

Теоретическое программирование

- Математические основы программирования;
- Теория схем программ;
- Семантическая теория программ;
- Теория параллельных вычислений;
- Прикладные задачи теоретического программирования.

Схемы программ

Программа – способ задания алгоритма.

Свойства программ:

- является конструктивным объектом;
- работает конечное время;
- характерны массовость и однозначность.

Схемы программ – математические модели программ.

Свойства схем программ:

- позволяют изучать свойства широких классов программ;
- сохраняют все свойства и особенности рассматриваемого класса программ;
- позволяют игнорировать несущественные свойства;
- изобразительно подобны программе.

Стандартные схемы программ

Класс стандартных схем **включает**:

- константы;
- простые переменные;
- выражения;
- операторы присваивания;
- условные операторы;
- метки;
- переходы на метки.

Класс стандартных схем **характеризуется**:

- базисом класса;
- структурой схем.

Базис В класса стандартных схем состоит:

- 4 счетных множества символов;
- множество операторов.

Множества символов:

1. Переменные:

$$X = \{x_1, x_2 \dots x_n; y, y_1, y_2 \dots; z, z_1, z_2 \dots\};$$

2. Функциональные символы:

$$F = \{f^{(0)}, f^{(1)}, f^{(2)} \dots; g^{(0)}, g^{(1)}, g^{(2)} \dots; h^{(0)}, h^{(1)}, h^{(2)} \dots\};$$

3. Предикатные символы:

$$P = \{p^{(0)}, p^{(1)}, p^{(2)} \dots; q^{(0)}, q^{(1)}, q^{(2)} \dots\};$$

4. Специальные символы:

старт, стоп, (,), := и т. д.

Множество операторов:

1) начальный оператор:

старт($x_1, x_2 \dots x_k$);

2) заключительный оператор:

стоп ($t_1, t_2 \dots t_n$);

3) оператор присваивания:

$x := t$;

4) условный оператор (тест);

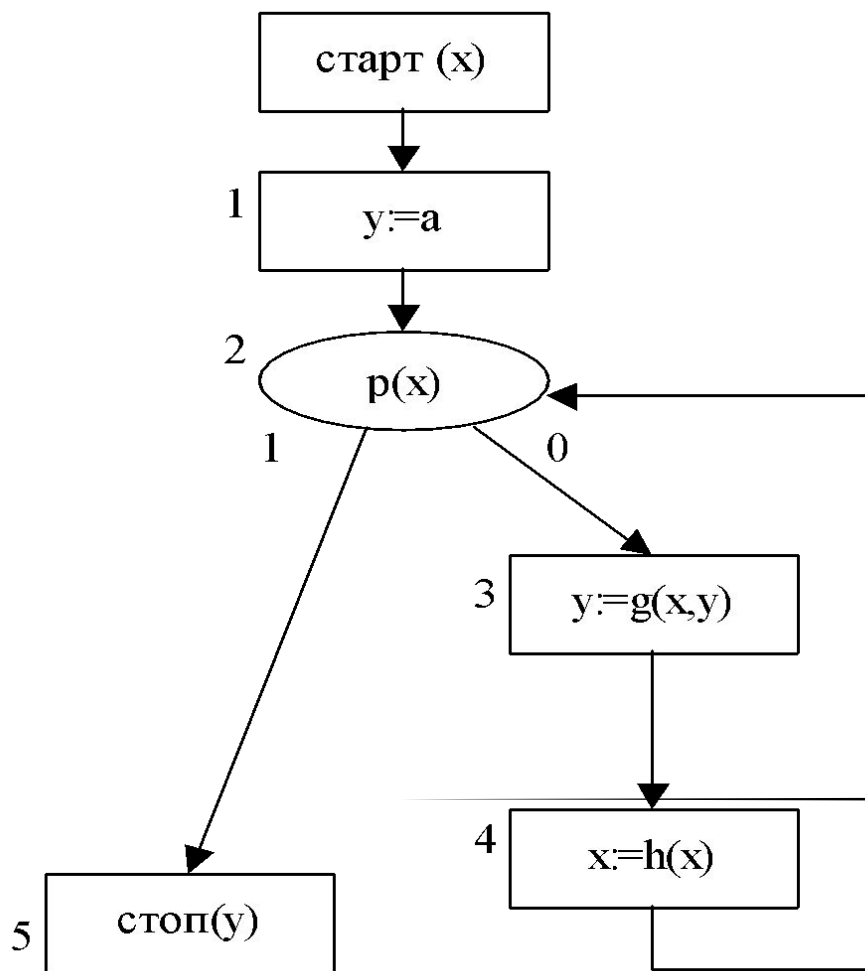
5) оператор петли.

Программа:

```
void main(void)
{ int x, y;
  cin>>x;
  y=1;
  while (x>0)
  { y=x*y;
    x--;
  }
  cout<<y;
}
```

Схема программы:

старт (x),
0: старт (x) на 1,
1: $y := a$ на 2,
2: если $p(x)$ то 5 иначе 3,
3: $y := g(x, y)$,
4: $x := h(x)$ на 2,
5: стоп (y).



Интерпретация:
 область интерпретации D -
 множество целых
 неотрицательных чисел;

$I(x)=4$; $I(y)=0$; $I(a)=1$;
 $I(g)=G$, где $G(d1,d2)=$
 $d1*d2$;

$I(h)=H$, где $H(d)=d-1$;

$I(p)=P$, где P - предикат

$P(d)=\begin{cases} 0, & \text{если } d>0 \\ 1, & \text{иначе} \end{cases}$

Конфигурация		U_0	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}
Метка		0	1	2	3	4	2	3	4	2	3	4	2	3
Значения	X	4	4	4	4	3	3	3	2	2	2	1	1	1
	Y	0	1	1	4	4	4	12	12	12	24	24	24	24

Рекурсивные схемы

$$FACT(x) = \begin{cases} 1, & \text{если } x = 0, \\ x * FACT(x - 1), & \text{если } x > 0. \end{cases}$$

FACT(x),

FACT(x)=если x=0 то 1 иначе x*FACT(x-1).

FACT(4)=если 4=0 то 1 иначе 4*FACT(4-1)=
=4*FACT(3)=4*(если 3=0 то 1 иначе 3*FACT(3-1))=
=4*3*FACT(2)=12*(если 2=0 то 1 иначе 2*FACT(2-1))=
=12*2*FACT(1)=24*(если 1=0 то 1 иначе 1*FACT(1-1))=
=24*1*FACT(0)=24*(если 0=0 то 1 иначе 0*FACT(0-1))=24

Базис РС включает:

- 4 счетных множества символов:
 - Переменные;
 - Функциональные символы;
 - Предикатные символы;
 - Специальные символы.
- Множество логических выражений.
- Множество термов.

Множество функциональных символов:

1. Множество базовых функциональных символов $(f^{(1)}, g^{(2)})$;
2. Множество определяемых функциональных символов $(F^{(1)}, G^{(2)})$.

Термы:

1. Простые термы

- Базовые термы;
- Вызовы ($F^{(n)}(t_1, t_2, \dots, t_n)$).

2. Условные термы:

если π то t_1 иначе t_2 .

Пример:

- базовые термы - $f(x, g(x, y))$; $h(h(a))$;
- вызовы - $F(x)$; $H(H(a))$; $F(H(x), f(x, y))$;
- условный терм
если $p(x, y)$ **то** $h(h(a))$ **иначе** $F(x)$.

Рекурсивное уравнение:

$$F^{(n)}(x_1, x_2, \dots, x_n) = t(x_1, x_2, \dots, x_n)$$

Рекурсивная схема: (t, M)

Рекурсивная программа: (R_s, I)

Примеры РС:

1) $R_{s1}: F(x),$

$$F(x) = \text{если } p(x) \text{ то } a \text{ иначе } g(x, F(h(x))).$$

2) $R_{s2}: A(x, y),$

$$A(x, y) = \text{если } p(x) \text{ то } f(x) \text{ иначе } B(x, y),$$

$$B(x, y) = \text{если } p(y) \text{ то } A(g(x), a) \text{ иначе } C(x, y);$$

$$C(x, y) = A(g(x), A(x, g(y))).$$

3) $R_{s3}: F(x),$

$$F(x) = \text{если } p(x) \text{ то } x \text{ иначе } f(F(g(x)), F(h(x))).$$

Протокол выполнения рекурсивной программы

$R_{s1}: F(x),$

$F(x) = \text{если } p(x) \text{ то } a \text{ иначе } g(x, F(h(x))).$

$I(x) = 4; I(a) = 1;$

$I(g) = G, \text{ где } G(d1, d2) = d1 * d2;$

$I(h) = H, \text{ где } H(d) = d - 1;$

$I(p) = P, \text{ где } P(d) = 1, \text{ если } d = 0, \text{ иначе } P(d) = 0.$

№ п/п	Значение терма для (R_s, I)
1	$F(4)$
2	$4 * F(3)$
3	$4 * (3 * F(2))$
4	$4 * (3 * (2 * F(1)))$
5	$4 * (3 * (2 * (1 * F(0))))$
6	$4 * (3 * (2 * (1 * 1))) = 24$

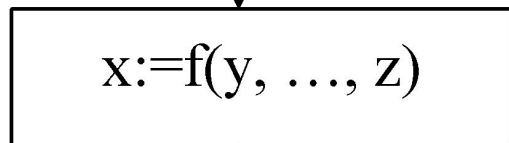
Трансляция схем программ

Теорема Маккарти: Класс стандартных схем транслируем в класс рекурсивных схем.

Алгоритм трансляции:

1. Точки сечения
 $i \Leftrightarrow F_i(x, y, \dots, z);$
старт $\Leftrightarrow F_1(x, y, \dots, z);$
стоп(x) $\Leftrightarrow x;$
2. Граф рассекается по точкам сечения;
3. Для каждого фрагмента строится рекурсивное уравнение:
 $F_i(x, y, \dots, z) = \dots$

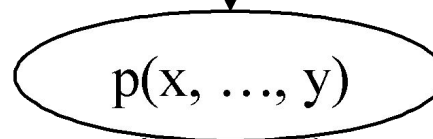
$t(f(y, \dots, z), y, \dots, z)$



$t(x, y, \dots, z)$

A)

если $p(x, \dots, y)$ то t_1 иначе t_2



1

t_1

0

t_2



B)

t t



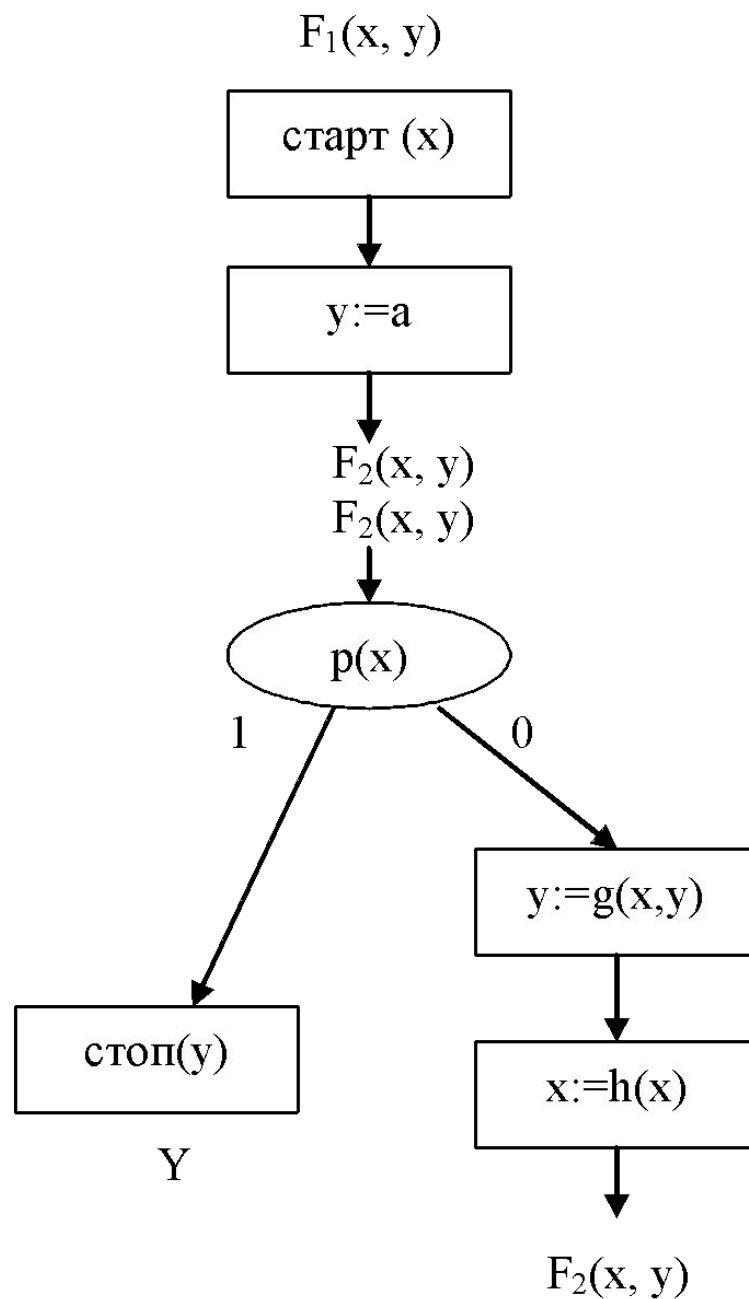
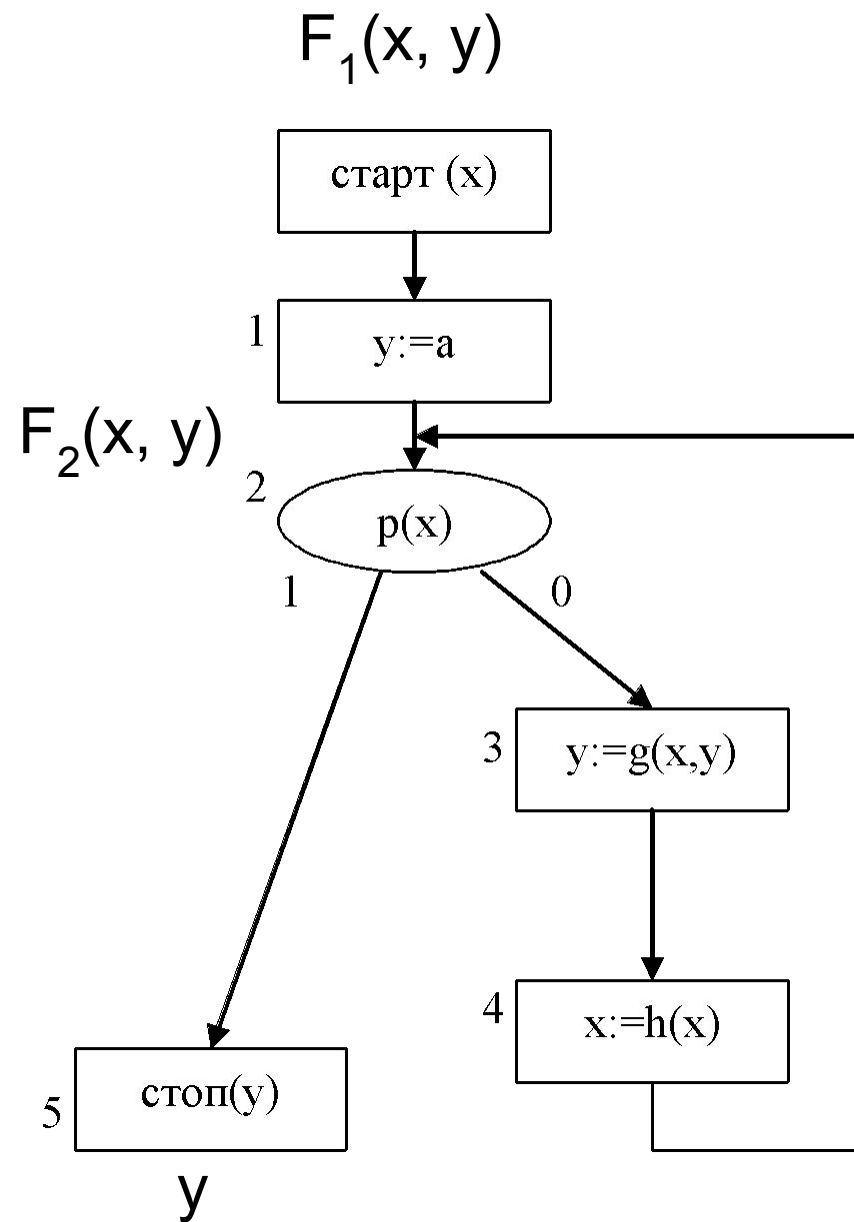
t

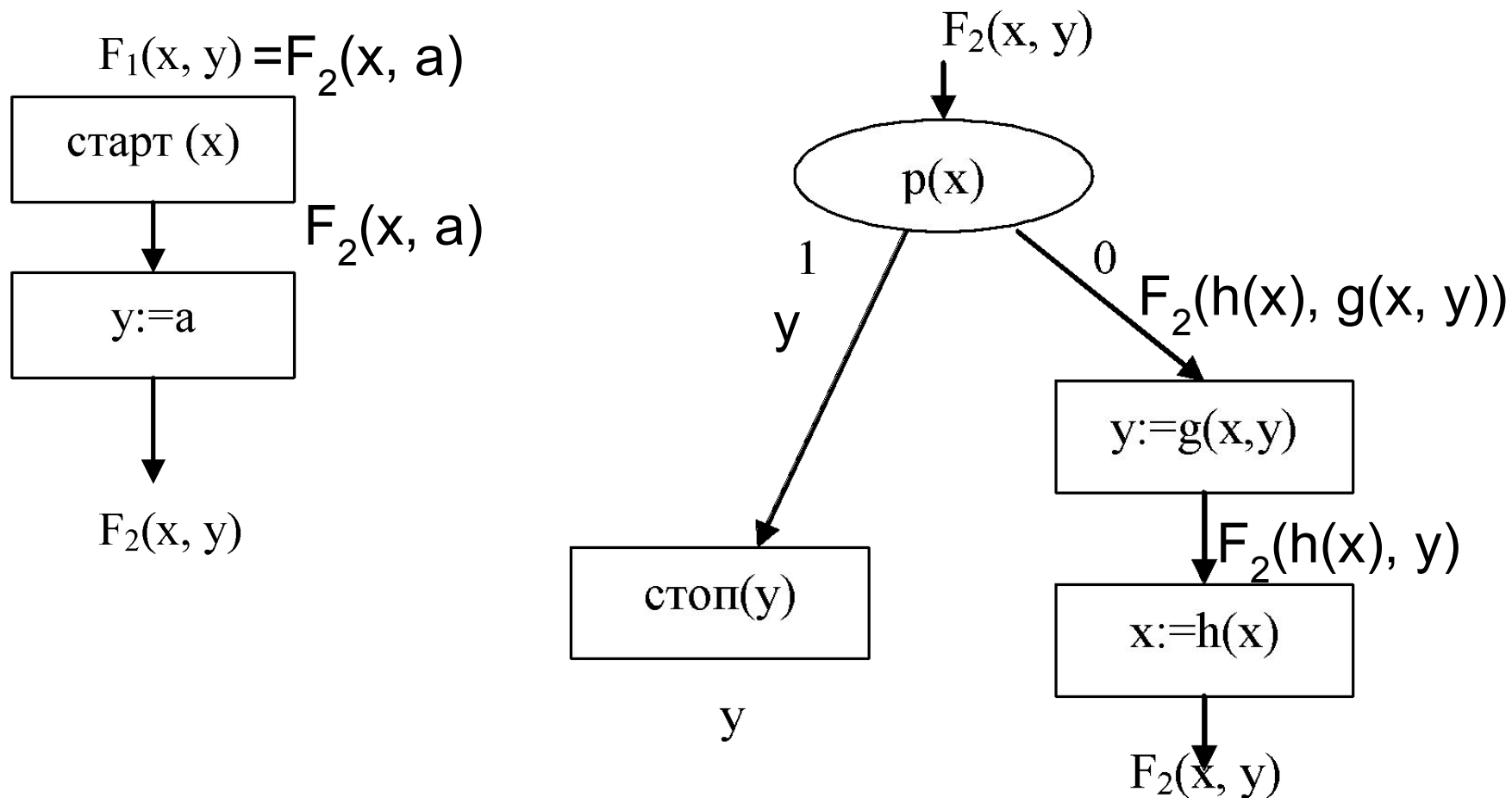


C)

$F_i(x, y, \dots, z) = t(x, y, \dots, z);$

Пример 1:

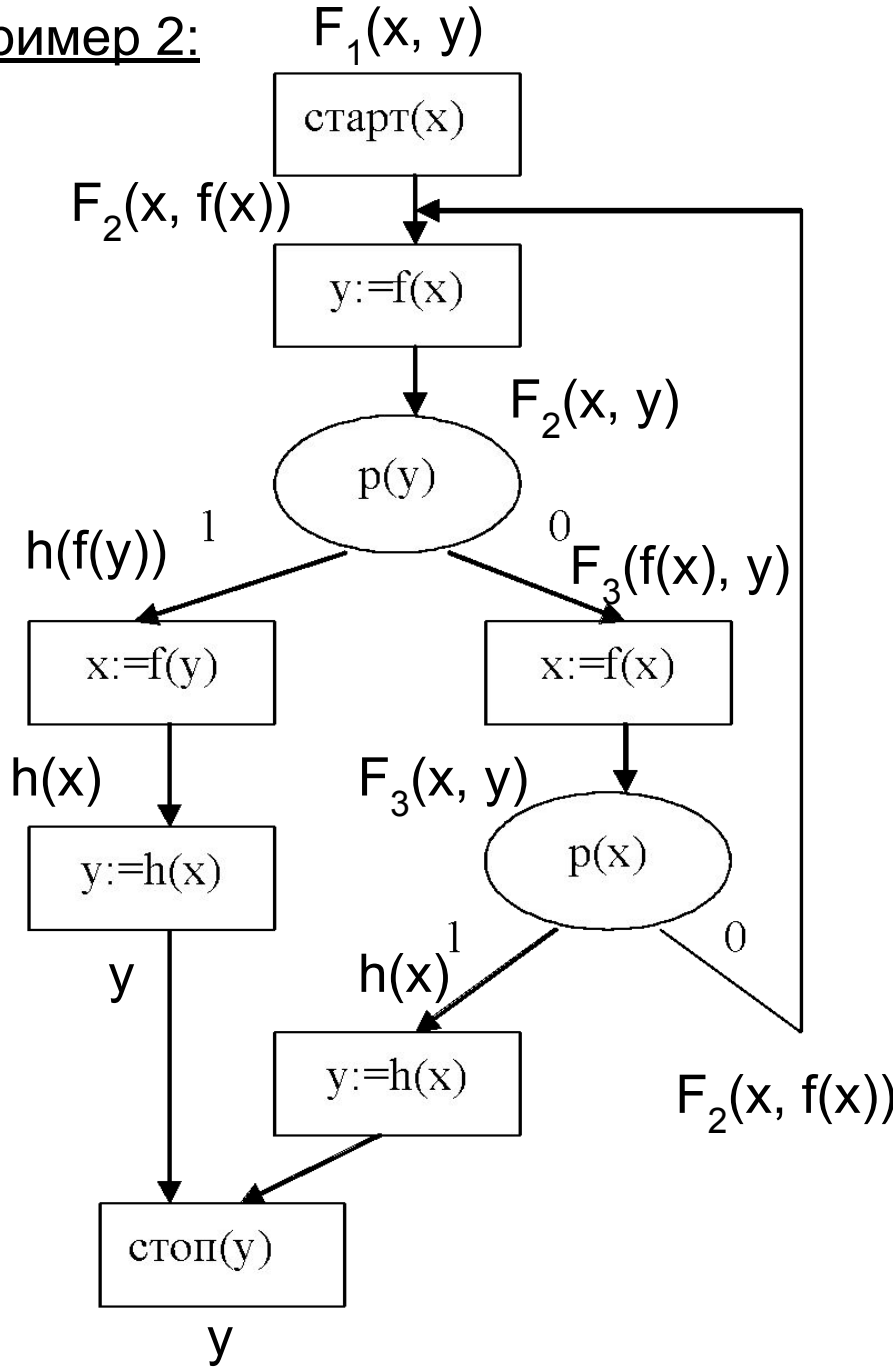




$$F_1(x, y) = F_2(x, a),$$

$$F_2(x, y) = \text{если } P(x) \text{ то } y \text{ иначе } F_2(h(x), g(x, y))$$

Пример 2:



$F_1(x, y) = F_2(x, f(x)),$

$F_2(x, y) = \text{если } p(y) \text{ то } h(f(y))$
 $\text{иначе } F_3(f(x), y),$

$F_3(x, y) = \text{если } p(x) \text{ то } h(x)$
 $\text{иначе } F_2(x, f(x)).$

Линейные унарные рекурсивные схемы

$F(x),$

$F(x) = \text{если } p(x) \text{ то } f(x) \text{ иначе } F(F(g(x))).$

$F(a),$

$F(x) = \text{если } p(x) \text{ то } x \text{ иначе } G(x),$

$G(x) = \text{если } q(x) \text{ то } f(F(f(x))) \text{ иначе } g(F(g(x))).$

Теорема (Патерсон-Хьюит). Класс линейных унарных рекурсивных схем транслируем в класс стандартных схем.

Схемы с процедурами

- Главная схема

$$x = F^{(n)}(y_1, y_2, \dots, y_n)$$

- Множество схем процедур.

Главная схема	Множество схем процедур
(старт(x), 1: $z := x$, 2: $u := a$, 3: $x := F(x, z, u)$, 4: $u := b$, 5: $z := F(z, x, u)$ 6: стоп(z))	$F(y, v, w) =$ (старт, 1: если $p(y)$ то 2 иначе 4, 2: $y := h(y)$, 3: $v := G(v, w)$, 4: если $q(w)$ то 5 иначе 6, 5: $y := v$, 6: стоп(y)) $G(t, r) =$ (старт, 1: если $q(r)$ то 2 иначе 3, 2: $t := r$, 3: стоп(t))

Трансляция рекурсивных схем в схемы с процедурами

- (старт $(y_1, y_2, \dots, y_n)$,
1: $y := t(y_1, y_2, \dots, y_n)$,
2: **стоп** (y)).
- $F_i(x_1, \dots, x_n) = \text{если } p(x_i, \dots, x_n) \text{ то } t_{i1} \text{ иначе } t_{i0}$

$F_i(x_1, \dots, x_n) = (\text{старт},$
1: $\text{если } p(x_{i1}, \dots, x_{in}) \text{ то } 2 \text{ иначе } k,$
2: $S(v, t_{i1}) \text{ на } m,$
 $k: S(v, t_{i0}),$
 $m: \text{стоп}(v)).$

$S(v, t) :$

а) $\text{если } t=x, \text{ то } S(v, t) \Rightarrow v:=x;$

б) $\text{если } t=\phi^{(n)}(t_1, \dots, t_n), \text{ то}$

$S(v, t) = \sigma_1, \sigma_2, \dots, \sigma_n, v:=\phi^{(n)}(z_1, \dots, z_n),$
 $\sigma_i = \begin{cases} z_i:=x, & \text{если } t_i - \text{переменная } x, \\ S(z_i, t_i) & \text{в противном случае.} \end{cases}$

Рекурсивная схема:

S: $F(x)$,

$F(x) = \text{если } p(x) \text{ то } x \text{ иначе } f(F(g(x)), F(h(x)))$

Схема с процедурами:

S: (старт (x),

1: $y := F(x)$,

2: стоп(y))

$F(x) =$ (старт,

1: если $p(x)$ то 2 иначе 3,

2: $v := x$ на 8,

3: $z := g(x)$,

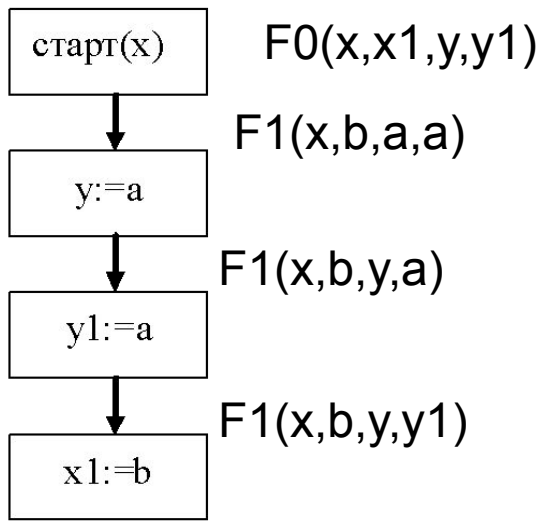
4: $z := F(z)$,

5: $u := h(x)$,

6: $u := F(u)$,

7: $v := f(z, u)$,

8: стоп(v)).



$F0(x, x1, y, y1)$

$F1(x, b, a, a)$

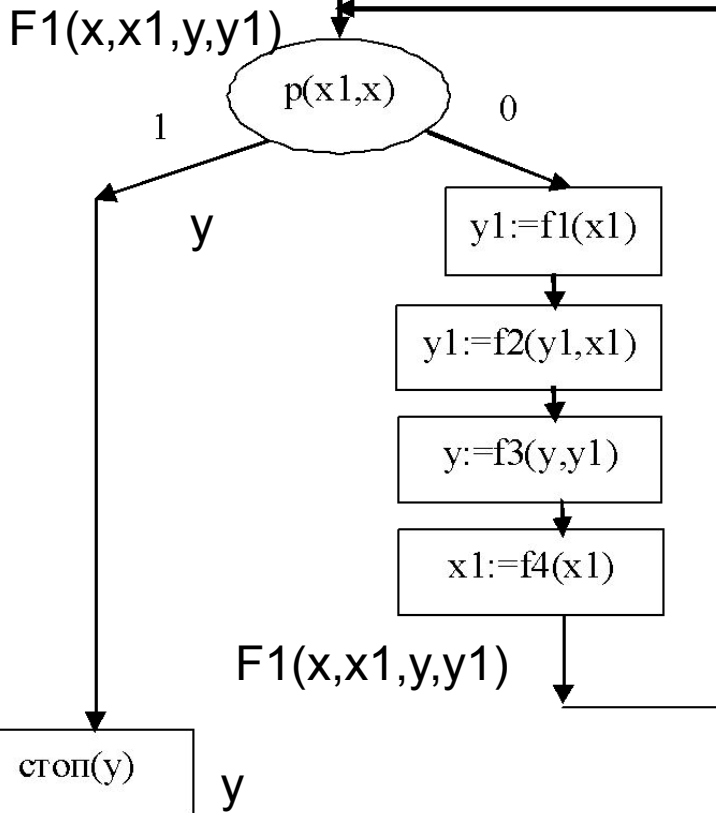
$F1(x, b, y, a)$

$F1(x, b, y, y1)$

$F0(x, x1, y, y1) = F1(x, b, a, a)$

$F1(x, x1, y, y1) =$ если $p(x1, x)$ то y
иначе

$F1(x, f4(x1), f3(y, f2(f1(x1), x1)),$
 $f2(f1(x1), x1))$



$F1(x, x1, y, y1)$

1

y

0

$y1 := f1(x1)$

$y1 := f2(y1, x1)$

$y := f3(y, y1)$

$x1 := f4(x1)$

$F1(x, x1, y, y1)$

$F1(x, f4(x1), f3(y, f2(f1(x1), x1)), f2(f1(x1), x1))$

$F1(x, f4(x1), f3(y, f2(y1, x1)), f2(y1, x1))$

$F1(x, f4(x1), f3(y, y1), y1)$

$F1(x, f4(x1), y, y1)$

$F0(x, x1, y, y1) = F1(x, b, a, a)$

$F1(x, x1, y, y1) =$ если $p(x1, x)$ то y

иначе $F1(x, f4(x1), f3(y, f2(f1(x1), x1)), f2(f1(x1), x1))$

S: (старт (x),

1: $x1 := b$;

2: $y := a$;

3: $y1 := a$;

4: $y := F1(x, x1, y, y1)$;

5: стоп (y))

$F1(x, x1, y, y1) =$ (старт,

1: если $p(x1, x)$ то 7 иначе 2

2: $y1 := f1(x1)$;

3: $y1 := f2(y1, x1)$;

4: $y := f3(y, y1)$;

5: $x1 := f4(x1)$;

6: $v := F1(x, x1, y, y1)$ на 8;

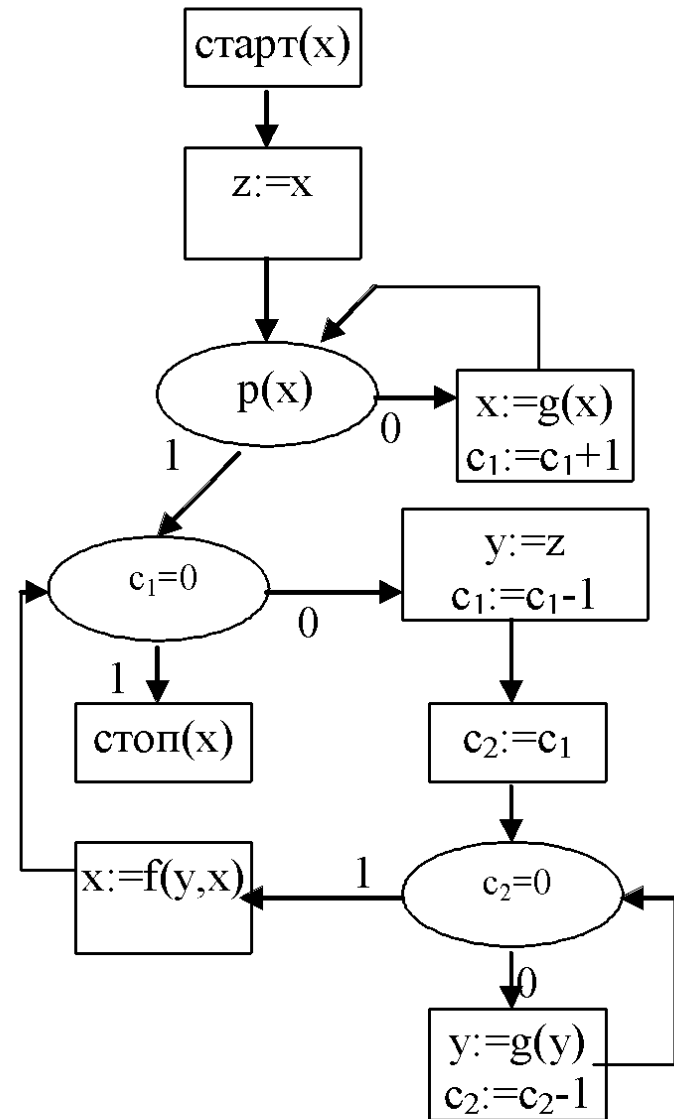
7: $v := y$;

8: стоп (v))

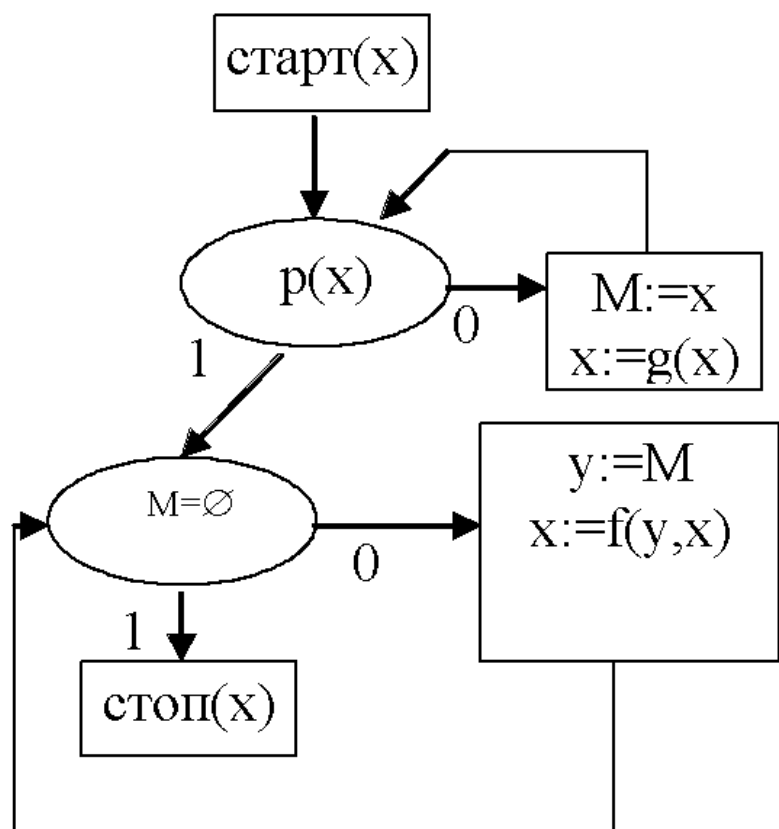
Обогащенные схемы

- класс счетчиковых схем;
интерпретированные операторы:
 $c := c + 1$; $c := c - 1$; $c = 0$; $c_1 := c_2$;

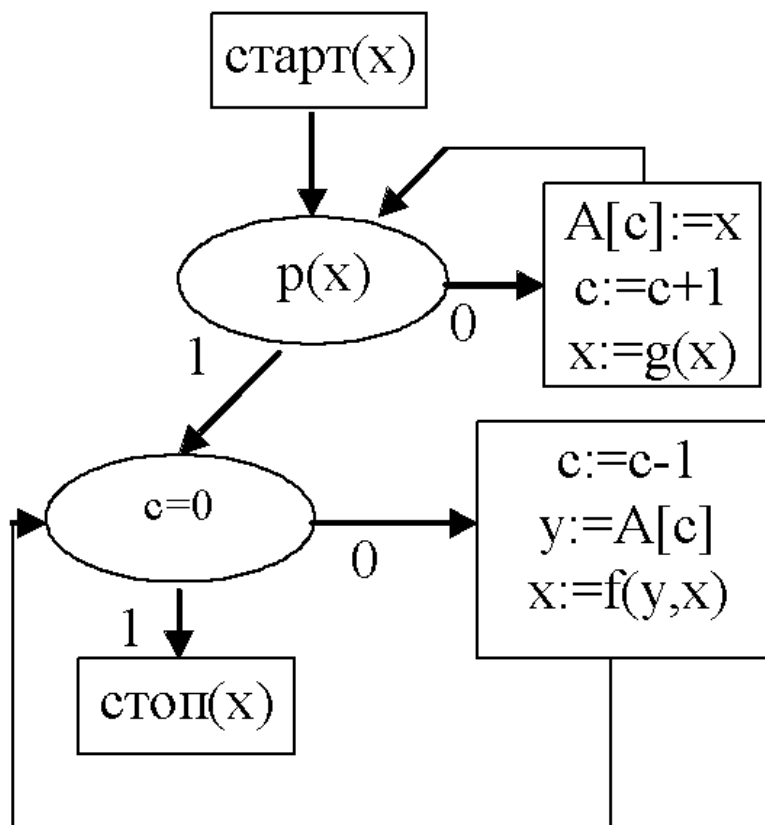
$F(x)$,
 $F(x) = \text{если } p(x) \text{ то } x$
 $\text{иначе } f(x, F(g(x)))$



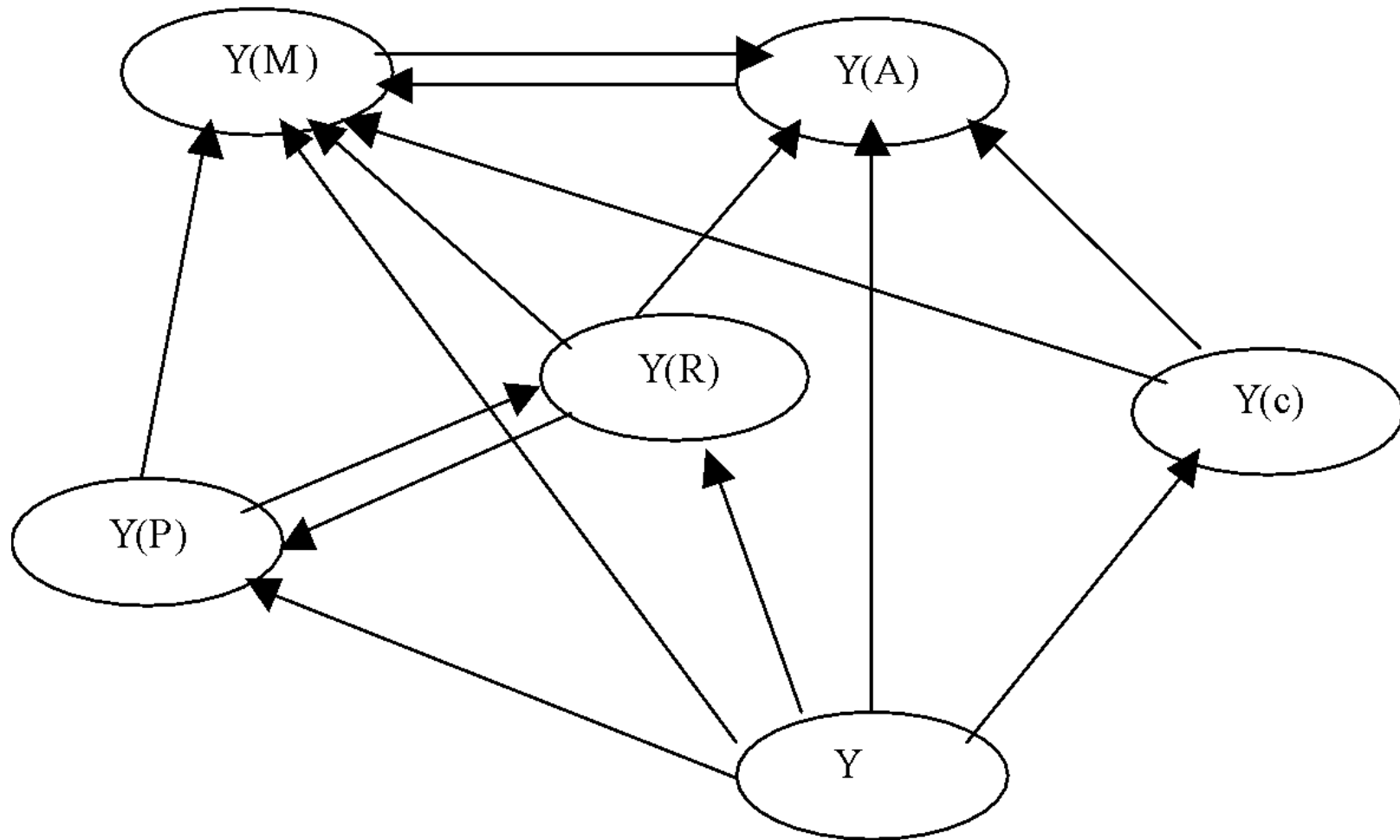
- класс магазинных схем;
интерпретированные
операторы:
 $M:=x$; $x:=M$; $M=\emptyset$;



- класс схем с массивами;
интерпретированные
операторы:
 $A[c]:=x$; $x:=A[c]$.



Трансляция обогащенных схем:



Y — стандартные схемы; Y(M) — магазинные схемы;
Y(R) — рекурсивные схемы; Y(A) — схемы с массивами;
Y(c) — счетчиковые схемы; Y(P) — схемы с процедурами.

Структурированные схемы

$(o_0, o_1, \dots, o_n)$

Специальные символы: **если, то, иначе, пока, цикл, конец.**

Три типа схемных операторов:

- простой оператор;
- условный оператор:
если π то σ_1 иначе σ_0 конец.
- оператор цикла:
пока π цикл σ конец
до π цикл σ конец.

Стандартная схема программы	Структурированная схема программы
сарт(x), 1: $y := f(x)$, 2: если $p(y)$ то 7 иначе 3, 3: $y := f(y)$, 4: если $p(y)$ то 5 иначе 7, 5: если $p(x)$ то 6 иначе 7, 6: $x := h(x)$ на 5, 7: стоп(x, y).	сарт(x), $y := f(x)$, если $p(y)$ то иначе $y := f(y)$, если $p(x)$ то пока $p(x)$ цикл $x := h(x)$ конец иначе конец конец, стоп(x, y).