

About the Generalized Reasoning Methods and their Use in Semiotic Systems

Prof.dr. Mihaela MUNTEAN, Conf.dr. Cornelia MUNTEAN
West University of Timișoara

In computational semiotics the problem is to emulate a semiosis cycle within a digital computer. This needs the construction of intelligent systems, able to perform intelligent behavior, such as sensorial perception, world modeling, value judgement and behavior generation. These intelligent systems could be generally implemented through object networks and the basic functions mentioned above could be obtained by generalization of some elementary knowledge operators. Based on the three main reasoning methods, deduction, induction and abduction, well known in the philosophy of science and used in AI systems, there were three new knowledge operators defined: knowledge extraction, knowledge generation and knowledge generation, operators that could be viewed as generalized interpretations of the standard reasoning procedures. This paper presents these new concepts and their connection, the current understanding of generalized deduction, induction and abduction and also how these operators could serve as the building blocks of universal intelligent systems.

Keywords: *Semiotic System, Knowledge Units, Deduction, Induction, Abduction*

1 Semiotica și sistemele inteligente

1.1. Introducere

În studiul sintactic, semantic și pragmatic al simbolurilor ca parte a vieții sociale [1], știință numită semiotică, se încearcă o emulare a ciclului de semioză (transfer și transformare a simbolurilor) cu ajutorul calculatorului digital. Acest lucru necesită construirea de sisteme inteligente, capabile de comportamente inteligente, ca percepția senzorială, modelarea mediului, analiza valorilor și generarea comportamentului. Modelarea matematică a unui astfel de sistem semiotic a fost în ultimul timp subiectul mai multor cercetări cu privire la relația dintre semiotică și sistemele inteligente.

Mai mulți cercetători și-au propus să descopere unitățile elementare sau minime de inteligență, adică un set minim de operatori care să facă posibilă construirea unui comportament inteligent în cadrul unui sistem inteligent. Astfel merită menționați Albus J. și lucrarea sa [2] și Meystel A. cu algoritmul său GFACS, prezentat în [3]. Ulterior, Ricardo R. Gudwin (în [4] și [5]) a propus un set alternativ de operatori, numiți extragerea cunoașterii, generarea cunoașterii și selecția cunoașterii, care ar putea fi percepuți ca o

generalizare a strategiilor de raționament deductiv, inductiv și respectiv a abducției.

În continuare lucrarea de față va prezenta aceste concepte noi, modul de înțelegere la ora actuală a deducției, inducției și abducției generalizate și felul în care operatorii propuși de Gudwin pot fi utilizați ca blocuri elementare pentru construirea sistemelor inteligente universale.

1.2. Unități de cunoaștere

Piese elementare de cunoaștere, numite și unități de cunoaștere, reprezintă, am putea spune, „un fir” de informație codat într-o structură.

Pentru a prezenta mai clar fundalul filozofic ce stă la baza unei astfel de definiții pentru unitățile de cunoaștere, vom porni de la considerentul că există o *lume reală*, compusă dintr-un set de fenomene dinamice continue care decurg în paralel. Desigur noi nu putem cunoaște *tot* din această lume reală, ci doar ceea ce putem percepe prin simțurile noastre, adică așa-numitul *mediu sensibil* al nostru sau *Umwelt*, cum este numit acest mediu în literatura de specialitate [5]. Acesta nu reprezintă însă realitatea, ci doar cea mai bună înțelegere a noastră cu privire la realitate. Astfel, simțurile noastre sunt sursa unei informații continue dar parțiale despre fenomenele

ce au loc în lumea reală. Din această sursă de informații se extrag așa-numitele **singularități**, adică acele grupuri de informații care pot fi cuprinse într-un singur concept (figura 1).

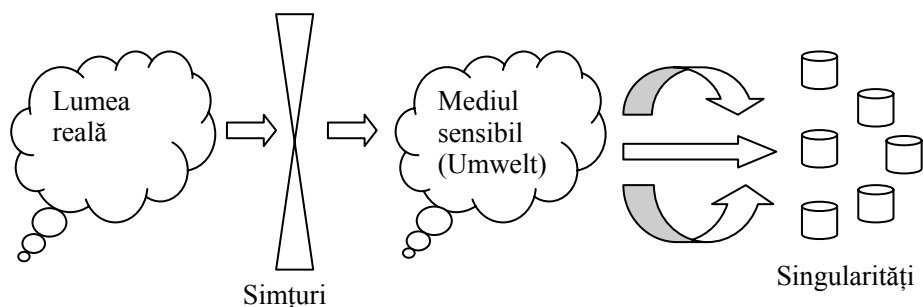


Fig.1. Extragerea singularităților

După ce aceste singularități sunt identificate în mediul sensibil, ele trebuie codificate pentru a deveni unități de cunoaștere, conform definiției enunțate mai sus. Această codificare necesită un **spațiu de reprezentare**

și o **structură de reprezentare** plasată în acel spațiu. Această structură poate fi abstractizată ca fiind o structură matematică de tipul numere (a), liste (b), arbori (c) și grafuri (d) – a se vedea figura 2.

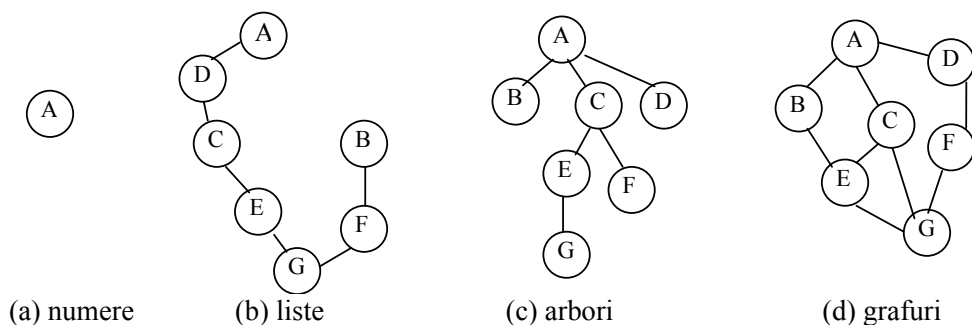


Fig.2. Structuri matematice

În continuare, ne putem închipui că spațiul de reprezentare conține un set de valori care ocupă fiecare un anumit loc în spațiul tridimensional. Pentru a forma unitățile de cunoaștere trebuie focalizată atenția asupra unui

anumit grup de valori, adică selectată o anumită regiune închisă a spațiului de reprezentare. În urma interpretării, singularitățile respective devin structuri, ca cele din figura 3.

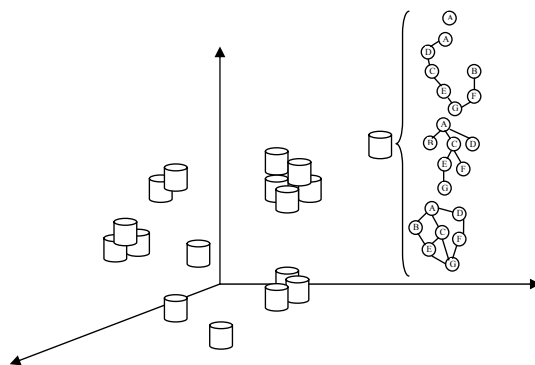


Fig.3. Spațiul de reprezentare conținând structuri matematice

Astfel, unitățile de cunoaștere se formează datorită acestor mecanisme de extragere a singularităților. Unități de cunoaștere mai complexe însă se formează aplicând operatorii de procesare a cunoștințelor despre care vom vorbi în continuare, și anume: extragerea, generarea și selecția cunoștințelor.

2. Operatorii elementari de cunoaștere și generalizarea lor

2.1. Generalizarea și specializarea

Unitățile de cunoaștere pot fi comparate unele cu altele prin intermediul unui operator de relație de abstractizare (\prec). Astfel, dacă două unități de cunoaștere a și b respectă relația $a \prec b$ spunem că b este o abstractizare sau o generalizare a lui a sau că a este o specializare a lui b . Aceste concepte sunt fundamentale pentru definițiile care vor urma în ceea ce privește extracția, generarea și selecția cunoașterii. Punctul cheie în înțelegerea relației de abstractizare sunt cele două modalități de definire a unui set de valori: Modalitatea *extensivă* cere listarea explicită a tuturor elementelor setului și poate fi aplicată doar pentru seturi finite de elemente, și modalitatea *intensivă*, care reprezintă un set infinit de valori utilizând un set finit de parametri. Această a doua modalitate implică o codificare capabilă să convertească din reprezentarea intensivă elemente în reprezentarea extensivă.

De exemplu, definim intensiv setul S ca fiind $S = \{(x, y) \in R^2 \mid y = 3x^2 + 4x - 2\}$. S va putea fi reprezentat ca fiind tuplul $(3, 4, -2)$, ceea ce codifică întreaga informație necesară pentru a reconstitui toate punctele (x, y) aparținătoare lui S . Să presupunem acum o unitate de cunoaștere $a = (1, 5)$ și o unitate de cunoaștere $b = (3, 4, -2)$. Dacă interpretăm a ca fiind o pereche în R^2 și b ca fiind parametri ce reprezintă setul infinit S , putem spune că $a \prec b$, deoarece unitatea de cunoaștere b cuprinde nu doar pe a , ci un întreg set de perechi care respectă aceeași relație. Observați că am putea avea de asemenea și o unitate de cunoaștere $c = (0, -2, 1, 5, 2, 18)$, care ar putea fi decodificată drept setul de perechi $T = \{(0, -2), (1, 5), (2, 18)\}$, și am avea de asemenea $a \prec b$ și $a \prec c \prec b$. Acest exemplu este doar un indiciu pentru a înțelege mai bine natura operatorii de abstractizare.

Modul în care s-a făcut codificarea setului S în tuplul $(3, 4, -2)$ poate fi privit ca un fel de operator de compresie a datelor. Fiecare unitate de cunoaștere b care poate fi extinsă la alte unități de cunoaștere a_i cu ajutorul unor interpretări particulare se spune că este o generalizare a acestora, iar a_i se spune că sunt specializări ale lui b .

2.2. Operatorii elementari de cunoaștere

Prezentăm în continuare setul minim de operatori propus de Gudwin [5] ca formând fundamentul conceptual pentru construirea sistemelor inteligente. Aceștia sunt operatorii sau, mai precis, clasele de operatori numite: **extracția cunoașterii, generarea cunoașterii și selecția cunoașterii**. Există o strânsă legătură între acești operatori de cunoaștere elementari și operatorii clasici de raționament, adică deducția, inducția și abducția.

Astfel, de exemplu, operatorul de extragere a cunoașterii poate fi asociat cu o deducție generalizată, în sensul că el cuprinde tot ce însemna operatorul tradițional de deducție, dar în plus permite orice tip de unitate de cunoaștere ca și intrare.

La fel stau lucrurile cu inducția generalizată, care include și unii operatori ce nu sunt acceptați în mod normal ca efectuând „inducție”. În acest sens operatorul de generare a cunoașterii generalizează ideea de inducție pentru a include orice tip de procedură care generează unități de cunoaștere care sunt mai abstracte decât intrările.

În fine, selecția cunoașterii poate fi comparată cu operatorul standard de abducție. Abducția este adeseori privită ca fiind inferența ce duce la explicația „cea mai bună”. Acea parte ce are de-a face cu generarea unui set de ipoteze este în mod clar o inducție (sau, mai bine spus, o inducție generalizată). Munca abductivă constă de fapt în evaluarea și selecția celei mai bune ipoteze.

➤ Extragerea cunoașterii și deducția

Presupunem o unitate de cunoaștere a și o unitate de cunoaștere b , astfel încât $a \prec b$. Atunci o funcție *fec* care reprezintă imaginea lui b (ținând cont că b este o structură, cum ar fi de exemplu un număr structurat) în a : $a = fec(b)$ se numește operator de extragere a cunoașterii.

În figura 4 am reprezentat un exemplu de extragere de cunoaștere. Dintr-un set P de unități de cunoaștere, numit „premisă”, operatorul extrage un set C de unități de cunoaștere numite „concluzie”. Numim această operație extragere de cunoaștere deoarece definiția extensivă a unităților de cunoaștere în C este un subset al definiției extensive de unități de cunoaștere în P. Astfel se „extrage” din P doar o parte din conținutul său semantic.

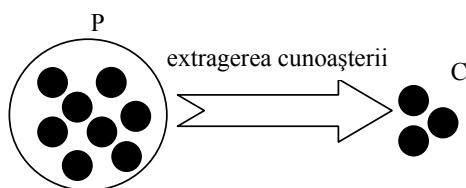


Fig. 4. Extragerea cunoașterii

Cea mai obișnuită formă deductivă este aplicarea lui *modus ponens*, ca în exemplul clasic de mai jos:

Dacă toți oamenii sunt muritori și Socrate e om rezultă că Socrate e muritor, ceea ce s-ar putea scrie simbolic:

$$om(x) \rightarrow muritor(x)$$

$$om(Socrate)$$

$$muritor(Socrate)$$

În acest caz cunoașterea că Socrate e muritor este conținută de asemenea și în premisa că toți oamenii sunt muritori și cea că Socrate este om. Astfel, $muritor(Socrate) \prec \{om(x) \rightarrow muritor(x) \wedge om(Socrate)\}$

Propozițiile sunt unități de cunoaștere statice, numite dicente, în sensul că ele există doar ca și date (- spre deosebire de tipurile de cunoaștere dinamice, cum sunt argumentele, și care nu există doar ca date, ci realizează de asemenea transformări în sistem). De obicei deducția se aplică doar la cunoașterea statică. Observăm că definiția operatorului de extragere de cunoaștere permite și includerea altor tipuri de cunoaștere, de vreme ce generalizează orice operație care explicitează cunoașterea ce există deja în premise. În acest sens putem numi operația de extragere de cunoaștere drept deducție generalizată.

Să observăm și cum calculul unei funcții este similar cu *modus ponens*: Cunoscând $f(x)$ și de asemenea x_0 , deducem $y_0=f(x_0)$. Astfel de-

ducția generalizată face exact ceea ce-i trădează numele, și anume generalizează ideea de deducție la alte tipuri de unități de cunoaștere.

➤ Generarea cunoașterii și inducția

Presupunem din nou o unitate de cunoaștere a și o unitate de cunoaștere b , astfel încât $a \prec b$ și o funcție fgc care reprezintă imaginea lui a pe b , deci: $b=fgc(a)$. În acest caz fgc este numit operator de generare a cunoașterii. De obicei acest operator nu este de tipul intrare unică/ieșire unică, ci conține un set de unități de cunoaștere de intrare și un set corespunzător de unități de cunoaștere de ieșire. Astfel, avem de exemplu situația: $(b_1, \dots, b_m) = fgc(a_1, a_2, \dots, a_n)$.

În figura 5 premisa P este colecția de unități de cunoaștere a_i și concluzia C este colecția de unități de cunoaștere b_i . Caracteristica de bază a acestui operator este faptul că definiția extensivă a unităților de cunoaștere din C conține neapărat elemente care inițial nu au fost conținute în definiția extensivă a unităților de cunoaștere din P, ci au fost adăugate pe parcursul procesului de generare a cunoașterii. Acest lucru se poate realiza fie prin combinarea unităților de cunoaștere, fie prin fuzionarea acestora sau transformarea, interpolarea, expansiunea topologică etc. (exemple: interpolarea funcțiilor, expansiunea topologică de la un număr la un set fuzzy etc.).

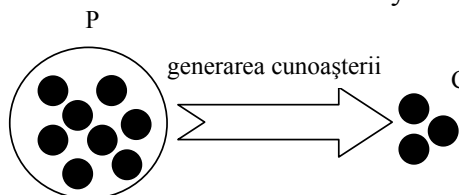


Fig. 5. Generarea cunoașterii

Se știe că inducția este procesul care generează o lege generală pe baza unor observații particulare, astfel că acestea devin instanțe particulare a respectivei teorii generale. Este deci vorba de un proces dinspre particular către universal, dinspre concret către abstract. Acest proces se numește generalizare. Procesul principal de inducție pornește cu un set de exemple (de propoziții, concepte etc.), care conduce la conceptul general care trebuie generat prin mai multe tehnici posibile. Acest lucru este similar cu procedura numită mai

sus generarea cunoașterii, cu o singură diferență: În generarea cunoașterii nu trebuie ca unitățile de cunoaștere generate să fie neapărat o generalizare a exemplurilor. Exemplele sunt utilizate doar ca un indicator către orice procedură care le folosește pentru a genera noi unități de cunoaștere. Astfel, de exemplu, procedurile de încrucișări și mutații genetice dintr-un algoritm genetic pot fi incluse în generarea cunoașterii, cu toate că în general acele procese nu se consideră a fi inducție. În acest caz generarea cunoașterii nu va putea fi numită inducție propriu-zisă, ci o formă de inducție generalizată.

➤ **Selecția cunoașterii și abducția**

Presupunem acum că avem un set de unități de cunoaștere (a_1, a_2, \dots, a_n) și un set de candidați pentru ieșire (b_1, b_2, \dots, b_m) . Presupunem de asemenea o funcție fsc care face o selecție între candidați: $b = fsc(a_1, a_2, \dots, a_n, c_1, c_2, \dots, c_m)$, în sensul că b este una dintre valorile c_i , iar valorile a_i sunt utilizate pentru a evalua și a putea face o alegere între c_i -uri. În acest caz fsc se numește operator de selecție a cunoașterii.

În figura 6 unitățile de cunoaștere a_i aparțin unui set P (premise), iar candidații c_i aparțin unui set H (numit set de ipoteze). Unitățile de cunoaștere b_i (mai multe în acest caz) sunt selectate dintre c_i -uri și sunt indicate prin C (concluzie). Să observăm că există acel caz particular când în H există un singur candidat c_i . În acest caz selecția devine validare sau, în caz că unitățile de cunoaștere din P nu vor putea fi utilizate pentru a valida noua unitate de cunoaștere pentru a fi ieșirea C, nu se va produce nici o ieșire.

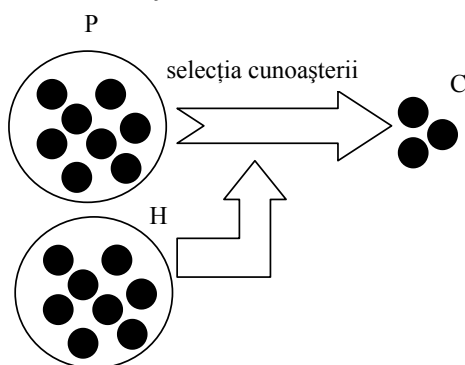


Fig. 6. Selecția cunoașterii

Abducția, introdusă în logica modernă de către filozoful Charles Peirce [7,8], este metoda de raționament cea mai puțin studiată până în prezent și adesea nu prea bine înțeleasă. Practic, diferiți cercetători au subliniat diferite aspecte ale abducției. Astfel, unii autori au înțeles abducția ca pe inversul deducției, alții s-au concentrat asupra întregului proces de explicare a raționamentului, ceea ce include detecția anomaliilor, verificarea ipotezelor și selecția celei mai bune ipoteze (a se vedea lucrarea [6]).

Un exemplu de raționament ce utilizează abducția ar fi următorul:

Deoarece calculatorul nu pornește și știind că dacă nu este curent la rețea calculatorul nu pornește, rezultă, conform abducției, că nu este curent la rețea. Desigur motivul pentru care calculatorul nu pornește ar putea fi și altul, și de aceea această metodă nu este o tehnică inferențială validă.

Abstractizând ceea ce se întâmplă de fapt în cazul abducției s-a ajuns la concluzia că abducția generalizată trebuie înțeleasă ca un pas de selecție/validare. Astfel Gudwin [4,5] a propus înțelegerea abducției nu ca un proces de detecție a anomaliilor (proces care ar consta practic mai mult din raționamente deducitive), nici ca un proces care generează ipoteze (ceea ce ar fi mai mult niște raționamente inductive), ci mai degrabă ca testarea și selecția acelor ipoteze.

3. Construirea sistemelor inteligente

Utilizând rețele obiectuale este posibil a se crea sisteme în care unitățile de cunoaștere sunt explicitate cu ajutorul obiectelor și în mod deosebit unitățile de cunoaștere argumentative sunt obiecte active.

Astfel, într-o rețea obiectuală unitățile de cunoaștere sunt reprezentate prin intermediul unor obiecte, adică niște unități elementare ce sunt în același timp și date și cod, plasate în anumite locații în rețea. Figura 7 prezintă o mică secțiune dintr-o rețea obiectuală.

Obiectele din locația activă din figura 7 sunt obiecte capabile să execute calcule. Ele fac o selecție între obiectele din locațiile de intrare, utilizează informațiile din acestea pentru a construi alte obiecte și a modifica propria lor

structură și eliberează noile obiecte create (sau transportă un obiect de intrare) în locațiile de ieșire. Obiectele utilizate pot fi lăsate din nou în locațiile de intrare, pot fi distruse sau pot fi transportate în locațiile de ieșire.

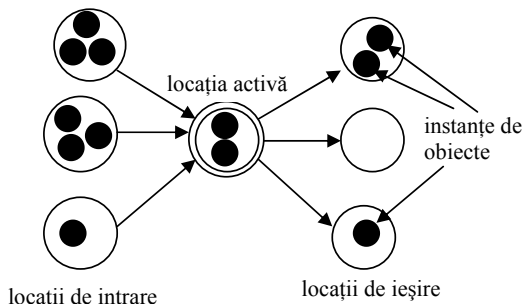


Fig. 7. Secțiune dintr-o rețea obiectuală

Două funcții principale sunt implicate în acest proces: Prima este selecția funcțiilor, care stabilește obiectele care să fie utilizate ca sursă de informație, iar a doua este funcția internă, care ia această informație împreună cu informația din interiorul structurii propriului obiect activ și o utilizează pentru a genera un nou obiect. Din punct de vedere conceptual atât funcția de selecție cât și funcția internă pot fi orice fel de funcție. Acest lucru face ca obiectul activ să fie o reprezentare perfectă a cunoașterii argumentative, capabile să efectueze fiecare sau toate din cele trei strategii de raționament generalizate: deducția, inducția și abducția. Funcția de selecție va efectua rolul abducției și funcția internă va prelua rolul deducției și inducției. Acest fapt face ca rețelele obiectuale să fie un instrument foarte bun de reprezentare și modelare a sistemelor inteligente.

O proprietate interesantă a unei rețele obiectuale este aceea că un obiect activ poate genera un alt obiect activ în același loc sau într-o altă locație în rețea. În acest caz partea activă a rețelei este posibil să fie modificată, mărită sau micșorată dinamic, datorită comportamentului rețelei. Această proprietate permite modelarea de sisteme care sunt capabile să-și schimbe propria structură, ca sisteme de învățare și sisteme adaptive. Exemple de rețele obiectuale care să implementeze sisteme fuzzy, rețele neuronale și algoritmi genetici sunt prezentate în lucrarea [4].

4. Concluzie

Această lucrare a prezentat unele concepte introduse relativ recent (în ultimii 10 ani) în filozofia cunoașterii și în domeniul mai nou al semioticii numerice. S-a insistat în mod deosebit asupra noilor concepte de deducție, inducție și abducție generalizate, acestea fiind propuse de unii cercetători drept componente elementare pe care să se bazeze construcția sistemelor inteligente universale și în mod special sistemele semiotice.

Bibliografie

1. Chandler, Daniel, *Semiotics for Beginners*, <http://www.aber.ac.uk/media/Documents/S4B/sem02.html>
2. Albus, J., *Outline for a Theory of Intelligence*, IEEE Transactions on SMC, vol.21, n.3, Mai-June 1991
3. Meystel, A., *Semiotic Modeling and Situation Analysis: An Introduction*, AdRem Inc., 1995
4. Gudwin R.R., Gomide, F.A.C., *Computational Semiotics: An Approach for the Study of Intelligent Systems – Part II Theory and Applications*, Technical Report RT-DCA09 – DCA-FEEC-UNICAMP, 1997
5. Gudwin, R.R., *On the Generalized Deduction, Induction and Abduction as the Elementary Reasoning Operators within Computational Semiotics*, Technical Report RT-DCA09 – DCA-FEEC-UNICAMP, 1997
6. Muntean M., Muntean C., *Reasoning Methods Used in AI: Some Philosophical, Theoretical and Practical Aspects*, Proceedings of the International Workshop IE&SI, 26-27 mai 2006, Timisoara, Editura Mirton, 2006
7. Yu, C.H., *Abduction, Deduction, and Induction: Their implications to quantitative methods*, 2005, Eprint: <http://seamonkey.ed.asu.edu/~alex/teaching/WBI/>
8. Yu, C.H., *Is there a Logic of Exploratory Data Analysis?*, Annual Meeting of American Educational Research Association, New Orleans, LA, April, 1994, Eprint <http://seamonkey.ed.asu.edu/~alex/pub/Peirce>