

Crearea automatizată a procesoarelor adaptabile de nivelul doi bazate pe nivelul unu cu utilizarea T-formalismul de interacțiune translatorică

Zinaida TODOROI, ASEM, Republic of Moldova, Manfred KUDLEK, Hamburg University, Germany, Diana MICUSA, USM, Republic of Moldova, Dumitru TODOROI, ASEM, Republic of Moldova

In this communication are studied the Adaptable Processors of the second level of translation complexity from the Adaptability Cube. The second level of translation complexity Adaptable Processors present the combinations of the first level of Adaptable Processors, obtained on the basis of Extensions Time-Realization methods, Model-Realization methods, and on the Adaptable Type-Translation methods. Present communication is concerned with the automatically obtaining of the Second Level Adaptable Processors of the Time-Model-Realization Type of extensions processing.

Introducere

Procesoarele adaptabile de nivelul doi de complexitate translatorică din cubul adaptabilității reprezintă combinații de procesoare de nivelul unu de tip Timp-Realizare și Model-Realizare a extinderilor și de Tip-Translatore adaptabile [Tod-03]. Procesoare-

le de nivel doi sunt prezentate prin intermediul procesoarelor adaptabile de tip Timp-Model-Realizare a extinderilor (Tabelul 1), de tip Tip-Model-Realizare a extinderilor (Tabelul 2) și de tip Tip-Timp-Realizare a extinderilor (Tabelul 3).

Tabel 1. Procesoare adaptabile de tip Timp-Model-Realizare a extinderelor

	Nivel-Nivel	Nivel-Direct	Nivel-Nivel-Direct
Pre-procesare	Preprocesor-N-N	Preprocesor-N-D	Preprocesor-N-N-D
Inter-procesare	Interprocesor-N-N	Interprocesor-N-D	Interprocesor-N-N-D
Post-procesare	Postprocesor-N-N	Postprocesor-N-D	Postprocesor-N-N-D

Tabel 2. Procesoare adaptabile de tip Tip-Model-Realizare a extinderelor

	Nivel-Nivel	Nivel-Direct	Nivel-Nivel-Direct
Compilare	Compiler-N-N	Compiler-N-D	Compiler -N-N-D
Interpretare	Interpreter-N-N	Interpreter-N-D	Interpreter -N-N-D
Compilare-interpretare	Interpreter-Interpreter-N-N	Interpreter-Interpreter-N-D	Interpreter-Interpreter-N-N-D

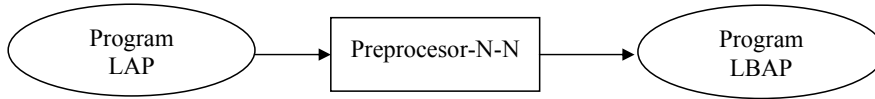
Tabel 3. Procesoare adaptabile de tip Tip-Timp-Realizare a extinderilor

	Preprocesare	Interprocesare	Postprocesare
Compilare	Compiler-Preprocesor	Compiler-Interprocesor	Compiler-Postprocesor
Interpretare	Interpreter-Preprocesor	Interpreter-Interprocesor	Interpreter - Postprocesor
Compilare-interpretare	Compiler-Interpreter-Preprocesor	Compiler-Interpreter-Interprocesor	Compiler-Interpreter-Postprocesor

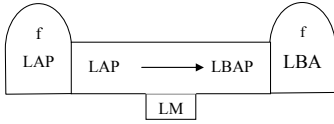
Sunt demonstrate procesele de obținere automatizată a procesoarelor adaptabile de nivelul doi de tip Timp-Model-Realizare a extinderilor în baza formalismului E-T-M de interacțiune a procesoarelor adaptabile. Demonstrarea obținerii automatizate a celorlalte procesoare de nivelul doi va fi prezentată în următoarele publicații.

1. Preprocesoarele N-N,-N-D, -N-N-D.

1.1. Preprocesorul-N-N realizează extinderile din programul în limbajul adaptabil până la realizarea elementelor de bază după modul de realizare a extinderilor cu procesoarele de tip nivel-nivel. Schema generală a acestui procesor adaptabil este următoarea:

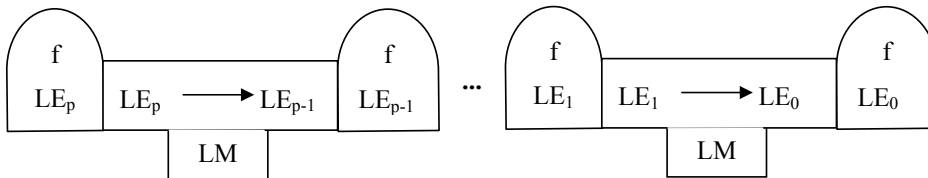


Utilizând formalismul Earley activarea acestui procesor adaptabil este prezentat prin intermediul următoarei formule:



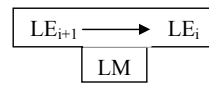
unde f -programul în limbajul LAP , $LAP \rightarrow LBAP$ -preprocesorul-N-N scris în limbajul mașină LM . Preprocesorul $LAP \rightarrow LBAP$ prezintă un set de procesoare de tip nivel-nivel.

Teorema Preprocesor-N-N: Dat fiind compilatorul (1) este suficient de

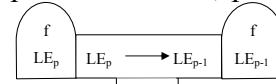


obținem preprocesorul-N-N, care realizează programul f în limbajul adaptabil de programare, compus din maxim p niveluri de extinderi, transformându-l în programul f scris în limbajul bază adaptabil de programare $LBAP$ (LE_0).

scris un set (2) de extinderi pentru a obține preprocesorul-N-N, prezentat

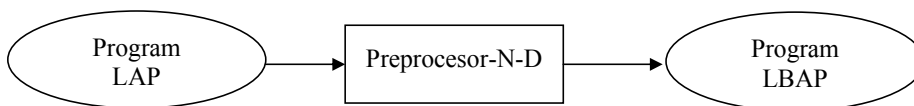


printr-un set (3) de preprocesoare-N-N scrise în limbajul mașinii LM .

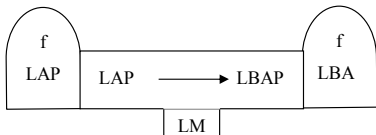


Demonstrație: Analog cu Lema N-N obținem un set (3) de procesoare adaptabile N-N, care, fiind puse în funcțiune consecutiv conform formulei (4),

1.2. Preprocesorul-N-D realizează extinderile din programul scris în limbajul adaptabil de programare direct în limbajul bază adaptabil de programare conform următoarei scheme generale:



Utilizând formalismul Earley funcționarea acestui preprocesor-N-D este prezentată prin următoarea formulă:

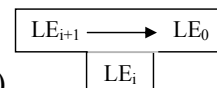


unde f - programul în limbajul LAP la intrarea preprocesorului (LAP) și la ieșirea lui ($LBAP$), și $LAP \rightarrow LBAP$ -preprocesorul-N-D scris în limbajul mașinii LM , acest procesor fiind reprezentat printr-un set de procesoare de tip N-D.

Teorema Preprocesor-N-D

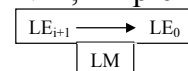
Dat fiind procesorul (1), este suficient de

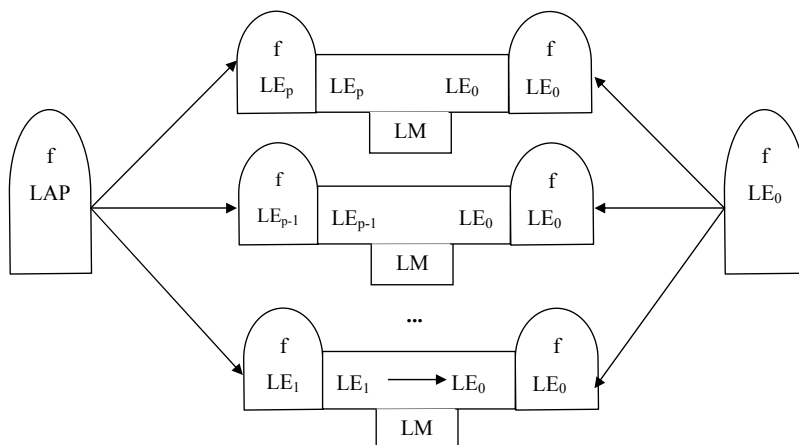
scris un set de extinderi (5), unde $i=0,1,2,\dots,p$; p -numărul maxim de niveluri ale LAP , pentru a obține preprocesorul-N-D, prezentat printr-un set (6)



de preprocesoare-N-D scrise în limbajul mașinii.

Demonstrație: Analog cu Lema N-D și Coralația N-D, obținem un set de preprocesoare-N-D, care, fiind puse în funcțiune conform următoarei formule (7):

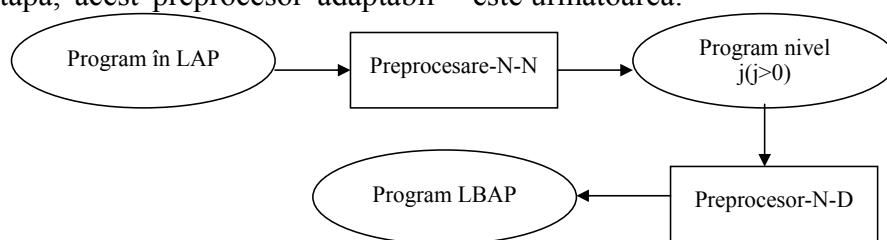




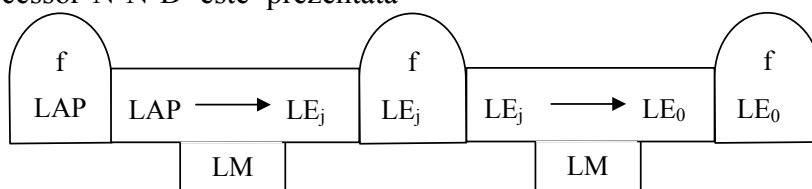
obținem preprocesorul-N-D, care realizează programul f în limbajul de programare LAP, compus din maxim p niveluri de extinderi, direct obținând, ca rezultat, același program f în limbajul LBAP.

1.3. Preprocesorul-N-N-D: realizează extinderile din programul adaptabil în două etape. La prima etapă, acest preprocesor adaptabil

efectuează realizarea extinderilor de niveluri $i=p, p-1, \dots, j$ ($j>0$) după metoda de preprocesare nivel-nivel. Programul, obținut după prelucrarea la prima etapă, la cea de-a doua etapă, este transformat într-un program în limbajul bază adaptabil de programare LBAP, direct după metoda de procesare nivel-direct. Schema generală a acestui procesor adaptabil este următoarea:



Utilizând formalismul E-T-M funcționarea prin intermediul următoarei formule: acestui preprocessor-N-N-D este prezentată



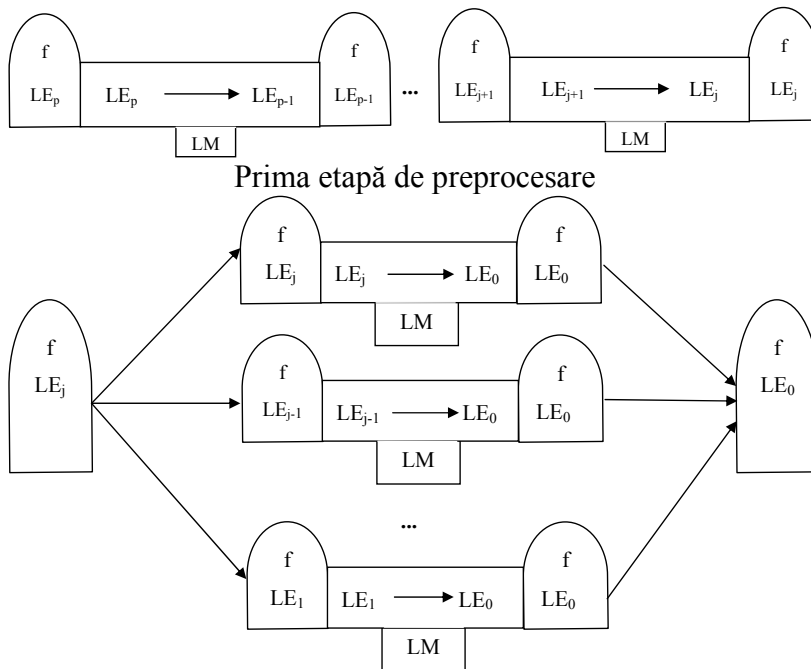
unde primul f -programul în LAP este transformat în f scris în limbajul adaptabil de nivel j (LE_j ; $j>0$) prin intermediul preprocesorului $LAP \rightarrow LE_j$ de tip N-N; cel de-al doilea f -programul în LE_j este transformat în f în LBAP (LE_0) prin intermediul preprocesorului $LE_j \rightarrow LE_0$ de tip N-D și LM-limbajul mașină.

Teorema Preprocesor-N-N-D

Dacă este dat procesorul (1), este suficient de creat un set (2) de extinderi pentru nivelurile $i=p-1, p-2, \dots, j$ ($j>0$) și un set (5) de extinderi pentru nivelurile $i=0, 1, 2, \dots, j-1$ cu scopul de a obține preprocesorul-N-N-D, prezentat prin intermediul a două seturi de tip (3) și respec-

tiv de tip (6) scrise în limbajul mașină.

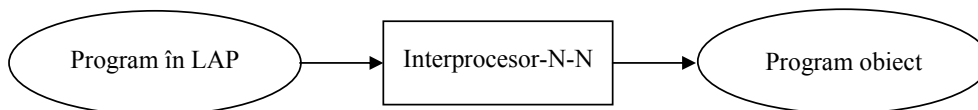
Demonstrație: Analog cu Teorema Preprocesor-N-N obținem $p-j$ preprocesoare-N-N și, conform Teoremei Preprocesor-N-D, obținem j preprocesoare-N-D. Primele $p-j$ preprocesoare-N-N „coboară” programul f în limbajul LAP de la nivelul p consecutiv până la nivelul j după metoda de preprocesare-N-N. Ultimele j procesoare preiau programul obținut la prima etapă de preprocesorul-N-N-D, prezentat de subpreprocesorul-N-N, și-l transformă direct în bază (LBAP). Preprocesorul-N-N-D funcționează conform următoarei formule (8):



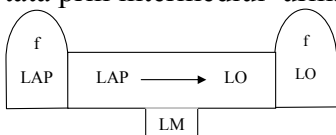
Preprocesoarele-N-N, -N-D și -N-N-D formează primul set de procesoare adaptabile de nivelul doi. Ele prezintă soluțiile pentru crearea procesoarelor adaptabile, prezentate prin rândul unu al tabelului 1. Aceste preprocesoare formează unul din următorii pași în obținerea automatizată a procesoarelor adaptabile prezentate în cubul adaptabilității.

2. Interprocesoarele-N-N, -N-D, -N-N-D.

2.1. Interprocesorul-N-N realizează elementele deviate din programul scris în LAP după modul N-N de realizare a extinderilor simultan cu elementele de bază (din LBAP), obținând programul obiect scris în limbajul mașinii reale sau abstracte. Schema generală a unui interprocesor-N-N este următoarea:



Utilizând formalismul Earley, activarea acestui interprocesor-N-N adaptabil este prezentată prin intermediul următoarei formule:



, unde f-programul în LAP este prezentat ca date de intrare la interprocesorul-N-N adaptabil LAP→LO, LO și LAP-limbaj obiect și limbaj adaptabil de programare respectiv, și LM-limbajul mașină.

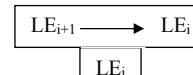
Teorema Interprocesor-N-N



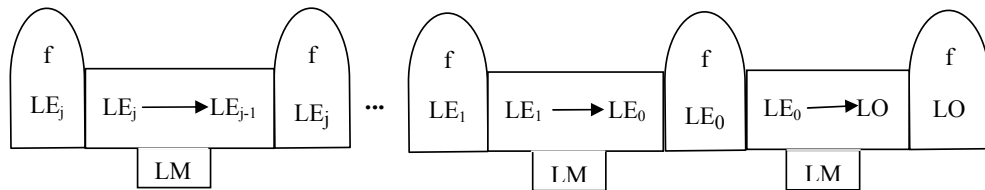
Fie dat procesorul (1) . Este necesar de obținut un set (2) de procesoare adaptabile de tip interprocesor-N-N:



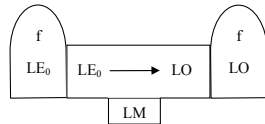
, unde $i=0,1,2,\dots,p-1$; p-nivelul LAP. Pentru soluționarea acestei probleme este suficient de creat setul (3) de extinderi:



Demonstrație: Analog cu Lema N-N obținem un set (2) de procesoare adaptabile N-N, care, fiind puse în funcțiune consecutiv pentru fiecare extindere de nivel j ($j=1,2,\dots,p$) conform formulei (4):

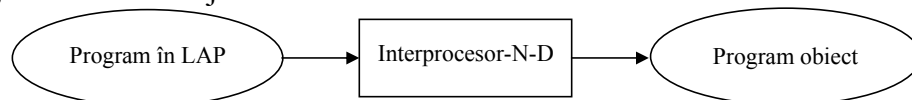


și simultan pentru fiecare element de bază

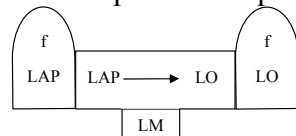


conform formulei (5):

obținem formula complexă pentru interprocesorul-N-N, compusă din formulele (4) și (5). Aceste formule sunt executate simultan asupra programului f în LAP. Limbajul LAP este compus din p niveluri de elemente derivate și din elemente în LBAP. Obținem programul f în limbajul obiect LO.



Utilizând formalismul E-T-M, activitatea acestui interprocesor-N-D adaptabil este pre-



zentată prin formula:

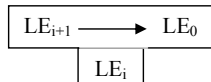
unde f - în LAP este programul inițial și în LO-programul rezultat, LAP, LO și LM-respectiv limbaj adaptabil, limbaj obiect și limbaj mașină.

Teorema Interprocesor-N-D

Fie dat procesorul (1) și este necesar de obținut un set (6) de procesoare adaptabile de tip



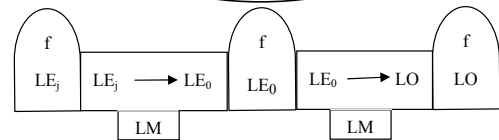
interprocesor-N-D: , pentru i=0,1,2,...,p, care împreună cu (1) formează (6) interprocesorul-N-D. Pentru soluționarea problemei este suficient de creat un set (7) de



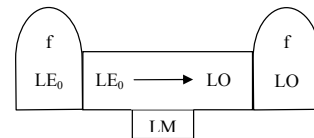
extinderi:

Demonstrație: Analog cu Lema N-D obținem setul (6) de procesoare adaptabile de tip N-D. Aceste procesoare, fiind puse în funcțiune pentru fiecare extindere de nivel j (j=1,2,...,p) din programul f în LAP, conform formulei

2.2. Interprocesorul-N-D realizează elementele derivate (extinderile) din programul inițial scris în LBAP după modul N-D de realizare a extinderilor, simultan cu elementele de bază (scrise în LBAP) din acest program, obținând, ca rezultat, programul rezultat (echivalent funcțional cu programul inițial) scris în limbajul obiect (LO)-limbaj al mașinii reale sau abstracte. Schema generală este următoarea:

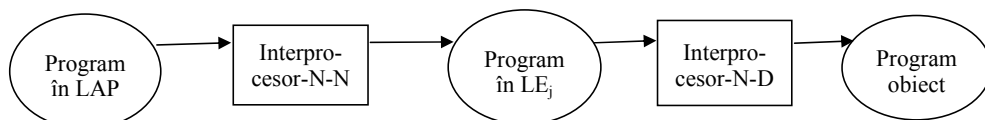


(8) și simultan procesorul (1) pentru fiecare element de bază conform formulei (9),

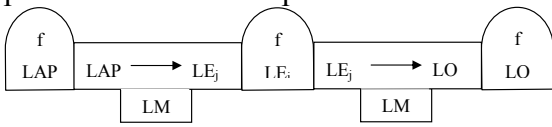


obținem formula complexă a interprocesorului-N-D. Acest interprocesor-N-D pentru fiecare element din programul f în LAP verifică apartenența elementului nivelului j sau bazei și, în dependență de rezultat, pune în funcțiune procesoarele conform (8) și respectiv (9), obținând ca rezultat programul f scris în limbajul obiect LO.

2.3. Interprocesorul-N-N-D realizează elementele derivate din programul scris în LAP parțial după modul N-N și în continuare după modul N-D de realizare a extinderilor simultan cu realizarea elementelor de bază (din LBAP), obținând programul obiect (intermediar; limbaj mașină reală sau abstractă). Schema generală a unui interprocesor-N-N-D este următoarea:



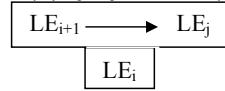
Utilizând formalismul E-T-M, activitatea acestui interprocesor-N-N-D adaptabil este prezentată prin formula:



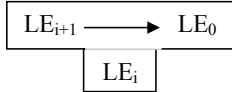
unde f - programul inițial în LAP este, la prima etapă, direct transformat în programul f în LE_j ($j > 0$) cu interprocesorul-N-N $LAP \rightarrow LE_j$ și apoi, la care de-a doua etapă direct transformat în programul f în LO (limbaj obiect) cu interprocesorul-N-D $LE_j \rightarrow LO$ și LM -limbaj mașină.

Teorema Interprocesor-N-N-D

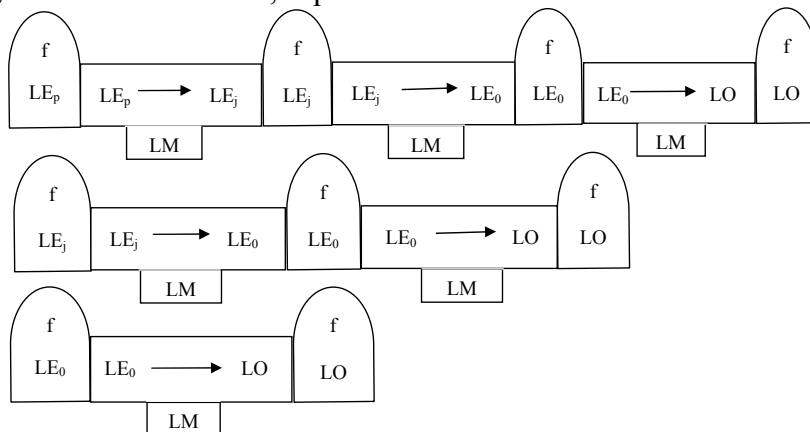
Fie dat procesorul (1) și șirurile (10) și (11)



de extinderi:

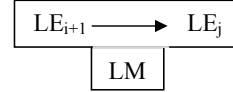


$i=j, j+1, \dots, p-1$; , pentru

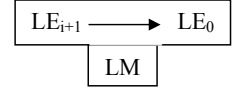


Prima formulă (14.1) funcționează pentru elementele derivate de nivel $i > j \geq 0$, formula (14.2)- pentru elementele derivate de nivel $j \geq 0$ și formula (14.3)- pentru elementele de bază. Interprocesarea-N-N (14.1) se efectuează pentru elementele extinderi mai mari ca j . Interprocesarea-N-D (14.2) se efectuează pentru extinderile de niveluri $i=1, 2, \dots, j$. Aceste două interprocesări (14.1) și (14.2) se efectuează simultan cu procesorul (14.3) asupra programului scris în LAP , formând interprocesorul-N-N-D adaptabil. Interprocesorul-N-N, interprocesorul-N-D și interprocesorul-N-N-D împreună formează al doilea set de procesoare de nivelul doi. Ele

$i=0, 1, 2, \dots, j-1$. Aceste șiruri (10) și (11) sunt suficiente pentru a obține interprocesorul-N-N-D adaptabil, prezentat prin intermediul șirului de procesoare (12) și (13):



, pentru $i=j, j+1, \dots, p-1$;



, pentru $i=0, 1, 2, \dots, j-1$.

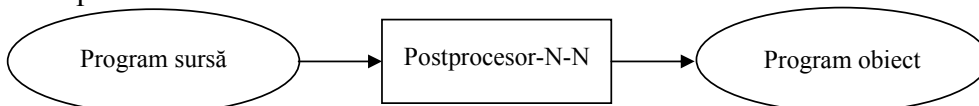
Demonstrație: Utilizând Teorema Interprocesor-N-N obținem primele $p-j$ interprocesoare-N-N (setul (12) de procesoare) și, utilizând Teorema Interprocesor-N-D obținem ultimele j interprocesoare-N-D (setul (13) de procesoare).

Interprocesorul-N-N-D funcționează conform următoarei formule (14):

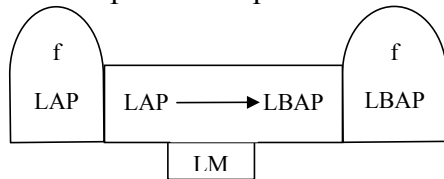
prezintă soluțiile pentru crearea automatizată a procesoarelor adaptabile, prezentate în celulele din rândul doi al Tabelului 1. Aceste procesoare formează al doilea pas în procesul de obținere automatizată a procesoarelor adaptabile din cubul adaptabilității.

3. Postprocesoarele-N-N, -N-D și -N-N-D.

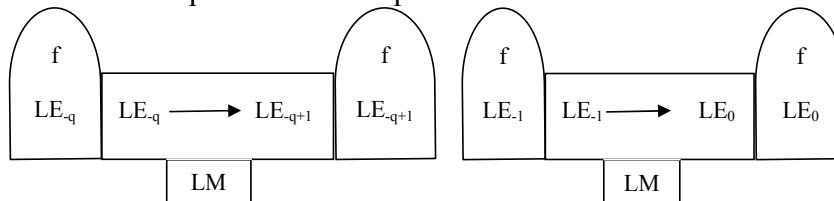
3.1. Postprocesorul-N-N realizează extinderile din programul în LAP după realizarea elementelor din bază ($LBAP$). Realizarea extinderilor se efectuează după modul de realizare de tip nivel-nivel. Schema generală a postprocesorului-N-N adaptabil este următoarea:



“Programul sursă” este compus din elementele LO funcțional echivalente elementelor LBAP (din programul LAP) deja realizate de procesorul LBAP→LO și elementele derivate (extinderi), care sunt realizate de postprocesorul-N-N. „Programul obiect”, în așa fel este compus din elementele LO, echivalente elementelor de bază din programul LAP, și din elementele LBAP, echivalente elementelor-extinderi din programul LAP. Utilizând formalismul Earley activizarea acestui procesor adaptabil este prezentată prin următoarea



formulă:
unde f în LAP este compus din elementele-extinderi și elementele LO, care sunt realizările elementelor de bază (LBAP), LAP→LBAP este un set de procesoare de tip



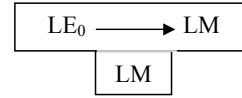
care, fiind pus în funcțiune conform formulei (18) cu programul f scris în limbajul adaptabil LAP, compus din extinderi mai jos ca baza, transformă (“coboară”) programul inițial în LAP în programul rezultat, scris în limbajul bază adaptabil de programare LBAP (LE0).

În continuare, programul rezultat f în LE0 (18) va fi realizat de procesorul LE0→LO ca și elementele de bază din programul LAP,

nivel-nivel și LO-limbajul obiect (intermediar, mașină reală, mașină abstractă).

Teorema Postprocesor-N-N

Fie dat procesorul (15). Este suficient de creat un set (16) de extinderi



, unde $i=0,-1,-2,\dots,-q$; q- numărul nivelurilor mai jos de bază a limbajului adaptabil de programare, pentru a obține

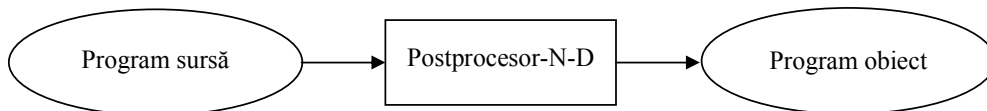


un set (17) de procesoare adaptabile, care formează postprocesorul-N-N necesar.

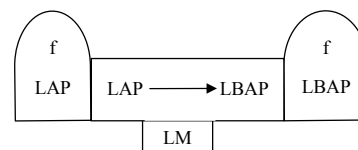
Demonstrație: Utilizăm Lema N-N pentru a obține setul (17) de procesoare de tip nivel-nivel, dat fiind setul (16) de extinderi de nivelul mai jos decât baza. Ca rezultat obținem postprocesorul-N-N:

deja realizat până la punerea în funcție a postprocesorului-N-N după formula (18).

3.2. Postprocesorul-N-D realizează extinderile mai jos de bază din programul în LAP după realizarea elementelor de bază. Realizarea extinderilor mai jos de bază se efectuează după modul de realizare (a extinderilor) de tip nivel-direct. Schema generală a postprocesorului-N-D adaptabil este următoarea:



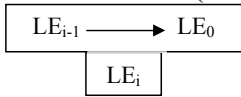
Din “programul sursă” postprocesorul, după modul de realizare nivel-direct, realizează extinderile de nivel i mai jos de bază, unde $i=-1,-2,\dots,-q$ și q- numărul nivelurilor de mai jos de bază a limbajului adaptabil de programare LAP. Utilizând formalismul E-T-M activizarea acestui postprocesor-N-D adaptabil este prezentată prin formula:



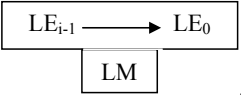
, unde f în LAP este compus din elementele de bază deja realizate și extinderile de niveluri mai jos de bază, care sunt realizate direct în bază de către un set de procesoare de tip nivel-direct.

Teorema Postprocesor-N-D

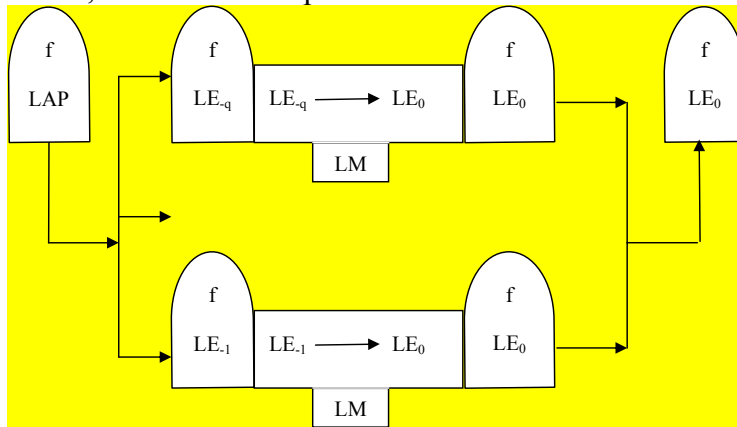
Fie dat procesorul (15) și un set (19) de extinderi de niveluri i ($i=-1,-2,\dots,-q$) mai jos de



bază: . Atunci este posibil de creat automatizat un set de procesoare



(20): , care formează pos-



tprocesorul-N-Dadaptabil necesar.

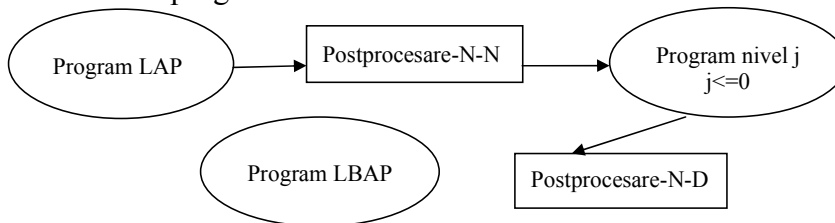
Demonstrație: Conform Lemei N-D, dat fiind procesorul (15) și setul de extinderi (18), se poate obține automatizat setul (19) de procesoare. Procesoarele (20) formează postprocesorul-N-D necesar, care funcționează conform următoarei formule (21):

Rezultatul evaluării formulei (21) este programul f în LBAP (LE_0), care va fi realizat de procesorul $LE_0 \rightarrow LO$ ca și elementele de bază din programul inițial f în LAP, deja realizate până la punerea în funcțiune a postprocesorului-N-D conform formulei (21).

3.3. Postprocesorul-N-N-D realizează extinderile mai jos de bază din programul în LAP

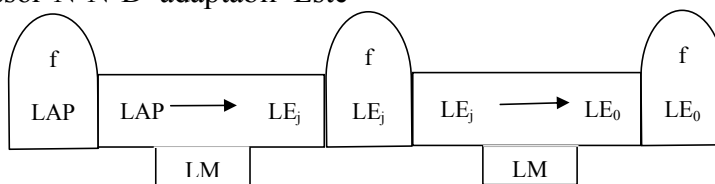
în două etape după realizarea elementelor de bază. Realizarea extinderilor parțial (etapa 1) este efectuată de procesoare de tip nivel-nivel și după aceasta (etapa 2) este efectuată de procesoarele de tip nivel-direct.

Schema generală a unui postprocesor-N-N-D este următoarea:



Utilizând formalismul E-T-M funcționarea acestui postprocessor-N-N-D adaptabil Este

prezentată prin formula următoare:



unde f -programul în LAP, care este transformat în programul f scris în LE_j ($-q < j <= 0$) prin intermediul postprocesorului-N-N ($LAP \rightarrow LE_j$); programul f în LE_j este transformat în f scris în LE_0 (LBAP) prin intermediul postprocesorului $LE_j \rightarrow LE_0$ de tip N-D și LM-limbaj mașină.

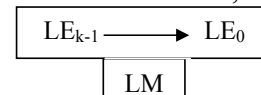
Teorema Postprocessor-N-N-D

Poate fi creat automatizat postprocesorul-N-

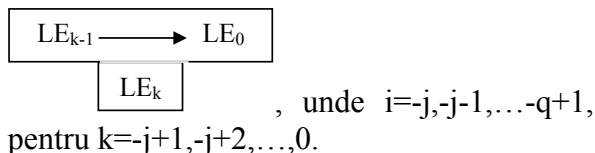
N-D, compus dintr-un set (22) de procesoare de tip N-N și dintr-un set (23) de procesoare



de tip N-D: , unde $i=-$

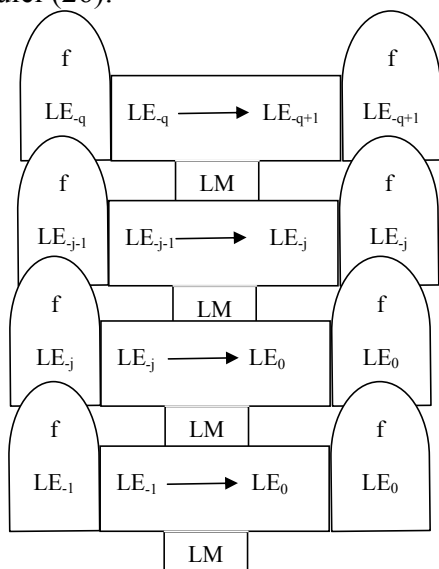


$q+1, \dots, j; j <= 0;$, pentru $k=-j+1, \dots, 0$, dacă sunt date inițial procesorul și șirurile (24) și (25) de extinderi:



Demonstrație: Conform Teoremei Postprocesor-N-N poate fi obținut șirul (22) de procesoare-N-N în baza setului (24) de extinderi. În baza Teoremei Postprocesor-N-D poate fi obținut șirul (23) de procesoare-N-D dat fiind setul (25) de extinderi. Ambele șiruri (22) și (23) de procesoare, în comun, formează șirul de procesoare adaptabile-componentele postprocesorului-N-N-D, cea ce era necesar de demonstrat.

Postprocesorul-N-N-D va funcționa conform formulei (26):



Faza 1: Primul set de procesoare de tip postprocesor-N-N efectuează procesarea de tip nivel-nivel.

Faza 2: Al doilea set de procesoare de tip postprocesor-N-D efectuează procesarea de tip nivel-direct.

Postprocesorul-N-N-D transformă programul LAP în două faze: faza 1: prin intermediul postprocesoarelor de tip N-N și faza 2: în continuare prin intermediul postprocesoarelor de tip N-D. În continuare programul rezultat f în LE_0 (26) va fi realizat de procesorul $LE_0 \rightarrow LO$ ca și elementele de bază din programul LAP, deja realizate până la punerea în funcțiune a postprocesorului-N-N-D conform formulei (26). Postprocesoarele-N-N, N-D și N-N-D formează ultimul set de procesoare adaptabile de nivelul doi, care sunt prezentate în Tabelul 1.

4. Concluzii și perspective

Au fost demonstrate posibilitățile de obținere automatizată a procesoarelor adaptabile din cubul adaptabilității de tip Timp-Model-Realizare de nivelul doi de complexitate translatorică. Următoarele două seturi de procesoare adaptabile de tip Tip-Timp-Realizare și Tip-Model-Realizare vor prezenta subiectul următoarelor publicații.

Referințe

[Tod-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, [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, To be published.

[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, Chisinau, p.361-368.