенты? Вопрос.

3. В каком случае система запускает алгоритм унификации? Каково назначение использования алгоритма унификации? Каков результат работы алгоритма унификации?

Пролог выполняет унификацию в двух случаях: когда цель сопоставляется с заголовком предложения или когда используется знак равенства, который является инфиксным предикатом (предикатом, который расположен между своими аргументами, а не перед ними).

Унификация двух термов — это основной шаг доказательства. В процессе работы система выполняет большое число унификаций. **Унификация** — операция, которая позволяет формализовать процесс логического вывода.

Унификация представляет собой процесс сопоставления цели с фактами и правилами базы знаний. Цель может быть согласована, если она может быть сопоставлена с заголовком какого-либо предложения базы.

Результатом его работы является последняя из построенных подстановок.

4. В каких пределах программы уникальны переменные?

Областью действия переменной в Прологе является одно предложение. В разных предложениях может использоваться одно имя переменной для обозначения разных объектов. Исключением является анонимная переменная. Каждая анонимная переменная — это отдельный объект.

5. Как применяется подстановка, полученная с помощью алгоритма унификации?

При согласовании переменные получают значения, указанные с другой стороны от знака «=», если переменные еще не были связаны. Переменные становятся связанными и после успешного согласования всех целевых утверждений, будет напечатано значение связанных переменных.

6. Как изменяется резольвента?

Резольвента - текущая цель, существующая на любой стадии вычислений. Резольвенты порождаются целью и каким-либо правилом или фактом, которые просматриваются последовательно сверху вниз. Если резольвента существует при наиболее общей унификации, она вычисляется. Если пустая резольвента с помощью такой стратегии не найдена, то ответ на вопрос отрицателен.

7. В каких случаях запускается механизм отката?

Откат дает возможность получить много решений в одном вопросе к программе.

Во всех точках программы, где существуют альтернативы, в стек заносятся точки возврата.

Если впоследствии окажется, что выбранный вариант не приводит к успеху, то осуществляется откат к последней из имеющихся в стеке точек программы, где был выбран один из альтернативных вариантов.

Выбирается очередной вариант, программа продолжает свою работу. Если все варианты в точке уже были использованы, то регистрируется неудачное завершение и осуществляется переход на предыдущую точку возврата, если такая есть.

При откате все связанные переменные, которые были означены после этой точки, опять освобождаются.