

# LIMBAJUL CYC (CYCL) ȘI LIMBAJUL NATURAL

Mironela Pîrnău

mironela@yahoo.com

Universitatea creștină „Dimitrie Cantemir”, București

**Rezumat:** Informațiile computerizate sunt stocate în multe forme, incluzând datele structurate (baze de date), semistructurate (foi de calcul, pagini WEB) și nestructurate (fișiere de tip text sau câmpuri de text). Cyc poate transforma aceste informații în cunoștințe utile. Cyc tratează fiecare înregistrare într-o bază de date ca și cum ar fi fost o declarație implicită, făcută într-o bază de date. Aceste declarații sunt apoi disponibile în timpul procesului de deducție. În mod similar, câmpurile de text pot fi citite folosind procesorul limbajului natural pentru a vedea dacă ele conțin declarații utile. Uneori declarațiile descriu despre ce este vorba în „text”. Cyc poate folosi aceste informații pentru a localiza și raporta orice informație pe care utilizatorul ar putea să o folosească pentru a răspunde la un anumit interogatoriu.

**Cuvinte cheie:** limbajul CycL, baze de date, set de instrucțiuni, formule.

## 1. Introducere în limbajul CycL

CycL (limbajul reprezentativ Cyc) este o modalitate de reprezentare largă și extraordinar de flexibilă a limbajului. Este, în mod esențial, o extensie a analizei predicatelor (FOPC—first order predicate calculus) și se poate ocupa și cu egalitățile, deducțiile, skolemizarea și unele caracteristici de rang secundar. Procesarea limbajului natural (Cyc NL) este una din cele mai studiate și mai mari provocări ale ingineriei software. Multe echipe au încercat să producă sisteme bazate pe limbajul natural, capabile să citească și să înțeleagă un simplu text. Sistemul Cyc NL are trei componente: lexiconul, analizatorul sintactic gramatical și interpretul semantic ceea ce face posibil ca Cyc să poată analiza negații și cuantificatori datorită interfețelor care permit oamenilor să facă deducții și să interogheze Cyc folosind limbajul natural în locul limbajului CycL.

### 1.1. Semantic Integration Bus

Informațiile computerizate sunt stocate în multe forme, incluzând datele care sunt structurate (baze de date), semistructurate (foi de calcul, pagini WEB) și nestructurate (fișiere de tip text sau câmpuri de text). Cyc poate transforma aceste informații în cunoștințe utile. Cyc tratează fiecare înregistrare într-o bază de date ca și cum ar fi fost o declarație implicită, făcută într-o bază de date. Aceste declarații sunt apoi disponibile în timpul procesului de deducție. În mod similar, câmpurile de text pot fi citite folosind procesorul limbajului natural pentru a vedea dacă ele conțin declarații utile. Uneori, declarațiile descriu despre ce este vorba în „text”. Cyc poate folosi aceste informații pentru a localiza și raporta orice informație pe care utilizatorul ar putea să o folosească pentru a răspunde la un anumit interogatoriu.

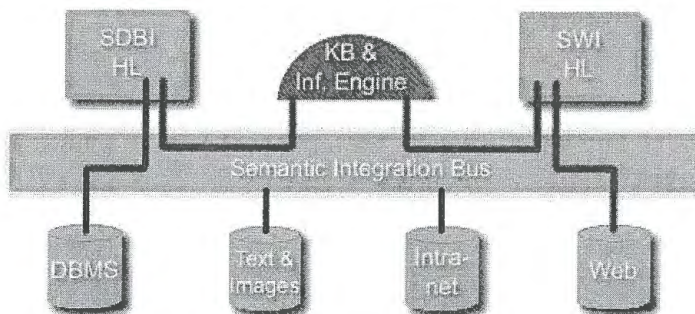


Figura 1. Semantic Integration Bus

În diagrama de mai sus, informațiile stocate într-o bază de date sau pe web sunt făcute disponibile motorului de inferențe ca declarații virtuale. Aceste seturi de declarații virtuale sunt administrate de module de niveluri euristice (heuristic level – HL). De exemplu, motorul de inferențe (inference engine –

Inf. Engine) emite o transmisie pe Bus. Un modul HL recunoaște că cererea are nevoie de o declarație care este stocată în spațiul de cunoștințe virtual. Modulul HL interceptează comanda, comunică cu baza de date (database – DB ), site-ul web sau altă sursă de cunoștințe, și întoarce legăturile la motorul de inferențe. Se continuă procesul de deducție, combinând informații din surse multiple.

## 2. Setul de instrumente Cyc

Sistemul Cyc include, de asemenea, o varietate de instrumente de interfață, care permit utilizatorului să caute, să editeze sau să extindă baza de cunoștințe Cyc pentru a pune întrebări motorului de inferențe, și pentru a interacționa cu limbajul natural și modulele de integrare a bazelor de date. Cel mai comun și mai des utilizat instrument este browser-ul HTML, care permite utilizatorului să vizualizeze baza de cunoștințe în mod hypertext. Paginile HTML, care descriu termenii Cyc, sunt generate din mers de sistemul Cyc. Fiecare pagină descrie un termen Cyc afișând toate declarațiile în care este implicat termenul respectiv, organizat potrivit unei scheme standard. Fiecare utilizare a unui termen cyc este un link HTML la o pagină HTML care descrie acel termen, astfel încât navigarea în baza de cunoștințe este ușoară urmărind o rețea de relații. Browser-ul HTML, de asemenea, include facilități pentru căutarea și editarea bazei de cunoștințe și pentru punerea unei întrebări motorului de inferențe. Alte instrumente de interfață HTML includ:

- un browser ierarhic care afișează orice subramură a unei structuri arborescente Cyc în format outline;
- un editor de lexicon, care asigură o modalitate de utilizare prietenoasă de a edita și extinde lexiconul Cyc;
- un analizator semantic CycL, care permite utilizatorului să experimenteze cu facilitățile limbajului natural analizând arbitrar în șirurile de text;
- o interfață pentru baze de date, care asigură o interfață cu modulul de integrare semantică Cyc;
- un browser WordNet, care permite utilizatorilor să vizualizeze WordNet în relație cu ontologia Cyc;
- un generator care generează reguli în limbajul natural.

Majoritatea constantelor Cyc au un nume unic, cum ar fi #Masa obiectelor sau #Bill. Constantele Cyc primesc prefixul #\$. Aceste caractere sunt, uneori, omise în documente care descriu limbajul CycL și, de asemenea, pot fi omise de unele instrumente ale unor interfețe. Numele constantei care urmează după \$ trebuie să îndeplinească următoarele reguli:

- toate numele de constante Cyc trebuie să aibă o lungime de cel puțin două caractere neincluzând caracterele prefixului # \$;
- numele constantei poate include orice literă mare sau literă mică, orice cifră, și simbolurile „-”, „\_”, „?”, „!”, „&”, sau „@” nu sunt permise;
- numele constantelor Cyc țin cont de literele mici și literele mari: #Foo nu este același lucru cu #foo; cu toate acestea este interzisă de sistem distingerea a două constante doar pe baza literei mari sau mici;
- toate numele predicatelor Cyc trebuie să înceapă cu literă mică;
- toate constantele nepredicat trebuie să înceapă cu literă mare; numele constantelor nepredicat pot, de asemenea, începe cu un caracter numeric (de exemplu, #3Mcorporation); de asemenea, se pot începe constante de tip predicat cu caractere numerice, dacă cineva aduce un argument destul de convingător pentru alegerea acestei variante;
- toate constantele de tip Cyc trebuie să fie compuse din unul sau mai multe cuvinte care au înțeles logic într-o secvență, fără pauze cu excepția liniuțelor sau sublinierilor (de exemplu: #Isa și #Smașinisport); o secvență de caractere numerice poate fi considerată un cuvânt, de exemplu #Sbirouidinfățadincorpul123; cu excepțiile menționate mai sus pentru numele predicatelor, toate cuvintele (ce nu conțin caractere numerice) trebuie să înceapă cu literă mare; un acronim poate fi considerat cuvânt, dar toate caracterele sale vor fi de același fel (de exemplu: se va scrie cu litere mici dacă acronimul începe cu numele unei constante predicat și cu litere mari dacă acronimul începe altfel.); liniuțele sunt folosite pentru a despărți părți ale numelor care restricționează sau redefinesc înțelesul numelor, de exemplu #Fruct - Cuvântul.

Înțelesul unei constante în limbajul CycL este determinat de declarațiile făcute în baza de cunoștințe. În mod convențional, alegem nume pentru constantele Cyc ce au un înțeles pentru utilizator (de exemplu: #Roșu sau #Culoarearșie), dar Cyc nu înțelege aceste șiruri de caractere. Baza de cunoștințe Cyc este

constituită dintr-un număr mare de declarații. Atunci când o formulă este introdusă cu succes în baza de cunoștințe, este stocată ca una din acestea. Fiecare declarație este compusă dintr-un număr de elemente:

- o formulă;
- o microteorie;
- o valoare de adevăr;
- o direcție (sau nivel de acces);
- un sprijin.

*Formulele* sunt formulele clasice Cyc, folosite pentru a declara fapte în baza de cunoștințe Cyc. Fiecare declarație este conținută într-o microteorie. O anumită formulă poate fi declarată într-una sau mai multe microteorii; în acest caz, vor exista declarații care au acea formulă în fiecare din aceste microteorii. De unde provin informațiile pentru declarații? Acest lucru depinde de originea declarației. Dacă o declarație este adăugată la baza de cunoștințe de către motorul de inferență ca rezultat al executării unei reguli, codul motorului de inferență decide ce microteorie concluzie ar trebui adăugată și o înregistrează. Dacă o declarație este rezultatul unei persoane sau a unui program extern de declarare a unei formule în baza de cunoștințe, la acel moment declaratorul trebuie să specifice în ce microteorie trebuie introdusă formula. Unele interfețe pentru introducerea cunoștințelor nu au nevoie ca utilizatorul să specifice o microteorie pentru noi declarații, și atunci se va încerca găsirea uneia potrivite fie se va folosi `#$BaseKB` ca default.

## 2.1. Valorile de adevăr

Fiecărei declarații îi este atașată o valoare de adevăr care îi indică gradul de adevăr. Limbajul CycL conține cinci posibile valori de adevăr, din care cele mai comune sunt: „default true” și „monotonically true”.

Declarațiile care sunt „monotonically true” sunt declarate ca fiind adevărate în orice situație. În cazul unei declarații „monotonically true” cu variabile universale în formulă, dacă un obiect este găsit pentru care declarația nu este adevărată, este semnalată o eroare. În cazul unei declarații de bază, care este „monotonically true”, dacă negația acelei formule este declarată în timpul inferării (în cadrul aceleiași microteorii) este semnalată o eroare. Declarațiile care sunt „default true”, în mod contrastant sunt adevărate în majoritatea cazurilor și pot fi rescrise. Dacă negația unei declarații de bază este declarată în aceeași microteorie în timpul inferenței nu este semnalată nici o eroare.

## 2.2. Direcțiile (nivelurile de acces)

Direcția este o valoare asociată cu fiecare declarație care determină când să fie executat procesul de deducție care implică declarația respectivă. Sunt trei valori posibile pentru direcții: înainte, înapoi și cod. Procesul de deducție care implică declarații cu direcția înainte este executat la momentul declarării (adică atunci când este adăugată o nouă declarație bazei de cunoștințe), în timp ce inferențierea pentru declarații cu direcția înapoi este amânată până când se produce un interogatoriu care permite inferențierea înapoi. Modulele HL au fost scrise pentru a suplini nevoia de inferare folosind declarațiile. Declarațiile cod nu pot fi editate prin interfața HTML. O modalitate de a vizualiza direcțiile ca o ierarhie este:

- :cod;
- variabilele declarate :cod sunt suportate direct de modulele HL;
- :înainte (forward);
- variabilele declarate înainte sunt folosite în procesul de deducție înainte. Toate declarațiile :cod sunt folosite implicit la acest nivel pentru că modulele HL care le suplinesc sunt folosite aici;
- :înapoi (backward);
- variabilele declarate înapoi sunt folosite numai în inferarea înapoi; toate declarațiile :înainte sunt, de asemenea, folosite la acest nivel și modulele HL suplinind declarațiile :cod sunt din nou folosite la acest nivel.

Fiecărei declarații îi este atașat un „sprijin”, care constă din una sau mai multe justificări care formează sprijinul pentru prezența declarației în baza de cunoștințe. În multe cazuri, cel puțin una din justificările sprijinite este locală, indicând că declarația a fost adăugată la baza de cunoștințe dintr-o sursă exterioară (de cele mai multe ori de un factor uman). În alte cazuri, o justificare sprijinită este o sursă care indică faptul că declarația a fost inferată și care subliniază pasul final al unui argument sau strategie de

inferare, care sprijină declarația. Sprijinul este un element al declarației, care nu trebuie specificat de factorul care introduce cunoștințele în momentul introducerii lor. Este creat și updatat automat.

### 3. Concluzii

Fundamentarea limbajului Cyc a fost lungă și anevoioasă, dar a început să dea roade. Începută ca proiect în 1984, Cyc este acum o tehnologie de lucru cu aplicații în multe domenii ale lumii reale. Baza vastă de cunoștințe a Cyc permite executarea de sarcini care sunt cu mult peste capacitățile altor tehnologii software.

### Bibliografie

1. **MARCUS, M.P.:** A Theory of Syntactic Recognition for Natural Language, Cambridge, 1980.
2. **MCGRAW, K.L., K. HARBISON-BROGGS:** Knowledge Acquisition: Principles and Guidelines, Prentice Hall, Englewood-Cliffs, NJ., 1989.
3. **PARTRIDGE, D.:** Engineering Artificial Intelligence Software, Intellect, Oxford, 1992.
4. **SCHUMACHER, M.:** Objective Coordination in Multi-Agent System Engineering, Design and Implementation, Lecture Notes in Computer Science, , Springer - Verlag, 2001, Vol. 2039.
5. **TEODORESCU, H.N., D. MLYNEK, A. KENDEL, H. J. ZIMMERMANN:** Intelligent Systems and Interfaces, Kluwer Academic Publishers, Boston, 2000.
6. [6] <http://www.cyc.com>

**REVISTA ROMÂNĂ DE INFORMATICĂ ȘI AUTOMATICĂ**  
**În atenția colaboratorilor**

Articolele trimise la redacția Revistei Române de Informatică și Automatică (RRIA) spre publicare trebuie să reprezinte lucrări primare, nepublicate anterior. În caz de acceptare, ele nu pot fi republicate fără acordul redacției RRIA. Autorii trebuie să respecte următoarele reguli:

- ◇ articolul trebuie să fie însoțit de un referat întocmit de unul din membrii Colegiului de redacție: în caz de acceptare spre publicare, articolul nu poate fi republicat fără acordul redacției;
- ◇ prima pagină a articolului trebuie să conțină titlul cu majuscule al articolului, numele autorului, instituția unde lucrează și adresa de e-mail dacă este cazul;
- ◇ la începutul articolului, se va prezenta un rezumat de 10-20 de rânduri în limba română, urmat de cuvinte cheie;
- ◇ textul articolului, trimis în două exemplare, se acceptă dactilografiat la două rânduri și pe dischete de 5 inch, introdus cu un procesor de texte (Word 6.0, 7.0. ) și i se vor anexa următoarele:
- ◇ lista de simboluri folosită în text: ea nu va fi publicată, dar va servi la identificarea și la reproducerea corectă a simbolurilor folosite de autor;
- ◇ figurile executate în Word, într-un fișier separat de articol, folosind ca poziție "Move Object with Text" (sau grup), vor fi numerotate cu cifre arabe și vor purta fiecare câte o explicație: poziția fiecărei figuri se va preciza de autor în cuprinsul articolului;
- ◇ tabelele executate pe calculator, într-un fișier separat de articol, puse în frame-uri, vor fi numerotate cu cifre romane: în cazul unor tabele de dimensiuni mari, se poate prezenta conținutul lor pe scurt în textul articolului sau în anexe, precizându-se că pot fi solicitate direct autorului;
- ◇ formulele matematice se vor numerota la dreapta, astfel: (1), (2)...
- ◇ referințele bibliografice se scriu la sfârșitul articolului și se citează în text cu numere de ordine, separate prin paranteze drepte: numele autorilor și titlurile lucrărilor citate se reproduc în întregime, iar titlurile revistelor se pot prescurta, de exemplu:

1. **HAIMES, S.Y.:** Multiple-criteria Decision Making. În: IEEE Trans. on Syst. Man & Cybern., SMC-25, nr. 3, 1985, pp. 25-28.

- ◇ pentru cărțile citate, se trece autorul, titlul, editura, localitatea, anul apariției, numărul de pagini:

1. **MICHALEWICZ, Z.:** Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, New York, 1994, 120p.

- ◇ mențiuni precum „Comunicare personală” sau „Lucrare republicată” se acceptă în text, dar nu și la Referințe bibliografice:

Autorul vor beneficia de un exemplar al revistei în care va apărea articolul și 15 copii ale articolului publicat.

**CUPON DE ABONAMENT**

Nr. de înregistrare \_\_\_\_\_ Data \_\_\_\_\_

Firma (persoana) \_\_\_\_\_

Localitatea \_\_\_\_\_ Cod poștal \_\_\_\_\_

Str. \_\_\_\_\_ Nr. \_\_\_\_\_

Telefon \_\_\_\_\_

Titlul publicației: **REVISTA ROMÂNĂ DE INFORMATICĂ ȘI AUTOMATICĂ**

Cost ab./an: **600.000 lei + 9% TVA** conform Legii 343/2006, Art. 140, alineat (2), lit. b, pentru modificarea și completarea Legii 571/2003 privind Codul fiscal.

Nr. abonamente \_\_\_\_\_

Plata a fost efectuată prin dispoziție de plată nr. \_\_\_\_\_ din \_\_\_\_\_

sau mandat poștal nr. \_\_\_\_\_ din \_\_\_\_\_

Plata se va face anticipat, prin livrarea sumei în **contul RO22 BRDE 445 SV 12240144450** al Institutului Național de Cercetare - Dezvoltare în Informatică - ICI, București.

Formularul completat se va expedia pe adresa:

**INSTITUTUL NAȚIONAL DE CERCETARE-DEZVOLTARE ÎN INFORMATICĂ - ICI**

Compartimentul *Ediția ICI*

Bd. M-șal. Al. Averescu, nr. 8-10, sector 1, cod 011455, București,

telefon: +40-1-316.37.78 (director), 316.07.36, 316.12.59/289, fax: 40-021-316.05.39, telex: 11.891 icpci r  
adresa de e-mail: [haiduc@u3ici.ro](mailto:haiduc@u3ici.ro) sau [www.ici.ro/revista/rria.html](http://www.ici.ro/revista/rria.html)

Vă mulțumim pentru colaborare!