

## Crearea primului nivel de procesoare adaptabile utilizând formalismul E-T-M de interacțiune translatorică a procesoarelor adaptabile

Dumitru TODOROI (AESM, Republic of Moldova), todoroi@ase.md  
 Manfred KUDLEK (Hamburg University, Germany),  
 Zinaida TODOROI (AESM, Republic of Moldova),  
 Toader JUCAN ("Al.I.Cuza" University, Iași, Romania),  
 Diana MICUȘA (Moldova State University, Republic of Moldova)

*The present paper comprises the development of the results: (a) Development of fundamental [Tod-92, Tod-98, Tod-01] methods, models, procedures and algorithms concerning creation and evaluation of Adaptable Informational Technologies Basis; (b) Investment and construction, analysis and development of Adaptable languages and processors' System Basis and its sub-systems [Tod-92]; (c) Development of the E-T-M formalisms for adaptable processors' interactions [Tod-98, Tod-99]; (d) Analysis and discussion of its possibility to be implemented in construction and algorithmic proving of automatically creation of Adaptable Processors of the first, second, and third levels of translation complexity will be provided [Tod-01]; (e) Algorithmic demonstration of lemmas of automatically creation of the first level's Adaptable Processors; and (f) Discussion about the first level of Adaptable Processors possibility to be implemented in the base software and hardware of information technologies development industrial branch. Present Project evaluates from the Fulbright 1997/1998 research project No. 22131 "Societal Information Systems' Adaptable Tools", performed in the University of Omaha, Nebraska, USA.*

**Keywords:** formalism, compiler, interpreter, automation.

### Introducere

Lucrarea actuală este consacrată cercetărilor procesoarelor adaptabile de nivelul unu de complexitate translatorică ale cubului adaptabilității. Procesoare adaptabile de nivelul unu prezintă setul de procesoare de tip Model-Realizare, Timp-Realizare și Tip-Translatare adaptabilă a extinderilor. Procesoarele adaptabile de tipul Timp-Realizare și Tip-Translatare adaptabilă a extinderilor sunt definite prin intermediul formalismului Earley-Todoroi-Micușa (E-T-M)[Mic-02] de interacțiune translatorică a procesoarelor adaptabile.

Procesoarele de tip Model-Realizare sunt prezentate de procesoarele adaptabile de tip Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct. Procesoarele adaptabile de tip Nivel-Nivel realizează extinderile prin metoda «de coborîre» de pe un nivel al limbajului adaptabil de programare (LAP) pe altul pînă la atingerea nivelului bazei limbajului adaptabil de programare (BLAP). Procesoarele adaptabile de tip Nivel-Direct «coboară» elemente-

le-extinderi din programul scris în LAP transformindu-l direct în programul scris în limbajul BLAP. Procesoarele adaptabile de tip Nivel-Nivel-Direct conțin module translatoice de procesare de tip Nivel-Nivel și Nivel-Direct. Prin intermediul formalismului E-T-M de interacțiune translatorică a procesoarelor adaptabile sunt demonstrate afirmațiile de obținere automatizată a procesoarelor adaptabile de tip Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct.

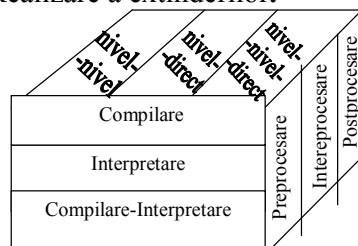
Procesoarele adaptabile din cubul adaptabilității (Fig.1) sunt împărțite în trei niveluri de complexitate translatorică. Procesoare adaptabile de nivelul unu prezintă setul de procesoare de tip Timp-Realizare, Model-Realizare și Tip-Translatare adaptabilă a extinderilor.

Nivelul unu de procesoare adaptabile de tip Timp-Realizare al extinderilor îl reprezintă procesoarele adaptabile: Pre-procesoarele, Inter-procesoarele și Post-procesoarele. Aceleași nivel unu aparțin și procesoarele adaptabile de tip Model-Realizare ale extinderilor:

realizarea extinderilor după metodele Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct. Al treilea set de procesoare adaptabile de nivelul unu îl constituie procesoarele adaptabile de tip Tip-Traducere adaptabilă: Compilare adaptabilă, Interpretare adaptabilă și Compilare-Interpretare adaptabilă a extinderilor.

Procesoarele adaptabile de nivelul unu de tip Nivel-Nivel realizează extinderile prin metoda «de coborîre» de pe un nivel al limbajului adaptabil de programare (LAP) pe altul pînă la atingerea nivelului bazei limbajului adaptabil de programare (BLAP). Procesoarele adaptabile de tip Nivel-Direct «coboară» elementele-extinderi din programul scris în LAP transformîndu-l direct în programul scris în limbajul BLAP. Procesoarele adaptabile de tip Nivel-Nivel-Direct sunt create în baza modulelor translatorice de procesare de tip Nivel-Nivel și Nivel-Direct.

Procesoare adaptabile de nivelul doi [ZTod-03] prezintă combinații de procesoare adaptabile de nivelul unu de tip Timp-Realizare, Model-Realizare și Tip-Traducere adaptabilă a extinderilor. Procesoarele de nivelul doi sunt prezentate prin intermediul procesoarelor adaptabile de tip Timp-Model-Realizare, de tip Tip-Model-Realizare și de tip Tip-Timp-Realizare a extinderilor.



**Fig.1.** Cubul adaptabilității

Teoremele de obținere automatizată a procesoare adaptabile de nivelul doi de complexitate translatorică de tip Timp-Model-Realizare a extinderilor sunt demonstrate în lucrarea [ZTod-03] prin intermediul formalismului E-T-M de interacțiuni translatorice ale procesoarelor adaptabile.

Procesoarelor adaptabile de nivelul trei de complexitate translatorică din cubul adaptabilității sunt reprezentate prin intermediul procesoarelor de tip Procesoare-Adaptabile-Compilatoare, Procesoare-Adaptabile-Interpretoare și Procesoare-Adaptabile-Compila-

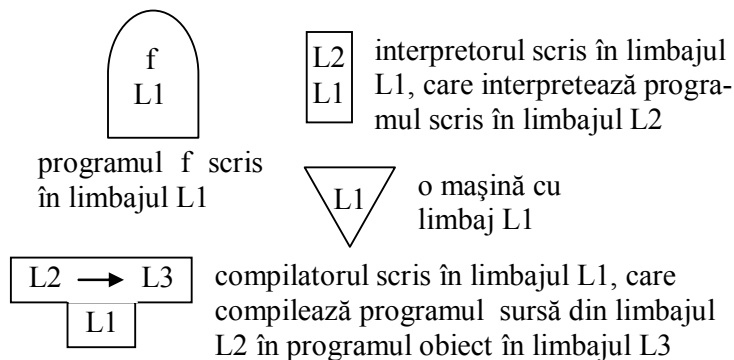
toare-Interpretare. Procesoarele-Adaptabile-Compilatoare sunt prezentate în lucrarea [Mic-03] prin intermediul Pre-procesoarelor-Compilatoare adaptabile, Inter-procesoarelor-Compilatoare adaptabile și Post-procesoarelor-Compilatoare adaptabile. Teoremele de obținere automatizată a acestor procesoare adaptabile de nivelul trei de complexitate translatorică sunt demonstrate în lucrarea [Mic-03] prin intermediul formalismului E-T-M de interacțiuni translatorice ale procesoarelor adaptabile.

**1. Formalismul Earley-Todoroi-Micușă (E-T-M) de interacțiuni translatorice a procesoarelor adaptabile.**

Cercetînd cubul adaptabilității (Fig.1), se poate de evidențiat 9 tipuri de procesoare adaptabile de nivelul unu de complexitate translatorică, 27 tipuri de procesoare adaptabile de nivelul doi și 27 tipuri de procesoare adaptabile de nivelul trei [Tod-92]. Studiului acestor tipuri de procesoare sunt consacrate un set de lucrări [Tod-98-Tod-01]. Lucrarea actuală este efectuată în deplină concordanță cu lucrările [ZTod-03, Mic-03]. Ea se referă la studierea posibilităților creării automatizate a procesoarelor adaptabile de nivelul unu de complexitate translatorică prin intermediul formalismului E-T-M [Mic-02] de interacțiuni translatorice a procesoarelor adaptabile.

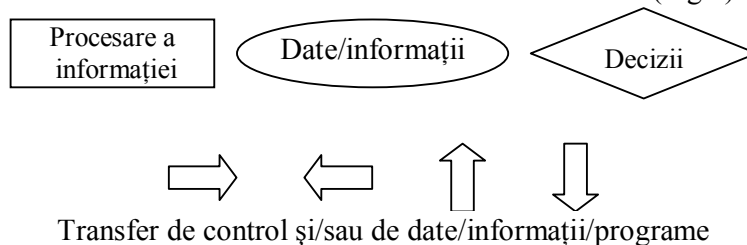
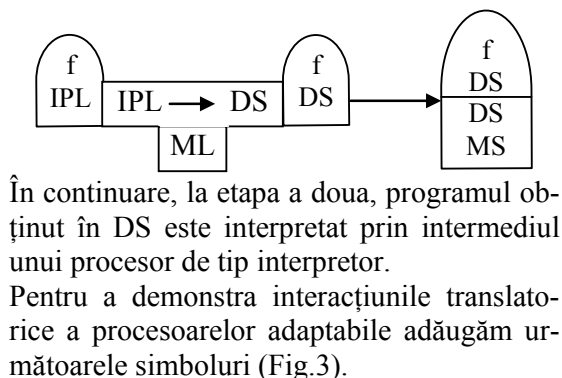
Sunt identificate 27 de scheme de procesoare adaptabile. Tipurile de traducere sunt prezentate prin metodele de compilare, interpretare și mixte: compilare-interpretare. După modul de realizare a extinderilor, procesoarele sunt grupate în procesoare de tip nivel-nivel, nivel-direct și mixt: nivel-nivel-direct. După ordinea (timpul, locul) de realizare corelatorică extinderi-elemente de bază, procesoarele sunt de tip preprocesor, interprocesor și postprocesor. Ca rezultat al acestor cercetări de mai sus s-au obținut procesoarele adaptabile acumulate și demonstrate de cubul adaptabilității (Fig.1).

Automatizarea proceselor de implementare a acestor procesoare adaptabile va fi demonstrată utilizînd tehnica de compilare-compilare. Pentru aceste interacțiuni va fi folosit T-formalismul sau formalismul Earley, simbolurile de bază ale căruia sunt următoarele:



**Fig.2. T-Simbolurile de interacțiuni translatorice (Earley).**

Aceste scheme de bază ale formalismului Earley [Ear-70] pot fi folosite la descrierea multor sisteme de programare. Unul din exemplele de utilizare a acestui formalism este sistemul IPL. Sistemul IPL este un procesor compus dintr-un compilator și un interpretor. Programul sursă scris în IPL, la prima etapă, este convertit într-o structură de date DS prin intermediul unui procesor de tip compilator.



**Fig.3. Simboluri de interacțiuni translatorice a procesoarelor adaptabile**

Prin intermediul simbolurilor, prezentate în fig.2 și fig.3, care împreună alcătuiesc formalismul E-T-M [Mic-02] de interacțiuni translatorice ale procesoarelor adaptabile vor fi demonstrate afirmațiile de creare automatizată a procesoarelor adaptabile de nivelul unu de complexitate translatorică a procesoarelor adaptabile din cubul adaptabilității. Formalismul E-T-M este utilizat și la demonstrarea posibilităților de creare automatizată de procesoare adaptabile de nivelul doi [ZTod-03] de complexitate translatorică ale cubului adaptabilității (Fig.1) și de procesoare adaptabile de nivelul trei [Mic-03].

## 2. Procesoarele adaptabile. Primul nivel

Utilizăm formalismul E-T-M pentru a demonstra interacțiunile translatorice cu proce-

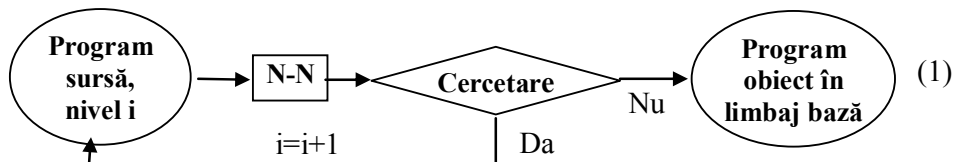
soarele simple de nivelul unu: procesare de tip Nivel-Nivel, procesare Nivel-Direct, procesare Nivel-Nivel-Direct, pre-procesare, inter-procesare, post-procesare, compilare adaptabilă, interpretare adaptabilă și compilare-interpretare adaptabilă a extinderilor. Aceste procesoare formează baza atomară a cubului adaptabilității (Fig.1.) de procesoare adaptabile.

### 2.1. Procesoarele N-N, N-D și N-N-D

Procesoarele adaptabile în baza modului de realizare a extinderilor sunt create conform procesării lor de tip nivel-nivel (N-N), nivel-direct (N-D) și nivel-nivel-direct (N-N-D).

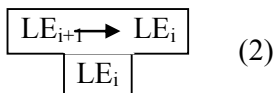
#### 2.1.1. Procesoarele de tip nivel-nivel (N-N)

Schema generală a acestui tip de procesor este:



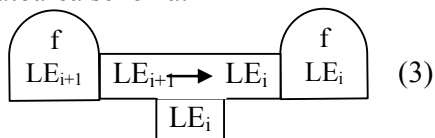
În această schemă programul în limbajul adaptabil este prezentat prin „Program sursă, nivel i”, procesorul de tip nivel-nivel este prezentat prin intermediul procesorului „N-N”, „Cercetare” definește dacă rezultatul translației mai conține extinderi (Da/Nu). Ca rezultat se obține „Programul obiect în limbaj bază”.

Utilizând formalismul E-T-M acest procesor este prezentat printr-un șir de procesoare de tip:

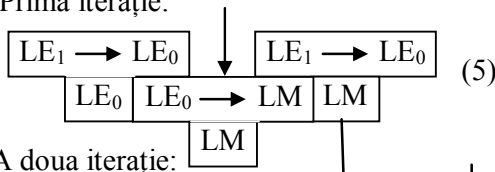


unde  $i = 0, 1, 2, \dots, p-1$  ( $p$ -numărul nivelurilor limbajului adaptabil de programare).

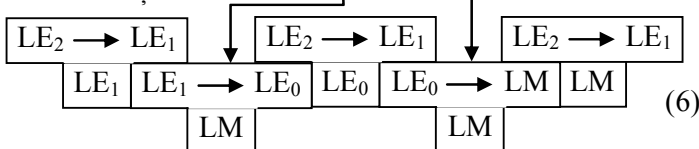
Unul din aceste procesoare de tip nivel-nivel traduce extinderile de nivelul superior ( $i+1$ ) din programul adaptabil, scris în limbajul adaptabil de programare  $f$ , obținând, ca rezultat, același program  $f$  cu extinderi doar de nivel cel mult  $i$ . Acest procesor este scris prin intermediul limbajului adaptabil de nivel  $i$ . Lucrul acestui procesor este prezentat prin următoarea schemă:



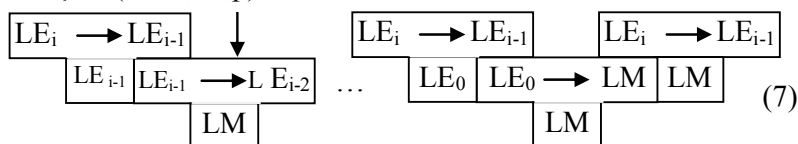
Prima iterație:



A doua iterație:



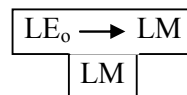
Iterația  $i$  ( $i=1,2,\dots,p$ ):



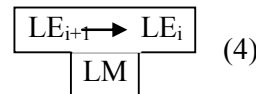
Prin urmare, dacă avem o mașină cu limbajul de programare  $LE_i$ , atunci prin intermediul acestui translator-compiler, programul  $f$  în limbajul  $LE_{i+1}$  se transformă în programul  $f$  în limbajul de programare  $LE_i$ . Acest translator este doar un set de definiții a extinderilor de nivel  $i+1$  prin intermediul limbajului de programare de nivel  $i$  ( $LE_i$ ).

Demonstrăm obținerea acestui set de procesoare (2) utilizând formalismul Earley.

**Lema N-N.** Dacă este dat un procesor adaptabil



este suficient de scris un set de extinderi (2), pentru a obține automatizat următorul set (4) de procesoare:



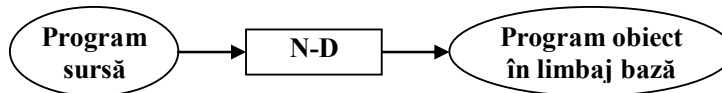
unde  $LE_{i+1}$  și  $LE_i$  sînt limbaje adaptabile de nivelul  $i+1$  și  $i$ ,  $LE_0$  este limbajul bază și  $LM$  este limbajul mașină.

Demonstrație:

Prin urmare, s-a obținut șirul de translație-procesoare de tip nivel-nivel, ceea ce era necesar de demonstrat.

**2.1.2. Procesoarele de tip nivel-direc (N-D)**

Schema generală a acestui tip de procesor este următoarea:



Prin intermediul acestui tip de procesor adaptabil oricare "Program sursă" în limbajul adaptabil de programare este translatat într-un "Program obiect în limbajul bază".

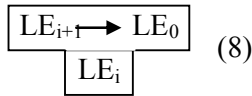
este suficient de scris un set de extinderi pentru a obține automatizat următorul set (9) de procesoare adaptabile de tip nivel-direc (N-D):



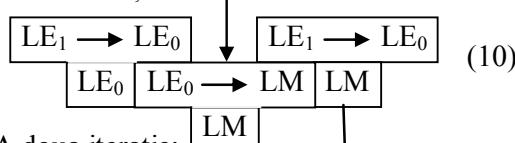
**Lema N-D.** Dacă este dat un procesor adaptabil

unde  $LE_{i+1}$  și  $LE_i$  sînt limbaje adaptabile de nivelul  $i+1$  și  $i$ ,  $LE_0$  este limbajul bază și  $LM$  este limbajul mașină.

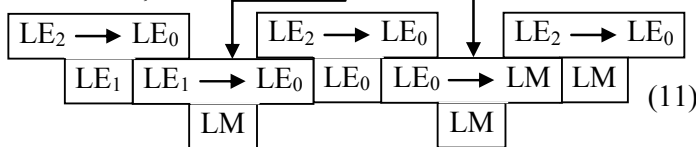
Demonstrație:



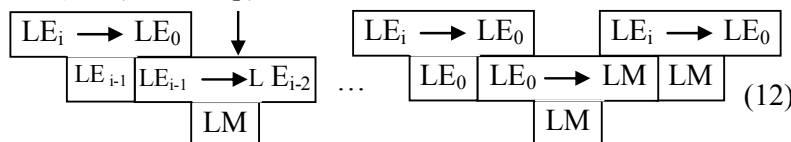
Prima iterație:



A doua iterație:



Iterația  $i$  ( $i=1,2,\dots,p$ ):



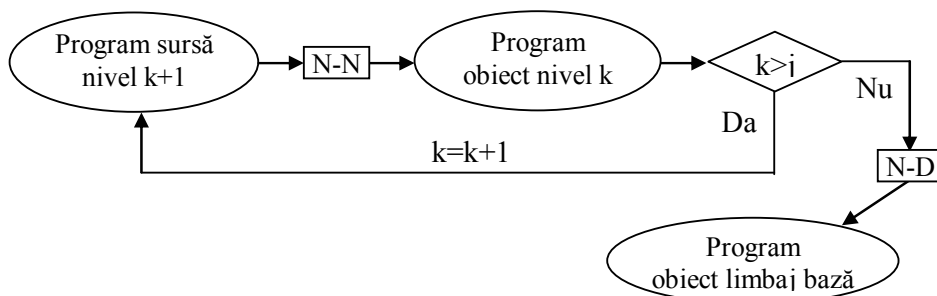
Prin urmare, obținem șirul (9) de translație-procesoare adaptabile de tip nivel-direc, ceea ce era necesar de demonstrat.

de extinderi pentru a obține setul de procesoare (9) utilizând doar numai limbajul de nivel  $i$  ( $LE_i$ ), unde  $i = 0,1,2,\dots,p$ , și  $p$ -nivelul limbajului adaptabil de programare.

**Corelație N-D.** Pentru a scrie șirul de extinderi de nivel  $i+1$  de tipul (8) este necesar de știut și utilizat limbajul adaptabil de programare de nivel  $i$  și limbajul bază. Este ușor de demonstrat, că e suficient de scris un set (2)

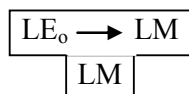
**2.1.3. Procesoare de tip nivel-nivel-direc**

Schema generală a acestui tip de procesor este următoarea:



Prin intermediul procesoarelor adaptabile “N-N” programul sursă adaptabil “coboară” după modul de traducere nivel-nivel până la obținerea “programului obiect” în limbajul adaptabil (intermediar) de nivel j. La etapa următoare acest program este transformat într-un “program obiect în limbajul bază” prin intermediul procesoarelor adaptabile “N-D” de tip nivel-direct.

**Lema N-N-D:** Dacă este dat un procesor adaptabil



este suficient de scris un set (2) de extinderi ( $i=j, j+1, \dots, p$ , p-numărul de niveluri ale limbajului adaptabil de programare și  $j \geq 0$ ) și un set (8) de extinderi ( $i=j-1, j-2, \dots, 0$ ) pentru a obține automatizat un set (4) de procesoare adaptabile de tip N-N și un set (9) de procesoare adaptabile de tip N-D.

Demonstrare: Această leamnă este ușor de demonstrat. Pentru aceasta este suficient de utilizat Lema N-N pentru a obține șirul (4) de procesoare adaptabile N-N scrise în limbajul mașinii LM și, în continuare, Lema N-D pen-

tru a obține setul (9) de procesoare adaptabile N-D scrise în limbajul mașinii LM.

**Corelația N-N-D:** Se poate demonstra, că, utilizând Lema N-N, Lema N-D și Corelația N-D pentru a obține setul (4) de procesoare adaptabile N-N și setul (9) de procesoare adaptabile N-D, care formează în comun setul (14) de procesoare adaptabile N-N-D:

$$\begin{array}{c} \boxed{LE_{i+1} \rightarrow LE_i} \quad \boxed{LE_j \rightarrow LE_0} \\ \boxed{LM} \quad \quad \quad \boxed{LM} \end{array} \quad (14)$$

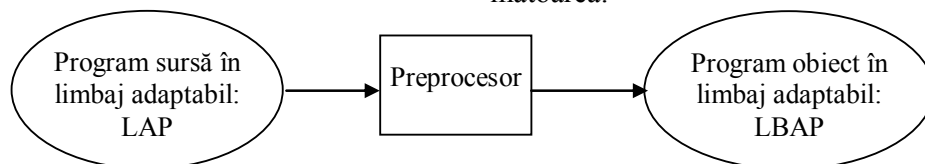
unde  $i=p-1, p-2, \dots, j, j \geq 0$ , p-numărul de niveluri ale limbajului adaptabil de programare, pentru aceasta este suficient de scris doar un set (2) de extinderi.

**2.2.Preprocesoarele, interprocesoarele și postprocesoarele**

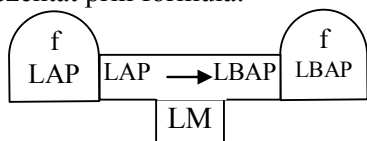
Procesoarele adaptabile de tip preprocesor, interprocesor și postprocesor realizează extinderile (elementele derivate) din programul scris într-un limbaj adaptabil de programare respectiv: până la realizarea elementelor de bază, în paralel cu realizarea acestora și după realizarea elementelor de bază.

**2.2.1.Preprocesoarele adaptabile**

Schema generală a unui preprocesor este următoarea:



Utilizând formalismul Earley preprocesorul este prezentat prin formula:



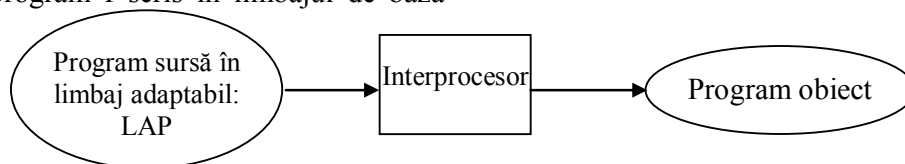
Prin punerea în funcție a acestui preprocesor adaptabil, programul f scris în limbajul adaptabil de programare LAP este transformat în același program f scris în limbajul de bază

adaptabil de programare. Acest procesor adaptabil este scris în limbajul mașinii LM.

Este interesantă interacțiunea preprocesorului adaptabil cu celelalte componente ale cubului adaptabilității (Fig.1.).

**2.2.2.Interprocesoarele adaptabile**

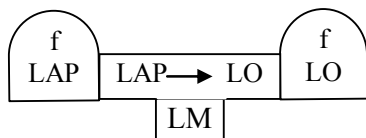
Schema generală a unui interprocesor este următoarea:



Interprocesorul realizează simultan elementele derivate (extinderile) și elementele de bază a „programului scris în limbajul de programare LAP”, obținând ca rezultat „programul

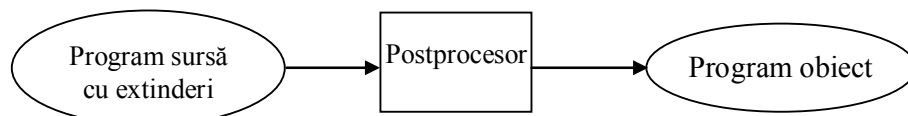
obiect” scris în limbajul mașinii sau, în caz general, într-un limbaj intermediar.

Utilizând formalismul Earley obținem formula următoare:



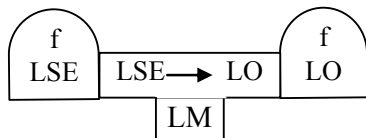
unde  $f$  este programul scris în limbajul adaptabil de programare LAP, LO-limbajul obiect și LM-limbajul mașinii.

Punerea în funcție a interprocesorului, prezentată prin formula de mai sus, cu programul  $f$  în LAP, ca date de intrarea lui, se obți-



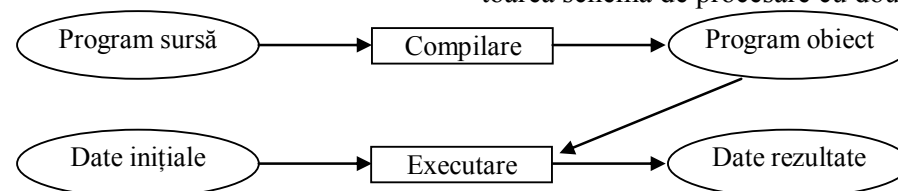
Postprocesorul realizează elementele derivate (extinderile) după realizarea elementelor de bază a limbajului de bază adaptabil de programare. „Programul sursă cu extinderi” este transformat în „programul obiect” fără extinderi. „Programul sursă” poate fi un cod în limbajul mașinii sau abstracte (limbaj sursă) cu extinderi. „Programul obiect”-rezultatul lucrului postprocesorului adaptabil este compus din codul în limbajul mașinii reale sau abstracte, copiate din „programul sursă”, și codul-echivalent extinderilor din programul sursă traduse de postprocesor.

Utilizând formalismul E-T-M obținem formula următoare:



unde  $f$  este programul scris în limbajul mașinii abstracte sau reale cu extinderi nerealizate LSE, LO-limbajul obiect în limbajul mașinii abstracte sau reale cu codul-echivalent al extinderilor realizate-limbajul obiect LO și LM-limbajul mașinii.

### 3.Procesoarele adaptabile de Tip-translatore.



Utilizând formalismul E-T-M un compilator este prezentat prin formula următoare (1):

ne programul  $f$  în limbajul mașinii sau, în mod general, în limbajul obiect intermediar LO. Se realizează simultan elementele derivate (de nivel) și elementele limbajului de bază adaptabil de programare LBAP.

### 2.2.3.Postprocesoarele adaptabile

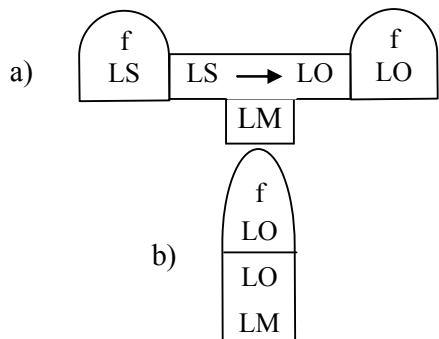
Schema generală a unui postprocesor este următoarea:

Procesoarele din cubul adaptabilității (Fig.1) în combinație formează nivelul trei de procesoare adaptabile. Ele sunt create în baza procesoarelor adaptabile de nivelul unu și doi. Sunt cunoscute, prin urmare, 27 de procesoare adaptabile de nivelul trei, care formează cubul adaptabilității.

Pentru a crea unele din aceste procesoare adaptabile este necesar de utilizat formalismul E-T-M pentru a demonstra și a treia componentă a acestor procesoare-componenta, bazată pe Tip-translatore, precum: compilatoare adaptabile, interpretoare adaptabile și de compilare-interpretare adaptabile. Prima componentă o constituie modelul de realizare a extinderilor: nivel-nivel, nivel-direct și nivel-nivel-direct. A doua componentă a procesoarelor din cubul adaptabilității (Fig.1) o constituie ordinea, consecutivitatea, locul, timpul, în care sunt realizate extinderile: până la realizarea elementelor de bază (preprocesare), în paralel cu realizarea elementelor de bază (interprocesare) și după realizarea elementelor de bază (postprocesare).

### 3.1.Compilare, interpretare și compilare-interpretare cu formalismul E-T-M.

3.1.1.Compilatorul este prezentat prin următoarea schemă de procesare cu două faze:



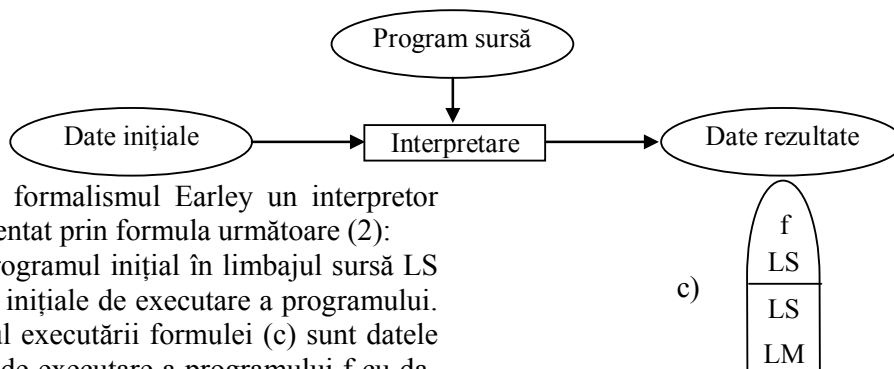
Prima fază: compilarea programului f scris în limbajul sursă LS, obținându-se programul f

în limbajul obiect LO (de obicei LO este limbajul mașinii LM).

A doua fază: interpretarea programului f în limbaj mașină LM.

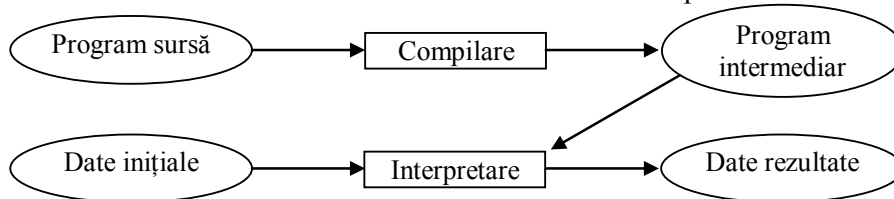
În continuare vom accentua atenția asupra 1-ei faze de compilare, care, din considerente de economie, va fi numită compilator (a).

**3.1.2. Interpretorul** este prezentat prin următoarea schemă de procesare:

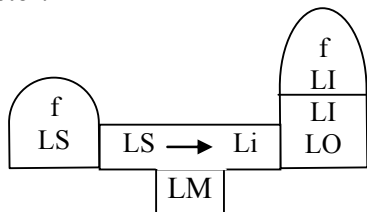


Utilizând formalismul Earley un interpretor este prezentat prin formula următoare (2): unde f-programul inițial în limbajul sursă LS cu datele inițiale de executare a programului. Rezultatul executării formulei (c) sunt datele rezultate de executare a programului f cu datele inițiale ca date de intrare.

**3.1.3. Compilator-interpretorul** într-o formă simplă este prezentat prin intermediul următoarei scheme de procesare:



Utilizând formalismul Earley obținem următoarea formulă (d) pentru un compilator-interpretor:



unde programul f în limbaj sursă LS este compilat în programul f în limbaj intermediar LI, care este interpretat de către interpretorul limbajului intermediar LI.

metodei Timp-Realizare a extinderilor (E-T-I-M), (2) a metodei de Nivel-Realizare a extinderilor (E-L-I-M), și (3) a metodei de Tip-Translate adaptabilă (P-T-I-M). Este demonstrată posibilitatea realizării automatizate a procesoarelor adaptabile de nivelului unu în baza formalismului E-T-M de interacțiuni translatorice a procesoarelor adaptabile Crearea automatizată a nivelului doi de procesoare adaptabile și a nivelului trei vor constitui baza comunicărilor [ZTod-03] și [Mic-03].

Discuțiile pe marginea actualelor comunicări sperăm vor constitui baza posibilității creării în Republica Moldova a industriei modulelor translatorice a procesoarelor adaptabile. Analiza posibilității implementării mijloacelor adaptabile ale realizării limbajului natural

**4. Concluzii.**

Prima parte a actualului Proiect de cercetări este consacrată tehnologiei [Tod-03] de realizare a extinderilor din punct de vedere (1) a



de procesare a informației va constitui baza concluziilor de implementare a lor în cadrul procesului de instruire a postuniversitarilor în primul rând și a universitarilor în rândul doi.

### Bibliografie

- [DTod-03]. Dumitru Todoroi. The Model Type of Extensions Realization of the Adaptability Cube' First Level of Translation Complexity' Adaptable Processors. // The Sixth International Conference on Economic Informatics IE'03, Romania, Bucharest, May 8-11, 2003
- [ZTod-03]. Zinaida Todoroi. The Second Level Adaptable Processors of the Extensible Time-Models Realization. // The Sixth International Conference on Economic Informatics IE'03, Romania, Bucharest, May 8-11, 2003
- [Mic-03]. Diana Micușă. The Adaptability Cube' Third Level of Translation Complexity' Adaptable Processors-Compilers. // The Sixth International Conference on Economic Informatics IE'03, Romania, Bucharest, May 8-11, 2003
- [Tod-02]. Dumitru Todoroi, Zinaida Todoroi. The Romanian Information Language MULTIMEDIA Dictionaries for European Community. // Proc. of International Conf. "Globalization and University' Economics Education", Vol.2, Iasi, October 24-25, 2002, p. 233-250.
- [Mic-02]. Diana Micușă, Toader Jucan, Dumitru Todoroi. The E-T-M formalism for NLP Adaptable Processors' Interactions. // Proc. of International Conf. "Globalization and University' Economics Education", Vol.2, Iasi, October 24-25, 2002, p. 200-218.
- [Tod-01]. Dumitru Todoroi, Zinaida Todoroi, Diana Micușă. Natural Language Processing based on Computerized Romanian Language. // Proc. of International Conf. "E-Business. Theory and Practice", ASEM, 2001, Chișinău, p.361-368.
- [Tod-00] Dumitru N. Todoroi, Sufi Nazem, Teodor Jucan, Diana D. Micușă. Future is Past multiplied by the Present: Software development by the Transition to a Full Information Society. // Acta Academia, 2000, Chișinău: Evrica, - pp 53-62.
- [Tod-99] Dumitru N. Todoroi, Stanley J. Hille, Michael Mulder, Teodor Jucan, Diana Micușă, Robert Bernier, Hesham El-Rewini. The Full Information Society Initial Development Period. // Acta Academia, 1999, Chișinău: Evrica, -pp 16-28.
- [Tod-98]. Dumitru Todoroi, Sufi Nazem, Teodor Jucan, Diana Micușă. Transition to a Full Information Society: Stage Development. // Working Paper No. 98-2, March, 1998, UNO, College of Business Administration, Omaha, Nebraska, USA, - 38 p.
- [Den-97]. Beyond Calculation : The Next Fifty Years of Computing. // Edited by Peter J. Denning and Bob Metcalfe, Copernicus, 1997 Springer-Verlag New York, Inc., 350 p.
- [IFI-94]. IFIP'94 Congress. // Proceedings of the IFIP'94 Congress. Volumes 1, 2, 3. Munich, Germany, 1994. - 1347 p.
- [Tod-92]. D.Todoroy. Computer Science. The Adaptable Programming. The Basis Conceptions. // Ed: ASEM, Chișinău, 1992. - 76 p.
- [Ear-70]. T. Earley and Ko., The formalism for translation interactions. // Comm. ACM, 13, 2 (Feb. 1970), pp. 70-81.