ALGEBRA ALGORITMOV

Kyjevská algebraicko-logická škola kybernetiky

Ústav kybernetiky V.M.Gluškova NAN Ukrajiny, Kyjev

Prof. G.E. Cejtlin

ALGEBRA ALGORITMOV

Modely.

Algebry.

Algebraické systémy

ALGEBRA ALGORITMOV

Modely.

Modely a algebry - základné pojmy matematiky

Konštrukcia tzv. *logicko-funkcionálnych modelov*

PREREKVIZITY

- relácia, funkcionálna relácia, či funkcia, alebo predikát (ako špeciálny prípad funkcie)
- *usporiadanie*, definované ako *binárna relácia* na niektorej univerzálnej množine prvkov; *čiastočné usporiadanie*, *úplné (lineárne) usporiadanie* $\prec$.
- **U** -univerzálna množina na ktorej je definovaná relácia $\prec$;
- SEQ - množina postupností prvkov z **U**
- postupnosť $M \in SEQ$ a $M = (a_1, a_2, ...a_n)$.
- usporiadaná M ak platí $a_i \prec a_j$, pre všetky i a j také, že $i \prec j, 1 \geq i, j \leq n$
- USQ- množina všetkých usporiadaných postupností na **U** ; $USQ \subset SEQ$.

PREREKVIZITY

Príklad 0.1

Definujeme si binárnu reláciu
$SORT \subset SEQ \times USQ$,
ktorá definuje priradenie medzi (vo všeobecnosti neusporiadanými) postupnosťami zo SEQ a usporiadanými postupnosťami z USQ. Je jasné, že jedna usporiadaná postupnosť z USQ môže zodpovedať viacerym postupnostiam zo SEQ. Relácia $SORT$ priraďuje každej postupnosti $M \in SEQ$ (vzor)jedinečnú postupnosť $M^{'} \in USQ$ (obraz), ktorá vznikne z M permutáciou prvkov tak, aby tieto boli lineárne usporiadané. Tak napríklad vzorom $M_1 = (5, 2, 1, 6, 5)$ a $M_2 = (1, 2, 5, 6, 5)$ zodpovedá jediný obraz-postupnosť $M_3 = (1, 2, 5, 5.6)$; tu $M_1, M_2 \in SEQ$ a $M_3 \in USQ$. S binárnou reláciou $SORT$ je asociovaná unárna (s jedným argumentom) funkcia $sort$ taká, že $sort(M) = M^{'}$ práve vtedy, keď postupnosti M a $M^{'}$ sú v relácii $SORT$, t.j. ak $M, M^{'} \in SORT$.

□

(Koniec príkladu)

PREREKVIZITY

$$\mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}_1, a_{i+1}, ..., a_n\mathbf{K} \tag{1}$$

Elementárne predikáty

Všetky nižšie uvádzané predikáty sú definované na označenej postupnosti

$$d(Y_1, K) = 1 \Leftrightarrow_{df} \mathbf{M}{=}\mathbf{M}^{'}{:}\mathbf{H}a_1, a_2, ..., a_n, \mathbf{Y}_1\mathbf{K} \tag{2}$$

$$\ell > r|Y_1 = 1 \Leftrightarrow_{df} \mathbf{M} = \mathbf{M}^{'}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}_1, a_{i+1}, ..., a_n\mathbf{K} \wedge a_i > a_{i+1} \tag{3}$$

$$UM = 1 \Leftrightarrow_{df} \mathbf{M} = \mathbf{M}^{'}{:}\mathbf{H}\mathbf{Y}_1, a_1, a_2, ..., a_n\mathbf{K} \wedge a_i < a_{i+1}, i = 1, 2, ..., n - 1 \tag{4}$$

Elementárne operátory

$$E \Leftrightarrow_{df} E(\mathbf{M}) = \mathbf{M} \tag{5}$$

(6)

$$P(Y_1) \Leftrightarrow_{df} \mathbf{M}_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}_1, a_{i+1}, ..., a_n\mathbf{K} \wedge \mathbf{M}_2 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, a_{i+1}, \mathbf{Y}_1, a_{i+2}, ..., a_n\mathbf{K} \wedge P(Y_1)(M_1) = M_2 \tag{7}$$

(8)

$$TRANSP(\ell, r) \Leftrightarrow_{df} \mathbf{M}_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}_1, a_{i+1}, ..., a_n\mathbf{K} \wedge \mathbf{M}_2 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_{i+1}, \mathbf{Y}_1, a_i, ..., a_n\mathbf{K} \wedge TRANSP(\ell, r)(M_1) = M_2 \tag{9}$$

(10)

$$UST(Y_1, H) \Leftrightarrow_{df} \mathbf{M}_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}_1, a_{i+1}, ..., a_n\mathbf{K} \wedge \mathbf{M}_2 = \mathbf{M}{:}\mathbf{H}\mathbf{Y}_1a_1, a_2, ..., a_i, a_{i+1}, ..., a_n\mathbf{K} \wedge UST(Y_1, H)(M_1) = M_2 \tag{11}$$

LOGICKO-FUNKCIONÁLNE MODELY

$$\widetilde{M} = (A; SIGN_\pi)$$

kde

- A je báza (množina dát)
- $SIGN_\pi = \{\pi_i | \in \mathbf{I}\}$, sa volá *signatúra* pozostávajúca z predikátov π na množine A a $\mathbf{I}$ je množina indexov.

Všeobecná schéma použitia modelu $\widetilde{M}$

$$V, R \subseteq A$$

kde V predstavuje množinu vstupných
R množinu výstupných dát

Formulovanú úlohu je možno predstaviť ako niektoré zobrazenie (funkciu)

$$f^{(n)} : V^n \to R$$

$$(d_1, d_2, ..., d_n, r) \in F^{(n+1)} \Leftrightarrow f^{(n)}(d_1, d_2, ..., d_n = r$$

$$J^{(n+1)}(d_1, d_2, ..., d_n, r) = 1 \Leftrightarrow (d_1, d_2, ..., d_n, r) \in F^{(n+1)} \qquad (12)$$

Pre zadanú úlohu je možno zostrojiť predikát typu (12) pre každú zodpovedajúcu povahe úlohy funkciu $f_i^{(n_i}$. To završuje konštrukciu modelu ako formalizácie zadania úlohy.

LOGICKO-FUNKCIONÁLNE MODELY

Príklad 0.2

Zostrojíme LFM $\widetilde{M}_1 = (SEQ, SIGN_1)$, ktorý bude definovaný pre úlohu triedenia postupností $M \in SEQ$.
Konštrukcia elementárnych predikátov signatúry $SIGN_1$, k vyjadreniu 2-miestneho charakteristického predikátu *S-O-R-T*.

- unárny predikát UM (viď), pre ktorý platí, že $UM(M) = 1 \Leftrightarrow M \in USQ$
- binárny predikát $D : SEQ^2 \rightarrow \{0, 1\}$ a taký , že $D(M, M^{'}) = 1 \Leftrightarrow |M| = |M^{'}|$, kde $|Z|$ je dĺžka ľubovoľnej postupnosti $Z \in SEQ$.
- binárny predikát $R : SEQ^2 \rightarrow \{0, 1\}$ a taký , že $R(M, M^{'}) = 1 \Leftrightarrow ||M|| = ||M^{'}||$, kde $||Z||$ je množina prvkov v ľubovoľnej postupnosti $Z \in SEQ$.
- binárny predikát $B : SEQ^2 \rightarrow \{0, 1\}$ a taký , že $B(M, M^{'}) = 1 \Leftrightarrow \mathbf{I}(q, M) = \mathbf{I}(q, M^{'})$, pre každý prvok $q \in ||M|| \cap ||M^{'}||$, kde $\mathbf{I}$(q,Z) je počet výskytov prvku q v postupnosti $Z \in SEQ$. Inými slovami, predikát $B(M, M^{'}) = 1$ práve vtedy, ak počty výskytov spoločných prvkov v postupnostiach M a $M^{'}$ sú rovnaké.

Potom

$$SIGN_1 = \{UM, D, R, B\}$$

Predikát S-O-R-T sa dá v LFM $\widetilde{M}_1 = (SEQ, SIGN_1)$ vyjadriť nasledujúcou konjunkciou :

$$S-O-R-T(M, M^{'}) = D(M, M^{'}) \wedge R(M, M^{'}) \wedge B(M, M^{'}) \wedge U(M^{'}) \quad (13)$$

pričom $M \in SEQ, M^{'} \in USQ$.

□

(Koniec príkladu)

ALGEBRY.

Algebra - základné pojmy

- **osnova (základná množina) algebry,**
- **signatúra algebry,**
- **systém vytvárajúcich prvkov (generátorov),**
- **bázy algebry,**
- **superpozície,**
- **axiomatický systém a ďalšie.**

ALGEBRY.

Operácie

$$f : A^n \to A$$

$$f(x_1, x_2, ..., x_n)$$

Funkciu $f(x_1, x_2, ..., x_n)$ voláme *(n-árnou) operáciou* definovanou na množine *A*.

Univerzálna algebra

$$\Omega = \{F_i(x_1, x_2, ..., x_n) | i = 1, 2, ..., k\}$$

$$\widetilde{A} = (A; \Omega)$$

kde :

- ***A* sa volá *osnova (základňa)* algebry,**
- **Ω je *signatúra* algebry.**

ALGEBRY.

Príklad 0.3

Ako príklad operácie uvedieme množinu prirodzených čísel $N = \{1, 2, ..., n, ...\}$ **a funkciu** $f(x, y) = x + y$ **dvoch argumentov x,y.**

$$f(x, y) = x + y$$

$$N = \{1, 2, ..., n, ...\}$$

Ako ďalší príklad uvedieme algebru boolovských funkcií

$$\widetilde{ABF} = (BF(n); \Omega_{ABF})$$

- kde osnovou je množina všetkých boolovských funkcií (b.f.) n premenných BF(n)
- signatúra Ω_{ABF} pozostáva z 3 operácií: $x \wedge y$, $x \vee y$ a negácie $\overline{x}$.

GENERÁTORY A BÁZY ALGEBRY.

Generátory algebry

V algebre $\widetilde{A} = (A; \Omega)$ množina $S \subseteq A$ a t.ž. každý prvok základnej množiny *A* je možno vytvoriť z prvkov množiny *S* s využitím operátorov zo signatúry Ω je *množinou vytvárajúcich prvkov*, alebo *množinou generátorov* algebry $\widetilde{A}$.

Báza algebry

Systém *S* generátorov algebry $\widetilde{A} = (A; \Omega)$ z ktorého nie je možné vylúčiť žiadny prvok $a \in S$ z *S*, aby sa pritom nenarušila vlastnosť $S - \{a\}$ byť systémom generátorov algebry $\widetilde{A}$ sa volá *bázou* algebry $\widetilde{A}$.

ALGEBRY.

Termy algebry

Algebra $\widetilde{A} = (A; \Omega)$ s množinou generátorov $G \subseteq A$:

- **každý prvok $x_i \in G$ je termom algebry $\widetilde{A}$;**
- **pre každú n-árnu operáciu $f_i^{(n)}$ výraz $t = f_i^{(n)}(x_1, x_2, ..., x_n)$, kde pre každé j $1 \leq j \leq n$ $x_j \in G$, je termom algebry $\widetilde{A}$. Budeme používať označenie $t(x_1, x_2, ..., x_n)$, aby sme zvýraznili , že ide o term závislý od n argumentov (n-árny term)**
- **pre každú n-árnu operáciu $f_i^{(n)}$ a termy $t_1, t_2, ..., t_n$ je $t = f_i^{(n)}(t_1, t_2, ..., t_n)$ termom algebry $\widetilde{A}$;**
- **dva termy $t = t(x_1, x_2, ..., x_n)$ a $t' = t'(x_1, x_2, ..., x_n)$ budeme považovať za ekvivalentné, ak platí, že pre každú n-icu argumentov $x_1, x_2, ..., x_n$ $t(x_1, x_2, ..., x_n)$ a $t'(x_1, x_2, ..., x_n)$ budú vytvárať (generovať) rovnaký prvok algebry $\widetilde{A}$. Také vzťahy medzi termami sa volajú *vzťahmi totožnosti*, alebo jednoducho *totožnosti*.**

Axiomatická charakteristika algebry

- **Pomocou vzťahu totožnosti sa dajú charakterizovať vlastnosti operácií algebry $\widetilde{A}$. Základné z týchto vlastností sa formulujú ako *axiómy*, zatiaľ čo ostatné vlastnosti sa dajú odvodiť z axióm.**

ALGEBRY.

Axiomatická charakteristika boolovej algebry

$$\widetilde{B} = (B; \{+, \cdot, ^- \})$$

zákon asociatívnosti	$(x \circ y) \circ z = x \circ (y \circ z) = x \circ y \circ z$
	kde symbol $\circ$ je symbolom operácie $+$, alebo $\cdot$
zákon komutatívnosti	$x \circ y = y \circ x$
zákon idempotentnosti	$x \circ x = x$
zákon distributívnosti $\cdot$ nad $+$	$(x + y) \circ z = (x \circ z) + (y \circ z)$
zákon distributívnosti $+$ nad $\cdot$	$(x \circ y) + z = (x + z) \circ (y + z)$
zákon negácie negácie	$\overline{\overline{x}} = x$
zákon pohltenia (absorbcie)	$x + x \cdot y = x$
	$x \cdot (x + y) = x$
pravidlá de Morgana	$\overline{x \cdot y} = \overline{x} + \overline{y}$
zákon vylúčenia tretieho	$x + \overline{\overline{x}} = 1$
zákon protirečenia	$x \cdot \overline{\overline{x}} = 0$
	kde 0 a 1 sú konštanty b.a.
axiómy pre konštanty	$\overline{1} = 0, \overline{0} = 1$
	$1 \cdot x = x, 0 + x = x$
	$1 + x = 1, 0 \cdot x = 0$

MNOHO-DRUHOVÉ ALGEBRAICKÉ SYSTÉMY

- **Mnoho-druhové algebry zohrávajú významné miesto v programovaní, napríklad pri algebraických špecifikáciach abstraktných dátových typov (ADT). Sú zovšeobecnením pojmov *model* a *algebra*.**
- **Význačná črta: operácie a predikáty sú *polymorfnej* povahy.**

Definícia 0.1

Mnoho-druhovým algebraickým systémom **voláme sytém** $AS = (Osnovy;$ **Signatúra), kde** $Osnovy = \{A_i \,|i \in \mathbf{I}\}$ **a Signatúra** $= \{SIGN_{\Pi} \cup SIGN_o$ **ktorá je daná zjednotením predikátov a operácií definovaných na množine osnov.**

Príklad 0.4

$$\begin{aligned} SEQ(1) &= \{M(1)_i \,|i \in I\} \\ SEQ(2) &= \{M(2)_j \,|j \in J\} \\ SEQ(3) &= \{M(3)_k \,|k \in K\} \end{aligned}$$

- $M(1)_i : Ha_1a_2...a_\ell Y(1)a_{\ell+1}...a_nK$ je (označená) číselná postupnosť;
- $M(2)_j : Ht_1t_2...t_iY(2)a_{i+1}...a_mK$ je (označená) postupnosť symbolov;
- $M(3)_k : Hz_1z_2...z_rY(3)z_{r+1}...z_pK$ je (označená) postupnosť záznamov;

MNOHO-DRUHOVÉ ALGEBRAICKÉ SYSTÉMY

Predikáty

1) $\ell > r|Y(q) = 1 \Leftrightarrow_{df} \mathbf{M}(\mathrm{q})_r = \mathbf{M}^{'}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}(q), a_{i+1}, ..., a_n\mathbf{K} \wedge a_i > a_{i+1}$
2) $d(Y(q), K) = 1 \Leftrightarrow_{df} \mathbf{M}(\mathrm{q})_r{=}\mathbf{M}^{'}{:}\mathbf{H}a_1, a_2, ..., a_n, \mathbf{Y}(q)\mathbf{K}$
3) $UM = 1 \Leftrightarrow_{df} \mathbf{M}(\mathrm{q})_r = \mathbf{M}^{'}{:}\mathbf{H}\mathbf{Y}(q), a_1, a_2, ..., a_n\mathbf{K} \wedge a_i < a_{i+1}, i = 1, 2, ..., n-1$

Operátory

4) $E \Leftrightarrow_{df} E(\mathbf{M}(q)_r) = \mathbf{M}(q)_r$

5) $P(Y(q)) \Leftrightarrow_{df} \mathbf{M}(q)_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}(q), a_{i+1}, ..., a_n\mathbf{K}\wedge$
$\mathbf{M}(q)_2 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, a_{i+1}, \mathbf{Y}(q), a_{i+2}, ..., a_n\mathbf{K}\wedge$
$P(Y_1)(M(q)_1) = M(q)_2$

6) $L(Y(q)) \Leftrightarrow_{df} \mathbf{M}(q)_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}(q), a_{i+1}, ..., a_n\mathbf{K}\wedge$
$\mathbf{M}(q)_2 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ...a_{i-1}, \mathbf{Y}(q), a_i, , a_{i+1}, ..., a_n\mathbf{K}\wedge$
$P(Y_1)(M(q)_1) = M(q)_2$

7) $TRANSP(\ell, r) \Leftrightarrow_{df} \mathbf{M}(q)_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}(q), a_{i+1}, ..., a_n\mathbf{K}\wedge$
$\mathbf{M}(q)_2 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_{i+1}, \mathbf{Y}(q), a_i, ..., a_n\mathbf{K}\wedge$
$TRANSP(\ell, r)(M(q)_1) = M(q)_2$

8) $UST(Y(q), H) \Leftrightarrow_{df} \mathbf{M}(q)_1 = \mathbf{M}{:}\mathbf{H}a_1, a_2, ..., a_i, \mathbf{Y}(q), a_{i+1}, ..., a_n\mathbf{K}\wedge$
$\mathbf{M}(q)_2 = \mathbf{M}{:}\mathbf{H}\mathbf{Y}(q)a_1, a_2, ..., a_i, a_{i+1}, ..., a_n\mathbf{K}\wedge$
$UST(Y(q), H)(M(q)_1) = M(q)_2$

□
(Koniec príkladu)

MNOHO-DRUHOVÉ ALGEBRAICKÉ SYSTÉMY

Algebra algoritmov

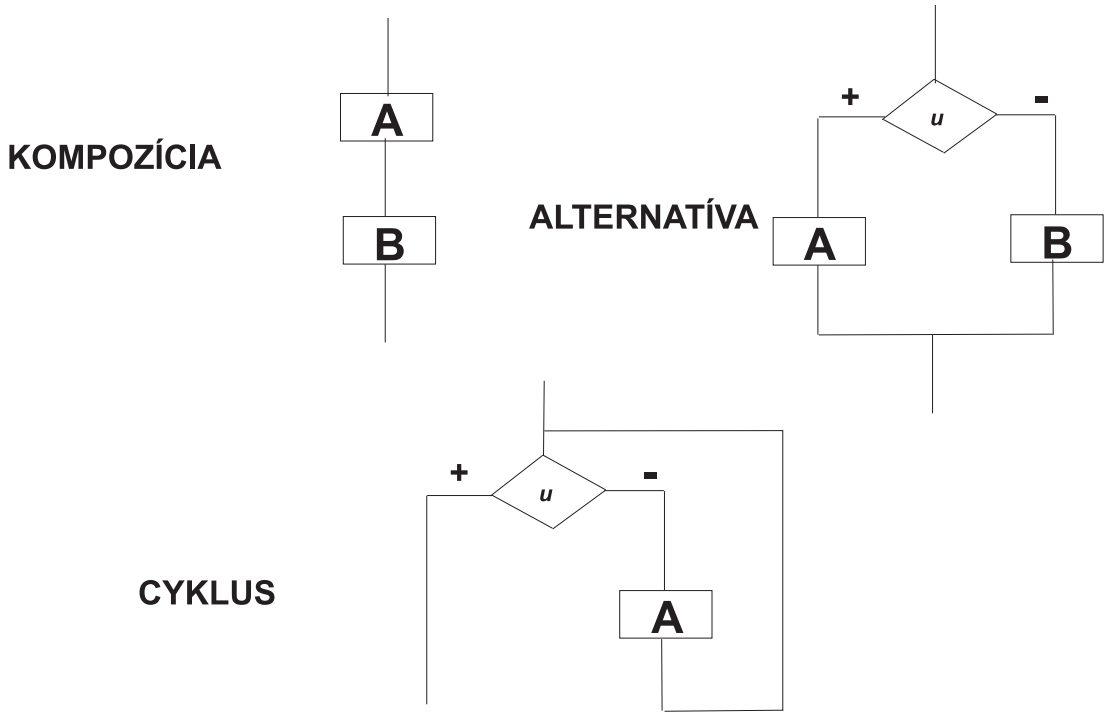
Algebra algoritmov je orientovaná na analytické vyjadrenie algoritmov, t.j. na vyjadrenie algoritmu ako niektorej formuly v takej algebre.

$$DA = (\{YC, OP\}; SIGN)$$

- **YC,OP** sú osnovy algebry DA, pričom YC je množina predikátov (podmienok) definovaných na spracovaných dátach, OP je množina operátorov (operácií) na spracovanie dát
- Signatúra operácií **SIGN** je definovaná na osnovách YC,OP. Do SIGN patria logické (boolovské) operácie: ***dizjunkcia*** ($\vee$), ***konjunkcia*** ($\wedge$) a ***negácia*** ($\neg$)- tieto sú definované na osnove YC. Na osnove OP sú definované operácie: **kompozícia** ($*$), **alternatíva** a **cyklus**. Pre tieto operácie budeme používať tieto označenia :

kompozícia	$A * B$
alternatíva	$([u]\, A, B)$
cyklus	$\{[u]\, A\}$

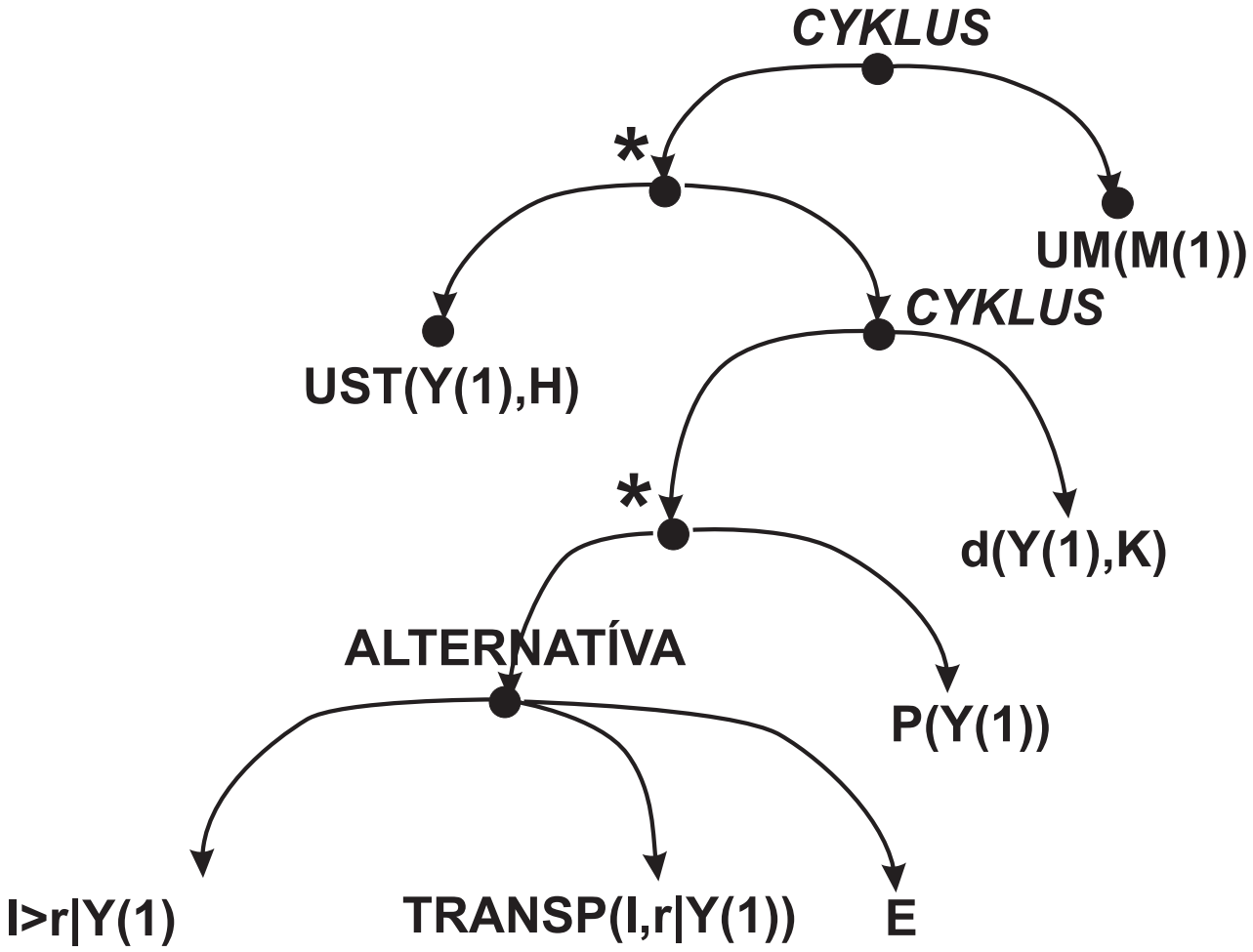
MNOHO-DRUHOVÉ ALGEBRAICKÉ SYSTÉMY



Obrázok 1: Grafická reprezentácia operáciií algebry DA

MNOHO-DRUHOVÉ ALGEBRAICKÉ SYSTÉMY

$$Bubble ::= \{[UM(M(1))]\,\{[d(Y(1),K)]\,([\ell > r|Y(1)]\,TRANSP\,(\ell,r,|Y(1)))\,,E) * \\ * P\,(Y(1)\} * UST(Y(1),H)\}$$



Obrázok 2: Grafová reprezentácia termu Bubble

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

CIELE:

- **tvorba algebier algoritmov;**
- **formálny opis štruktúrnych a neštruktúrnych schém algoritmov;**
- **tri tvary formálneho opisu:**
  1. **analytický;**
  2. **lingvistický;**
  3. **grafový.**

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Formalizované projektovanie algoritmov

Hlavný objekt štúdia

- **schémy algoritmov;**
- **superpozícia-vkladanie jednej schémy namiesto prvkov v druhej schéme;**
- ***rozvinutie* schémy (evolúcia):projektovane algoritmu zhora-nadol (top-down);**
- ***zvinutie*schémy (involúcia, konvolúcia)- prechod k formalizovanej špecifikácii vyššej úrovne.**

Príklad 0.5

Návrh zloženej schémy Π, ktorá je špecifikáciou štruktúry algoritmov určitej triedy.

$$\begin{aligned} \Pi &::= \{[u_1]\, A_1\}, \\ A_1 &::= \{[u_2]\, A_2 * D\}, \\ A_2 &::= A_3 * C, \\ A_3 &::= ([u]\, A, B), u ::= \overline{u}_2 \wedge u_3. \end{aligned}$$

Vykonáme teraz superpozíciu - zvinutie (konvolúciu) procesu Π, ktorý je nateraz uvedený ako postupnosť evolučných krokov (špecifikáciou A_1, A_2, A_3). Po postupnom nahradení premenných A_1, A_2, A_3 dostaneme formulu

$$\Pi ::= \{[u_1]\, \{[u_2]\, ([\overline{u}_2 \wedge u_3]\, A, B) * C * D\}\}$$

□

(Koniec príkladu)

- Schéma Π je príklad *neiterpretovanej* schémy;
- Prechod od neiterpretovaných schém k algoritmom je spojený s iterpretáciou operátorových a logických (predikátových) premenných v neiterpretovanej schéme.

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Príklad 0.6

Majme označenú postupnosť

$$\mathbf{M} : \mathbf{H}a_1, a_2, ..., a_i, Y_1, a_{i+1}, ..., a_n\mathbf{K}$$

Zavedieme si teraz čiastočnú interpretáciu premenných schémy Π

$$u_2 \rightarrow d(Y_1, K),\ C \Rightarrow P(Y_1),\ D \Rightarrow UST(Y_1, H)$$

kde symboly $\rightarrow$ a $\Rightarrow$ označujú interpretáciu zodpovedajúco logických a operátorových premenných v schémach. S touto čiastočnou interpretáciou schémy Π dostávame čiastočne interpretovanú schému $C\Pi$

$$C\Pi ::= \left\{[u_1]\left\{[d(Y_1, K)]\left(\left[\overline{d(Y_1, K)} \wedge u_3\right] A, B\right) * P(Y_1) * UST(Y_1, H)\right\}\right\}$$

□

(Koniec príkladu)

Príklad 0.7

Majme $\mathbf{M}$ číselnú postupnosť; zavedieme si interpretáciu logických a operátorových premenných schémy $C\Pi$.

$$u_1 \rightarrow UM,\ u_3 \rightarrow l > r|Y_1,\ A \Rightarrow TRANSP(l, r),\ B \Rightarrow E$$

S touto interpretáciou logických a operátorových premenných schémy $C\Pi$ získavame algoritmus

$$\begin{aligned} BUBBLE\ ::=\ & \left\{[UM]\left\{[d(Y_1, K)]\left(\left[\overline{d(Y_1, K)} \wedge l > r|Y_1\right] TRANSP(l, r), E\right) * \right.\right. \\ & \left.\left. * P(Y_1) * UST(Y_1, H)\right\}\right\} \end{aligned}$$

□

(Koniec príkladu)

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Dijkstrova algebra.

E.W.Dijkstra (1930- 2002)- holandský informatik, ETU

- štrukturované programovanie (without GOTO);
- konštrukcie:***sekvencia, alternatíva a cyklus***;
- Dijkstra ešte v r.1968 vo svojom dopise Akadémii vied USA navrhoval vytvorenie algebry algoritmov založenej na konštrukciach ***sekvencia, alternatíva a cyklus***.

Definícia 0.2

Dijkstrova algebra je 2-druhový algebraický systém $AD = (ACC, L(2); SIGN)$ osnovami ktorej sú: množina operátorov ACC, pozostávajúca zo štruktúrnych schém a množina boolovských funkcií L(2). Signatúru SIGN vytvárajú operácie sekvencia, alternatíva a cyklus (ktoré nadobúdajú hodnoty patriace do ACC) a boolovské operácie konjunkcia ($\wedge$), dizjunkcia ($\vee$), negácia ($\neg$)(ktoré nadobúdajú hodnoty patriace do L(2)). Premenné $A = \{A_1, A_2, ..., A_n\}$ a $U = \{u_1, u_2, ..., u_m\}$ sa využívajú na označenie zodpovedajúco elementárnych (bázových) operátorov a logických podmienok.

□

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

TRANSFORMÁCIE SCHÉM ALGORITMOV

G.E.Cejtlin (1940-)- ukrajinský informatik, IPS NAN Kyjev

$$\{[u]\,A\} = ([u]\,E, A * \{[u]\,A\}) \tag{14}$$

odvodené operácie:

$$\begin{aligned}
\Phi(u) &= ([u]\,E, N)\\
&\quad \textit{filtrácia}\\
\{A\,[u]\,B\} &= A * \{[u]\,B * A\}\\
&\quad \textit{zovšeobecnený cyklus } \textbf{DO-WHILE-DO}\\
\{E\,[u]\,B\} &= [u]\,B\\
&\quad \textit{zovšeobecnený cyklus } \textbf{WHILE-DO}\\
\{A\,[u]\,E\} &= \{A\,[u]\}\\
&\quad \textit{zovšeobecnený cyklus } \textbf{DO-WHILE}
\end{aligned}$$

Vlastnosti operácie alternatívy a cyklu

$$\{[u]\,A\} = \{[u]\,\Phi(\overline{u}) * A\} \tag{15}$$

$$\left\{[u]\,\Phi(\overline{u}) * \left(\left[\overline{u} \wedge u'\right] A, B\right) * C\right\} = \left\{[u]\,\Phi(\overline{u}) * \left(\left[u'\right] A, B\right) * C\right\} \tag{16}$$

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Príklad 0.8

Uvedieme teraz proces transformácie schémy Π z príkladu 0.5 do kompaktnejšieho tvaru s použitím rovností (15) a (16). Proces transformácie bude sprevádzaný komentármi, ktoré sú uvedené v komentárových zátvorkach (/* a */).

$$\Pi ::= \{[u_1]\,\{[u_2]\,([\overline{u}_2 \wedge u_3]\,A, B) * C * D\}\}$$

/* Použijeme rovnosť (15), k formovaniu filtra $\Phi(\overline{u})$ pred vloženou alternatívou*/

$$= \{[u_1]\,\{[u_2]\,\Phi(\overline{u}_2)\,([\overline{u}_2 \wedge u_3]\,A, B) * C * D\}\} =$$

/* Na základe rovnosti (16) dochádza k absorbcii (pohlteniu) prvého konjunktívneho súčiniteľa v podmienke vloženej alternatívy */

$$= \{[u_1]\,\{[u_2]\,\Phi(\overline{u}_2)\,([u_3]\,A, B) * C * D\}\} =$$

/* Použijeme rovnosť (15) v opačnom smere */

$$= \{[u_1]\,\{[u_2])\,([u_3]\,A, B) * C * D\}\}\,.$$

Realizovanú postupnosť transformačných krokov môžeme považovať za formálny dôkaz (odvodenie) rovnosti:

$$\left\{[u]\,(\left[\overline{u} \wedge u'\right]\,A, B) * C\right\} = \left\{[u]\,(\left[u'\right]\,A, B) * C\right\}.$$

□

(Koniec príkladu)

Ak teraz aplikujeme interpretácie premenných schémy Π ako v príkladoch 0.6 a 0.7 dostaneme kompaktnejšie vyjadrenie algoritmu BubbleSort:

$$BUBBLE ::= \{[UM]\,\{[d(Y_1, K)]\,([l > r|Y_1]\,TRANSP(l, r), E) * P(Y_1)\} * UST(Y_1, H)\}$$

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

ALGEBRA JANOVA (1958)

Definícia 0.3

Janovova algebra je 2-druhový algebraický systém

$$AJ = (\{AHC, L(2)\}; SIGN')$$

osnovami ktorej sú: množina AHC neštrukturovaných (označených i neoznačených) schém a množina boolovských funkcií L(2). Signatúru $SIGN'$ vytvárajú operácie kompozície $A*B$, neštrukturovaného prechodu $\Pi(u, F)$ a boolovské operácie konjunkcia ($\wedge$), disjunkcia ($\vee$), negácia ($\neg$)(ktoré nadobúdajú hodnoty patriace do L(2)).

□

1. Operátorové schémy algoritmov Janova (v ďalšom iba schémy Janova) sú založené na špeciálne označených postupnostiach (viď ďalej)
2. Schémy Janova patria do kategórie *neštrukturovaných* schém algoritmov;
3. Signatúra AJ si vyžaduje, vychádzajúc z povahy operátorových schém Janova, zavedenie dvoch operátorov: kompozícia $A * B$ a podmienený prechod $\Pi(u) \downarrow_m$, ktorý, pri pravdivej podmienke u, odovzdáva riadenie na značku (marker) m. V opačnom prípade sa realizuje, v realizovanej(spracovávanej) postupnosti operátorov, napravo od aktuálnej

pozície stojaci operátor. Operátor podmieneného prechodu $\Pi(u) \downarrow_m$ zodpovedá príkazy *GO TO* v programovacích jazykoch.

4. Nech **F** je neštrukturovaná logická schéma (v ktorej sú použité operácie kompozície a podmieneného prechodu), na ktorú možno nazerať ako na označenú symbolovú postupnosť, ktorá je označená dvomi markermi:**I** a m:. Marker **I** môže byť umiestnený buď na začiatku schémy **F**, alebo hneď za symbolom $*$, alebo $\downarrow$. Marker *m:* označuje ľubovoľný výskyt niektorého operátorového podvýrazu v schéme **F**. Podotýkame, že zavedenie takých označení v schéme **F** nenarúša poradie realizácie operátorov v nej. V tom prípade podmienený operátor možno interpretovať ako niektorú binárnu operáciu $\Pi(u, F)$, ktorá závisí od podmienky u a označenej schémy **F**. Výsledok aplikácie takej operácie je nová schéma $\mathbf{F}'$, v ktorej je marker **I** v schéme **F** nahradený symbolom $\Pi(u) \downarrow_m$.

Príklad 0.9

Nech je daná schéma $F_1 ::= A * B * C$. Vytvoríme označenú schému $F_1 ::= A * \mathbf{m:}B * \mathbf{I}\ C$. Aplikácia operácie $\Pi(u, F_1)$ na F_1 nám dá schému

$$F_1' = \Pi(u, F_1) ::= A * \mathbf{m:}B * \Pi(u) \downarrow_m C$$

Všimnime si, že aplikácia operácie $\Pi(u, F_1)$ na sekvenčnú schému F_1 priviedla k cyklickej schéme F_1'. Špeciálnym prípadom operácie $\Pi(u, F_1)$ pri u=1 je operácia bezpodmienkového prechodu $\Pi \downarrow_m$, ktorá závisí len od označenej schémy F_1.

□

(Koniec príkladu)

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Reprezentácia schém AD v AJ

$$
\begin{aligned}
([u]A, B) &= \Pi(u) \downarrow_m B * \Pi \downarrow_{m'} m : A * m^{'} : E, \\
\{[u]A\} &= m : \Pi(u) \downarrow_{m'} A * \Pi \downarrow_m *m^{'} : E, \\
\{A[u]\} &= m : A * \Pi(\overline{u}) \downarrow_m E, \qquad (17) \\
\{A[u]\} &= A * \left\{ [u]\, A^{'} \right\}, \qquad (18)
\end{aligned}
$$

kde $A = A^{'}$, alebo A sa líši od $A^{'}$ prítomnosťou značiek.

Veta 0.1 *V algebre AJ je vyjadriteľná ľubovoľná štruktúrna schéma, ktorá je vyjadriteľná v algebre AD;inými slovami platí, že $ACC \subset AHC$ a teda vyjadrovacia sila AJ je väčšia ako AD. Tu ACC je množina schém vyjadriteľných v AD a AHC je množina schém vyjadriteľných v zodpovedajúcej algebre Dijkstru AD algebre Janova AJ.*

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

Príklad 0.10

Ukážeme ako je možno pretransformovať štrukturovanú schému na neštrukturovanú pomocou aplikácie uvedených rovností.
Ako príklad si vezmeme algoritmus opisujúci v AD fungovanie abstraktnej tlačiarne:

PRINT ::=
TLAČ PRVÉHO RIADKU$*$
$\{[eof]$ **TLAČ AKTUÁLNEHO RIADKU**$\}$;

TLAČ PRVÉHO RIADKU ::=
$P(Y_1) *$ **TLAČ**$*$
$\{[eol]\, P(Y_1) *$ **TLAČ**$\} *$
CRL;

TLAČ AKTUÁLNEHO RIADKU ::=
TLAČ $* \{[eol]\, P(Y_1) *$ **TLAČ**$\} *$ **CRL**;

kde

- *eof* je predikát označujúci koniec súboru (**e**nd **o**f **f**ile);
- *CRL* je operácia zabezpečujúca návrat hlavy a posun o 1 riadok (**c**arriage **r**eturn and **l**ine);
- *eol* je predikát označujúci koniec riadku (**e**nd **o**f **l**ine).

Predpokladá sa, že v počiatočnom stave indikátor Y_1 sa nachádza naľavo od 1. symbolu súboru, ktorý bude vytlačený a pri prechode na nový riadok sa indikátor Y_1 nastavuje na 1. symbol nového riadku.

Prejdeme teraz k neinterpretovanej schéme S(PRINT), ktorá odráža štruktúru schémy PRINT:

$$\mathbf{S(PRINT)} ::= \mathbf{A} * \mathbf{B}\,\{[\mathbf{u}]\,\mathbf{A} * \mathbf{B}\} * \mathbf{C} * \left\{\left[\mathbf{u}^{'}\right]\,\mathbf{B} * \{[\mathbf{u}]\,\mathbf{A} * \mathbf{B}\} * \mathbf{C}\right\}$$

kde A::= $P(Y_1)$, B::= TLAČ, u::= eol, C ::= CRL, $u^{'} ::= eof$.
Pretransformujeme teraz schému S(PRINT) na jej neštrukturovaný ekvivalent s použitím vyššie uvedených rovností:
S(PRINT) = /*pretože ľavý kontext základného cyklu B $\{[u]\,A * B\} * C$ je totožný s jeho telom, použijeme rovnosť (18) v smere zprava doľava */

$$= \mathbf{A} * \mathbf{B}\left\{\mathbf{B} * \{[\mathbf{u}]\,\mathbf{A} * \mathbf{B}\} * \mathbf{C}\left[\mathbf{u}^{'}\right]\right\} =$$

/*Použijeme rovnosť (17) */

$$= \mathbf{A} * \mathbf{m}^{'} : \mathbf{B} * \{[\mathbf{u}]\,\mathbf{A} * \mathbf{B}\} * \mathbf{C} * \mathbf{\Pi}(\mathbf{u}^{'}) \downarrow_{\mathbf{m}'} =$$

/*Znovu použijeme rovnosť (18) */

$$= \left\{\mathbf{A} * \mathbf{m}^{'} : \mathbf{B}\,[\mathbf{u}]\right\} * \mathbf{C} * \mathbf{\Pi}(\mathbf{u}^{'}) \downarrow_{\mathbf{m}'} =$$

/*Napokon znovu použijeme rovnosť (17) */

$$= \mathbf{m} : \mathbf{A} * \mathbf{m}^{'} : \mathbf{B} * \mathbf{\Pi}(\mathbf{u}) \downarrow_{\mathbf{m}} \mathbf{C} * \mathbf{\Pi}(\mathbf{u}^{'}) \downarrow_{\mathbf{m}'} .$$

Ak zodpovedajúcim spôsobom interpretujeme získanú neštrukturovanú schému, získame ekvivalentnú reprezentáciu algoritmu PRINT v AJ.

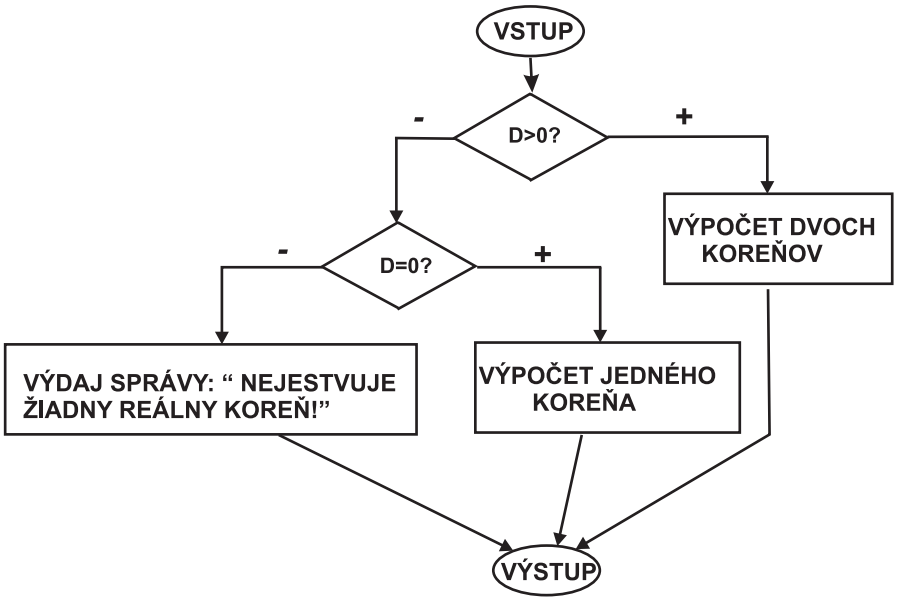
□

(Koniec príkladu)

ALGEBRY ALGORITMOV LOGIKY A SCHEMATOLÓGIE

GRAF-SCHÉMY ALGORITMOV

Kalužnin, L.A.(1923)- ruský matematik



Obrázok 3: Graf-schéma algoritmu riešenia kvadratickej rovnice.

Superpozícia g-s

$$G'' = G(A_j \Rightarrow G')$$

Algebra Kalužnina

$$AK = (\{OP, YC\}; SIGN)$$

- OP - množina operátorových premenných reprezentovaných v grafovej forme,
- YC-množina logických podmienok reprezentovaných v grafovej forme
- SIGN-signatúra operácií zahrňujúca:
  1. boolovské operácie,
  2. operáciu kompozície reprezentovanej prostredníctvom orientovanej hrany spájajúcej dva operátorové vrcholy graf-schémy;
  3. unárnu operáciu *reštrukturalizácie* grafu - prepnutie jednej z jeho orientovaných hrán vychádzajúcej z niektorého operátorového vrchola, alebo rozpoznávača a jej napojenie na niektorý iný vrchol graf-schémy (g-s) algoritmu s následným odstránením ´visiacich´- nedosiahnuteľných z koreňa g-s vrcholov.

operáciu kompozície reprezentovanej prostredníctvom orientovanej hrany spájajúcej dva operátorové vrcholy graf-schémy; unárnu operáciu *reštrukturalizácie* grafu - prepnutie jednej z jeho orientovaných hrán vychádzajúcej z niektorého operátorového vrchola, alebo rozpoznávača a jej napojenie na niektorý iný vrchol graf-schémy (g-s) algoritmu s následným odstránením ´visiacich´- nedosiahnuteľných z koreňa g-s vrcholov. Operácia reštrukturalizácie zodpovedá operácii prechodu známej zo schém Janova a vchodiacej do jej signatúry.

ALGORITMICKÉ ALGEBRY GLUŚKOVA (SAA)

Hlavné charakteristiky:

- orientácia na analytickú formu reprezentácie algoritmov;
- orientácia na akokoľvek hlbokú formalizovanú transformáciu analytickej reprezentácie algoritmov, špeciálne s cieľom optimalizovať algoritmy podľa zvolených kritérií;
- SAA sú získané z AD rozšírením jej signatúry o operáciu *prognozovania.*

$$\mathbf{AG} = (\{\mathbf{OP}, \mathbf{YC}\}; \mathbf{SIGN}'')$$
$$SIGN'' = SIGN \cup \{progn\}$$

$$\mathbf{u} = \mathbf{A} \bullet \mathbf{u}'$$

Táto operácia definuje predikát $u \in YC$ s vlastnosťou, že $u(m) = u'(m')$, kde $m' = A(m)$). $A \in OP$, $u' \in YC$ a $m, m' \in IM$‘; tu OP a YC je zodpovedajúco množina operátorov a množina podmienok a IM je informačná množina spracovávaných dát, na ktorej sú definované operátory z OP a podmienky z YC. Z toho plynie, že operácia prognozovania pozostáva z previerky (testu) podmienky u' po vykonaní operácie A. Táto previerka slúži ako prognóza o pokračovaní výpočtového procesu, ktorá sa realizuje prostredníctvom priradenia podmienke u v stave m(do vykonania operátora A) hodnoty u', ktorá sa vypočíta v stave m' do ktorého systém prejde po realizácii operátora A.

Príklad 0.11

Vysvetlíme si podstatu operácie prognozovania pri výpočte podmienky UM v príklade algoritmu BUBBLE′ .Takú previerku pravdivosti predikátu UM je treba uskutočniť v procese triedenia postupnosti, bez nutnosti zavedenia ďalších ukazovateľov. Použijeme k tomu operáciu prognozovania:

$$UM ::= SKAN \bullet [d(Y_1, K)]\,,$$
$$\text{kde SKAN} ::= \{[d(Y_1, K) \vee (l > r|Y_1)]\, P(Y_1)\}. \qquad (19)$$

Pri ´vstupe´ postupnosti $M \in SEQ$ na vstup predikátu UM realizuje sa prehliadanie (skanovanie) prvkov prostredníctvom premiestňovania ukazovateľa Y_1 doprava až do dosiahnutia značky **K**, alebo zafixovaní neusporiadanej dvojice (l,r). Po vykonaní tohto cyklu sa uskutoční previerka hodnoty predikátu $d(Y_1, K)$, ktorá sa priradí predikátu UM v súlade so sémantikou operácie prognozovania a Y_1 sa vracia k značke **H**.

□

(Koniec príkladu)

Zafixujeme si bázu $\mathbf{I} \in (OP \cup YC)$; *Interpretovanou regulárnou schémou*(PC) $F/I \in OP$ sa volá superpozícia operácií zo signatúry $SIGN''$ a prvkov bázy **I**, ktorá predstavuje zložený operátor (algoritmus) **F/I** v AG.
Ďalej uvádzané výsledky boli získané v rámci výskumov týkajúcich sa rozvoja formalizmu založeného na algebrách a formálnych gramatikách a ktoré sa vzťahujú k AG [**?**].

Veta 0.2 *[?] Každý algoritmus A (vrátane programu, alebo mikroprogramu) môže byť reprezentovný niektorou PC F/I v AG, t.j. A=F/I. To znamená, že algebry Gluškova z hľadiska svojej vyjadrovacej sily sú totožné so známymi algoritmickými systémami (ako napríklad Turingove stroje).*

□

Zvolíme si bázu algebry algoritmov ako množinu V operátorových a logických premenných $V = A \cup U$, kde $A = \{A_1, A_2, ..., A_m\}$ a $U = \{u_1, u_2, ..., u_n\}$; takú algebru budeme volať *neinterpretovanou.* Platí táto

Veta 0.3 *Pre neinterpretované algebry Dijkstry, Janova a Gluškova platia nasledujúce vzťahy vlastnej inklúzie:*$ACC \subset AHC \subset OP$, *kde ACC ,*

AHC a OP sú triedy operátorových schém vyjadriteľných v zodpovedajúcich algebrách.

□

Na formalizáciu nedeterministických a paralelných výpočtov sú orientované modifikované SAA, na ktorých je založená tzv. *štruktúrna schematológia*[**?**].
SAA Gluškova poslúžili ako prototyp programových logík a súčasným výskumom v oblasti funkcionálnych, algebraických a algebraicko-gramatických formalizmov a metódam transformačnej syntézy programov.

ALGEBRA LOGIKY A PROBLÉM FUNKCIONÁLNEJ ÚPLNOSTI

Motivácia:

Potreba zaoberať sa algebrou logiky je daná dvomi príčinami:

1. **princíp tvorby boolovských funkcií je použitý pri tvorbe algebier algoritmov; je vytvorená algebra schematológie ako metaalgebra, ktorá zahŕňa rôzne algebry algoritmov, vrátane tých, s ktorými sme sa oboznámili vyššie;**
2. **algebra logiky je významnou súčasťou algebry schematológie.**

ALGEBRA LOGIKY A PROBLÉM FUNKCIONÁLNEJ ÚPLNOSTI

PREREKVIZITY

1. **teória boolovských funkcií;**
2. **formy reprezentácie boolovských funkcií;**
3. **kanonické formy reprezentácie boolovských funkcií (undf,uncf);**
4. **úplné systémy boolovských funkcií;**

Ďalšie pojmy:

- **Majme b.f.** $f(x_1, x_2, ..., x_n)$**; o b.f.** ***f*** **budeme hovoriť, že *podstatne závisí* od** x_i **ak sa nájde aspon jedna n-1-ica premenných** $(a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n)$ **a taká, že**

$$f(a_1, a_2, ..., a_{i-1}, 0, a_{i+1}, ..., a_n) \neq f(a_1, a_2, ..., a_{i-1}, 1, a_{i+1}, ..., a_n) \quad (20)$$

  Ak premenná x_i **nie je podstatná pre *f* potom ju voláme *fiktívnou* premennou pre *f*.**

- **O dvoch b.f.** ***n*** **premenných** $f(x_1, x_2, ..., x_n)$ **a** $g(x_1, x_2, ..., x_n)$ **hovoríme, že sú *duálne*, ak platí,**

$$f(x_1, x_2, ..., x_n) = \overline{g}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n) \quad (21)$$

Príklad 0.12

Tabulka:B.f.2-premenných podstatne závislé od 2-premenných.

x	**y**	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Oznacenie		$\wedge$	$\vee$	—	$\downarrow$	$\oplus$	$\equiv$	$\Rightarrow$	$\not\Leftarrow$	$\Leftarrow$	$\Rightarrow$
0	**0**	0	0	1	1	0	1	1	0	1	0
0	**1**	0	1	1	0	1	0	1	1	0	0
1	**0**	0	1	1	0	1	0	0	0	1	1
1	**1**	1	1	0	0	0	1	1	0	1	0

kde

$f_1(x,y) = x \wedge y$ **−konjunkcia**
$f_2(x,y) = x \vee y$ **−disjunkcia**
$f_3(x,y) = x|y$ **−-Shaefferova čiarka**
$f_4(x,y) = x \downarrow y$ **−-Pearceova šípka**
$f_5(x,y) = x \oplus y$ **−súcet modulo 2**
$f_6(x,y) = x \equiv y$ **−ekvivalencia**
$f_7(x,y) = x \Rightarrow y$ **−implikácia**
$f_8(x,y) = x \not\Leftarrow y$ **−obrátená anti**
$f_9(x,y) = x \Leftarrow y$ **−obrátená imp**
$f_{10}(x,y) = x \not\Rightarrow y$ **− antiimplikác**

□

(Koniec príkladu)

Definícia 0.4

Algebraický systém AL=(BF;SIGN$_L$), kde BF je množina všetkých b.f. a SIGN$_L$ pozostáva zo superpozície, premenovania a stotožnenia (premenných), ako aj pridávania a odobratia fiktívnych premenných, voláme (dvojznačnou) algebrou logiky.

□

ALGEBRA LOGIKY A PROBLÉM FUNKCIONÁLNEJ ÚPLNOSTI

Ďalšie pojmy:

V ďalšom budeme pre undf b.f. $f(x_1, x_2, ..., x_n)$ **(??) používať označenie** D_f **a teda**

$$D_f = f(x_1, x_2, ..., x_n) = \sum_{i=0}^{2^{n-1}} r_i \cdot m_i(x_1, x_2, ..., x_n) \tag{22}$$

Majme D_f **pre b.f.** $\overline{f \equiv 0}$**; ak zameníme v (23) symboly** $+$ **za symboly operácie** $\oplus$**-*súcet modulo 2* dostaneme novú reprezentáciu *f*, ktorá sa nazýva *úplnou bisumárnou normálnou formou* (ubnf) b.f.** f **a budeme ju označovať ako** B_f**.**

$$B_f = f(x_1, x_2, ..., x_n) = \oplus_{i=0}^{2^{n-1}} r_i \cdot m_i(x_1, x_2, ..., x_n) \tag{23}$$

Nie je tažké vidieť, že zámena symbolov $+$ **za symboly operácie** $\oplus$ **nemení hodnotu reprezentovanej funkcie. To plynie z povahy mintermov (nadobúdajú hodnotu 1 práve na jednej n-ici boolovských hodnôt** $(a_1, a_2, ..., a_n)$**) a povahy operácií** $+$ **a** $\oplus$ **(sú totožné na všetkých 2-iciach argumentov okrem (1,1)).**

Potom platí tento

Dôsledok 0.1

Každá b.f. $f(x_1, x_2, ..., x_n)$ **je jednoznacne reprezentovatelná svojou ubnf.**

□

Dôsledok 0.2

Systém operácií pozostávajúci z konjunkcie ($\wedge$), súčtu modulo 2 ($\oplus$) a negácie je úplný.

□

Princíp duality umožnuje sformulovať tento

Dôsledok 0.3

Systém operácií pozostávajúci z disjunkcie ($\vee$), ekvivalencie ($\equiv$) a negácie je úplný.

□

Ako príklad uvedieme pre b.f. f_2 **z tab. 0.12 jednotlivé formy jej reprezentácie:**

$$\begin{aligned} f_2(x_1, x_2) &= \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2} + x_1 \cdot x_2 \qquad (udnf) \\ f_2(x_1, x_2) &= x_1 + x_2 \qquad\qquad\qquad\qquad\qquad (ucnf) \\ f_2(x_1, x_2) &= \overline{x_1} \cdot x_2 \oplus x_1 \cdot \overline{x_2} \oplus x_1 \cdot x_2 \qquad (ubnf) \end{aligned}$$

Úplné systémy operácií v AL možno považovať za signatúry operácií definovaných na množine b.f. BF. Prostredníctvom superpozície týchto operácií môže byť zostrojená ľubovoľná b.f. z BF. Množina BF spolu s vybratým úplným systémom b.f. sa volá *algebrou boolovských funkciií ABF*. Najznámejšími algebrami b.f. sú:

- **algebra Boola (AB) so signatúrou (úplným systémom)** $\wedge, \vee, \neg$**;**
- **algebra Žegalkina (AŽ) so signatúrou (úplným systémom)** $\wedge, \oplus,$ **konštanta**1

Algebra Boola vyhovuje všetkým zákonom Boolovej algebry, ktoré sme uviedli v kapitole ??. Tieto zákony tvoria základ teórie úplných foriem reprezentácie b.f., na ich základe bola dokázaná algoritmická rozhodnutelnost problému ekvivalencie b.f. a vyriešený problém minimalizácie b.f. (nájdenie najjednoduchšej formy reprezentácie b.f.).
Algebra Žegalkina AŽ=(BF;{$\wedge, \oplus,$ konštanta1}) - presnejšie operácie jej signatúry vyhovujú nasledujúcim zákonom:

zákon komutatívnosti	$(x \bullet y) = (y \bullet x)$
zákon asociatívnosti	$(x \bullet y) \bullet z = x \bullet (y \bullet z) = x \bullet y \bullet z$
	kde symbol $\bullet$ je symbolom operácie $\wedge$, ale
zákon distributívnosti	$(x \oplus y) \wedge z = (x \wedge z) \oplus (y \wedge z)$
zákon idempotentnosti konjunkcie	$x \wedge x = x$
zákon (pravidlo) krátenia	$x \oplus x = 0$

Vlastnosti konštánt:

$$x \wedge 1 = x;\ x \wedge 0 = 0;\ x \oplus 0 = x;\ x \oplus 1 = \overline{x} \quad (24)$$

Posledná z rovností (24)ukazuje, že negácia je odvodená operácia v AŽ.
Medzi AB a AŽ zaujímavú pozíciu zaujíma algebra so signatúrou: konjunkcia, suma modulo 2 a negácia. Bol rozpracovaný aparát tzv. *bisumárnych normálnych foriem*b.f. [?, ?] a študovaná problematika minimalizácie b.f. v rámci tejto algebry.
Majme b.f. $f(x_1, x_2, ..., x_n) \neq 0$ a nech f je reprezentovaná v ubnf.

$$B_f = f(x_1, x_2, ..., x_n) = \oplus_{i=0}^{2^{n-1}} r_i \cdot m_i(x_1, x_2, ..., x_n) \quad (25)$$

Zameníme teraz v každom minterme v (25) každý výskyt negovanej premennej $\overline{x}$ xa term $x \oplus 1$ podľa poslednej rovnosti z ((24)). Po otvorení zátvoriek a komprimácie konjunkcií na základe zákona idempotentnosti konjunkcie a odstránenia premenných podľa zákona krátenia získame napokon polynomiálnu reprezentáciu b.f. f v tvare

$$f = \oplus_{k-1}^{m} U_k \quad (26)$$

kde U_k je konjunktívny term. Reprezentácia (26) nesie názov *polynóm Žegalkina*(PŽ).

Príklad 0.13

Zostrojíme teraz PŽ pre b.f. f_2 **z príkladu 0.12; ubnf** f_2 **má tvar**

$$B_{f_2} = \overline{x} \wedge y \oplus x \wedge \overline{y} \oplus x \wedge y \tag{27}$$

Nahradením negovaných výskytov podľa (24) dostaneme

$$B_{f_2} = (x \oplus 1) \wedge y \oplus x \wedge (y \oplus 1) \oplus x \wedge y \tag{28}$$

Uplatnením zákona distributívnosti máme

$$B_{f_2} = x \wedge y \oplus y \oplus x \wedge y \oplus x \oplus x \wedge y \tag{29}$$

Uplatnením zákona krátenia napokon dostávame

$$B_{f_2} = x \oplus y \oplus x \wedge y \tag{30}$$

Tvar pravej strany (30) je PŽ.

□

(Koniec príkladu)

Platí táto

Veta 0.4 ***Ľubovoľná b.f.*** $f(x_1, x_2, ..., x_n) \neq 0$ ***sa dá jednoznačným spôsobom (s presnosťou do usporiadania konjunkcií) vyjadriť ako polynóm Žegalkina.***

Proof: Skutočne, predpokladajme , že pre b.f. $f(x_1, x_2, ..., x_n) \neq 0$ **sa nájdu dva rôzne PZ, napríklad** PZ_1 **a** PZ_2 **a** $PZ_1 \neq PZ_2$**, ktoré reprezentujú *f*. Pozrime sa na množinu (konjunktívnych) termov, ktoré sa súčasne nevyskytujú v** PZ_1 **a** PZ_2**; taká množina je neprázdna, lebo** $PZ_1 \neq PZ_2$**.**
Zvolíme si v tejto množine najkratšiu konjunkciu. Priradíme všetkým premenným tejto konjunkcie hodnotu 1 a všetkým ostatným premenným *f* hodnotu 0. Týme sme vytvorili n-icu $\widetilde{a}$ **na ktorej** $PZ_1(\widetilde{a}) \neq PZ_2(\widetilde{a})$**, čo protirečí predpokladu o existencii dvoch rôznych polynómov** PZ_1 **a** PZ_2 **reprezentujúcich rovnakú funkciu.**

□

Duálne tvrdenie môže buť sformulované aj pre algebru so signatúrou : disjunkcia, ekvivalencia a konštanta 0.

V AL existuje viac algebier b.f. so signatúrou predstavujúci úplný systém: algebra Schaefera (Schaeferova čiarka), algebra Piercea (Pierceova šípka). Dôkazy úplnosti systému operácií sú založené na technike vyjadrenia operácií známeho úplného systému prostredníctvom superpozícií testovaného na úplnosť systému . Také dôkazy budú námetom cvičení a samostatnej práce čitateľa.

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Triedy boolovských funkkcií

1. ***Zachovanie konštanty 0.*-trieda** T_0
2. ***Zachovanie konštanty 1.*-trieda** T_1
3. ***Samodualita.*- trieda *S***
4. ***Monotonnosť.*- trieda *M***
5. ***Linearita.*- trieda *L***

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Príklad 0.14

1. ***Zachovanie konštanty 0.*** **Z 2-argumentových b.f. sú tieto b.f. zachovávajúce 0:konjunkcia-**$(0 \wedge 0 = 0)$**; dizjunkcia-**$(0 \vee 0 = 0)$**; suma modulo 2-**$(0 \oplus 0 = 0)$**; antiimplikácia -** $(0 \not\Rightarrow 0 = 0)$**; opačná antiimplikácia -** $(0 \not\Leftarrow 0 = 0)$**;**
2. ***Zachovanie konštanty 1.*** **Z 2-argumentových b.f. sú tieto b.f. zachovávajúce 1:konjunkcia-**$(1 \wedge 1 = 1)$**; dizjunkcia-**$(1 \vee 1 = 1)$**; ekvivalencia-**$(1 \equiv 1 = 1)$**; implikácia -** $(1 \Rightarrow 1 = 1)$**; opačná implikácia -** $(11 \Leftarrow 1 = 1)$**;**
3. ***Samodualita.*** **Medzi samoduálne funkcie patrí *negácia*-**$(x = \overline{\overline{x}})$**; 3-argumentová b.f.** $f(x, y, z) = x \wedge y \vee y \wedge z \vee x \wedge z$**.**
4. ***Monotonosť.*** **Z 2-argumentových b.f. sú tieto b.f. monotonné:konjunkcia-**(f_1)**; dizjunkcia-**(f_2)**; monotonné sú aj konštanty 0 a 1.**
5. ***Lineárnosť.*** **Medzi elementárnymi 2-argumentovými b.f. sú lineárnymi b.f.** f_5 **(*suma mod 2*) a b.f.** f_6 **(*ekvivalencia*). Všetky ostatné 2-argumentové funkcie z tab. 0.12 nie sú lineárne b.f..**

□

(Koniec príkladu)

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Definícia 0.5

Dve n-ice hodnôt boolovských premenných $x_1, x_2, ..., x_n$ $\widehat{a} = (a_1, a_2, ..., a_n)$ **a** $\widehat{b} = (b_1, b_2, ..., b_n)$ **a také, že** $\widehat{a} \leq \widehat{b}$**, sa volajú susedné ak sa nájde jediná premenná** x_i **taká, že** $a_i = 0$ **a** $b_i = 1$**.**

□

Lemma 0.1 ***Ak b.f.*** $f(x_1, x_2, ..., x_n)$ ***nie je monotonná, potom sa nájdu také 2 susedné n-ice*** $\widehat{a}$ ***a*** $\widehat{a}'$ ***, že*** $\widehat{a} \leq \widehat{a}'$ ***a*** $f(\widehat{a}) > f(\widehat{a}')$

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Definícia 0.6

Rodina *K* b.f. vytvára *uzavretú triedu* (lebo hovoríme o *subalgebre AL*), ak pri ľubovoľných superpozíciach b.f. z triedy *K* a tiež premenovaniami a stotožneniami ich premenných vzniknú iba b.f. z *K* a žiadne iné.

□

Lemma 0.2 ***Triedy*** T_0, T_1, S, M, L ***sú uzavreté triedy b.f.***

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Veta 0.5 ***Majme SYST- systém b.f.; SYST je úplným systémom b.f. vtedy a len vtedy ak pre každú triedu*** T_0, T_1, S, M, L ***platí, že v SYST sa nachádza aspoň jedna b.f., ktorá nepatrí do danej triedy.***

Proof:

$(\Leftarrow)$***Nevyhnutnosť***
Nevyhnutná podmienka vyplýva z toho, že ak všetky funkcie v SYST patria aspoň do jednej z uvedených tried, tak potom systém b.f. SYST je neúplný v dôsledku lemy 0.2 o uzavretosti tried T_0, T_1, S, M, L**.**
$(\Rightarrow)$***Dostatočnosť.***
Podľa predpokladu SYST obsahuje b.f. $f_1 \notin T_0, f_2 \notin T_1, f_3 \notin S, f_4 \notin M, f_5 \notin L$**, ktoré nemusia byť nutne rôzne. Plán je nasledovný:**

a) **Najprv vytvoríme pomocou b.f.** f_1 **až** f_4 **konštanty 0,1 a negáciu;**

b) **potom pomocou b.f.** f_5 **vytvoríme konjunkciu a týme preukážeme, že systém b.f.SYST je redukovateľný k úplnému systému b.f. a tým je SYST úplný systém.**

Vezmime najprv b.f. $f_1 \notin T_0$**, čo znamená, že** $f_1(0, 0, ..., 0) = 1$**; označme si** $t = f_1(1, 1, ..., 1)$**. Môžu nastať 2 prípady:**

1. **t=0.V takom prípade stotožnením premenných v f_1 dostávame, že**

$$f_1(x, x, ..., x) = \overline{x} \tag{31}$$

Ďalej si zvolíme b.f. $f_3(x_1, x_2, ..., x_k) \notin S$**; to znamená, že sa nájdu dve protikladné k-ice** $\widetilde{a} = (a_1, a_2, ..., a_k)$ **a** $\widetilde{a}' = (\overline{a_1}, \overline{a_2}, ..., \overline{a_k})$ **na ktorých funkcia** f_3 **nadobúda rovnakú hodnotu. Rozbijeme teraz premenné funkcie** f_3 **na dve skupiny:**

- **-prvá:** $x_i \in skp1 \Leftrightarrow a_i = 0$**;**
- **-druhá:** $x_i \in skp2 \Leftrightarrow a_i = 1$**.**

Všetky premenné zo $skp1$ **stotožníme s** $\overline{x}$ **a tie zo** $skp2$ **stotožníme s** x**; dostaneme funkciu** $f_3(x)$ **a takú, že** $f_3(0) = f_3(1) = const$**. Dosadením získnej konštanty** ***const*** **do funkcie** $f_1(x, x, ..., x)$ **v (31) dostaneme druhú konštantu** $\overline{const}$**.**

2. **t=1. Po stotožnení premenných v** f_1 **dostaneme, že** $f_1(x, x, ..., x) = 1$ **a teda získavame konštantu 1. Zvolíme ľubovoľnú funkciu** $f_2 \notin T_1$ **(t.j.** $f_2(1, 1, ..., 1) = 0$**), a dosadíme do nej namiesto 1** $f_1(x, x, ..., x) = 1$**, čím dostaneme konštantu 0 podľa formuly**

$$f_2(f_1(x, x, ..., x), f_1(x, x, ..., x), ..., f_1(x, x, ..., x)) = 0 \tag{32}$$

K vytvoreniu negácie v danom prípade si zvolíme niektorú b.f. $f_4(x_1, x_2, ..., x_k) \notin M$**. Podľa lemy 0.1 pre nemonotonnú funkciu** f_4 **sa vždy nájdu 2 susedné k-ice** $\widetilde{a} = (a_1, a_2, ..., a_k)$ **a** $\widetilde{a}' = (a_1', a_2', ..., a_k')$ **, ktoré sa líšia iba v jednej položke (povedzme)** $a_i = 0$ **a** $a_i' = 1$ **(t.j.**$a_i < a_i'$ **), na ktorých** $f_4(\widetilde{a}) = 1$ **a** $f_4(\widetilde{a}') = 0$**. Dosadíme v** f_4 **namiesto premennej** x_i **premennú** $\boldsymbol{x}$ **; za**

všetky premenné x_ℓ, $\ell \neq i$ pre ktoré $a_\ell = 0$ dosadíme *konštantu 0* a za ostatné premenné *konštantu 1*. Dostaneme b.f. $f_4(x) = \overline{x}$.

Preukázali sme, že v oboch prípadoch sa dajú zostrojiť konštanty 0 a 1 a negácia.
Prejedeme teraz k zostrojeniu *konjunkcie*. Za tým účelom si vyberieme nelineárnu b.f. $f_5(x_1, x_2, ..., x_k) \notin L$, ktorá je reprezentovaná polynómom Žegalkina PZ, t.j. $f_5 = PZ$. Zvolíme si pevne premenné x a y podľa ktorých je f_5 nelineárna. Po vhodnom premenovaní premenných dostaneme:

$$f_5(x, y, z_1, z_2, ..., z_{r-2}) = x \wedge y \wedge q_1(z_1, z_2, ..., z_{r-2}) \oplus x \wedge q_2(z_1, z_2, ..., z_{r-2})$$

Podľa predpokladu o nelinearite f_5 podľa x a y je $q_1(z_1, z_2, ..., z_{r-2}) \neq 0$, čo znamená, že sa nájde r-2-ica $\widetilde{t} = (t_1, t_2, ..., t_{r-2})$ na ktorej $q_1(\widetilde{t}) = 1$. Vzhľadom k tomu, že disponujeme konštantami 0 a 1, dosadíme v (33) do f_5 za každú premennú z_i hodnotu t_i, čím získame toto vyjadrenie b.f. f_5 :

$$f_5(x, y, z_1, z_2, ..., z_{r-2}) = x \wedge y \oplus x \wedge c_1 \oplus y \wedge c_2 \oplus c_3 \qquad (34)$$

kde $c_i = q_i(t_1, t_2, ..., t_{r-2})$, $i = 1, 2, 3$. Vzhľadom na jestvujúcu konštrukciu negácie položíme $c_3 = 0$.
Konštrukcia konjunkcie bude závisieť od kombinácie hodnôt c_1 a c_2:

1. **$c_1 \neq c_2$. Konjunkciu z $f_5(x, y)$ vytvoríme dosadením do $f_5(x, y)$ negáciu jednej z jej premenných. Pri $c_1 = 1$ a $c_2 = 0$ $f_5(x, y, z_1, z_2, ..., z_{r-2}) = x \wedge y \oplus x$. Ak dosadíme**

namiesto y jej negáciu dostaneme

$$\begin{aligned} f_5(x,\overline{y}) &= x \wedge \overline{y} \oplus x \\ &= x \wedge (y \oplus 1) \oplus x \\ &= x \wedge y \oplus x \oplus x \\ &= x \wedge y \end{aligned} \tag{35}$$

2. $c_1 = c_2$**.Prípad** $c_1 = c_2 = 0$ **dáva triviálne** $f_5(x,y) = x \wedge y$**; v prípade,** $c_1 = c_2 = 1$ **ak dosadíme v** $f_5(x,y)$ **namiesto x a y ich negácie dostaneme**

$$\begin{aligned} f_5(\overline{x},\overline{y}) &= \overline{x} \wedge \overline{y} \oplus \overline{x} \oplus \overline{y} \\ &= (x \oplus 1) \wedge (y \oplus 1) \oplus (x \oplus 1) \oplus (y \oplus 1) \\ &= ((x \oplus 1) \wedge y) \oplus (x \oplus 1) \oplus (x \oplus 1) \oplus (y \oplus 1) \\ &= ((x \oplus 1) \wedge y) \oplus (y \oplus 1) \\ &= (x \wedge y) \oplus y \oplus (y \oplus 1) \\ &= (x \wedge y) \oplus 1 \quad \textbf{a definitívne} \\ f_5(\overline{x},\overline{y}) &= (x \wedge y) \oplus 1 \end{aligned} \tag{36} \tag{37}$$

Pravá strana (37) je negáciou $x \wedge y$ **a teda** $\overline{f_5(\overline{x},\overline{y})} = x \wedge y$**. Záverom môžeme konštatovať, že SYST je úplný systém b.f.**

□

VETA O FUNKCIONÁLNEJ ÚPLNOSTI (E.POST)

Dôsledok 0.4 ***Triedy*** T_0, T_1, S, M, L ***sú pre-úplné (sú maximálne subalgebry.)***

Dôsledok 0.5 ***(zoslabené kritérium úplnosti) Systém SYST pozostávajúci z konštánt 0 ,1 je úplný práve vtedy ak SYST obsahuje aspon jednu nemonotonnú a jednu lineárnu b.f.***

ALGEBRA ALGORITMIKY A JEJ APLIKÁCIE.

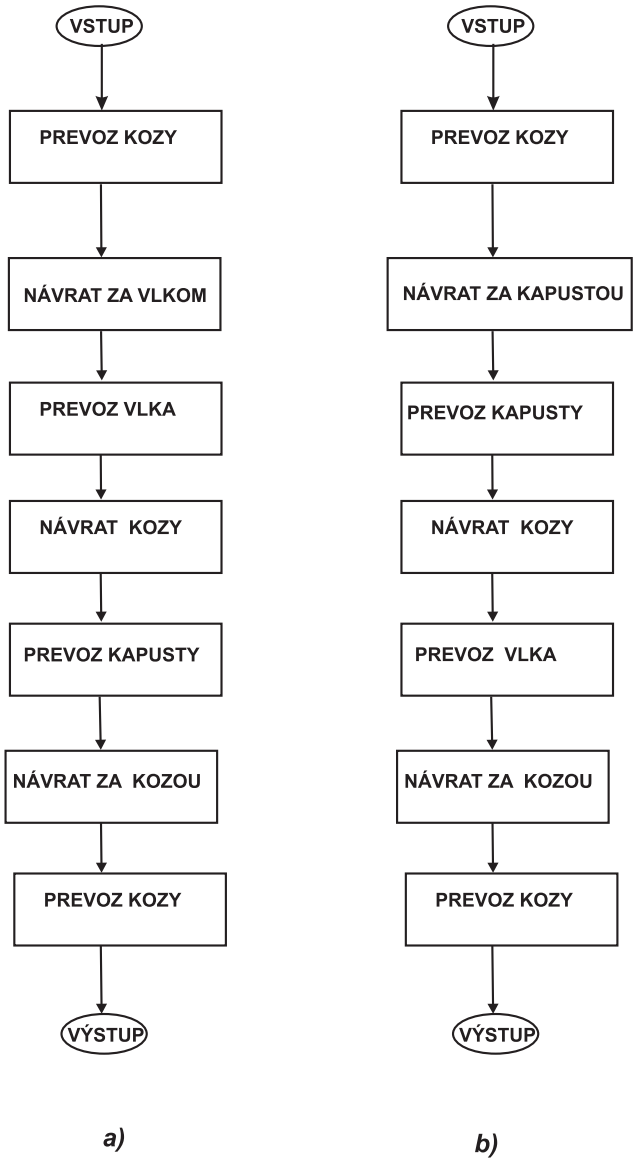
Ciele:

- **bude zostrojená algebra algoritmiky, ktorá zahŕňa ako svoju významnú zložku-algebru logiky;**
- **Budú preskúmané vlastnosti štruktúry subalgebier algebry algoritmiky (AA);**
- **Zvláštna pozornosť bude venovaná problému funkcionálnej úplnosti jej jednotlivých subalgebier, spojených s algebrami algoritmov ;**
- **Tvorba algebier algoritmov v rámci zodpovedajúcich subalgebier AA po analógii s tvorbou algebier b.f. v rámci AL**

ALGEBRA ALGORITMIKY A JEJ APLIKÁCIE.

Zovšeobecnené graf-schémy

- **Zovšeobecnená graf-schéma (zgs) $\widetilde{G}$ je označený graf ako v graf-schéme s tým rozdielom, že ak $\widetilde{G} = (V, \vec{E}, \ell)$, kde V-množina vrcholov, $\vec{E}$ -množina orientovaných hrán, potom každý vrchol zgs $\widetilde{G}$ (okrem vstupného a výstupného) je označený dvojicou *(A,u)*, kde *A* je operátorová premenná a *u* je predikátová premenná a okrem toho má dve vystupujúce hrany: plusovú (+) a mínusovú (-).**
- **Ak pri interpretácii I zgs $\widetilde{G}$ premenná *u*(vo vrchole *v* so značkou *(A,u)*,t.j $\ell(v) = (A, u)$) má hodnotu *u=const*, potom sa daný vrchol považuje za *operátorový* a pri ´vstupe´ do takého *v* prvku $m \in IM$ výsledok spracovania $m' = A(m)$ vystupuje po tej hrane vrchola *v*, ktorý zodpovedá hodnote premennej *u*;t.j. ak *u=1* výstup bude na plusovú (+) hranu, v opačnom prípade na mínusovú (-) hranu.**
- **Ak vrchol *v* má *A=E*(identický operátor), potom sa taký vrchol považuje za rozhodovcí (testovací) vrchol s podmienkou (predikátom) *u*.**
- **Vo všeobecnom prípade, ak $A \neq 0$ a $u \neq const$, to interpretujeme tak, že vo vrchole *v* s $\ell(v) = (A, u)$ sa realizuje operácia prognozovania $A \bullet [u]$ a to tak, že prvok $m \in IM$, ktorý vstupuje do *v* sa vytvorí kópia na ktorú sa aplikuje operátor *A* a v závislosti od hodnoty $u(m')$, kde $m' = A(m)$, pôvodný prvok *m* ´vystupuje´ z *v* po zodpovedajúcej hodnote $u(m')$ výstupnej hrane.**



Obrázok 4: Hlavolam prevozníka: graf-schéma

ALGEBRA ALGORITMIKY A JEJ APLIKÁCIE.

Veta 0.6 ***K ľubovoľnej PC F(I) v SAA=(OP,YC;SIGN) môže byť vytvorená ekvivalentná zgs*** $\widetilde{G}/I$ ***v lgebre ZGS -*** $F(I) = \widetilde{G}/I$***, naopak.***

□

Dôsledok 0.6 ***Ľubovoľný algoritmus (alebo program) je vyjadriteľný v algebre ZGS prostredníctvom zodpovedajúcej PF.***

□

Tento dôsledok vyplýva z vety 0.6 a vety Gluškova o *regularizácii* ľubovoľných algoritmov programov v SAA.

Veta 0.7 ***Nech AZG - algebra zgs; potom je AZG izomorfná s algebrou Gluškova AG.***

□

ALGEBRA ALGORITMIKY A JEJ APLIKÁCIE.

Zhrnutie.

- **Na vytvorenú AZG môžu buť použité, resp. rozšírené už spomínané predtým výsledky štruktúrnej schematológie.**
- **Koncepčná príbuznosť algebraických špecifikácií otvára možnosť spoločného používania grafových a analytických opisov (reprezentácií) algoritmov v procese mnohoúrovňového projektovnia tried algoritmov a programov ;**
- **Podobná interpretácia má svojim dôsledkom využitie výhod a nivelizáciu nedostatkov prítomných tak v grafových ako aj v analytických špecifikáciach algoritmov, ak sú využívané oddelene.**
- **Medzi výhody analytických špecifikácií algoritmov patria:**
  1. **kompaktnosť,**
  2. **detailnosť,**
  3. **orientácia na transformáciu špecifikovaných objektov,**
  4. **neobmedzené možnosti na použitie komentárov orientovných na objasnenie niektorých stránok špecifikácie,**
- **Medzi výhody grafických špecifikácií algoritmov patrí to, že grafové reprezentácie sú transparentnejšie v porovnaní s analytickými, avšak jej využiteľnosť je iba pre jednoduché systémy.Žiaľ táto výhoda sa stáva nevýhodou pri prechode k projektovniu zložitých algoritmov, ktoré sú základom pri projektovaní aplikačných programových systémov rôznej orientácie.**

Odporúčanie

- **Je rozumné a výhodné využívať spolu grafové a analytické špecifikácie na každej úrovni projektovania programového systému.**
- **Taký spôsob spočíva spravidla v projektovní relatívne malých algoritmov a programov.**
- **Vzájomne prepojené poúrovňové projektovnie v termínoch grafových a analytických špecifikácií takých algoritmov a programov je**
  1. **založené na možnosti automatizovaného prechodu od jednej formy k druhej,**
  2. **stimuluje prehĺbenie analýzy projektovaných algoritmov**
  3. **umožňuje transformáciu a syntézu programov s využitím nástrojov na ich generovanie.**

ALGEBRA ALGORITMIKY A APLIKAČNÉ SUBALGEBRY.

2-úrovňový algebraický systém

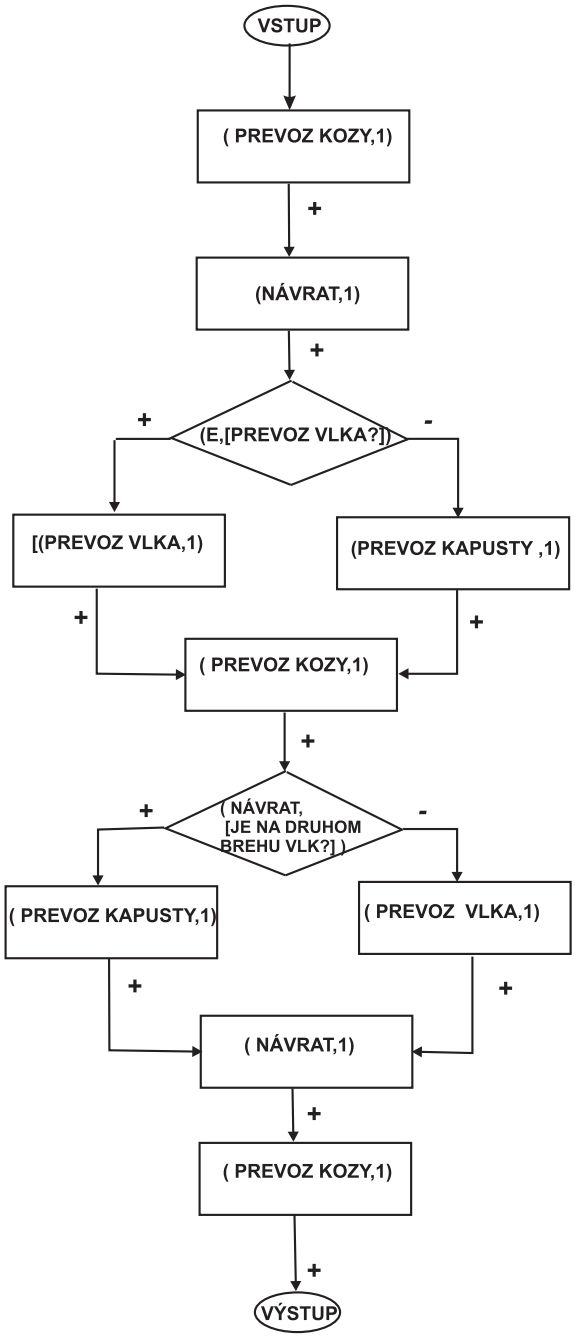
- **metaalgebra algoritmiky;**
- **m-druhový algebraický systém (m $\geq$ 2).**

m-druhový algebraický systém

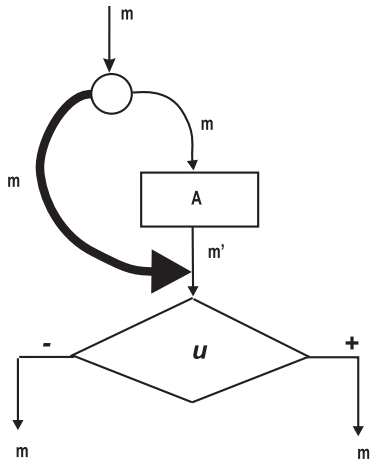
- **formalizácia ADT;**
- **asociácia s konkrétnymi predmetnými oblasťami(triedenie, vyhľadávanie, jazykové procesory atp.).**

$$MAS = (\{D_i | i \in I\}; SIGN_O, SIGN_\Pi)$$

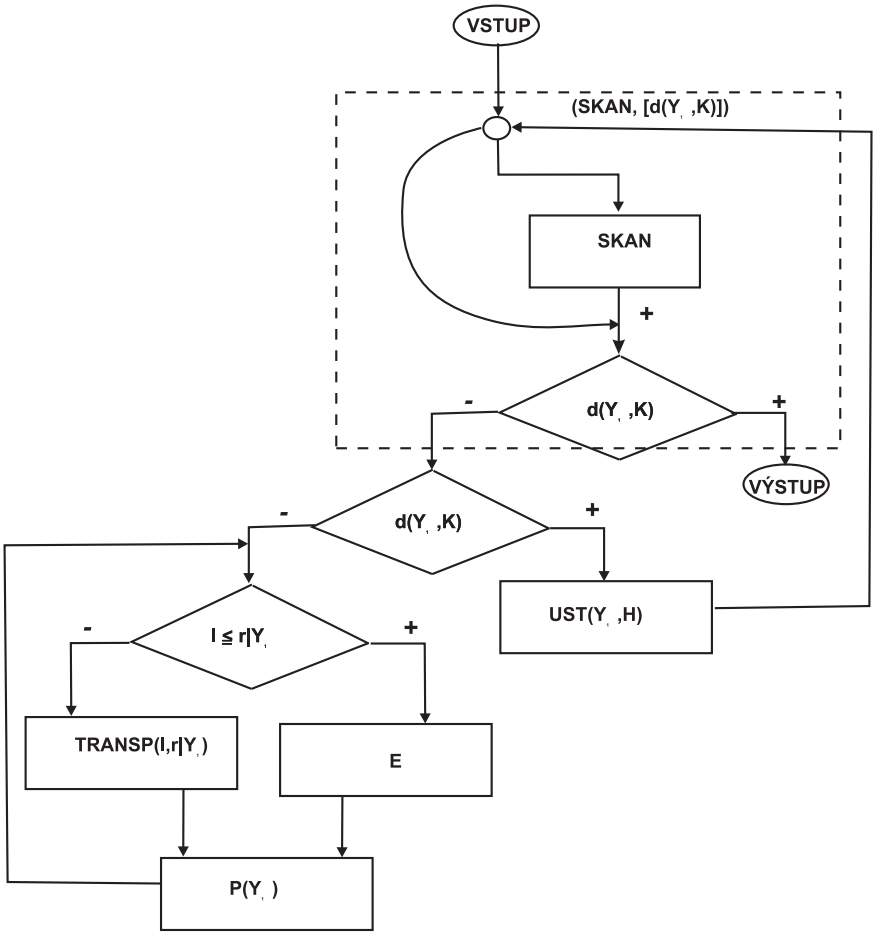
- D_i**- osnovy;**
- $SIGN_O, SIGN_\Pi$ **- zoskupenia operácií a predikátov definovaných na druhoch** D_i



Obrázok 5: Hlavolam prevozníka:zovšobecnená g-s.



Obrázok 6: Zovšobecnená g-s operácie *prognoz*



Obrázok 7: Zovšobecnená g-s algoritmu BubbleSort

Literatúra