

# EVALUAREA CONSISTENȚEI UNUI LIMBAJ DE INTERACȚIUNE OM-CALCULATOR CU AJUTORUL MODELELOR CONEXIONISTE

dr.ec. Costin Pribeanu  
drnd.mat. Laurențiu Leuștean

*Institutul Național de Cercetare-Dezvoltare, ICI București*

**Rezumat:** Confirmarea ipotezei că ușurința în învățare este dependentă de consistența limbajului, se poate face pe cale experimentală, testând diferite variante ale aceleiași sarcini de lucru. O modalitate mai puțin costisitoare este utilizarea unei rețele neurale simple, antrenată să învețe o gramatică sarcină acțiune. În cadrul acestui articol, se prezintă două experimente de acest tip și o comparație a rezultatelor cu cele obținute prin estimare pe baza unei gramatici sarcină-acțiune.

**Cuvinte cheie:** interacțiune om-calculator, rețele neurale, gramatici sarcină-acțiune.

## 1. Introducere

Pentru a evalua posibilitatea de utilizare, în special din punctul de vedere al ușurinței învățării, este necesară analizarea cunoștințelor cerute pentru a opera cu o interfață. O formă de reprezentare eficientă o constituie tehnicile bazate pe notații formale, capabile să rescrie sarcini simple, în specificații de acțiuni. Tehnicile din această categorie sunt utile în compararea unor opțiuni de proiectare sau a unor interfețe în ceea ce privește ușurința în învățare. Posibilitatea de utilizare se poate evalua prin numărarea regulilor, adâncimea (detalierea) descrierii și numărul de comenzi pe care trebuie să le cunoască utilizatorul.

Payne și Green [1] propun o gramatică sarcină-acțiune, denumită TAG (Task Action Grammar), definită ca fiind un dispozitiv formal, care pornește de la o descriere a sarcinii, și un set de componente semantice. "Outputul" produs este lista acțiunilor necesare pentru a efectua sarcina de lucru.

În esență, notația este o transformare (o derivare) a sarcinii utilizatorului în secvențe de acțiuni, de aici și denumirea de gramatică sarcină-acțiune. Simbolurile de start ale gramaticii sunt sarcini, iar simbolurile terminale - specificații ale acțiunilor. În mod corespunzător, termenul limbaj sarcină-acțiune desemnează domeniul de aplicabilitate al acestor reprezentări: o interfață om-calculator, indiferent dacă este bazată pe limbaj de comandă sau pe manipulare directă.

În TAG se presupune că unei sarcini simple îi corespunde un concept din universul utilizatorului. Aceste concepte corespund unei reprezentări a cunoștințelor bazate pe trăsături semantice. Aceste trăsături pot fi privite ca variabile restricționate ca tip în sensul că marja valorilor posibile este definită

printr-un  $n$ -tuplu, de regulă, cu un număr de valori mic.

Ca metrică pentru estimarea ușurinței în învățare, TAG utilizează numărul de scheme (reguli de nivel înalt) și numărul de reguli de rescriere. Cu cât numărul de scheme ale regulilor este mai redus, cu atât limbajul este mai consistent. O metodă de evaluare mai precisă a efectelor pozitive ale consistenței asupra ușurinței în învățare este cea experimentală, prin care se compară diferite variante ale sarcinii de lucru.

O modalitate mai puțin costisitoare este utilizarea unei rețele neurale simple, antrenată să învețe o gramatică sarcină-acțiune. În acest tip de experiment, consistența limbajului este variabila independentă, iar timpul de învățare este variabila măsurată.

În cadrul acestui articol, se prezintă două experimente care au avut ca obiectiv verificarea ipotezei potrivit căreia ușurința în învățare este influențată pozitiv de consistența limbajului de interacțiune. În secțiunea următoare, se descriu trăsăturile experimentului. Rezultatele experimentelor sunt descrise în secțiunile 3 și 4. În Secțiunea 5, este făcută o discuție a ultimului experiment, comparând rezultatele cu cele obținute prin evaluarea gramaticii sarcină-acțiune.

## 2. Trăsături ale experimentului

Exemple de astfel de experimente sunt descrise de Bowden et al [2], Plunket și Marchman [3], Green și Doubleday [4]. În cadrul acestei lucrări au fost efectuate două experimente de acest tip, urmând metoda dată de Green și Doubleday [4], care testează consistența pe un exemplu propriu, evaluată inițial pe baza specificației TAG, și cel dat de Bowden et al. și experimentat, de asemenea, de Green și Doubleday.

Algoritmul folosit a fost algoritmul de învățare supervizată Backpropagation [5]. Acest algoritm dă o metodă de ajustare a ponderilor într-o rețea neurală feed-forward. Scopul ajustării ponderilor este producerea ieșirilor corecte pentru un set de exemple de antrenament, constând în date de intrare și ieșirile dorite. Algoritmul are doi parametri

- un număr mic de dispozitive astfel încât să se evite confuzia utilizatorului neprofesionist;
  - naturalețe în utilizarea dispozitivelor menționate;
  - situarea la o înălțime corespunzătoare, care să permită și accesul persoanelor cu cerințe speciale (handicapați, copii etc.);
  - dispozitivele de ieșire trebuie să fie ușor de abordat: monitorul trebuie să fie de o dimensiune care să permită lizibilitate oricărui tip de utilizator, iar sistemul de redare a sunetului trebuie să fie de foarte bună calitate;
  - securitatea trebuie să fie asigurată prin folosirea unui design corespunzător și a unor materiale adecvate, robuste;
  - manipularea operațiilor trebuie să fie condusă de un set de operații simple, inteligibile, bazate pe reguli ergonomice.
2. **IONIȚĂ, A. & col.:** Model experimental de Baze de Date Geografice de Referință, TR ICI 1997.
  3. **IONIȚĂ, A.:** Punct de vedere privind managementul informației geografice în contextul informatizării administrației publice locale, Seminar, Tg. Mureș, sept. 1997.
  4. **SCHMIDT, B., STREIT, U., UHLENKUKEN, CHR.:** Visualisation Strategies and Techniques for High-Dimensional SpatioTemporal Data. În: Proc. Of JEC'97 Viena, aprilie 1997.
  5. **BULUGU, P.:** București. Ghid turistic și stradal, ed. CARTOPETRU București, 1998.
  6. \* \* \* PROFILE, winter '97/98, TAROM ROMANIAN AIR TRANSPORT.
  7. **IONIȚĂ, A. & col.:** Model experimental pentru chioșc de informare a cetățeanului dintr-un oraș, utilizând Baze de Date Geografice de Referință, TR ICI 1998.
  8. **RAWN SHAH:** Suggestions for Information Kiosk-Systems using World Wide Web. În: The World Wide Web Information Kiosk Special Interest Group, 1994.

## Bibliografie

1. **IONIȚĂ, A. & col.:** Recomandări de integrare a bazelor de date geografice de referință, TR ICI, 1996.



ajustabili: rata de învățare  $\varepsilon$  și termenul momentum  $\alpha$ .

Rata de învățare poate varia foarte mult (de la 100 la 0.01, de exemplu). Dacă se observă o convergență lentă, rata de învățare se mărește. Dacă eroarea nu se îmbunătățește după mai multe epoci, rata de învățare se micșorează. Termenul "momentum" este introdus pentru a încorpora în actualizarea ponderilor influențe din epocile trecute. Acest termen îmbunătățește convergența algoritmului. Mai mult, folosirea acestui termen permite ca rate de învățare diferite să producă aproximativ aceiași timpi de convergență.

De asemenea, numărul nodurilor ascunse poate fi ajustat. Numărul optim de noduri ascunse diferă în funcție de problemă. În general, un număr prea mic de noduri ascunse face ca rețeaua să nu fie capabilă să învețe, iar un număr prea mare face ca rețeaua să nu poată generaliza bine.

În acest experiment, primul test pentru exemplul al doilea a fost făcut cu trei noduri ascunse. Întrucât, în multe cazuri, rețeaua nu a fost capabilă să învețe, numărul nodurilor ascunse s-a mărit de la trei la șase.

În experimentul nostru, ponderile au fost inițializate cu valori aleatoare, uniform distribuite în intervalul [-1,1]. A fost folosită rata de învățare  $\varepsilon = 0.5$  și termenul momentum  $\alpha = 0.5$ .

### 3. Evaluarea consistenței

În cadrul primului experiment, a fost ales un exemplu de sarcină de lucru, specifică unui editor de texte, executată cu ajutorul a trei variante: prima consistentă, a doua inconsistentă și a treia arbitrară. Ultimele două variante au fost create modificând specificația consistentă, realizată cu ajutorul descrierii TAG.

Specificația TAG pentru varianta consistentă este următoare:

#### DICȚIONARUL SARCINILOR SIMPLE

mută-cursorul-un-caracter-înainte

Ctrl-F {direcție=înainte, unitate=caracter}

mută-cursor-ul-un-caracter-înapoi

Ctrl-B {direcție=înapoi, unitate=caracter}

mută-cursorul-un-cuvânt-înainte

Esc-F {direcție=înainte, unitate=cuvânt}

mută-cursorul-un-cuvânt-înapoi

Esc-B {direcție=înapoi, unitate=cuvânt}

#### SCHEMELE REGULILOR

1. sarcină[direcție, unitate]

→ simbol[unitate]+literă[direcție]

2. simbol [unitate=cuvânt] → "Esc"

3. simbol [unitate=caracter] → "Ctrl"

4. literă [direcție=înainte] → "F"

5. literă [direcție=înapoi] → "B"

Prin modificarea comenzii "mută-cursorul-un-cuvânt-înapoi" se obține o variantă inconsistentă. Prin atribuirea unor valori oarecare fiecărei comenzi se obține varianta arbitrară. Maparea trăsăturilor semantice la nivel sintactic și lexical este descrisă în tabelul de mai jos.

Tabelul 1 Seturile de comenzi pentru sarcina de lucru

Trăsături semantice		(a) consistent		(b) inconsistent		(c) arbitrar	
înainte	caracter	Ctrl	F	Ctrl	F	Ctrl	A
înainte	cuvânt	Esc	F	Esc	F	Esc	B
înapoi	caracter	Ctrl	B	Ctrl	B	Shift	C
înapoi	cuvânt	Esc	B	Ctrl	Z	Nul	D

Experimentul va testa modul în care rețeaua învață maparea trăsăturilor semantice pe simboluri lexicale. Se observă din tabel că sunt 4 intrări posibile (înainte, înapoi, caracter și cuvânt) și 10 ieșiri (Ctrl, Esc, Shift, Nul, A, B, C, D, F și Z).

Nu contează, în acest experiment, ordinea mapării celor două elemente de lexic. De asemenea, nu au importanță valorile luate de către ieșiri, chiar dacă, într-un model mental al utilizatorului, acestea au o semnificație suplimentară, cum ar fi, de exemplu, caracterul mutual exclusiv al trăsăturilor înainte / înapoi.

Valorile ieșirilor și ale intrărilor în rețea pentru cazul (a), limbaj consistent sunt prezentate în tabelul de mai jos.

Tabelul 2. Valorile de intrare și ieșire pentru cazul (a) – limbaj consistent

Comanda	Intrări				Ieșiri									
	fw	Bw	c	w	Ctrl	Esc	Shf	nul	F	B	Z	A	C	D
C fw	1	0	1	0	1	0	0	0	1	0	0	0	0	0
W fw	1	0	0	1	0	1	0	0	1	0	0	0	0	0
C bw	0	1	1	0	1	0	0	0	0	1	0	0	0	0
W bw	0	1	0	1	0	1	0	0	0	1	0	0	0	0

Rețeaua, având 4 noduri de intrare, 3 noduri ascunse și 9 noduri de ieșire a fost antrenată pentru învățarea celor trei limbaje cu seturile de date utilizând drept criteriu de oprire cea mai mare diferență între ieșirea observată și ieșirea țintă. Acest criteriu a fost stabilit la valoarea 0.01. Eroarea medie pătratică a fost, de asemenea, calculată, dar nu a fost utilizată ca un criteriu de oprire. Efortul de învățare a fost măsurat în număr de epoci.

Având în vedere că algoritmul de învățare este stocastic, rezultatele obținute din rulări succesive sunt diferite. Din acest motiv, au fost efectuate 40 de rulări pentru fiecare caz. Rezultatele arată un grad ridicat de semnificație ( $F_{2, 120}=578.933$ ,  $p=0.01$ ,  $r=0.9526$ ). Sinteza rezultatelor este prezentată în tabelul 3.

**Tabelul 3. Rezultate medii pentru efortul de învățare**

Varianta	Efort de învățare	Raport
(a) consistent	154.225	1.0
(b) inconsistent	339.475	2.2
(c) arbitrar	443.875	2.8

Pentru același tip de experiment, Green și Doubleday [4] raportează următoarele rezultate: media efortului de învățare: 800, 2000 și 2670, obținute ca urmare a 6 rulări. Aceasta revine la un raport de 2.5 pentru inconsistent/consistent și 3.3 pentru arbitrar/consistent.

#### 4. Evaluarea ortogonalității

În cadrul celui de al doilea experiment, a fost ales un caz mai complicat, în care prima variantă este ortogonală și cea de a doua organizată.

**Tabelul 5. Valorile de intrare și ieșire pentru cazul (a) – limbaj ortogonal**

Comanda	Intrări						Ieșiri									
	m	d	f	b	c	w	meta	esc	del	→	←	C	D	W	H	
m f c	1	0	1	0	1	0	0	1	0	1	0	1	0	0	0	
m b c	1	0	0	1	1	0	0	1	0	0	1	1	0	0	0	
d f c	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	
d b c	0	1	0	1	1	0	0	0	1	0	1	1	0	0	0	
m f w	1	0	1	0	0	1	0	1	0	1	0	0	0	1	0	
m b w	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	
d f w	0	1	1	0	0	1	0	0	1	1	0	0	0	1	0	
d b w	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	

**Tabelul 4. Seturile de comenzi pentru sarcina de lucru**

Comanda	(a) ortogonal			(b) organizat	
	esc	→	C	meta	C
mută caracter înainte	esc	→	C	meta	C
mută caracter înapoi	esc	←	C	esc	C
șterge caracter înainte	del	→	C	meta	D
șterge caracter înapoi	del	←	C	esc	D
mută cuvânt înainte	esc	→	W	meta	W
mută cuvânt înapoi	esc	←	W	esc	W
șterge cuvânt înainte	del	→	W	meta	H
șterge cuvânt înapoi	del	←	W	esc	H

Ortogonalitatea este un caz particular de consistență, în care trăsăturile semantice sunt mapate pe simboluri lexicale independente (ortogonale). Distincția a fost făcută de Bowden et al [2], care au considerat că numai definirea pe baza trăsăturilor semantice din TAG nu conduce decât la o consistență parțială, pe care ei au denumit-o organizare.

Maparea trăsăturilor semantice, la nivel sintactic și lexical, este descrisă în tabelul de mai sus.

Se observă din tabel că sunt 6 intrări posibile (mută, șterge, înainte, înapoi, caracter și cuvânt) și 9 ieșiri (meta, esc, del, →, ←, C, D, W, H). Experimentul a utilizat o rețea cu 6 noduri de intrare, 6 noduri ascunse și 9 noduri de ieșire.

Valorile ieșirilor și intrărilor în rețea pentru cazul (a), limbaj ortogonal sunt prezentate în tabelul de mai jos.



Au fost efectuate 30 de rulări pentru fiecare caz. Rezultatele arată un grad ridicat de semnificație ( $F_{1,60}=48.467$ ,  $p=0.01$ ,  $r=0.914$ ). Sinteza rezultatelor este prezentată în Tabelul 6.

**Tabelul 6. Rezultate medii pentru efortul de învățare**

Varianta	Efort de învățare	Raport
(a) ortogonal	78.900	1.0
(b) consistent	330.167	4.1

Pentru același tip de experiment, cu o rețea cu 20 de noduri de intrare, 20 de noduri de ieșire și 20 de noduri ascunse, Green și Doubleday [4] raportează următoarele rezultate: media efortului de învățare: 3533 și 5433, obținute ca urmare a 6 rulări.

## 5. Discuție

În continuare, se poate face și o analiză comparativă între cele două variante din cel de al doilea caz pe baza specificației TAG.

Specificația TAG pentru varianta ortogonală este următoarea

### DICȚIONARUL SARCINILOR SIMPLE

mută-cursorul-un-caracter-înainte

esc→C { direcție=înainte, operație=mută, unitate=car }

mută-cursor ul-un-caracter-înapoi

esc←C { direcție=înapoi, operație=mută, unitate=car }

mută-cursorul-un-cuvânt-înainte

esc→W { direcție=înainte, operație=mută, unitate=cuv }

mută-cursorul-un cuvânt-înapoi

esc←W { direcție=înapoi, operație=mută, unitate=cuv }

șterge -un-caracter-înainte

del→C { direcție=înainte, operație=șterge, unitate=car }

șterge -un-caracter-înapoi

del←C { direcție=înapoi, operație=șterge, unitate=car }

șterge -un-cuvânt-înainte

del→W { direcție=înainte, operație=șterge, unitate=cuv }

șterge -un cuvânt-înapoi

del←W { direcție=înapoi, operație=șterge, unitate=cuv }

### SCHEMELE REGULILOR

1. sarcină[operație, direcție, unitate]  
→ simbol[operație]+simbol[direcție]+literă[unitate]
2. simbol [operație=mută] →“esc”
3. simbol[operație=șterge] →“del”
4. \*simbol [direcție=înainte] →“→”
5. \*simbol[direcție=înapoi] →“←”
6. literă [unitate caracter] →“C”
7. literă [unitate cuvânt] →“W”

În această specificație, regulile 4 și 5 nu sunt luate în calcul pentru estimarea ușurinței în învățare, având în vedere caracterul intuitiv al simbolurilor utilizate (vezi, pentru o discuție mai cuprinzătoare, Payne și Green [1]).

Specificația TAG pentru varianta consistentă este următoare:

### DICȚIONARUL SARCINILOR SIMPLE

mută-cursorul-un-caracter-înainte

meta-C { direcție=înainte, operație=mută car }

mută-cursor ul-un-caracter-înapoi

esc-C { direcție=înapoi, operație=mută car }

mută-cursorul-un-cuvânt-înainte

meta-W { direcție=înainte, operație=mută cuv }

mută-cursorul-un cuvânt-înapoi

esc-W { direcție=înapoi, operație=mută cuv }

șterge -un-caracter-înainte

meta-D { direcție=înainte, operație=șterge car }

șterge -un-caracter-înapoi

esc-D { direcție=înapoi, operație=șterge car }

șterge -un-cuvânt-înainte

meta-H { direcție=înainte, operație=șterge cuv }

șterge -un cuvânt-înapoi

esc-H { direcție=înapoi, operație=șterge cuv }

### SCHEMELE REGULILOR

1. sarcină[direcție, unitate]  
→ simbol[direcție]+literă[operație]
2. simbol [direcție=înainte] →“meta”
3. simbol[direcție=înapoi] →“esc”
4. literă [operație=mută car] →“C”

5. literă [operație=mută cuv] →“W”
6. literă [operație=șterge car] →“D”
7. literă [operație=șterge cuv] →“H”

Comparația dintre cele două specificații arată că limbajul ortogonal este mai ușor de învățat, întrucât necesită o schemă și 4 reguli, față de cel consistent, care necesită o schemă și 6 reguli. Trebuie, totuși, remarcat faptul că se face apel la o subtilitate de estimare, care este de natură subiectivă.

## 6. Concluzii

Concluziile care se desprind din acest experiment confirmă ipoteza lui Green și Doubleday [4], conform căreia efectul consistenței se află localizat mai mult în componenta de recunoaștere a patern-urilor decât în cea de învățare din arhitectura cognitivă a utilizatorului. În același timp, experimentul confirmă rezultatele obținute prin teste, utilizând subiecți umani, care atestă efectul pozitiv pe care îl are consistența limbajului asupra efortului de învățare.

Comparația dintre cele două specificații (ortogonală și consistentă) arată o limitare a tehnicii TAG în estimarea efortului de învățare pe baza numărului de reguli. În același timp, este un argument în plus că rețelele neurale pot furniza o modalitate de evaluare mai eficientă și mai puțin costisitoare a efectelor pozitive pe care le are regularitatea (consistența) limbajului asupra efortului cognitiv.

## Bibliografie

1. **PAYNE, S.J., T.R.G.GREEN:** Task action grammars: a model of the mental representation of task languages. În: Human-Computer Interaction, 2, 1986, pp. 93-133.
2. **BOWDEN, E.M., SOUGLAS, S.A., STANDFORD, C.A.:** Testing the principle of orthogonality in language design. În: Human Computer Interaction, 4(2), 1989, pp. 95-120.
3. **PLUNKETT, K., MARCHMAN, V.:** U-shaped learning and consistency effects in a multi-layered perceptron: implication for child language acquisition. În: Cognition, 38, 1991, pp. 43-102.
4. **GREEN, T.R.G., DOUBLEDAY, A.:** Connectionist modelling of consistency effects in task action languages. În: Oppermann, R., Bagnara, R., Benyon, D. (Eds.) Proceedings of ECCE-7, 1994.
5. **RUMELHART, D., MCCLELLAND, J.,** PDP Research Group: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 1, MIT Press, Cambridge, 1986.