Mecanisme Formale pentru Specificarea Limbajelor.

Cum să citesc această carte?

Republicarea acestei cărți, după 40 ani de la prima apariție, are drept scop, printre altele, să pună în evidență reziliența procesului de evoluție a gândirii umane prin crearea calculatorului și a folosirii lui la rezolvarea problemelor. Acest proces are două etape: prima etapa a început odată cu folosirea calculatorului în inginerie și matematică prin dezvoltarea tehnologiei de integrare a procesului de rezolvare a problemelor umane în procesul de calcul executat de mașină. A doua etapă a început odată cu dezvoltarea conceptului de inteligență artificială, prin integrarea procesului de calcul executat de mașină, adică a tehnologiei de utilizare a calculatorului, în procesul de rezolvare a problemelor executat de creierul uman. Prima etapă este caracterizată de dezvoltarea termenului numit Information Technology, pe scurt IT. A doua etapă are drept scop dezvoltarea și fundamentarea termenului de Inteligență Artificială, pe scurt AI. În 1983, când a fost publicată prima dată această carte, IT era în fașă, și deci cartea a fost dedicată discutării mecanismelor formale care au facilitat apariția și fundamentarea termenului de IT. Acum, după 40 de ani, cartea este încă actuală, dar se cristalizează din ce în ce mai pregnant conceptul de AI. De aceea, republicarea unei ediții revăzute a acestei cărți este justificată, dar cere elemente noi care să justifice mersul curent at cercetării în domeniul tehnicii de calcul. Aceste elemente le-am introdus prin adăugarea preambulului și a capitolului 8.

Preambulul cărții arată nevoia reorientării cercetării în domeniul IT către simplificarea metodologiei de utilizare a calculatorului, astfel ca el (calculatorul) să devină un asistent al creierului uman. Aceasta se poate obține prin dezvoltarea unei metodologii de utilizare a calculatorului folosind limbajul natural al utilizatorului. Capitolul 8 al cărții arată un model de implementare a

metodologiei postulate în preambul. Pentru a arata continuitatea ideilor discutate in carte reproducem în
această ediție abstractul cărții publicate în 1983 și adăugăm capitolului 8 cu titlul: *Calculatorul ca Instrument al Creierului Uman, Un proiect cu potențial nelimitat.*

Capitolul 8 debutează cu o dezbatere care justifică nevoia dezvoltării unei noi metodologi de utilizare a calculatorului pentru mânuirea abstracțiilor în procesul de gândire umană, așa cum acest proces se pare că este executat de creierul uman. Noua metodologie de utilizare a calculatorului se bazează pe utilizarea limbajului natural al utilizatorului în procesul de rezolvare a problemelor, așa cum folosesc toate celelalte metodologii de utilizare a instrumentelor umane dezvoltate cu scopul amplificării capabilității omului in mânuirea elementelor fizice (automobilul pentru mânuirea mișcării, ochelarii pentru mânuirea imaginii, medicamentele pentru mânuirea agenților patogeni, etc.). Fiindcă toate celelalte tehnologii de utilizare a instrumentelor umane se folosesc cu ajutorul limbajului natural, și calculatorul trebuie folosit la fel. Deci trebuie dezvoltată o metodologie de folosire a calculatorului folosind limbajul natural, paralelă cu metodologia actuală, în care calculatorul este utilizat folosind limbajele de programare. Folosirea limbajelor de programare se datorește faptului că limbajele naturale sunt ambigue și deci nu pot fi translate corect in limbajul mașinii.

Observația noastră este că ambiguitatea limbajului natural este transparentă pentru utilizatorul calculatorului. Aceasta fiindcă utilizatorul calculatorului folosește un limbaj al domeniului în care el utilizează calculatorul. Acest limbaj l-am numit Limbajul Algoritmic al Domeniului, pe scurt DAL. Manipularea termenilor domeniului cere de asemenea o reprezentare formală a cunoștințelor domeniului. Aceasta se obține prin reprezentarea domeniului de utilizare al calculatorului folosind o ontologie a domeniului, unde cunoștințele sunt memorate sub forma de perechi (Termen, Semnificație) unde Termen este termenul din DAL (limbajul natural folosit) iar Semnificație este un Web Service care reprezintă cunoștințele denumite de Termen. Astfel, domeniul problemei se poate păstra într-un fișier la care are acces deopotrivă calculatorul și utilizatorul său.

Procesul de reprezentare a domeniului de aplicație al calculatorului sub forma unei ontologii este de fapt un proces de formalizare a domeniului discutat în capitolul 8 sub numele de Emanciparea Computațională a Domeniului. De aceea, apare nevoia unei mașini virtuale a cărei memorie este ontologia domeniului iar instrucțiunile sunt Web Services. Creierul poate mânui această mașină executând algoritmi pe internet, astfel calculatorul devenind de fapt un asistent al creierului. Capitolul 8 ilustrează dezvoltarea acestei metodologii de utilizare a calculatorului, descriind implementarea ei pentru domeniul utilizării calculatorului la învățarea algebrei într-o clasă de liceu.

Întrebarea cititorului „cum să citesc această carte” are mai multe răspunsuri. Noi, având în vedere conținutul cărții și potențialul utilizării materialului prezentat, dăm următorul răspuns. Cititorul interesat simplu în utilizarea calculatorului în procesul de rezolvare a problemelor sale, ignorând miriadele de software cunoscute ca Software Tools, folosite pentru transformarea problemei și a soluției ei în program executabil, ar trebui să citească preambulul și capitolul 8.

Dacă cititorul este interesat în informatică, adică în știința rezolvări problemelor cu calculatorul, atunci el ar trebui să citească și să înțeleagă înainte de toate capitolul 1. Acest capitol discută un concept global de limbaj care este destinat a fi utilizat în timpul comunicației dintre diferiți comunicatori. După cunoștințele noastre, acest concept este unic și inedit. Meritul lui este că reprezintă o sinteză a conceptului de limbaj folosit în domeniul informaticii, și că atât limbajul natural uman cât și diferite alte limbaje artificiale, cum sunt limbajele de programare, sunt văzute în acest capitol ca și cazuri particulare ale conceptului de limbaj discutat în acest capitol. Deci conceptul de limbaj prezentat în capitolul 1 este robust si prezintă o bază solidă pentru manipularea limbajului ca obiect de calcul în diferitele aspecte ale acestei probleme.

În funcție de interesul cititorului, acesta poate continua abordând capitolul 2 sau capitolul 6. Capitolul 2 discută conceptul de gramatică ca mecanism de specificare a unui limbaj prin generarea elementelor lui, precum și Mașina Turing ca mecanism de specificare a unui limbaj prin recunoașterea elementelor lui. Algoritmic vorbind, gramatica este văzută ca un algoritm de generare a elementelor unui limbaj printr-un proces de rescriere a unui simbol dat, folosind reguli de re-scriere similare cu regulile de rescriere a propozițiilor unui limbaj natural din simbolul numit *fraza*, folosind categorii sintactice ca substantiv, verb, adjectiv, adverb, etc. Pe de altă parte, Mașina Turing apare ca algoritm de recunoaștere a stringurilor unui limbaj folosind reguli de recunoaștere care reduc aceste stringuri folosind operații simple de substituire a sub-stringurilor unui string similare cu regulile simple de calcul. Mai departe, în funcție de interesul său, cititorul poate continua cu capitolul 3 unde se discută algoritmii de analiză a unui limbaj, care au la bază gramatica care specifică limbajul respectiv. Acești algoritmi constituie fundamentul tehnologiei curente de tehnică de calcul. Datele acestor algoritmi sunt:

1. Un string de analizat.

2. Un pachet de reguli gramaticale de forma *ParteStângă* = *ParteDreaptă*, unde *ParteStângă* este un simbol numit categorie sintactică, iar *ParteDreaptă* este o combinație de categorii sintactice și stringuri date numite terminale.
3. O categorie sintactică numită axiomă sau simbol de start.

Ipoteza acestor algoritmi este că stringul de analizat este un element al limbajului specificat de către regulile de specificare date. Aceasta înseamnă că stringul de analizat se poate regenera folosind regulile gramaticale date. Procesul de regenerare constă din construirea arborelui de derivare al striungului de analizat, în care o categorie sintactică C identificată de regula de specificate C = ParteDreaptă se poate substitui cu ParteDreaptă iar substringul identificat de ParteDreaptă se poate substitui cu simbolul C. Astfel, algoritmul de construire al arborelui de derivare al stringului dat, consumă acest string secvențial, simbol cu simbol, citindul de la stânga spre dreapta sau de la dreapta spre st*â*nga, in timp ce construiește arborele lui de derivare. Această operație se poate conduce după una din două strategii: de sus in jos, (adică Top-Dopwn), incep*â*nd cu rădăcina acestui arbore marcată de axiomă, sau de jos in sus (adică Bottom-Up), incep*â*nd cu frontiera sa a cărei noduri sunt marcate de cuvintele (sau simbolurie) stringului dat. Deoarece aceste doua strategii sunt unic determinate de datele problemei, procesul de construire al arborelui de derivare al stringului dat este bine definit si stringul astfel generat este obținut prin concatenarea frunzelor acestui arbore.

Dacă cititorul este mai degrabă interesat în abordarea limbajului de analizat pe baza unor algoritmi de recunoaștere a elementelor lui de tipul Mașinii Turing atunci cititorul este sfătuit să treacă de la capitolul 2 la capitolul 4.

Capitolul 4 este dedicat discuției algoritmilor similari celor dezbătuți în capitolul 3, dar care au la bază un mecanism de calcul bazat pe Mașina Turing. Operația de baza a acestor algoritmi este procesul cunoscut sub numele de *patern matching in strings*. Acest tip de algoritmi sunt mai puțin discutați în literatură. Deci, aici considerăm că este un câmp fertil de cercetare și dezvoltare pentru studenți și cercetători. Autorul și studenții lui de la Universitatea Statului Iowa, Iowa City, USA, au inițiat o metodologie de dezvoltare a software-lui folosind algoritmi similari a căror implementare se bazează pe ”patern matching”. Avantajul acestor algoritmi este că sunt mult mai eficienți deoarece pot utiliza in paralel toate regulile din pachetul de specificare și toate simbolurile stringului dat, eliminând ipoteza operației secvențiale. Firește, aici este vorba de o cercetare originală în care regulile din pachetul care specifică limbajul sunt ordonate în clase de ierarhie, așa cum este explicat în capitolul 4. Informația pe care se bazează acești algoritmi este cunoscută sub numele de *conte*xte și *non-contexte*. *Contextel*e sunt perechi de stringuri care dacă includ ParteDreaptă a unei reguli de specificare atunci aceasta specifică ParteStângă a acestei reguli. *Non-contextele* sunt perechi de stringuri care dacă includ ParteDreaptă a unei reguli de specificare atunci aceasta nu specifică ParteStângă a acestei reguli. Capitolul 5 al cărții conține o colecție de algoritmi care se utilizează pentru calcularea *contextelor* și *non-contextelor* regulilor de specificare.

Capitolul 6 al cărții conține o metodologie originală de manipulare a abstracțiilor de calcul. Ideea fundamentală a acestei metodologii este considerarea abstracțiilor de calcul ca o

reflectare a lumii fizice. Universul fizic este considerat ca o listă deschisă de modele ierarhice, în care un model fizic se construiește luând ca baza un alt model fizic. Modelul algebric actual de manipulare a abstracțiilor nu distinge nivele ierarhice de modele. Adică, structurile algebrice actuale folosite ca modele ale abstracțiilor de calcul nu sunt ierarhizate așa ca cele fizice. Legătura dintre algebrele folosite ca modele de manipulare a abstracțiilor este o legătură funcțională, care prezervă operațiile algebrice, fără a considera reprezentarea unui model în termenii unui alt model dat. Modelele algebrelor heterogene generalizează relația funcțională dintre algebre, dar nu discută reprezentarea elementelor unei algebre heterogene ca elemente ale altei algebre heterogene, care nu aparține cu necesitate aceleiași clase de similaritate. Ori, algoritmii folosiți de diferite sisteme informatice, cum ar fi de pildă algoritmii

de traducere a unui limbaj de programare în termenii unui alt limbaj de programare, necesită o astfel de relație între abstracțiile de calcul manipulate. Metodologia de manipulare a abstracțiilor discutată în capitolul 6 al acestei cărți pune la baza relațiilor dintre modelele abstracțiilor manipulate de informatică o astfel de relație, în care abstracțiile se construiesc sistematic luând ca bază alte abstracții, după cum universul fizic este construit având ca bază un model ierarhic. De exemplu, pentru construirea unei case ca model fizic se folosesc alte modele fizice cum sunt fundația, pereții, acoperișul, etc., fiecare dintre aceste modele fiind tratat similar. Relația fundamentală care stă la baza ierarhiei modelelor fizice este abstractizarea. Aceasta înseamnă că pentru a construi un model fizic folosind alte modele fizice ca și componente, prin abstractizare se *uită*, adică se neglijează, toate proprietățile componentei folosite în afară de aceea de a fi o componentă a

modelului construit. Pentru a folosi un model fizic într-un anumit scop se face *adjuncția* proprietăților modelului de folosit în contextul în care acesta este folosit. Aceste relații de *uitare* și de *adjuncție* sunt formalizate în această carte folosind Ierarhia Algebrelor Heterogene (în original: Heterogeneous Algebraic Structures, pe scurt HAS hierarchy). În capitolele 6 și 7 ierarhia HAS este pusă la baza construcției modelelor abstracte de calcul și sunt studiate proprietățile acestor modele folosind *operațiile derivate* din algebrele heterogene, reprezentate în abstracțiile de calcul cu ajutorul *macro-operațiilor* din limbajele de programare. Acest studiu, fiind o premieră, oferă un câmp deschis atât cercetărilor teoretice, cât si aplicațiilor practice. În particular, folosirea calculatorului ca un instrument al creierului uman devine astfel baza evoluției nelimitate a software-ului în procesul evoluției umane.

Prof. Teodor Rus,
august 29, 2023

4