O APLICAȚIE A TEORIEI CATEGORIILOR ÎN BAZE DE DATE

ing. Mihai Marchidann

Institutul de Cercetări în Informatică

Rezumat: Lucrarea de față încearcă să expună într-un mod cât mai accesibil rezultatele din [1] și să propună eventual și alte direcții de abordare și utilizare ale acestora. Scopul materialului este de a prezenta o demonstrație a decidabilității problemei echivalenței a două specificații de date, utilizând teoria categoriilor conform [1].

Se prezintă o formalizare a conceptului de specificație de date într-un context categorial, definindu-se noțiunea de MD-sketch. Se demonstrează că, pentru oricare două MD-sketch-uri, problema echivalenței (prin intermediul categoriilor lor de modele) este decidabilă. Dându-se o modalitate de translatare a specificațiilor de date în MD-sketch-uri cu categoriile de modele echivalente, rezultă că problema echivalenței specificațiilor de date este decidabilă. Noțiunile prezentate pot constitui un punct de plecare în studiul bazelor de date în condiții de incertitudine.

Cuvinte cheie: teoria categoriilor, baze de date, categorie, monosursă, sursă dublă, MD-sketch, specificație de date, model.

1. Introducere

O problemă de mare importanță în domeniul bazelor de date distribuite este *integrarea vizualizărilor*. Aceasta presupune o abordare modularizată în concepția aplicațiilor mari, care utilizează un anumit număr de scheme bogate în informație semantică.

Integrarea vizualizărilor presupune:

- realizarea unei scheme globale a bazei de date distribuite, pornind de la schemele inițiale;
- integrarea schemelor (vizualizărilor) ce modelează fiecare câte o aplicație sau o parte a aplicației ce utilizează baza de date.

În cazul de față, interesează un aspect esențial al problemei realizării schemei globale. Schema globală va trebui să conțină toată semantica schemelor locale inițiale. Datorită faptului că anumite părți ale lumii reale pot fi reprezentate (nu neaparat în mod identic) de mai multe scheme locale, dacă se dorește ca redundanțele din schema globală rezultată să fie minime, vor trebui găsite acele părți ale schemelor care induc aceste redundanțe.

Apare deci imperios necesară posibilitatea de a decide în cadrul procesului de integrare dacă două scheme sunt echivalente din punct de vedere semantic, adică dacă modelează aceeași parte a lumii reale.

Deoarece termenul de schemă a bazei de date, utilizat cu înțelesul de "structură" a acesteia, este generic, se va folosi în continuarea lucrării termenul de *specificație de date* [1]. Acesta reprezintă o concretizare a noțiunii de schemă într-un cadrul mai particular, ce ține de faza proiectării conceptuale a bazei de date. Un exemplu de specificație semantică de date, foarte folosit în proiectarea bazelor de date, este cel al lui Chen: diagramele Entitate Relație [7].

În continuare, se va vorbi despre problema echivalenței specificațiilor de date. Pentru o specificație de date se definește un model al său. Modelele unei specificații de date încorporează informația semantică a acesteia. Datorită dificultăților de formalizare a conceptului de echivalență la acest nivel, se introduc mecanisme categoriale. Problema echivalenței se va pune în termenii echivalenței categoriilor de modele corespunzatoare. În acest cadru categorial, se lucrează cu anumite structuri algebrice (sketch-uri) pentru care problema echivalenței este riguros definită și în legătură cu care pot fi obținute anumite rezultate utile. Ca și la specificațiile de date, echivalența a două sketch-uri se bazează pe echivalența categoriilor de modele corespunzătoare.

Categoria modelelor unui sketch (sketch limită) este caracterizată prin noțiunea de categorie accesibilă (local prezentabilă sau categorii accesibile complete).

Problema decidabilității echivalenței a două categorii accesibile (local prezentabile) este deschisă.

Se poate pune problema existenței unei clase de categorii accesibile, care să includă categoriile modelelor specificațiilor de date și să permită astfel definirea echivalenței acestor specificații.

Răspunsul la problema precedentă este afirmativ. În [1] s-a definit în limbaj categorial conceptul de specificație de date astfel încât categoria modelelor sale de fie accesibilă. Diagramele Entitate-Relație pot fi translatate în aceste specificații de date (SD).

Clasa categoriilor accesibile, corespunzătoare specificațiilor de date, poate fi definită de clasa categoriilor modelelor unui tip particular de sketch (caz special de sumă finită de sketch-uri) numit MD-sketch. În clasa categoriilor modelelor de MD-sketch-uri, problema decidabilității echivalenței este rezolvată pozitiv.

În plus, există un algoritm de translatare a specificațiilor de date în MD-sketch-uri.

Rezultă că se poate construi o schemă de decizie cu ajutorul careia să se decidă dacă două specificații de date sunt sau nu echivalente.

Rezultatele ce vor fi prezentate, evidențiază utilitatea formalizării categoriale a unor concepte (e.g. specificații de date) din domeniul bazelor de date. După cum se arată, pe această bază se pot dezvolta algoritmi utili în automatizarea procesului de integrare a vizualizărilor.

2. Preliminarii. MD-sketch-uri

Pentru început, se vor prezenta o serie de definiții ale unor concepte necesare înțelegerii materialului.

2.1. Concepte categoriale

Definiție:

Spunem că am definit o categorie C dacă am dat următoarele:

1. O clasă de obiecte (notată cu $|C|$).
2. Pentru fiecare pereche X,Y de obiecte din $|C|$ ezistă o mulțime de morfisme Hom(X,Y) (sau C(X,Y)) ale cărei elemente sunt morfismele categoriei de la X la Y.
3. Operația de compunere a morfismelor din C: o: $C(X,Y) \times C(Y,Z) \to C(X,Z)$, cu următoarele proprietăți:
   a) 'o' este asociativă: $\forall$ f, g, h cu $f \in C(X,Y)$, $g \in C(Y,Z)$, $h \in C(X,Z)$, avem ca h o (g o f) = (h o g) o f.
   b) există morfismul identic: $\forall X \in |C|$, $\exists$ $1_X \in C(X,X)$ astfel încât:
      - $\forall f: Y \to X$, $1_X \circ f = f$, și
      - $\forall g: X \to Z$, $g \circ 1_X = g$.

   c) pentru $\forall$ X,X',Y,Y' obiecte din $|C|$ cu $(X,Y) \neq (X',Y')$, avem că: $C(X,Y) \cap C(X',Y') = \varnothing$.

Se definește $Hom_C := \cup\{C(A,B)/(A,B) \in |C| \times |C|\}$, *clasa morfismelor* lui C.

Observații:

- Se poate folosi și notația diagramatică de compunere a funcțiilor: în loc de "g o f" se scrie "f; g", în care domeniul lui g face parte din imaginea lui f.
- Orice morfism are un singur domeniu și un singur codomeniu.
- Morfismul 1_A este determinat în mod unic de A, $\forall A \in |C|$.
- Înmulțirea morfismelor este o operație parțială în clasa Hom_C.

Exemplu:

Fie $(P,\leq)$ o mulțime parțial ordonată. Se va defini categoria C asociată astfel:

- $|C| := P$,
- $C(X,Y) := \begin{cases} \{(X,Y)\}, & \text{dacă } X \leq Y \\ \varnothing, & \text{altfel.} \end{cases}$

Se definește compunerea morfismelor în categoria C: $\forall X,Y,Z \in |C|$, o: $C(X,Y) \times C(Y,Z) \to C(X,Z)$ astfel:

- dacă $C(X,Y) = \varnothing$ sau $C(Y,Z) = \varnothing$, atunci: o := $\perp$, unde prin '$\perp$' am notat aplicația zero;
- dacă $C(X,Y) \neq \varnothing$ sau $C(Y,Z) \neq \varnothing$, atunci: (Y,Z) o (X,Y) := (X,Z).

Definiții:

- Un obiect $A \in |C|$ se numește *obiect inițial* al categoriei C dacă pentru $\forall X \in |C|$, atunci $\exists!$ un morfism f: $A \to X$.
- Un obiect $A \in |C|$ se numește *obiect final* al categoriei C dacă pentru $\forall X \in |C|$, $\exists!$ un morfism f: $X \to A$.

Definiție: Fie C și C' două categorii. Un functor (sau functor covariant) de la C în C', notat cu

$F: C \to C'$, este definit astfel:

- o aplicație $F: |C| \to |C'|$, $X \mapsto FX$ (imaginea obiectului X în C' prin F), o aplicație pe morfisme

$F: C(X,Y) \to C'(FX,FY)$, $\forall X,Y \in |C|$, astfel încât:

* $\forall f \in C(X,Y)$, $\forall g \in C(Y,Z)$, avem

$F(g \circ f) = Fg \circ Ff$.

* $\forall X \in |C|$, avem $F1_X = 1_{FX}$.

Fie $U: T \to S$ un functor. Pentru $\forall T_1,T_2 \in |T|$, functorul U induce o aplicație

$U_{T,S}: Hom_T(T_1,T_2) \to Hom_S(U(T_1),U(T_2))$.

Definiție:

Spunem că U este *fidel*, respectiv *plin* sau cofidel, dacă și numai dacă (dnd) aplicația $U_{T,S}$ indusă este injectivă, respectiv surjectivă.

Exemplu:

Unul dintre cele mai cunoscute tipuri de functori este *functorul de uitare* (de subiacentă):

Fie C o categorie *concretă* (adică există un functor fidel de la C în *Set*). Se definește:

$U_C = U : C \to Set$, astfel:

- $X \in |C|$, $X \mapsto UX$, unde UX este mulțimea suport subiacentă lui X;
- $\forall f \in C(X,Y)$, $Uf: UX \to UY$ este definită astfel: $(Uf)(x) := f(x)$, $\forall x \in UX$.

Se poate spune că $Uf = f$ (notație "formală"). Semnificația ar fi că U "uită" în *Set* morfismele dintre obiectele categoriei C.

Definiție:

Fie functorul $F: I \to C$, unde I este categorie *mică* (suportul este o mulțime).

Dacă $\{\alpha_i\}_{i\in I}$ este familie de morfisme în C, cu $\alpha_i: F(i) \to C$ și L este un obiect al lui C, atunci spunem că $(\{\alpha_i\}_i, L)$ este *colimită* (sau *limită inductivă, limită directă)* pentru F dacă îndeplinește:

a) $(\{\alpha_i\}_i, L)$ este *con inductiv*: pt. orice $m: i \to j$ morfism al lui I din Hom_I ($m \in I(i,j)$), avem: $\alpha_i = \alpha_j \circ F(m)$ (adică triunghiurile sunt comutative). Se notează cu $F(m) \in Hom_C$ imaginea prin F a morfismului m.

b) $(\{\alpha_i\}_i, L)$ este *con inițial*: $\forall(\{\beta_i\}_i, M)$ un alt con inductiv, $\exists!\ s: L \to M$ morfism în C astfel încât: $s \circ \alpha_i = \beta_i$.

$$\begin{tikzcd}
F(i) \arrow[rr, "F(m)"] \arrow[dr, "\alpha_i"] \arrow[ddr, "\beta_i"'] & & F(j) \arrow[dl, "\alpha_j"'] \arrow[ddl, "\beta_j"] \\
& L \arrow[d, dotted, "\exists! s"] & \\
& M &
\end{tikzcd}$$

Prin dualitate se definesc și noțiunile de *con proiectiv* și *limită proiectivă* (limita inversă).

2.2. MD-sketch-uri

Fie I o mulțime finită de indici.

Definiție:

Perechea $(X,(f_i)_{i\in I})$ se numește *sursă finită* în categoria C dacă:

a) X este obiect al C

b) $f_i: X \to Y_i$ familie de morfisme în C, pentru $Y_i \in |C|$, $\forall i\in I$.

Se notează o sursă cu $(X,(f_i)_i)$, cu $f_i: X \to Y_i$ sau și mai simplu cu $(f_i)_i$.

Definiție: O sursă $(X, (f_i)_i)$ se numește mono-sursa (familie monomorfă cu aceeași sursă) dacă din $f_i \circ x = f_i \circ y$, $\forall i$, rezultă $x = y$.

Definiție: Pentru o sursă $f_i: X \to Y_i$, familia indexată de obiecte $(Y_i)_i$ se numește baza sursei (este o diagramă discretă).

Definiție: Într-o categorie C, se numește o sursă dublă o pereche (f_i,g_i) de două surse $f_i: X \to Y_i$ și $g_i: Z \to Y_i$ în C, care au aceeași bază $(Y_i)_i$.

Definiție: O sursă dublă $(f_i,g_i)_i$ într-o categorie C se numește de disjuncție dacă limita diagramei corespunzătoare în C este obiect inițial în C.

Observație: În FinSet - categoria mulțimilor și a aplicațiilor finite - această condiție înseamnă că **nu** $\exists\ x\in X$ și **nu** $\exists\ z\in Z$ a.i. $f_i(x) = g_i(z)$, $\forall$ I.

În continuare, se presupune că C este o subcategorie a lui FinSet. Ipoteza aceasta de lucru în FinSet (subcategorie în categoria mulțimilor Set) este importantă deoarece transformările între sketch-uri conservă categoriile de modele în Set.

Definiție: Fie C o categorie și $\{A_i\}_{i\in I}$ o familie de obiecte din C. Se spune că un obiect P din C este suma finită (sau produs finit) al lui $\{A_i\}_{i\in I}$ dacă $\exists$ $(\pi_i: P \to A_i)_{i\in I}$ o familie de morfisme din C cu proprietatea de universalitate:

- pentru $\forall\ P' \in |C|$, $\forall\ (\alpha_i: P' \to A_i)_{i\in I}$, familie de morfisme din C, $\exists!$ un morfism $\gamma: P' \to P$ astfel încât diagrama următoare comută:

$$\begin{tikzcd}
P \arrow[rr, "\pi_i"] & & A_i \\
\forall P' \arrow[u, dotted, "\exists! \gamma"] \arrow[urr, "\forall \alpha_i"'] & &
\end{tikzcd}$$

Morfismele π_i se numesc proiecțiile canonice ale lui P pe A_i.

Definiție: Se numește un MD-sketch un triplet (C,M,D) unde:

- C este o categorie finită (i.e. $|C|$ este o mulțime finită de obiecte),
- M este o mulțime finită de surse din C,
- D este o mulțime finită de surse duble din C.

Definiție: Se numește un *model* al unui MD-sketch, un functor de la C în *FinSet*, care duce orice sursă din M într-o monosursă și orice sursă dublă într-o sursă dublă de disjuncție.

Observații:

- o sursă μ se mai numește *condiție de monicitate*,
- o sursă dublă $\delta=(f,g)$ se mai numește *condiție de disjuncție*.

Modelele lui (C,M,D) sunt acei functori F din Fun $(C,FinSet)$ pentru care imaginile Ff și Fg sunt disjuncte pentru $\forall\delta=(f,g)$ sursă dublă din D.

Definiție:

1) O condiție de monicitate μ se spune că *este satisfacută* de un functor F (not. $F \models \mu$) dnd $F(\mu)$ este monosursă.
2) O condiție de disjuncție δ se spune că *este satisfăcută* de functorul F (not. $F \models \delta$) dnd $F(\delta)$ este sursă dublă de disjuncție.

Pentru un MD-sketch (C,M,D), se definește mulțimea condițiilor satisfăcute de orice model al lui (C,M,D) prin $E(C,M,D):= M \cup D$. Va trebui însă găsită o reprezentare omogenă a lui M și D. Se poate spune atunci că un model pentru MD-sketch-ul (C,M,D) este $\forall\ F \in \mathrm{Fun}(C,FinSet)$ cu $F \models E(C,M,D)$, adică $F \models e$, $\forall\ e \in E(C,M,D)$.

Observație: Deoarece MD-sketch-urile sunt structuri sintactice (diagrame) într-o categorie fixată, se poate încerca încadrarea MD-sketch-urilor în contextul teoriei instituțiilor din logica ecuațională, după cum urmează:

Definiție: Se numește *instituție* un tuplu $I=(Sign$, Sen, MOD, $\models)$, unde:

- $Sign$ este categoria signaturilor;
- Sen: $Sign \rightarrow Set$, functorul care asociază unei signaturi mulțimea tuturor propozițiilor peste acea signatură;
- MOD: $Sign \rightarrow \mathrm{Cat}^{op}$, functorul care asociază fiecărei signaturi $\Sigma \in |Sign|$ o categorie ale cărei obiecte sunt Σ-modele, iar morfismele sunt morfisme de Σ-modele; Cat^{op} este duala categoriei categoriilor;
- $\models_\Sigma \subseteq |\mathrm{MOD}(\Sigma)| \times \mathrm{Sen}(\Sigma)$, $\forall \Sigma \in |Sign|$, o relație numită Σ-satisfacere astfel încât: $\forall \varphi: \Sigma \rightarrow \Sigma'$ morfism în $Sign$, să există condiția de satisfacere:

(CS): $M' \models_{\Sigma'} \mathrm{Sen}(\varphi)(e) \Leftrightarrow \mathrm{MOD}(\varphi)(M') \models_\Sigma e$, pentru $\forall M' \in |\mathrm{MOD}(\Sigma')|, \forall e \in \mathrm{Sen}(\Sigma)$.

Fie (C,M,D) un MD-sketch. Pentru încadrarea MD-sketch-urilor în contextul instituțional, vor trebui urmărite următoarele: scrierea conceptului de MD-sketch-ului sub forma unei signaturi $\Sigma \in |Sign|$. Pentru aceasta trebuie ca sursele din M și sursele duble din D să fie scrise sub forma de signaturi. Odată realizat acest lucru, înseamnă că s-a concretizat o formă pentru categoria $Sign$. Cu această categorie precizată, se va urmări ca $\mathrm{MOD}(\Sigma)$-categoria modelelor lui Σ să fie subcategorie (plină) în $\mathrm{Fun}(C,FinSet)$. Se definește un Σ-model, un model al MD-sketch-ului Σ, ca fiind un functor $(F: C \rightarrow FinSet) \in \mathrm{MOD}(\Sigma)$ cu $F \models E(C,M,D)$. Vor trebui urmărite atât uniformitatea satisfacerii de către un model (atât a surselor, cât și a surselor duble), cât și validitatea condiției de satisfacere în acest context. Pentru detalii legate de teoria instituțiilor se poate consulta [15].

Definiție: Se numește *subcategorie plină* a unei categorii C, o categorie S astfel încât:

- $|S| \subseteq |C|$,
- $\forall\ X,Y \in |S|$, $S(X,Y) = C(X,Y)$ (la subcategorie simplă aveam $S(X,Y) \subseteq C(X,Y)$).
- $\forall\ X,Y,Z \in |S|$, compunerea din S, o_S: $S(X,Y) \times S(X,Y) \rightarrow S(X,Y)$ este restricția operației de compunere din C.
- $\forall\ X \in |S|$, avem $1_X \in S(X,X)$.

Observații:

Categoria modelelor unui MD-sketch (C,M,D), notată cu $\mathrm{Mod}(C,M,D)$, este o subcategorie plină a lui Fun(C,FinSet) - categoria functorilor de la C în FinSet. $\mathrm{Mod}(C,M,D)$ conține deci acei functori din $\mathrm{Fun}(C,FinSet)$ care satisfac atât sursele din M cât și sursele duble din D. Dacă se elimină condiția de satisfacere a surselor duble din D, se obține $\mathrm{Mod}(C,M)$ - subcategoria plină a lui $\mathrm{Fun}(C,FinSet)$ ce conține toți acei functori care duc sursele în monosurse.

Observații:

- Cerința ca o sursă să fie monosursă poate fi reformulată ca un anumit con să fie con-inductiv.
- Pentru a avea și condiția de disjuncție, se pot adăuga un obiect nou în C, apoi săgeți (morfisme) de la acest nou obiect la toate obiectele vechi; condiția de disjuncție poate fi reformulată prin impunerea ca noul obiect să fie obiect inițial și ca limita sursei duble să fie tot acest nou obiect.
- Se pot considera MD-sketch-uri și în alte categorii, dar pentru ceea ce interesează este nevoie doar de modelele din *FinSet*.

În continuare, se prezintă proprietăți ale MD-sketch-urilor.

Definiție:

Fie un functor $U: C \rightarrow D$. Spunem că functorul $T: D \rightarrow C$ este *adjunctul* (la stânga) al lui U dacă:

$(\forall\ D \in |D|)\ (\exists\ \eta: D \rightarrow U(T(D)))\ (\forall\ C \in |C|)$

$(\forall\ f: D \rightarrow U(C))\ (\exists!\ f^{\#}: T(D) \rightarrow C)$ astfel încât: $U(f^{\#}) \circ \eta = f$.

Observații:

- D se numește categoria *slabă* a acestei diagrame
- $f^{\#}$ este unica extensie a morfismului f.

Definiție: O *subcategorie* plină C' a lui C se numește *reflectivă* în C dacă functorul de incluziune de la C' în C are un adjunct la stânga.

Lema: Mod(C,M) este o *subcategorie epi-reflectivă* a lui Fun($C,FinSet$). În plus, există un algoritm care să calculeze *reflexia* unui functor $F: C \to FinSet$.

2.3. Algoritmul de calcul al reflexiei unui functor:

Mod(C,M) este o subcategorie plină a lui Fun($C,FinSet$).

1) Fie $X \in |C|$ fixat. Se definește o relație $^{X}R^{F} \subseteq X \times X$ astfel: $x \, {^{X}R^{F}} \, y$ *iff* $\exists$ o sursă $f_i: X \to Y_i$ în M, astfel încât $Ff_i(x) = Ff_i(y)$, $\forall\, i \in I$.

Se notează cu M_X mulțimea acelor surse din M cu domeniul X, adică $\{\mu \in M \mid \mu = (X,(f_i)_i)\}$.

Formal avem $^{X}R^{F} := \bigcup_{\mu \in M_X} {^{X}R_{\mu}}$, unde

$$^{X}R_{\mu} := \{(x,y) \in X \times X \,/\, \mu = (f_i: X \to Y_i) \in M_X,\ Ff_i(x) = Ff_i(y),\ \forall\, i \in I\} \subseteq X^2.$$

Deși relațiile $^{X}R_{\mu}$ sunt relații de echivalență, în general reuniunea acestora, $^{X}R^{F}$, nu este o relație de echivalentă. De aceea, se construiește cea mai mică relație de echivalență din E(X) - mulțimea relațiilor de echivalență din X^2 - care să conțină $^{X}R^{F}$:

$$^{X}\equiv_{F} := \langle {^{X}R^{F}} \rangle := \bigcap \left(R^{F} \in E(X) \mid R^{F} \supseteq {^{X}R^{F}} \right)$$

Avem că: $^{X}\equiv_{F} \in E(X)$.

Reflexia lui F, notată R(F), va fi câtul lui F în raport cu relația de echivalență $\equiv_F := (^{X}\equiv_F)_{X \in |C|}$. Se obține astfel

$$F/\equiv_F : C \to FinSet.$$

Avem:

- $X \in |C|$, se definește $F/_{\equiv F}(X) := F(X)/_{X\equiv F}$;
- $\forall\, \mu \in Hom_C(X,Y)$ se definește $F/\equiv_F(\mu)$: $F/_{\equiv F}(X) \to F/_{\equiv F}(X)$ ca fiind unicul morfism $F(\mu)^{\#}$ care închide următoarea diagramă comutativă:

$$\begin{array}{ccc} F(X) & \xrightarrow{\pi_X} & F(X)/_{X\equiv F} \\ F(\mu) & & F(\mu)^{\#} \\ F(Y) & \xrightarrow{\pi_Y} & F(Y)/_{Y\equiv F} \end{array}$$

S-a definit astfel functorul $F/_{\equiv F}$ care este *reflexia lui F*, notată cu R(F), de-a lungul incluziunii I. $F/_{\equiv F}$ este model.

Rezultă următoarea diagramă:

R

$\to$

Fun(C,$FinSet$) $\xleftarrow{I}$ Mod(C,M) $\xleftarrow{J}$ Mod(C,M,D)

unde I și J sunt incluziuni, iar R este adjunctul la stânga al lui I.

Fie $F: I \to C$, și $G: C \to D$ doi functori, unde I este categorie mică. Fie D și C două obiecte în categoriile D și respectiv C.

Definiție:

Se spune că G *reflectă colimitele* dacă pentru $\forall$ $(\{\alpha_i\}_i, D)$ colimită al lui GoF, $\exists$ $(\{\beta_i\}_i, C)$ colimita al lui F astfel încât:

- $G(\beta_i) = \alpha_i$, $\forall\, i \in I$, și
- $G(C) = D$.

Definiție:

Se spune că G conservă colimitele (comută cu colimitele) dacă $\forall\, F: I \to C$, $\forall$ $(\{\beta_i\}_i, C)$ colimită a lui F, atunci $(\{G(\beta_i)\}, G(C))$ este colimită a lui GoF.

Observație: Comută cu colimitele înseamnă că duce colimite în colimite. Este suficient să se arate pentru una singură.

Au loc următoarele rezultate:

Lema: Dacă $F = J(G)$ pentru un anumit G, și dacă $\alpha: F' \to F$ este un morfism în Mod(C,M), atunci și F' este în imaginea lui J.

(dem):

Presupunem prin absurd că F' nu este în imaginea lui J, adică nu există $G' \in$ Mod(C,M,D) (G' model) astfel încât $F' = J(G') = G'$ (pentru că J este incluziunea canonică). Rezultă că F' nu este un model al lui (C,M,D), deci $\exists\ \delta = (f_i, g_i) \in D$, cu $f_i: X \to Y_i$, $g_i: Z \to Y_i$, astfel încât să **nu** avem $F' \models \delta$.

Avem $\alpha: F' \to F$ (de fapt $\alpha: |C| \to Hom_{FinSet}$) transformare naturală:

- $\alpha_X := \alpha(X) \in Hom_{FinSet}(F'(X), F(X))$
- diagrama comutativă:

$$\begin{tikzcd} F'(X) \arrow[r, "\alpha_X"] \arrow[d, "F'(f_i)"'] & F(X) \arrow[d, "F(f_i)"] \\ F'(Y_i) \arrow[r, "\alpha_{Y_i}"] & F(Y_i) \end{tikzcd}$$

adică: $\alpha_{Y_i} \circ F'(f_i) = F(f_i) \circ \alpha_X$ (*)

Din F' nu satisface δ, rezultă că $\exists\, x' \in F'(X)$ și $\exists\, z' \in F'(Z)$ astfel încât:

(**) $F'(f_i)(x') = F'(g_i)(z')$, pentru $\forall\, i \in I$.

Notăm:

- $x := \alpha_X(x') \in F(X)$, unde $x' \in F'(X)$.
- $z := \alpha_Z(z') \in F(Z)$, unde $z' \in F'(Z)$.

Din condiția (**) rezultă că (α_{Y_i} o $F'(f_i)$)(x')= (α_{Y_i} o $F'(g_i)$)(z'). Folosind (*) deducem:

$F(f_i) \circ \alpha_X(x') = F(g_i) \circ \alpha_Z(z')$,

deci $F(f_i)(x)=F(g_i)(z)$, $\forall$ $i\in I$. Rezultă că functorul F nu satisface pe δ, deci F nu este un model și nu este în imaginea lui J. Contradicție cu ipoteza. În consecință și functorul F' este în imaginea lui J.

Definiție: Fie C o categorie, iar S o categorie mică ($|S|$ este o mulțime). Numim o *diagramă* D în categoria C un functor D: $S \to C$.

Lemă: Dacă functorul F este fidel și plin (cofidel) atunci F reflectă colimitele.

Lemă: *J conservă și reflectă colimitele.*

(dem)

- Conform lemei anterioare, J reflectă colimitele deoarece J este fidel și plin.

Vrem să arătăm că J conservă colimitele.

$$\mathrm{Fun}(C,\mathit{FinSet}) \xleftarrow{I} \mathrm{Mod}(C,M) \xleftarrow{J} \mathrm{Mod}(C,M,D)$$
$$\overset{R}{\to}$$

Fie D diagrama cu colimita (λ_i, L) în *Mod(C,M,D)*, atunci J(D) este o diagrama în Mod(C,M). Deoarece Mod(C,M) este *co-complet*, diagrama J(D) are colimita. Vom arăta că $(\{J(\lambda_i)\}_i, J(L))$ este colimita diagramei J(D).

Fie (λ_i', L') o colimită a diagramei J(D) (am arătat că există). Deoarece este $(\{J(\lambda_i)\}_i, J(L))$ co-con în Mod(C,M) (triunghiurile sunt comutative deoarece sunt imagini ale unor triunghiuri comutative din Mod(C,M,D)), din proprietatea de universalitate a colimitei L' avem că $\exists$ α: $L' \to J(L)$ morfism în Mod(C,M) astfel încât $\alpha \circ \lambda_i' = \lambda_i$, $\forall i$. Din lema anterioară rezultă că și colimita L' este în imaginea lui J. Datorită faptului că J reflectă colimitele, rezultă că există o colimită G în Mod(C,M,D) cu $J(G)=L'$. Cum G și L sunt izomorfe (deoarece, colimitele sunt unice pană la un izomorfism), atunci și imaginile lor $L'=J(G)$ și $J(L)$ sunt izomorfe.

Pentru a introduce conceptul de MD-sketch normal introducem:

Definiție: Un functor $F\in$ Fun($C,FinSet$) se spune că este *reprezentabil* dacă:

- $\exists$ $A\in|C|$ un obiect
- $\exists$ φ: $H^A \to F$ izomorfism natural, unde H^A este *functorul de bază*

$H^A=H(A,\_):C \to \mathit{FinSet}$ definit astfel:

* $H^A(X)=\mathrm{Hom}(A,X)$, $\forall$ $X\in|C|$.

* $H^A(\alpha):H^A(X) \to H^A(Y)$, $\forall$ $\alpha:X \to Y$, ($\alpha\in\mathrm{Hom}(X,Y)$), cu $H^A(\alpha)(\xi)=[H(A,\_)(\alpha)](\xi)=\alpha\xi, \forall\xi\in H^A(X)$.

Se spune că (A, φ) este o *reprezentare* a lui F (sau că reprezintă pe F). Pentru amănunte se poate consulta [2].

Definiție: Un *MD-sketch* (C,M,D) se numește *normal* (sau în formă normală) dacă toți *functorii reprezentabili* cu sursa C sunt modele (duc surse în monosurse și surse duble în surse duble disjuncte).

Lemă: Pentru orice MD-sketch (C,M,D) există un algoritm cu ajutorul căruia să obținem un MD-sketch normal (C',M',D') astfel încât categoriile Mod(C,M,D) și Mod(C',M',D') să fie echivalente.

Observație:

- Forma normală se obține prin modificarea MD-sketch-ului inițial astfel încât să fie îndeplinite condițiile de monicitate și de disjuncție.
- Construcția poate fi facută deoarece categoria C este finită.

Definiție: Fie $A\in|C|$ un obiect. Definim functorul

Hom(A,_): $|C| \to$ Fun($C,FinSet$) astfel:

- $\mathrm{Hom}(A,\underline{B}):=\mathrm{Hom}_C(A,B)\in\mathrm{Fun}(C,\mathit{FinSet})$
- $\mathrm{Hom}(A,\underline{f})$: $\mathrm{Hom}(A,\underline{C}) \to \mathrm{Hom}(A,\underline{D})$, unde $f\in\mathrm{Hom}_C(C,D)$. Avem pentru $\forall g\in\mathrm{Hom}_C(A,C)$ $\mathrm{Hom}(A,\underline{f})(g):=f \circ g$,

Propoziție: Dacă (C,M,D) este un MD-sketch normal și $\mu = (f_i: X \to Y_i)$ o sursă finită din C, atunci este decidabil dacă μ este dusă într-o monosursă în fiecare model al lui (C,M,D).

(dem):

Fie următoarea diagramă:

$$\mathrm{Mod}(C,M) \xleftarrow{J'} \mathrm{Mod}(C,M\cup\{\mu\}) \xleftarrow{J''} \mathrm{Mod}(C,M,D)$$
$$\xleftarrow{\quad\quad J \quad\quad}$$

Dacă μ este dusă într-o monosursă de fiecare functor din Mod(C,M,D) atunci avem:

- Dintr-o lemă anterioară,

  J: Mod(C,M,D) $\to$Mod(C,M) conservă colimitele, unde $J = J' \circ J''$.

- J' şi J'' reflectă colimitele pentru că sunt incluziuni pline (adică $\forall$A'',B''$\in$ | Mod(C,M,D) |,$\forall$A',B'$\in$ | Mod($C,M\cup\{\mu\}$)|, avem că $J'_{A',B'}$ şi $J'_{A'',B''}$ sunt injecţii surjective).

Se consideră diagrama din Mod($C,M\cup\{\mu\}$):

Hom(f_i,_): Hom(Y_i,_) $\rightarrow$ Hom(X,_),

Hom(fi,_) este morfism în duala categoriei functorilor Hom(A,_), unde A$\in|C|$. Diagrama are drept colimită pe Hom(X,_). Deoarece J' reflectă colimitele, lui Hom(X,_) îi corespunde o colimită în Mod(C,M,D) (care este tot Hom(X,_) - pentru că J' este incluziune). Din cauză că J=J'oJ'' este injecţie şi conservă colimitele, atunci colimita în Mod(C,M) a diagramei date este tot Hom(X,_).

Rezultă că o colimită din Mod($C,M\cup\{\mu\}$) este colimită şi în Mod(C,M).

Reciproc, dacă avem colimita din Mod(C,M) este Hom(X,_), se demonstrează că μ poate fi dusă într-o monosursă de fiecare functor din Mod(C,M,D).

S-a demonstrat astfel că problema deciziei dacă μ este dusă într-o monosursă în fiecare model al (C,M,D) echivalează cu colimita în Mod(C,M) a diagramei (prezentate in Mod($C,M\cup\{\mu\}$)) este izomorfă cu Hom(C,_).

Concluzie (algoritm):

Pentru decizie e suficient să se calculeze colimita diagramei în Mod(C,M) şi să se verifice dacă este izomorfa cu Hom(X,_).

Observaţie: Se pot calcula colimitele diagramelor finite din Mod(C,M) astfel: se calculează colimita în Fun(C,*FinSet*) şi apoi se construieşte conform algoritmului prezentat reflexia acestei colimite de-a lungul incluziunii I.

Propoziţie:

Dacă (C,M,D) este un MD-sketch normal şi δ=(f_i,g_i) este o sursă dublă finită din C, atunci este decidabil dacă δ este dusă într-o sursă dublă de disjuncţie în fiecare model al lui (C,M,D).

(dem):

Fie diagrama următoare din Mod(C,M):

Hom(f_i,_): Hom(Y_i,_) $\rightarrow$ Hom(X,_), şi

Hom(g_i,_): Hom(Y_i,_) $\rightarrow$ Hom(Z,_).

S-a arătat în lema anterioară că se pot calcula colimite de diagrame finite în Mod(C,M) (cu ajutorul reflecţiei de-a lungul incluziunii în Fun(C,*FinSet*)), deci se poate calcula colimita (λ_i,F) a diagramei în Mod(C,M). Există două posibilităţi:

a) F satisface toate sursele duble (condiţiile de disjuncţie) din D. Aceasta înseamnă că F este un model al lui (C,M,D) care nu satisface pe δ, deci δ nu este satisfacută de toate modelele.

b) F nu satisface toate condiţiile de disjuncţie. Înseamnă că F nu este un model. Se va demonstra că orice functor F' care nu satisface pe δ nu este un model. Va rezulta că orice functor care este model că satisface pe δ, deci δ este satifacută de modelele. Fie F' un functor oarecare care nu satisface pe δ. Se construieşte un co-con (λ_i',F') prin adaugare de săgeţi în diagrama menţionată anterior. Din proprietatea de universalitate a colimitei F, rezultă că $\exists$ α: $F \rightarrow F'$ morfism în Mod(C,M). Presupunând prin absurd că F' este model, atunci F' este în imaginea incluziunii J de la Mod(C,M,D) la Mod(C,M). Conform unei leme anterioare, rezultă că şi F este în imaginea incluziunii J, deci F este un model. Contradicţie.

Concluzie (algoritm):

Propoziţia ne asigură de faptul că putem decide dacă δ este satisfăcută de toate modelele, calculând colimita în Mod(C,M) (s-a arătat că este calculabilă) a diagramei de mai sus şi verificând apoi dacă toate condiţiile disjuncte din D sunt satisfăcute în vârful F al acestei colimite.

Definiţii:

- O categorie se numeşte *schelet* iff nu există nici o pereche de obiecte izomorfe.
- O categorie se numeşte *Cauchy-completă* iff fiecare morfism idempotent este *secţionabil*.

Definiţie: Un morfism $\alpha\in Hom_C$ (A,B) este *secţionabil* (inversabil al dreapta) dacă $\exists$ α''$\in$ Hom_C (B,A) cu α'' o $\alpha=1_B$.

Lemă: Pentru orice MD-sketch normal (C,M,D) se poate calcula (exista un algoritm) un MD-sketch normal echivalent (C',M',D') pentru care C' este *schelet* şi *Cauchy-completă*.

Lemă: Fiecare model M: C $\rightarrow$ FinSet al unui MD-sketch normal (C,M,D) este o colimită a *modelelor reprezentabile*.

(dem)

Am amintit că fiecare obiect din Fun(C,*FinSet*) este o colimită de functori reprezentabili. Cum aceşti functori sunt modele (deoarece un MD-sketch normal presupune că toţi functorii reprezentabili sunt modele), rezultă că acesta (obiectul) este o colimită de modele. Deoarece $I \circ J$: Mod(C,M,D) $\rightarrow$ Fun($C,FinSet$) dintr-o diagramă anterioară este fidel şi plin, deci reflectă colimitele, atunci există un model colimită în Mod(C,M,D) a cărei imagine prin IoJ să fie obiectul amintit. Deoarece IoJ este incluziune, rezultă că obiectul este un model, deci modelul este o colimită de modele reprezentabile (functori reprezentabili care sunt modele). Parcurgând obiectele lui Mod(C,M,D) şi privindu-le ca obiecte în Fun($C,FinSet$), cu ajutorul raţionamentului anterior, se deduce concluzia lemei.

2.4. TEOREMA: *Echivalenţa MD-sketch-urilor este decidabilă.*

(dem):

Fie (C,M,D) şi (C',M',D') două MD-sketch-uri. Se poate cosidera că aceste MD-sketch-uri sunt normale cu C şi C' *schelet* şi *Cauchy-complete*, datorită lemelor anterioare. Se spune că *două MD-sketch-uri sunt echivalente* dacă există un izomorfism i: $C \rightarrow C'$ astfel încât:

a) pentru fiecare μ din M şi pentru fiecare δ din D avem că i(μ) este monosursă în fiecare model al lui (C',M',D') şi i(δ) este sursă dublă de disjuncţie în fiecare model al lui (C',M',D'). Se poate verifica acest lucru deoarece i(μ) şi i(δ) sunt surse şi respectiv surse duble în C', iar problema referitoare la satisfacerea acestora de toate modelele lui (C',M',D') este decidabilă.

b) pentru fiecare μ' din M' şi pentru fiecare δ' din D' avem că $i^{-1}(\mu')$ este monosursă în fiecare model al lui (C,M,D) şi i(δ') este sursă dublă de disjuncţie în fiecare model al lui (C,M,D). Problema este echivalentă cu cea de la punctul a).

Observaţie:

Datorită faptului că C şi C' sunt finite, rezultă că se pot enumera toate izomorfismele dintre ele şi folosind cele două propoziţii de mai sus putem decide echivalenţa.

Rămâne de verificat că echivalenţa MD-sketch-urilor, aşa cum a fost dată aici, este compatibilă (necesară şi suficientă) cu echivalenţa categoriilor de modele asociate.

O sarcină interesantă ar fi rescrierea acestei teoreme în limbajul instituţiilor din logica ecuaţională.

3. Specificaţii de date

Definiţie:

O specificaţie de date (SD) este un triplet (S,M,A) unde:

1) S este categorie finită
2) M este mulţime finită de surse din S
3) A: $S' \rightarrow$ FinSet este functor, unde S' este categoria discretă, obţinută din S (aceeaşi mulţime de obiecte, dar are doar morfisme identitate).

Observaţie: A este de fapt o funcţie din obiectele lui S în clasa mulţimilor finite.

Definiţie: Un model al unei SD (S,M,A) este perechea (M,λ) unde:

1) M: $S \rightarrow$ *FinSet* este functor cu M(μ) monosursă pentru orice sursa $\mu \in M$.
2) λ: M o $I \rightarrow A$ o *transformare naturală*

$$S' \xrightarrow{I} S \xrightarrow{M} FinSet, \qquad S' \xrightarrow{M \circ I} FinSet$$

$$\Big\downarrow \lambda$$

$$S' \xrightarrow{A} FinSet,$$

unde I este incluziune.

Un *exemplu* de SD ar putea fi:

– categoria S este perechea (G,E), unde G este un graf, iar E este o mulţime de ecuaţii;
– functorul A = 1 (functorul de *uitare* de la S' în *FinSet* - "uită" morfismele).

Cu acest tip de SD se pot specifica doar tipurile de entităţi (obiectele lui S) şi nişte constrângeri structurale, date de setul de ecuaţii.

Observaţii:

- aceste SD-uri sunt nişte MD-sketh-uri cu set vid de condiţii de disjuncţie (disjointness conditions);
- dacă S este categorie discretă (fără morfisme nonidentitate), modelele sunt doar mulţimi cu tip;
- săgeţile din categoria S specifică dependenţe existenţiale.

Exemplu:

$$G = \begin{array}{c} \text{COMPUTER} \\ \big\downarrow \\ \text{LOCATION} \end{array}, \quad E = \varnothing, \quad M = \varnothing$$

Această SD specifică, de exemplu, că fiecare calculator (entitate de tip COMPUTER) trebuie să aibă o locaţie asociată cu el.

Exemplu:

O sursă cu n săgeţi poate fi privită ca o multirelaţie de aritate n:

$$G = \begin{tikzcd} & \text{CONNECTION} \arrow[dl, "c"'] \arrow[dr, "p"] & \\ \text{COMPUTER} & & \text{PRINTER} \end{tikzcd},$$

$E = \varnothing$, $M = \varnothing$

Fiecare tip de entitate CONNECTION este asociată cu un cuplu de entități (x,y) în care x este de tip COMPUTER, iar y este de tip PRINTER. În concluzie, CONNECTION este o multirelație peste COMPUTER și PRINTER.

Observație: Pentru ca multirelația să fie normală (să nu existe duplicate), trebuie ca sursa corespunzatoare să fie monosursă. Rezultă:

M={(CONNECTION,(c,p))}. În felul acesta nu se permite definirea de două ori a aceleiași multirelații. Restul componentelor rămân la fel.

Dacă vrem să introducem niște constrângeri, acestea se pot face prin introducerea unor ecuații în S (în E). În exemplul următor, SD-ul va specifica faptul că imprimantele și calculatoarele pot fi conectate doar dacă se află în același loc (au aceeași locație).

$$G = \begin{tikzcd} & \text{CONNECTION} \arrow[dl, "c"'] \arrow[dr, "p"] & \\ \text{COMPUTER} \arrow[dr, "l1"'] & & \text{PRINTER}, \arrow[dl, "l2"] \\ & \text{LOCATION} & \end{tikzcd}$$

E = { l2 o p = l1 o c },

M = M={(CONNECTION,(c,p))}.

Dacă vrem să stocăm și atributele entităților, cu ajutorul functorului A putem să specificăm pentru fiecare tip de entitate (nod al lui S) setul valorilor atributelor pe care entitățile de acel tip le pot avea. În fiecare model M, toate entitățile de tip C (adică toate elementele din mulțimea M(C)) vor fi etichetate cu elemente din A(C) (etichetarea pe entități o face transformarea naturală λ).

Exemplu:

$$G = \begin{tikzcd} & \text{CONNECTION} \arrow[dl, "c"'] \arrow[dr, "p"] & \\ \text{COMPUTER} \arrow[dr, "l2"'] & & \text{PRINTER} \;\ldots.\; \{\text{matrix,laser}\} \arrow[dl, "l1"] \\ & \text{LOCATION} & \end{tikzcd}$$

E = {l2 o p = l1 o c},

M = {(CONNECTION, (c,p))}.

Linia punctată din figura de mai sus reprezintă functorul A pentru nodul din graf care este legat de o *mulțime finită*.

De exemplu: A(PRINTER) este mulțimea {matrix,laser}. Aceasta înseamnă că într-un model al acestei specificații, fiecare entitate de tip PRINTER va fi etichetată cu o valoare din mulțimea {matrix,laser} care se mai numește și mulțimea (valorilor) atributelor nodului PRINTER. Dacă un nod n al grafului nu are o legatură de acest tip la o mulțime, atunci se spune ca A(n) este o *mulțime terminală*.

Un ultim *exemplu*:

$$G = \begin{tikzcd} \text{M-CONNECTION} \arrow[d, "m"'] \arrow[dr, "c1"] & & \text{L-CONNECTION} \arrow[dl, "c2"'] \arrow[d, "l1"] \\ \text{MATR-PRINTER} \arrow[dr, "l2"'] & \text{COMP.} \arrow[d, "l3"] & \text{LAS-PRINTER} \arrow[dl, "l4"] \\ & \text{LOCATION} & \end{tikzcd}$$

E = {l2 o m = l3 o c1, l3 o c2 = l4 o l1}.

M = { (M-CONNECTION,(m,c1)),
(L-CONNECTION,(c2,l1)) }.

În această specificație, două tipuri de entități diferite sunt folosite pentru matrix-printers și laser-printers, în loc de a folosi un singur tip de entitate cu un atribut.

Ceea ce trebuie văzut este faptul că anumite categorii modele sunt într-adevăr echivalente pentru a evita că anumite specificații diferite să se suprapună fie și parțial.

Observații:

- Din aceste exemple reiese că, specificațiile de date, așa cum sunt ele definite aici, reprezintă un mod intuitiv de specificare a structurii unei baze de date.
- Modelul (M,λ) al ultimei SD este o posibilă instanță a bazei de date a cărei structură este dată de diagrama.
- Diagramele Entitate-Relație pot fi translatate în specificațiile de date definite aici, deci pot fi reformulate ca specificații de date categoriale.

Observații:

- Functorul M indică numărul de entități ale aceluiași tip și cum sunt asociate între ele.
- Transformarea naturală λ asociază câte o valoare de atribut pentru fiecare entitate.

Definiție: Un *homomorfism* între două modele (M, λ) și (M', λ') ale unei specificații de date (S,M,A) este o transformare naturală α: M → M' astfel încât λ' o αI = λ (o transformare naturală între M și M' care este compatibilă cu etichetarea).

Avem M, M': $S \to FinSet$, deci se mai poate scrie α: $|S| \to \mathrm{Hom}_{C^{S}}$ cu:

- $\alpha_X \in \mathrm{Hom}_{C^{S}}(M(X),M'(X)), \forall X \in |S|$.
- $\forall \varphi \in \mathrm{Hom}_{C}(X,Y)$ aveam diagrama comutativă: $\alpha_Y \circ M(\varphi) = M'(\varphi) \circ \alpha_X$.

$$\begin{tikzcd} M(X) \arrow[r, "\alpha_X"] \arrow[d, "M(\varphi)"'] & M'(X) \arrow[d, "M'(\varphi)"] \\ M(Y) \arrow[r, "\alpha_Y"] & M'(Y) \end{tikzcd}$$

Observație: Modelele și homomorfismele de modele ale unei specificații $F=(S,M,A)$ formează o categorie numită *categoria modelelor lui F*, notată Mod(F). Un model M (obiect din Mod(F)) duce orice sursa $\mu \in M$ într-o mono-sursă.

Definiție: Se spune că două specificații de date sunt echivalente dacă și numai dacă categoriile modelelor sunt echivalente.

Observație: Pentru orice specificație de date $F=(S,M,A)$ se poate defini un MD-sketch numit *translația* lui F.

Teorema: Categoria modelelor specificației $F=(S,M,A)$ este echivalentă cu categoria modelelor corespunzătoare translației lui F.

Având acest rezultat, s-a obținut un algoritm de decizie a echivalenței oricăror specificații de date.

4. Concluzii:

S-a prezentat în această lucrare problema decidabilității specificațiilor de date așa cum a fost ea abordată în [1].

Schema de decizie a echivalenței a două specificații de date peste o categorie C constă din translatarea acestora în MD-sketch-uri echivalente (la nivelul categoriilor de modele) și apoi prin aplicarea algoritmului de testare a echivalenței MD-sketch-urilor.

Echivalența MD-sketch-urilor se face prin parcurgerea izomorfismelor între categoriile asociate acestora și prin verificarea dacă imaginile tuturor surselor și surselor duble, dintr-un MD-sketch prin izomorfismul considerat la un moment dat, și inversul sunt satisfăcute în orice model al celuilalt MD-sketch.

Teoria prezentată poate să constituie un punct de plecare pentru:

- modelarea bazelor de date în contextul categorial prezentat din perspectiva logicii ecuaționale;
- extinderea teoriei la studiul bazelor de date în condiții de incertitudine prin considerarea în locul categoriei *FinSet* a categoriei mulțimilor fuzzy cu suport finit *FuzzFinSet* în sens Goguen [14]:
  - $|FuzzFinSet|$ este clasa tuturor mulțimilor fuzzy A $: X \to [0,1]$, unde X este o mulțime finită.
  - dacă A: $X \to [0,1]$ și B $: Y \to [0,1]$ atunci $\mathrm{Hom}_{FuzzFinSet}(A,B)$ este mulțimea aplicațiilor f: $X \to Y$ astfel încât $B \circ f \geq A$.

În încheiere, aduc mulțumiri domnului profesor Paul Flondor și domnului dr. mat. Mircea Sularia pentru încurajările și sprijinul acordat.

Bibliografie:

1. **PIESSENS, F., STEEGMANS, E.**: Categorial Data-Specifications,Theory and Applications of Categories, Vol. 1, No. 8, 1995, pp. 156-173.
2. **PURDEA, I.**: Tratat de Algebră Modernă, vol. II, Editura Academiei RSR, 1982.
3. **BARR, M., WELLS, C.**: Category Theory for Computing Science, Prentice Hall International Series in Computer Science, Englewood Cliffs, N.Y., 1990.
4. **BORCEAUX, F.**: Handbook of categorial algebra I, Cambridge University Press, 1993.
5. **ADAMEK, J., ROSICKY, J.**: Finitary sketches. În: Journal of Symbolic Logic".
6. **CADISH, B., DISKIN, Z.**: Algebraic graph-based approach to management of multibase systems, I: Schema integration via sketches and equations. În: Proc. of NGITS'95, June 1995.
7. **CHEN, P.P.**:The Entity-Relationship Model - Towards a Unified View of Data. În: ACM Transactions on Database Sysyems, Vol. 1, No. 1, 1976, pp. 9-36.
8. **DISKIN, Z., CADISH, B., BEYLIN, I.**: Algebraic graph-based approach to management of multibase systems, II: Algebraic aspects of schema integration. În: Proc. of ADBIS'95, Moscow, June 1995.
9. **MAKKAI, M.**: Generalized sketches as a framework for completeness theorems. În: Journal of Pure and Applied Algebra.
10. **TEOREY, T. J.**: Database Modeling and Design, Morgan Kaufmann Publishers, Inc., 1990.
11. **TUJIN, C.**: Data modeling from a categorical perspective", PhD thesis, Univ. of Antwerpen, Belgia, 1994.
12. **RUDEANU, S.**: Fundamentele Algebrice ale Informaticii, note de curs, Univ. București.
13. **CĂZĂNESCU, V. E.**: Teoria Categoriilor, note de curs, Univ. București.
14. **NEGOIȚĂ, C.V., RALESCU, D.A.**: Fuzzy Sets and Systems, Birkhauser Verlag, Boston, 1977.
15. **GOGUEN, J., BURSTALL, R.** Institutions: Abstract model theory for specification and programming. În: Journal of the ACM, Jan. 1992.