

Ինֆորմատիկա

УДК 519.682

Մ.Մ. ԱՂԱՍՅԱՆ

ՊՐՈՑԵԴՈՒՐԱՅԻՆ ԾՐԱԳՐԵՐԻ ԴԵՆՈՏԱՑԻՈՆ ՍԵՄԱՆՏԻԿԱ
(ՓՈՓՈԽԱԿԱՆՆԵՐԻ ՔԱՆԱԿԻ ՄԻՆԻՄԱԼԱՑՄԱՆ ԽՆԴՐԻ ԼՈՒԾՈՒՄ)

Ներածություն: Պրոցեդուրային ծրագրերի դենոտացիոն սեմանտիկա սահմանելու համար օգտագործվում է [1] աշխատանքում նկարագրված լեզուն, որում ծրագիրը իրենից ներկայացնում է հավասարումների համակարգ: [1]-ում պրոցեդուրային կամայական ծրագրի համապատասխանության մեջ է դրվում ֆունկցիոնալ հավասարումների համակարգ, այնպիսին, որի համար գոյություն ունի փոքրագույն լուծում և այդ լուծման գլխավոր ֆունկցիան (առաջին ֆունկցիան) համընկնում է պրոցեդուրային ծրագրով հաշվարկվող ֆունկցիայի հետ: Այդ հավասարումների համակարգի յուրաքանչյուր հավասարման մեջ մասնակցում են պրոցեդուրային ծրագրի բոլոր փոփոխականները: Այս աշխատանքում առաջարկված է ալգորիթմ, որը վերը նշված ֆունկցիոնալ հավասարումների համակարգի յուրաքանչյուր հավասարման մեջ փոփոխականների քանակը մինիմալացնում է՝ ստանալով առաջինին համարժեք համակարգ:

1. Օգտագործվող սահմանումները և արդյունքները: Դիտարկենք որևէ պրոցեդուրային ծրագրավորման լեզու L : Դիցուք M -ը այդ լեզվի առարկաների բազմությունն է, C -ն բազիսային գործողությունների բազմությունը, իսկ P -ն ինչ-որ ծրագիր է գրված L լեզվով, որը օգտագործում է $x_1, x_2, \dots, x_n, z$ ($n \geq 0$) փոփոխականները, որտեղ z -ը ելքի փոփոխական է: Ենթադրենք, որ P -ն օգտագործում է միայն ներմուծման, արտածման, վերագրման, անցման և պայմանական օպերատորները:

Նկարագրենք, թե ինչպես է S ֆունկցիոնալ հավասարումների համակարգը համապատասխանեցվում P ծրագրին:

Յուրաքանչյուր օպերատորի, բացի ներմուծման և արտածման օպերատորներից, համապատասխանեցնենք ֆունկցիոնալ փոփոխական և հավասարում: Նկարագրենք այդ հավասարումները, ելնելով օպերատորների տեսքից:

Մտցնենք նշանակում՝ $x_1 \dots x_n = \bar{x}$:

Դիտարկենք վերագրման օպերատորը: Ենթադրենք նրան համապատասխանեցված է F ֆունկցիոնալ փոփոխականը: Վերագրման օպերատորը կարող է ունենալ հետևյալ տեսքերը.

1. $x := m$,
2. $x := x_s$,
3. $x := f(x_{i_1}, \dots, x_{i_k})$,

որտեղ $x \in \{x_1, \dots, x_n, z\}$, $m \in M$, $1 \leq s \leq n$, $x_{i_j} \in \{x_1, \dots, x_n\}$, $j = 1, \dots, k$, $k \geq 1$,
 $f: M^k \rightarrow M$:

Եթե $x = z$, ապա վերագրման օպերատորին համապատասխանեցվող հավասարումը կունենա հետևյալ տեսքը.

1. $F = \lambda \bar{x}[m]$,
2. $F = \lambda \bar{x}[x_s]$,
3. $F = \lambda \bar{x}[f(x_{i_1}, \dots, x_{i_k})]$:

Եթե $x = x_i$, $i \in \{1, \dots, n\}$ և վերագրման օպերատորից հետո կատարվող օպերատորին համապատասխանեցված է F' ֆունկցիոնալ փոփոխականը, ապա վերագրման օպերատորին համապատասխանեցվող հավասարումը կունենա հետևյալ տեսքը.

1. $F = \lambda \bar{x}[F'(x_1, \dots, x_{i-1}, m, x_{i+1}, \dots, x_n)]$,
2. $F = \lambda \bar{x}[F'(x_1, \dots, x_{i-1}, x_s, x_{i+1}, \dots, x_n)]$,
3. $F = \lambda \bar{x}[F'(x_1, \dots, x_{i-1}, f(x_{i_1}, \dots, x_{i_k}), x_{i+1}, \dots, x_n)]$:

Դիտարկենք պայմանական օպերատորը, որում ստուգվում է B տրամաբանական արտահայտությունը: Ենթադրենք պայմանական օպերատորին համապատասխանեցված է F ֆունկցիոնալ փոփոխականը, B -ի դրական արժեքի դեպքում կատարվող օպերատորին համապատասխանեցված է F' ֆունկցիոնալ փոփոխականը, իսկ B -ի բացասական արժեքի դեպքում կատարվող օպերատորին՝ F'' ֆունկցիոնալ փոփոխականը: Այդ դեպքում պայմանական օպերատորին համապատասխանեցվող հավասարումը կունենա հետևյալ տեսքը.

$$F = \lambda \bar{x}[\text{if } B \text{ then } F'(\bar{x}) \text{ else } F''(\bar{x})]:$$

Դիտարկենք անցման օպերատորը: Ենթադրենք նրան համապատասխանեցված է F ֆունկցիոնալ փոփոխականը, իսկ այն օպերատորին, որին պետք է կատարվի անցում, համապատասխանեցված է F' ֆունկցիոնալ փոփոխականը: Այդ դեպքում անցման օպերատորին համապատասխանեցվող հավասարումը կունենա հետևյալ տեսքը.

$$F = \lambda \bar{x}[F'(\bar{x})]:$$

Ներմուծենք որոշ նշանակումներ:

Դիցուք $G(Q, E)$ -ն կողմնորոշված գրաֆ է, որտեղ Q -ն գագաթների բազմությունն է, իսկ E -ն կողերի (աղեղների) բազմությունը: Եթե $(p, q) \in E$, որտեղ $p, q \in Q$, ապա կզրենք $p \rightarrow q$: Եթե $q_0, \dots, q_n$ հաջորդականությունը ներկայացնում է ճանապարհ $G(Q, E)$ գրաֆում, կասենք, որ q_n գագաթը հասանելի է q_0 գագաթից և կնշանակենք $q_0 \xrightarrow{n} q_n$:

2. Փոքրագույն տարրի կառուցման ալգորիթմ: Դիցուք ունենք V վերջավոր բազմությունը, $G(Q, E)$ կողմնորոշված գրաֆը, որի յուրա-

քանչյուր q գազաթի համար սահմանված է երեք բազմություն՝
 $In_q^0 \subset V, Out_q^0 \subset V, In_q \subset V$, ընդ որում՝ $In_q^0 \subset In_q$:

Նշանակենք $IN_G^0 = \{In_q^0 \mid q \in Q\}$, $IN_G = \{In_q \mid q \in Q\}$, $OUT_G^0 = \{Out_q^0 \mid q \in Q\}$:

Ֆիքսենք IN_G^0, OUT_G^0 բազմությունները: Կասենք, որ IN_G բազմությունը անհակասելի է, եթե $\forall q \in Q, \forall v \in In_q, \forall p \in Q$ և եթե $p \rightarrow q$, ապա $v \in In_p \cup Out_p^0$: Կասենք IN_G -ն հիմնավորված է, եթե $\forall p \in Q, \forall v \in In_p \setminus In_p^0$, $\exists q \in Q$ այնպիսին, որ $p \xrightarrow{k} q, k \geq 1, v \in In_q^0, \exists q_0, \dots, q_k \in Q$ այնպիսին, որ $q_0 = p, q_k = q, q_i \rightarrow q_{i+1}, v \notin Out_{q_i}^0, i = 0, \dots, k-1$:

Նկարագրենք ալգորիթմ, որը տրված $G(Q, E), IN_G^0, OUT_G^0$ -ի համար կառուցում է անհակասելի, հիմնավորված IN_G բազմություն:

Ալգորիթմ: Անհակասելի, հիմնավորված IN_G բազմության կառուցում:

Մուտք: V վերջավոր բազմություն, $G(Q, E), IN_G^0, OUT_G^0$:

Ելք: IN_G անհակասելի, հիմնավորված բազմություն:

Քայլ1: $IN_G = IN_G^0$:

Քայլ2: $G(Q, E)$ գրաֆի կամայական p գազաթի համար կատարել

Քայլ3:

p գազաթից հասանելի կամայական q գազաթի համար կատարել Քայլ4, այնուհետև կատարել Քայլ6:

Քայլ4: $Outs_{pq} = V$: p գազաթից q տանող կամայական պարզ ճանապարհի համար՝ $q_0, \dots, q_k \in Q, q_i \rightarrow q_{i+1}, i = 0, \dots, k-1$, որտեղ $q_0 = p, q_k = q$, կատարել Քայլ5:

Քայլ5: $Outs_{pq} = Outs_{pq} \cap (Out_{q_0}^0 \cup \dots \cup Out_{q_{k-1}}^0)$:

Քայլ6: $In_p = In_p \cup (In_q^0 \setminus Outs_{pq})$:

Ալգորիթմը վերջավոր է, քանի որ $G(Q, E)$ գրաֆում գազաթների քանակը, հետևաբար նաև պարզ ճանապարհների քանակը վերջավոր է: Եթե $G(Q, E)$ գրաֆում Q -ն n -րդ կարգի է, ապա քայլերի քանակը ալգորիթմում կլինի $\frac{n^2(2^{n-2})!}{(2^{n-2} - n)!}$: Ապացուցենք, որ այս ալգորիթմով ստացված IN_G բազմությունը հիմնավորված և անհակասելի է:

Նախ ցույց տանք, որ այն հիմնավորված է:

Վերցնենք $\forall p \in Q$ և $\forall v \in In_p \setminus In_p^0$: Կառուցման ալգորիթմից հետևում է, որ $\exists q \in Q$ այնպիսին, որ $p \xrightarrow{k} q, k \geq 1, v \in In_q^0, v \notin Outs_{pq} \Rightarrow \exists q \in Q$ այնպիսին, որ $p \xrightarrow{k} q, k \geq 1, v \in In_q^0, \exists q_0, \dots, q_k \in Q$ այնպիսին, որ $q_0 = p, q_k = q, q_i \rightarrow q_{i+1}, v \notin Out_{q_i}^0, i = 0, \dots, k-1$:

Յույց տանք, որ IN_G -ն անհակասելի է:

Վերցնենք $\forall q \in Q, \forall v \in In_q$ և $\forall p \in Q$: Եթե $p \rightarrow q$ և $v \in In_q^0$, ապա ըստ կառուցման ալգորիթմի
 $v \in In_q^0, p \rightarrow q \Rightarrow v \in In_q^0, Outs_{pq} \subset Out_p^0, (In_q^0 \setminus Outs_{pq}) \subset In_p \Rightarrow v \in In_p \cup Outs_{pq}$,
 $Outs_{pq} \subset Out_p^0 \Rightarrow v \in In_p \cup Out_p^0$:

Եթե $v \notin In_q^0$ և $p \rightarrow q$ ապա կառուցման ալգորիթմից և IN_G -ի հիմնավորվածությունից հետևում է, որ $(\exists s \in Q$ այնպիսին, որ $q \xrightarrow{k} s, v \in In_s^0, v \notin Outs_{qs})$,
 $(p \rightarrow q) \Rightarrow \exists s \in Q$ այնպիսին, որ $p \xrightarrow{k+1} s, v \in In_s^0, v \notin Outs_{qs}$,
 $Outs_{ps} \subset Out_p^0 \cup Outs_{qs}, (In_s^0 \setminus Out_{ps}) \subset In_p \Rightarrow v \in In_p \cup Out_p^0$:

Ապացույցը ավարտված է:

Դիցուք ունենք IN'_G և IN''_G -ը: Կասենք $IN'_G < IN''_G$, եթե $\forall p \in Q, In'_p \subset In''_p$, որտեղ $In'_p \in IN'_G, In''_p \in IN''_G$:

Դիցուք $G(Q, E)$ գրաֆի գագաթների քանակը n է, և ֆիքսված է գագաթների որևէ համարակալում: Այդ դեպքում IN_G -ն կարող ենք ներկայացնել հետևյալ տեսքով.

$$IN_G = \langle In_{q_1}, \dots, In_{q_n} \rangle, \text{ որտեղ } In_{q_i} \in Q, In_{q_i} \in 2^V, i = 1, \dots, n :$$

Դիցուք $\mathfrak{R} = \{IN_G | IN_G \text{ -ն անհակասելի է} \}$:

Այս նշանակումից երևում է, որ $\mathfrak{R} \subset \mathfrak{I}$, որտեղ $\mathfrak{I} = \underbrace{2^V \times \dots \times 2^V}_n$:

Քանի որ $\mathfrak{I}$ -ն մասնակի կարգավորված բազմություն է, որտեղ մասնակի կարգը պատկանելիության հարաբերությունն է, ապա այդպիսին է նաև $\mathfrak{R}$ -ը:

Պնդում: Եթե IN_G բազմությունը անհակասելի է և հիմնավորված, ապա այն $\mathfrak{R}$ մասնակի կարգավորված բազմության փոքրագույն տարրն է:

Ապացույց: IN_G -ն անհակասելի է, հետևաբար պատկանում է $\mathfrak{R}$ -ին:

Վերցնենք կամայական $IN'_G = \{In'_q | q \in Q\} \in \mathfrak{R}$: Յույց տանք, որ $IN_G < IN'_G$:

Վերցնենք $\forall p \in Q$, և $\forall v \in In_p$:

Եթե $v \in In_p^0$, ապա $v \in In'_p$, քանի որ $In_p^0 \subset In'_p$:

Եթե $v \notin In_p^0$, ապա օգտվելով IN_G -ի հիմնավորվածությունից և IN'_G -ի անհակասելիությունից կարող ենք գրել՝ $\forall p \in Q, \forall v \in In_p \setminus In_p^0 \Rightarrow \exists q \in Q$ այնպիսին, որ $p \xrightarrow{k} q, k \geq 1, v \in In_q^0, \exists q_0, \dots, q_k \in Q$ այնպիսին, որ $q_0 = p, q_k = q, q_i \rightarrow q_{i+1}, v \notin Out_{q_i}^0, i = 0, \dots, k-1 \Rightarrow v \in In'_{q_i}, i = 0, \dots, k-1 \Rightarrow v \in In'_p$:

Պնդումը ապացուցված է:

3. Փոփոխականների քանակի մինիմալացում ֆունկցիոնալ հավասարումների համակարգում: Նշված խնդրի լուծման համար պետք է կատարել

հետևյալ գործողությունները: Համարժեք ձևափոխությունների միջոցով վերացնել P ծրագրում եղած ցիկլի և ընտրման օպերատորները: Ֆիքսել փոփոխականների որոշակի հերթականություն: Կառուցել ֆունկցիոնալ հավասարումների S համակարգ (տես հոդվածի 1-ին բաժինը): Կառուցել $G(Q, E)$ կողմնորոշված գրաֆ, որի գագաթների բազմությունը S համակարգի ֆունկցիոնալ փոփոխականների բազմությունն է, և երկու գագաթ միացված են կողմնորոշված կողով, երբ S -ում գոյություն ունի հավասարում, որի ձախ կողմում օգտագործվում է առաջին գագաթի ֆունկցիոնալ փոփոխականը, իսկ աջ կողմում՝ երկրորդ գագաթի ֆունկցիոնալ փոփոխականը: Որպես V բազմություն վերցնել P ծրագրում մասնակցող բոլոր փոփոխականների բազմությունը: Գրաֆի յուրաքանչյուր q գագաթի համար ստեղծել երկու բազմություն՝ In_q^0 և Out_q^0 : In_q^0 բազմությունը ստեղծել տվյալ գագաթին համապատասխան օպերատորում (այն հավասարման համապատասխան օպերատորը, որի ձախ կողմում օգտագործվում է տվյալ գագաթի ֆունկցիոնալ փոփոխականը) մասնակցող բոլոր այն փոփոխականներից, որոնց արժեքը օգտագործվում է այդ օպերատորում, իսկ Out_q^0 բազմությունը՝ տվյալ գագաթին համապատասխան օպերատորում մասնակցող այն փոփոխականներից, որոնք արժեք են ստանում այդ օպերատորում: Այնուհետև, երկրորդ բաժնում բերված ալգորիթմի միջոցով կառուցել IN_G անհակասելի, հիմնավորված բազմությունը: S համակարգի յուրաքանչյուր հավասարման մեջ λ -ից հետո թողնել միայն այն փոփոխականները, որոնք հանդիսանում են այդ հավասարման ձախ կողմի ֆունկցիոնալ փոփոխականի (մշ. q) In_q բազմության տարրերը, իսկ հավասարման աջ կողմում օգտագործվող կամայական p ֆունկցիոնալ փոփոխականի արգումենտում՝ In_p բազմության տարրերը կամ նրանց վերագրվող արտահայտությունները (եթե ընթացիկ հավասարումը վերագրման օպերատորներին համապատասխան հավասարումն է)՝ ամենուրեք պահպանելով փոփոխականների վերը սահմանված հերթականությունը:

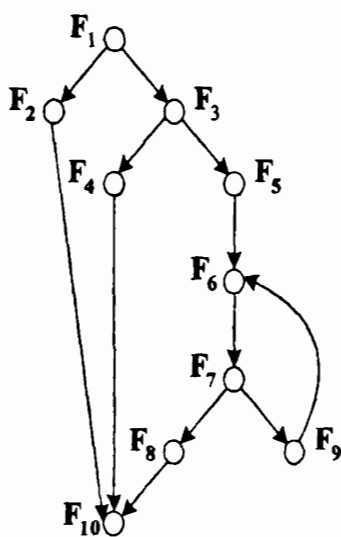
Այս ալգորիթմը P ծրագրին և նրան համապատասխանեցված ֆունկցիոնալ հավասարումների S համակարգի համար միարժեք կառուցում է $G(Q, E)$ գրաֆը և IN_G^0, OUT_G^0 բազմությունները: Նկատենք, որ IN_G բազմության անհակասելիությունից հետևում է S համակարգի կոռեկտությունը, այսինքն կամայական ֆունկցիոնալ հավասարման մեջ օգտագործվող կամայական փոփոխական հանդիսանում է այդ հավասարման λ -ին հաջորդող փոփոխականների կարգավորված բազմության տարր, իսկ հիմնավորվածությունից՝ այդ համակարգում ոչ անհրաժեշտ փոփոխականների բացակայությունը: Այսպիսով, ստեղծված համակարգը կլինի կոռեկտ, և յուրաքանչյուր հավասարման մեջ կամայական փոփոխականի հեռացումը հավասարումների համակարգը կդարձնի ոչ կոռեկտ:

4. Օրինակ: Վերցնենք x -րդ Ֆիբոնաչիի թիվը հաշվարկող հետևյալ ծրագիրը.

```
#include <iostream.h>
Void main()
```

{ Int x,y,u,v,i,z;	
Cin >> x;	
If (x==0)	F_1
z = 0;	F_2
Else	
if (x==1)	F_3
z = 1;	F_4
Else	
{u = 0; v = 1; i = 2;	F_5
a: y = u + v;	F_6
If(x==i)	F_7
z = y;	F_8
Else	
{u = v; v = y; i = i + 1; goto a;}}	F_9
Cout << z;}	F_{10}

Որպեսզի օրինակը կարճ լինի, ծրագրում եղած հաջորդական վերագրման և անցման այն օպերատորները, որոնք կարելի է կատարել զուգահեռ, դիտարկենք որպես մեկ բարդ օպերատոր և նրանց համապատասխանեցնենք մեկ ֆունկցիոնալ հավասարում: Այդ ծրագրի օպերատորներին համապատասխանեցված ֆունկցիոնալ փոփոխականները նշված են ծրագրի աջ կողմում: Վերցնենք $V = \{x, y, u, v, i, z\}$ վերջավոր բազմությունը: Նկարում պատկերված են պրոցեդուրային ծրագրից կազմված կողմնորոշված գրաֆը, IN_G^0 և OUT_G^0 բազմության տարրերը:



$IN_{F_1}^0 = \{x\};$	$OUT_{F_1}^0 = \emptyset;$
$IN_{F_2}^0 = \emptyset;$	$OUT_{F_2}^0 = \{z\};$
$IN_{F_3}^0 = \{x\};$	$OUT_{F_3}^0 = \emptyset;$
$IN_{F_4}^0 = \emptyset;$	$OUT_{F_4}^0 = \{z\};$
$IN_{F_5}^0 = \emptyset;$	$OUT_{F_5}^0 = \{u, v, i\};$
$IN_{F_6}^0 = \{u, v\};$	$OUT_{F_6}^0 = \{y\};$
$IN_{F_7}^0 = \{x, i\};$	$OUT_{F_7}^0 = \emptyset;$
$IN_{F_8}^0 = \{y\};$	$OUT_{F_8}^0 = \{z\};$
$IN_{F_9}^0 = \{y, v, i\};$	$OUT_{F_9}^0 = \{u, v, i\};$
$IN_{F_{10}}^0 = \{z\};$	$OUT_{F_{10}}^0 = \{z\};$

Այնուհետև, օգտագործելով հոդվածի երկրորդ կետում նկարագրված ալգորիթմը, կառուցել անհակասելի, հիմնավորված IN_G բազմությունը, որը բաղկացած է հետևյալ տարրերից.

$$\begin{aligned} In_{F_1} &= \{x\}; & In_{F_6} &= \{x, u, v, i\}; \\ In_{F_2} &= \emptyset; & In_{F_7} &= \{x, y, v, i\}; \\ In_{F_3} &= \{x\}; & In_{F_8} &= \{y\}; \\ In_{F_4} &= \emptyset; & In_{F_9} &= \{x, y, v, i\}; \\ In_{F_5} &= \{x\}; & In_{F_{10}} &= \{z\}; \end{aligned}$$

$x \in In_{F_9}$, քանի որ, ինչպես երևում է նկարից, $F_9 \rightarrow F_6 \rightarrow F_7, x \in In_{F_7} \setminus Outs_{F_9F_7}$, որտեղ $Outs_{F_9F_7} = Out_{F_9}^0 \cup Out_{F_6}^0 = \{u, v, i, y\}$:

Պահպանելով որևէ հերթականություն, օրինակ, V -ի սահմանման ժամանակ փոփոխականների հերթականությունը, և օգտագործելով IN_G -ն (տես 3-ը)՝ հավասարումների համակարգում փոփոխականների քանակը միմիայնացնելուց հետո կստանանք հավասարումների հետևյալ համակարգը.

$$\begin{aligned} F_1 &= \lambda x [if \ x = 0 \ then \ F_2 \ else \ F_3(x)] \\ F_2 &= F_{10}(0) \\ F_3 &= \lambda x [if \ x = 1 \ then \ F_4 \ else \ F_5(x)] \\ F_4 &= F_{10}(1) \\ F_5 &= \lambda x [F_6(x, 0, 1, 2)] \\ F_6 &= \lambda x \ u \ v \ i [F_7(x, u + v, v, i)] \\ F_7 &= \lambda x \ y \ v \ i [if \ x = i \ then \ F_8(y) \ else \ F_9(x, y, v, i)] \\ F_8 &= \lambda y [F_{10}(y)] \\ F_9 &= \lambda x \ y \ v \ i [F_6(x, v, y, i + 1)] \\ F_{10} &= \lambda z [z]: \end{aligned}$$

Օրինակը ավարտված է:

Համակարգային ծրագրավորման ամբիոն

Ստացվել է 25.12.2000

ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Нигиян С.А. – Программирование, 1991, №5, с. 77–86.

М.М. АГАСЯН

ДЕНОТАЦИОННАЯ СЕМАНТИКА ПРОЦЕДУРНЫХ ПРОГРАММ (РЕШЕНИЕ ЗАДАЧИ МИНИМИЗАЦИИ ЧИСЛА ПЕРЕМЕННЫХ)

Резюме

В работе описан алгоритм, который каждой программе, написанной на процедурном языке, ставит в соответствие такую функциональную систему

уравнений, которая определяет ее денотационную семантику, и каждое уравнение этой системы имеет минимальное число переменных. Алгоритм использует математическую модель, для которой доказано существование наименьшего элемента.

M.M. AGHASIAN

THE DENOTATIONAL SEMANTICS OF PROCEDURAL PROGRAMS (THE SOLUTION TO THE PROBLEM OF VARIABLES' NUMBER MINIMIZATION)

Summary

In this article an algorithm is described, which brings every program, written in procedural language, to conformity with such a system of functional equations, that defines its denotational semantics and in each equation the number of variables is minimal. The algorithm uses the mathematical model, for which the existence of the least element is proved.