

**UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**REZUMATUL TEZEI DE DOCTORAT
SISTEME INTELIGENTE HIBRIDE
ÎN DIAGNOZA MEDICALĂ
HYBRID INTELLIGENT SYSTEMS
IN MEDICAL DIAGNOSIS**

Conducător științific: Prof. Dr. Ing. IOAN DUMITRACHE

Doctorand: Asist.mat.inf. SABINA MUNTEANU

2006

Mamei mele

Mulțumiri

Aș dori să mulțumesc în primul rând Lui Dumnezeu, fără de Care nimic nu este posibil, și familiei mele pentru sprijinul neprețuit pe care mi l-a acordat.

Adresez cele mai sincere mulțumiri domnului profesor Ioan Dumitrache, pentru susținerea, înaltul profesionalism și răbdarea de care a dat dovadă în îndrumarea acestei lucrări, și medicilor Lidia Drăgan și Cristina Morar care mi-au fost de mare ajutor în realizarea aplicației medicale.

Le mulțumesc tuturor prietenilor și celor care au avut încredere în mine.

Autoarea

Glosar

ALP Abductive Logic Programming: Programare Logică Abductivă;

ANFIS Adaptive Neural Fuzzy Inference System: Sistem de inferență fuzzy neural adaptiv;

(A)TMS (Assumption-based) Truth Maintenance System: Sistem de menținere a consistenței (bazat pe ipoteze);

BCP (Boolean Constraint Propagation) Propagarea Booleană a Constrângerilor;

(D)CSP (Dynamic) Constraint Satisfaction Problem: Problemă de Satisfacere a Constrângerilor;

KBANN Knowledge-Based Artificial Neural Network: Rețele neurale bazate pe cunoștințe;

NFC Neuro-Fuzzy Classifier: Clasificator neuro-fuzzy;

RBF (network) Radial Basis Function: Rețea cu funcții bază radiale;

SLD (NF) Single Linear Derivation (with Negation as Failure): Derivare singulară lineară.

Cuprins

I.Introducere	8
II.Specificul problemelor de diagnoză medicală.....	8
2.1.Etape ale procesului de diagnoză.....	8
2.2.Tipuri și surse de erori în diagnoza medicală.....	9
2.3.Concluzii și contribuții.....	11
III.Starea în domeniul diagnozei asistate de calculator.....	11
3.1. Diagnoza prin metode statistice.....	11
3.1.1.Diagnoza folosind cunoștințe asociative.....	11
3.1.1.1.Rețelele neurale și clasificarea.....	12
3.1.1.2.Algoritmii genetici și diagnoza medicală.....	14
3.2. Modele intermediare de diagnoză.....	15
3.2.1.Modele probabiliste structurate. Rețele bayesiene.....	15
3.2.1.1.Inferență probabilistă în rețele bayesiene.....	15
3.2.1.2.Bayesianism și cauzalitate.....	17
3.2.1.3.Sistemul de diagnoză medicală CASNET.....	18
3.2.2. Sisteme de inferență fuzzy.....	19
3.2.3.Modele simbolice.....	20
3.3. Diagnoza pe bază de model.....	22
3.3.1.Abducția și tehnicile bazate pe consistență	22
3.3.2.Metode de diagnoză specifice sistemelor dinamice	23
3.4.Concluzii și contribuții.....	25
IV.Sisteme inteligente hibride de diagnoză medicală.....	25
4.1.Tehnici inteligente hibride.....	25
4.2.Tehnici inteligente hibride în diagnoza medicală.....	27
4.2.1.Hibridizare neuro-simbolică	27
4.2.1.1.Rețele neurale bazate pe cunoștințe.....	27
4.2.1.2.Rețele neurale comitet.....	28
4.2.2.Hibridizare neuro-fuzzy	29
4.2.2.1.ANFIS (Adaptive Neural Fuzzy Inference Systems).....	29
4.2.2.2.Clasificatorul neuro-fuzzy NFC.....	29
4.2.2.3.Perceptronul Fuzzy Multistrat	29
4.2.2.4.Neuroni fuzzy logici	30
4.3.Un exemplu de hibridizare combinativă în diagnoza medicală.....	30
4.3.1.Sistemul CHECK.....	30
4.3.2.Comparație între sistemele CHECK și CASNET.....	32
4.4.Sistemul hibrid DiaMed.....	32

4.5.Concluzii și contribuții.....	33
V.Selectarea ipotezelor de diagnostic prin decizie fuzzy în DiaMed.....	33
5.1.Sisteme de inferență fuzzy în diagnoza medicală.....	34
5.2.Decizie fuzzy în diagnoza medicală.....	35
5.3.Evaluarea capacității de discriminare a unui test de diagnoză.....	39
5.4.Clasificare prin decizie fuzzy.....	39
5.5.Concluzii și contribuții.....	40
VI.Sisteme de argumentare directă și algoritmi de satisfacere a constrângerilor în modelarea raționamentului nemonoton.....	41
6.1.Sisteme de argumentare directă.....	42
6.2.Algoritmi DCSP.....	46
6.2.1.Backtracking dinamic pentru CSP.....	46
6.2.2.CSP pentru sisteme de argumentare.....	47
6.2.3.Algoritmi de satisfacere a constrângerilor dinamici.....	48
6.3.Concluzii și contribuții.....	49
VII.Relația abducției cu sistemele de argumentare, tehnicile de satisfacere a constrângerilor și sistemele de menținere a consistenței (TMS).....	49
7.1. Calculul abducției folosind TMS. Legătura între CSP și abducție.....	50
7.2.CSP în calculul semanticii admisibile din argumentație.....	51
7.3.Avantaje ale argumentelor asupra TMS.....	52
7.4.Concluzii și contribuții.....	53
VIII.Argumentarea deciziei de diagnostic în DiaMed.....	53
8.1.Reprezentarea cunoștințelor.....	53
8.2.Modelarea problemei în cadrul DCSP.....	56
8.3.Algorithm.....	59
8.4.Exemplu.....	61
8.5.Concluzii și contribuții.....	64
IX.Aplicații ale algoritmului DiaMed în diagnoza sistemelor dinamice	65
9.1.Decizie fuzzy sau sisteme de inferență fuzzy.....	65
9.2.Studiu comparativ a două abordări pentru sistemele complexe.....	67
9.3.Concluzii și contribuții.....	68
X.Concluzii și direcții de dezvoltare.....	68
Bibliografie selectivă.....	70
Anexa 1. Modelul medical	77

I.Introducere

Diagnoza este unul dintre aspectele cele mai importante ale problemelor complexe ale lumii reale. Începând cu diagnoza tehnică, ce trebuie să asigure productivitatea și siguranța oamenilor în mediile în care omul interacționează cu mașina, și continuând cu diagnoza medicală pentru identificarea și tratarea bolilor, avem o multitudine de situații ce relevă utilitatea problemei pe care dorim să o abordăm în cadrul acestei lucrări. Siguranța centralelor nucleare, pilotarea de avioane în medii dificile, sistemele de producție (și lista poate continua pe câteva pagini) sunt de neconceput fără un sistem de monitorizare care să depisteze rapid și precis dacă sistemul sub observație începe să se îndepărteze de comportamentul normal.

Formularea unei probleme de diagnoză. Un diagnostic este, informal, "cea mai plauzibilă combinație de ipoteze care explică toate datele prezente" (definiția abducției din [Byl91]). Această definiție are nevoie de trei elemente de bază: o noțiune de "explicație", un mecanism de generare a ipotezelor de diagnostic, și un algoritm de discriminare pe bază de testare. O caracteristică a raționamentului de diagnoză este că pornește cu un set incomplet de date inițiale, pe baza cărora elaborează tentativ direcții de investigare, care sunt focalizate cu ajutorul unui plan de testare (completarea informațiilor inițiale până când ajung la un profil recognoscibil dintr-un set de prototipuri). Pe parcursul algoritmului, o parte din ipotezele generate pot fi infirmate de noile evidențe, și pot apare ipoteze noi.

Definiția dată mai sus face problema, în cazul general, intratabilă (NP-dificilă) ([Byl91]), sursa principală a acestei intratabilități fiind interacțiunile semnificative care pot să apară între elementele unor ipoteze compuse (cum este cazul și în diagnoza medicală). De exemplu, în medicină, prezența simultană a două boli poate să mascheze prezența unor simptome altfel manifeste și semnificative pentru fiecare boală în parte și să introducă simptome specifice doar apariției combinate; o altă situație dificil de formalizat este cea a diagnosticului diferențial, în care prezența unor anumite simptome/combinații de simptome într-un context dat poate să anuleze o ipoteză plauzibilă, discriminând între ea și un diagnostic alternativ, și invalidând raționamentul ipotetic dezvoltat până în punctul respectiv. Intratabilitatea problemei este depășită cel mai adesea prin folosirea unor euristici. Acestea fie simplifică spațiul problemei căutând o soluție exactă pentru spațiul simplificat, fie se mulțumesc să ofere o soluție aproximativă pentru problema inițială, după cum se va vedea și în cadrul lucrării de față.

II.Specificul problemelor de diagnoză medicală

"Nu putem pretinde unui domeniu mai multă exactitate decât cuprinde natura sa"
(Aristotel)

2.1.Etape ale procesului de diagnoză

În diagnoza medicală nu există reguli clare și precise după care experții se ghidează, și care să poată fi traduse ușor algoritmic: raționamentul depinde în cea mai mare măsură de experiența, cunoștințele și intuiția expertului. Plecând de la un set de simptome inițiale,

medicul încearcă, după clarificarea simptomelor, reducerea bolilor la un set restrâns de posibilități (i.e. **diagnosticul clinic**). Aceasta este o operație de clasificare în care medicul trebuie să completeze petele albe sau să ignore unele simptome particulare, -adică recunoașterea unei forme puternic afectată de zgomot, în limbajul inteligenței artificiale. Rolul diagnosticului clinic este de a indica medicului ce investigații paraclinice trebuie să efectueze, pentru decizia finală. Desigur, pot exista mai multe etape de testare. Logica ce stă la baza acestui proces este o combinație între logica formală nemonotonă, și logica internă a fenomenelor (ce modelează domeniul). O etapă foarte importantă în diagnosticare constă în **argumentarea diagnosticului**. În abordarea prezentă, am apelat la elemente de teoria argumentației pentru a modela această etapă.

Întrucât domeniul abordat este o lume dinamică inaccesibilă de o complexitate încă departe de a fi înțeleasă de către noi, incertitudinea este inevitabilă, ca și erorile pe care le determină ([RusNor02]). Așa cum observa autorul în [Res88], fenomenele din patologia umană sunt complexe, neliniare, discontinue și variabile, astfel că diagnosticul este cea mai dificilă problemă în practica medicală.

2.2. Tipuri și surse de erori în diagnoza medicală

O problemă ce poate apare în practică este diagnosticarea greșită (cu o frecvență ce variază între 8% și 40 %). Cauzele erorilor pot proveni din trei direcții principale: *erorile pacientului*, (erori de comunicare - unele simptome nu sunt raportate din jenă/frică sau fiind considerate nerelevante, sau sunt raportate simptome false cu un anumit interes), *erorile testelor de laborator* (testele sunt supuse unor limitări din cauza cărora pot obține un fals negativ sau un fals pozitiv; în plus, unele teste eșuează la pacienții care suferă de o boală suplimentară, într-un proces nedorit de *mascare a simptomelor* întâlnit când este prezentă o anumită combinație de boli), și, în fine, *erorile medicului*, unde sistemele expert pot fi de ajutor întrucât aceste erori provin (dincolo de unele limitări umane cum ar fi oboseala etc.) din neluarea în considerare la timp a unor alternative de diagnostic mai rare și deci mai puțin cunoscute.

Tipurile de diagnostic greșit întâlnite în practică variază de la *diagnosticarea greșită a subtipurii* (spre exemplu, în diabet), *boală cauzatoare ignorată, ignorarea complicațiilor sau a cauzelor medicamentoase*, până la situații mai dificile, în care se *ignoră stările înrudite* (unele boli tind să apară grupat, chiar dacă nu se cauzează între ele; este vorba probabil de o cauză comună declanșatoare pentru toate, nu întotdeauna clară, întrucât informația medicală are încă multe lacune). În fine, unul dintre cele mai dificile cazuri este cel în care se *ignoră o boală nerelaționată* (cel mai probabil din pricina mascării unor simptome). În aceste condiții, rolul calculatorului poate fi definit de două componente: 1) ajută medicul să își reamintească rapid stări patologice neobișnuite și 2) sugerează ipoteze alternative și îngustează spațiul de căutare mai repede și mai bine decât expertul uman.

Există boli mai dificil de diagnosticat decât altele. Astfel, unele boli sunt *supra-diagnosticate*, în timp ce altele sunt *sub-diagnosticate* (neglijate): spre exemplu, sindromul colonului iritabil este deseori diagnosticat în locul bolii cum ar fi boala Chron sau boala Celiac, - mai rare și mai ușor neglijate. Un rol important în sub-diagnosticare îl au simptomele nespecifice, vagi, lipsa testelor definitive (care să tranșeze

foarte clar în favoarea unui anumit diagnostic), lipsa unor cunoștințe temeinice asupra bolii și a mecanismelor sale, sau apariția simptomelor abia în stadii foarte avansate ale bolii (cazul “ucigașilor tăcuți”, cum ar fi glaucomul, hipotirodismul, hipertensiunea, osteoporoza, cancerul de plămâni, atacul de cord ș. a.) Între bolile dificil de diagnosticat se deosebesc câteva grupe cunoscute: *bolile mintale, emoționale și de comportament* (aici dificultatea provine din lipsa unor criterii foarte clare și universal valabile), *bolile digestive*: simptomele sunt vagi și accesul în diferite puncte ale tractului digestiv este anevoios, *bolile cu simptome vagi* ca: stare de rău, oboseală (de exemplu: fibromialgia, lupusul, boala Lyme, scleroza multiplă etc.) și *bolile rare* (cu care medicii nu sunt familiarizați).

Am menționat mai sus ca sursă de dificultate a diagnosticului faptul că este o problemă dinamică: acesta se construiește treptat, pe măsura descoperirii și prelucrării informațiilor descoperite. Cauzele principale pentru care informațiile nu duc automat la diagnostic ar fi [Res88]:

- Există puține semne patognomonice (semne evidente care să sugereze în special o anumită boală), și “diagnosticul rezultă din **modul în care se combină diferitele semne și simptome** mai puțin patognomonice” [Rîm83] (discontinuitatea este o consecință a acestei situații; diferențe mici în mulțimea de simptome pot duce la diferențe semnificative de diagnostic- adică boli neînrudite);
- Majoritatea semnelor și simptomelor pot apare în foarte multe boli (idem nota precedentă);
- Bolnavul nu expune spontan toate semnele, acestea fiind descoperite treptat de medic (dinamism);
- Una și aceeași boală se prezintă individualizat la fiecare pacient, în funcție de terenul întâlnit (variabilitate).

Prezentăm în continuare, mai în detaliu, câteva exemple clasificate de situații dificile.

Boli mascate. În această situație, manifestările clinice ale unor boli pot să fie atât de atipice încât să se confunde cu tabloul clinic al altor boli. Spre exemplu, în unele cazuri, infarctul miocardic poate evolua sub forma unor dispepsii (greață, balonare, diaree), din cauza dereglărilor circulatorii ce produc congestii asupra unor organe din cavitatea abdominală. De asemenea, ulcerul gastro-duodenal poate evolua sub forma unei pancreatite cronice, iar cancerul bronho-pulmonar sub masca unei pneumonii, sau a unei hipertirodii, datorită secreției de hormon tireo-stimulant de către celulele canceroase. Unele forme de cancer bronho-pulmonar pot evolua ca sindroame nefrotice, iar cancerul pancreatic poate evolua sub masca unei litiaze biliare sau a unui diabet, și acestea sunt doar câteva exemple. Soluția în acest caz stă în găsirea corectă a cauzei bolii.

Boli asociate. Este situația în care o boală apare însoțită de complicațiile sale. Multe din “măștile” menționate mai sus, sunt, de fapt, boli autentice, spre exemplu: astmul însoțit de consecința sa -emfizemul, retinopatia și glomeruloscleroza ce complică diabetul, infecția urinară și insuficiența renală ce pot apare în litiaza renală, sau pancreatita acută din cadrul litiazei biliare.

Boli concomitente. Acestea sunt boli simultane ce pot evolua independent, fără legături patogenice importante, și situația este una dintre cele mai confuze.

2.3. Concluzii și contribuții

Capitolul propune o imagine de ansamblu a procesului de diagnoză medicală. Principala sa contribuție stă în faptul că, relevând o parte din dificultățile acestei diagnoze, oferă o justificare teoretică a necesității și adecvării abordărilor pe care le vom folosi pentru această problemă în cadrul lucrării de față.

III. Starea în domeniul diagnozei asistate de calculator

3.1. Diagnoza prin metode statistice

Metodele statistice aproximează inferența logică pe cunoștințe formale, prin asocieri superficiale cauze-efecte, furnizând soluții aproximative. În general sunt eficiente, dar au slabe capacități de explicare a concluziilor.

3.1.1. Diagnoza folosind cunoștințe asociative

Diagnoza poate fi privită în termenii clasificării de forme (clasificarea este operația elementară care stă la baza diagnozei). Clasificarea prin metode metrice presupune construirea unei mulțimi de *funcții de discriminare* [DudHarSto00]: $g_i(x)$, $i=1, \dots, c$, iar clasificatorul atribuie vectorul de caracteristici x clasei ω_i dacă $g_i(x) > g_j(x)$ pentru toți $j \neq i$. De obicei aceste funcții sunt construite pe bază de probabilități condiționale sau măsuri fuzzy- ținând cont de incertitudinile ce apar frecvent într-o problemă de clasificare. Funcțiile de discriminare definesc regiunile de decizie (granițele de decizie) asociate diferitelor categorii.

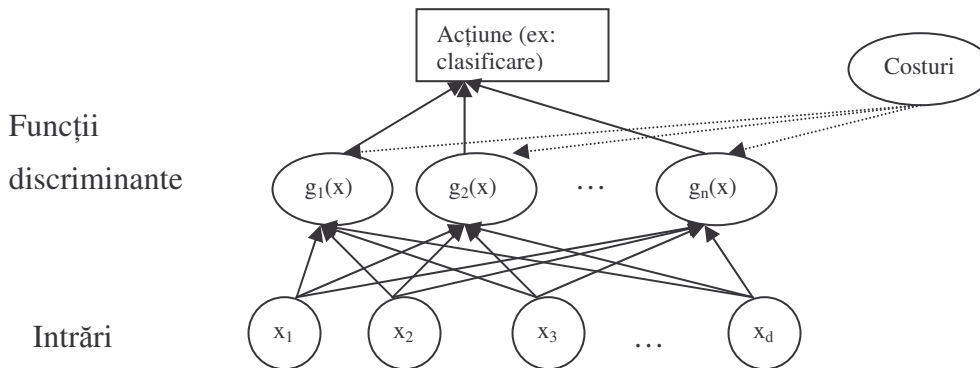


Fig. 3.1. Schema generală a unei clasificări

Abordările statistice care urmează pot fi toate privite din acest punct de vedere. Deși diagnoza pe bază de model logico-simbolic (Secțiunea 3.3.) poate trata probleme complexe, nelineare, variabile în timp, nu există încă o abordare unificată satisfăcătoare. S-a ajuns astfel la *tehnicele inteligente*: acestea cuprind modele conexiuniste structurate (rețelele bayesiene sau rețelele neurale fuzzy) sau nestructurate (rețelele neurale

“clasice” = de tip cutie neagră), modele logico-simbolice pe bază de reguli, cum sunt sistemele expert (acestea, deși în principiu pot conține reguli ce descriu comportamentul cauzal de profunzime, sunt de obicei folosite pentru a reprezenta euristici ce condensează expertiza umană în cunoștințe superficiale). Sistemele de inferență fuzzy sunt tot sisteme pe bază de reguli, dar au un mecanism de inferență propriu, specializat pentru lucrul cu informații imprecise și incerte, ceea ce le conferă un statut aparte. Rețelele neurale sunt un exemplu tipic de raționament inductiv. Ele construiesc modele aproximative “ascunse” și distribuite în ponderile rețelei, pornind de la mulțimi de exemple asociate cu domeniul de interes. Rețelele bayesiene, cauzale și posibiliste sintetizează într-un model compact inferențe statistice.

Un avantaj în plus al tehnicilor inteligente stă în faptul că acestea și-au dovedit deja atât limitările, avantajele, cât și posibilitățile de traducere/comunicare de la un formalism la altul și pot fi combinate în sisteme hibride în cadrul cărora cooperează pentru a îmbunătăți performanțele sistemului (câteva exemple vor fi prezentate spre sfârșitul secțiunii.)

Așadar, tehnicile inteligente prezentate în continuare, deși nu se bucură de claritatea și expresivitatea diagnozei pe bază de model, pot trata eficient probleme complexe, nelineare, variabile în timp, unul dintre atuurile lor principale fiind faptul că lucrează, de obicei, cu aproximări ale domeniului (cunoștințe de suprafață), aproximări care simplifică mult inferența, dar în același timp fac necesare mecanismele de tratare a incertitudinii și impreciziei.

3.1.1.1.Rețelele neurale și clasificarea. Una dintre cele mai cunoscute și mai clasice metode de clasificare este regula lui Bayes, despre care s-a demonstrat că minimizează probabilitatea medie a erorii de clasificare [DudHarSto00]. Dezavantajele sale principale sunt: necesitatea cunoașterii apriorice a probabilităților (acestea pot fi însă estimate din date) și restrictivitatea ipotezei de independență între clase.

Primul dezavantaj este înlăturat în rețelele neurale, modelele nestructurate ce realizează regresie nelineară, multe arhitecturi fiind recunoscute ca aproximatori universali. Dacă mulțimea de exemple este reprezentativă pentru domeniul ales, rețeaua are bune capacități de generalizare în spațiul problemei. Unul din avantajele principale ale rețelelor neurale este capacitatea de învățare, ele ilustrând foarte bine inferența *inductivă*. Rețelele sintetizează modelul unui domeniu din exemplele de învățare, și, odată cu modelul, construiesc implicit și algoritmul asociat problemei considerate (fiind astfel utile în situațiile în care algoritmi specializați nu există). Aceste cunoștințe (model + algoritm) sunt reprezentate distribuit în ponderile numerice asociate conexiunilor sinaptice. Fiind o situație particulară de regresie nelineară, ele sunt, în plus, foarte tolerante la zgomotul din datele de intrare.

Rețelele folosesc, așadar, cunoștințe superficiale (de suprafață), pentru un raționament bazat pe asocieri (nu pe un model cauzal de profunzime). Prin urmare, prețul plătit pentru avantajele de mai sus stă în lipsa facilităților de explicare a rezultatului obținut (element critic pentru diagnoza medicală). Acest dezavantaj a fost parțial ameliorat de unele arhitecturi moderne de rețele neurale, dintre care câteva vor fi prezentate în continuare. Un alt dezavantaj al rețelelor neurale, privite prin prisma diagnozei medicale, este faptul

că în cazul sistemelor complexe, ce necesită rețele neurale mari, apar dificultăți în antrenare datorate minimelor locale.

Perceptronul multistrat. Este una dintre cele mai cunoscute și utilizate arhitecturi feedforward, și folosește învățare supervizată de tipul corectare a erorii [Hay98]. Unul din aspectele interesante ale algoritmului de antrenare (backpropagation) este faptul că acesta calculează de fapt probabilitățile condiționale a posteriori și se poate aplica în final regula de decizie Bayes, rețeaua folosind pentru clasificare funcții discriminant implicit construite, și constituind un exemplu particular al aplicării teoriei deciziei în recunoașterea formelor.

Exemplu de clasificare cu ajutorul unui perceptron multistrat. Un dezavantaj al rețelelor neurale, privite prin prisma diagnozei medicale, este dat de discontinuitatea cunoștințelor medicale. Cazuri apropiate ca tablou al simptomelor pot face parte din clase diferite, dar este foarte probabil că o rețea neurală le va confunda. În plus, sistemele complexe, de dimensiuni mari sunt o problemă pentru o rețea neurală, din cauza minimelor locale. Am ales totuși o pereche de boli (*ciroza* și *insuficiența cardiacă*) și am încercat să le clasificăm cu ajutorul unui perceptron multistrat. S-au putut urmări avantajele învățării din exemple, rapidității calculului paralel la partea de recunoaștere, și robusteții la forme afectate de zgomot. Între dezavantaje, în afară de lipsa facilităților de explicare, s-au adăugat restricțiile prohibitive impuse de necesarul de memorie și de timp de antrenare, ca și lipsa datelor suficiente și semnificative pentru a lucra cu toate cele 30 de boli considerate în aplicația finală.

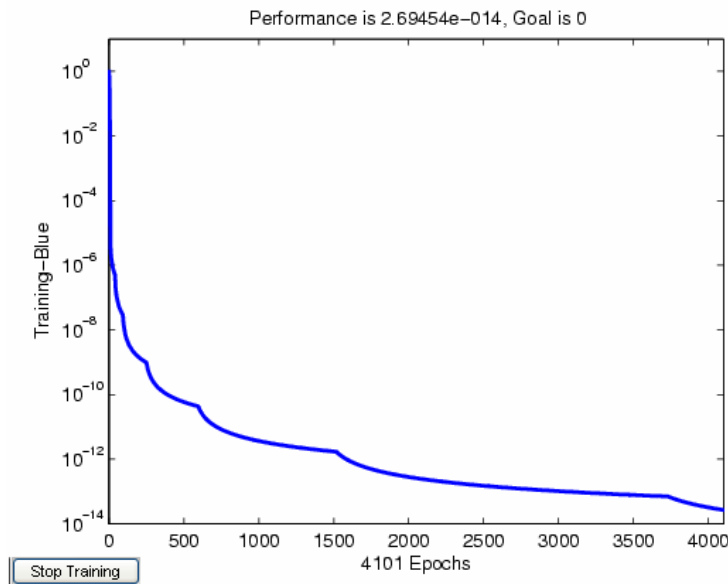


Figura 3.2. Performanțele rețelei după 4000 de epoci

Rețeaua pe care am folosit-o are 28 de intrări (cele 28 de simptome ale bolilor), 15 neuroni pe stratul intermediar și două ieșiri (1 și 0 pentru ciroză, 0 și 1 pentru insuficiență). Stratul intermediar și cel de ieșire folosesc drept funcție de transfer tangenta sigmoidă, iar funcția de performanță utilizată a fost media erorilor pătratice. Am antrenat rețeaua cu algoritmul de optimizare numerică Levenberg-Marquardt, preferat ca fiind o metodă mai rapidă de antrenare, comparativ cu metodele de gradient descendent sau

gradient descendent cu termen-moment. Performanțele după aproximativ 4000 de epoci de antrenare (batch) sunt redată de Figura 3.2.

Performanțele de recunoaștere în cadrul cirozei au fost de 100%, iar în cazul insuficienței cardiace de 76% recunoașteri corecte, pe un set de exemple nefolosit la antrenare. Aceste rezultate sunt destul de aproape de situația reală, în care insuficiența cardiacă este uneori confundată cu ciroza, după cum s-a mai menționat.

3.1.1.2. Algoritmii genetici și diagnoza medicală. Algoritmii genetici ocupă un loc aparte în cadrul tehnicilor inteligente. Dacă tehnicile tradiționale de rezolvare algoritmică cum ar fi greedy, backtracking etc. (căutare deterministă), și călirea simulată / căutarea tabu (căutare probabilistă) au în comun faptul că folosesc un singur punct al spațiului de căutare pentru explorarea de la fiecare iterație, algoritmii genetici sunt o tehnică inteligentă ce îmbunătățește aceste abordări prin faptul că procesează o populație de soluții simultan, introducând conceptul de *competiție* între soluțiile unei populații.

Noțiunea de “vecinătate” a doi indivizi-soluție din algoritmii genetici conduce la necesitatea definirii unor operatori de recombinare corecți relativ la semantica problemei, problemă dificilă în cele mai multe cazuri, și sursă principală de limitări a acestei tehnici. Se deosebesc două tipuri principale de probleme, diferențiate pe criteriul adecvării (neadecvării) la folosirea algoritmilor genetici:

I. Probleme de căutare a unei soluții optime într-un spațiu continuu de soluții, spațiul soluțiilor posibile ocupând o pondere semnificativă din spațiul total al problemei (spațiu continuu de căutare).

Aici este adecvată și utilă utilizarea algoritmilor genetici, după cum s-a demonstrat deja în cadrul a numeroase aplicații (a se vedea, spre exemplu, [MichFog04] și [Mich96]). În acest caz, constrângerile problemei nu sunt complexe și prohibitive pentru aplicarea acestor algoritmi, și sunt utilizate în slujba optimizării prin aplicarea de penalizări și funcții de reparație pe funcția de evaluare a soluțiilor a algoritmului genetic (folosirea de constrângeri pentru optimizare). Alegerea a priori a unei anumite metode evoluționare pentru o problemă de optimizare neliniară este încă o problemă deschisă. Caracteristicile care fac o problemă cu constrângeri dificilă pentru algoritmii genetici nu sunt clare, se știe doar că ele depind de topologia subspațiului fezabil (cu cât spațiul este mai puțin conex, problema este mai dificilă), și de caracteristicile funcției de evaluare.

II. Probleme de găsim a unei soluții (nu neapărat optimă). Acestea sunt probleme în care constrângerile reduc semnificativ spațiul soluțiilor (spațiu discret de căutare, de exemplu, problema damelor pe tabla de șah).

În acest caz, se abordează de obicei problema CSP (Constraint Satisfaction Problem) cu metode specifice de rezolvare, iar algoritmii genetici sunt folosiți pentru îmbunătățirea acestor metode (optimizare în slujba rezolvării constrângerilor). Acești algoritmi genetici speciali calculează o măsură a constrângerilor încălcate pentru a evalua calitatea soluțiilor [Par94].

Ca urmare a acestor observații, am considerat greoaie și neadecvată utilizarea algoritmilor genetici într-o problemă de diagnoză medicală, ce are întotdeauna un spațiu discret de căutare (fiind discontinuă). Dificultățile apar chiar din partea de modelare și

determinare de operatori adecvați, iar rezultatele sunt prea slabe și costisitoare în raport cu specificul acestor probleme. În plus, complexitatea și amploarea domeniului medical nu se potrivește cu viteza redusă de procesare a algoritmilor genetici.

3.2. Modele intermediare de diagnoză

Structurile prezentate în această secțiune reprezintă o etapă intermediară între metodele asociative și diagnoza pe bază de model. Le-am denumit “intermediare” întrucât, deși au capacitatea de a reprezenta natural și precis modelul profund al unui sistem, sunt de obicei folosite în modelarea cunoștințelor empirice, sau a modelelor parțiale, din motive de eficiență.

3.2.1. Modele probabiliste structurate. Rețele bayesiene

Rețelele bayesiene sunt un formalism ce are la bază ca metodă de calcul regula lui Bayes, și ca reprezentare a cunoștințelor un model al domeniului redus la relațiile de influență directă (aceste influențe nu sunt neapărat cauzale; după cum se va argumenta mai jos, probabilitățile condiționale nu sunt cele mai potrivite în modelarea cauzalității.)

3.2.1.1. Inferență probabilistă în rețelele bayesiene. Dacă în semantica abordărilor logice valorile de adevăr se referă direct la starea lumii, probabilitățile sunt măsuri ale încrederilor/presupunerilor/ ipotezelor noastre relativ la lume. Așadar, avantajul raționamentului probabilist asupra logicii este că un agent inteligent poate lua decizii chiar atunci când nu există suficiente informații care să demonstreze că acțiunea aleasă va da cu certitudine rezultatele dorite.

Inferența probabilistă reprezintă un mecanism de inferență pentru raționamentul aproximativ, care deși este exponențial în cel mai rău caz, este eficient în multe situații practice. Ideea de bază este că distribuția de probabilitate produsă de un model poate răspunde practic la orice întrebare legată de domeniul respectiv. Deși abordarea devine intratabilă la un număr mare de variabile, în contextul aplicării regulii lui Bayes independența condițională dintre variabile simplifică mult calculul interogărilor și reduce numărul de probabilități condiționale ce trebuie specificate.

Definiție[RusNor02]. O rețea bayesiană este o structură de date ce reprezintă dependențele între variabilele domeniului și dă o descriere concisă a distribuției de probabilitate produsă.

Nodurile rețelei reprezintă variabile aleatoare; nodurile rețelei X și Y sunt legate printr-un arc ce pornește din X dacă X are influență directă asupra lui Y în domeniul modelat. Fiecare nod are asociată o tabelă de probabilități condiționale ce cuantifică efectul părinților asupra nodului. Topologia rețelei se poate interpreta ca o bază abstractă de cunoștințe ce reprezintă structura generală a proceselor de dependență din cadrul domeniului, rețeaua fiind definită complet de această topologie și tabelele de probabilități condiționale.

Orice intrare din probabilitatea produs (și deci orice interogare) se poate calcula din rețea după formula [RusNor02]:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | Parents(x_i)) \quad (3.1)$$

dar în general:

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1) \cdot P(x_{n-1}, \dots, x_1) = P(x_n | x_{n-1}, \dots, x_1) \cdot P(x_{n-1} | x_{n-2}, \dots, x_1) \dots P(x_2 | x_1) \cdot P(x_1) \\ &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1); \quad (3.2) \end{aligned}$$

deci pentru a avea (3.17) este necesar:

$$\forall i: P(X_i | X_{i-1}, \dots, X_1) = P(X_i | Parents(X_i)) \quad (3.3)$$

Relația 3.3 are loc dacă se numerotează cu un algoritmul corespunzător nodurile rețelei: $Părinți(X_i) \in \{X_{i-1}, \dots, X_1\}$, ideea fiind că orice nod este independent de predecesorii din rețea fiind dați părinții (depinde doar de părinți).

Deși există algoritmi liniari pentru rețelele de tip poliarbore (în care există cel mult o cale nedirecționată între oricare două noduri), pentru rețelele multiplu conectate inferența probabilistică exactă rămâne NP-dificilă. Pentru astfel de rețele generale se încearcă reducerea lor la poliarbore, aplicarea algoritmului de bază pentru fiecare poliarbore și combinarea rezultatelor, conform unei strategii de tip “divide et impera”. Algoritmii generalizați sunt în principal de patru tipuri: metode de clustering, metode pe bază de mulțimi tăietură, algoritmii de simulare stocastică, și metodele de căutare pentru rețele de tip “noisy-or”. Din păcate, aproximările stocastice ale inferenței probabiliste sunt deasemenea NP-dificile [HuaHen96], ca și celelalte tipuri de algoritmi generalizați, cu excepția căutării în cadrul rețelelor noisy-OR. Rețelele de tip “noisy OR” modelează relații logice cu zgomot, ce adaugă incertitudine la abordarea logică (spre deosebire de nodurile deterministe ale rețelelor bayesiene clasice, unde probabilitățile valorilor sunt specificate exact de valorile părinților, fără incertitudine.)

Ex. Febră=Răceală sau Gripă sau Malarie (în logică: $A(\text{devărat}) = A \text{ sau } A \text{ sau } A$)

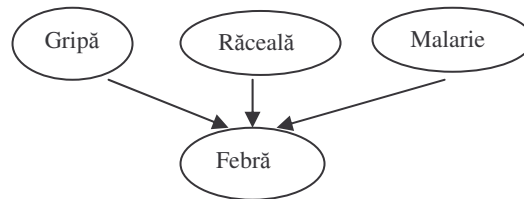


Figura 3.3.Exemplu de mini-rețea noisy-OR

Dacă presupunem că:

- toate cauzele sunt listate ;

- fiecare cauză are o șansă independentă de a produce efectul;
- orice previne Gripa de a produce Febră e independent de ceea ce previne Malaria de a produce Febră,

atunci parametrii zgomot pot fi priviți ca posibili inhibitori ai dependențelor modelate în probabilitățile condiționale; astfel, dacă $P(\text{Febră}|\text{Răceală})=0.4 \Rightarrow P(\text{inhibitori})=1-0.4=0.6$.

Regula de calcul: probabilitatea ca ieșirea să fie falsă este produsul parametrilor zgomot pentru părinții cu valoarea “adevărat”. Prin urmare, nu mai avem tabele de probabilități condiționale ci maximum k parametri (unde k=numărul de părinți). Este o modelare diferită a ideii de “undercutting defeaters” [Poll92, Poll02]: o implicație poate avea excepții de utilizare.

Metode de inferență pe bază de căutare. Ideea acestei metode este de a căuta instanțieri parțiale sau complete ale spațiului produs aposteriori, folosite apoi pentru estimarea valorilor variabilelor de interes. Succesul metodelor de căutare depinde de doi factori [HuaHen96]:

- în distribuția produs, masa majoritară a probabilității este concentrată într-o mică fracție a variabilelor ipoteze (după cum este și cazul în diagnoză) ;
- există reguli eficiente de reducere masivă a spațiului de căutare.

Proprietatea 2 depinde de clasa de rețele bayesiene. Astfel, Max Henrion a dezvoltat un algoritm simplu (- TopN) pentru inferență în rețele noisy-OR cu două nivele [HuaHen96]. De exemplu, pentru rețeaua din Figura 3.3, algoritmul găsește cele mai probabile N instanțieri ale nivelului boli. O generalizare a algoritmului TopN pentru rețele noisy-OR multinivel este dată de către algoritmul TopEpsilon [HuaHen96], care enumeră instanțierile complete ale rețelei cu o probabilitate produs mai mare decât un prag fixat. Totuși, presupunerile făcute în definirea rețelelor noisy-OR devin nerealiste în multe cazuri particulare.

3.2.1.2. Bayesianism și cauzalitate. Cele mai riguros apropiate de diagnoza pe bază de model cauzal profund sunt rețelele cauzale. Acestea sunt rețele bayesiene în care părinții unui nod reprezintă cauzele sale directe (din modelul natural al domeniului). În construirea unei rețele bayesiene, adăugarea nodurilor pornind de la cauzele “de bază” este o recomandare care îmbunătățește mult eficiența modelului, dar acesta nu este singurul avantaj: după cum demonstrează Poole, rețelele cauzale pot fi modelate ca un program logic cu probabilități asociate diferitelor ipoteze, fiind astfel echivalente cu viziunea logică asupra abducției.

După cum observa autorul în [Pea01b], statistica este domeniul care se ocupă de descoperirea asocierilor ce pot apărea între variabile, sub forma unor coeficienți de corelație, folosind tehnici de regresie și estimare, și orice concept statistic poate fi exprimat în termenii unei distribuții de probabilitate (subiectivă sau bazată pe frecvențe). În plus, statistica nu surprinde decât relațiile statice din cadrul lumii modelate. Condițiile dinamice care țin de modificările posibile ale variabilelor trebuie modelate într-un formalism nou, specific cauzalității. O idee care decurge de aici este că probabilitățile

condiționale nu sunt totdeauna potrivite pentru modelarea cauzalității. În sprijinul acestei afirmații, un argument în plus este dat și de următorul exemplu.

Conform teoriei probabilității, dacă H reprezintă o ipoteză și E o mulțime de evidențe, și formalizăm cauzalitatea cu ajutorul probabilității condiționale, atunci:

$$P(H|E) = 1 - P(H'|E) \quad (3.4)$$

implică o relație cauză-efect între E și H' când există o relație cauză-efect între E și H .

Un exemplu simplu din [GiaRil94] ne arată că această asociere nu este corectă:

$$\text{“Dacă } P(a \text{ absolvi} \mid \text{nota “A” la cursul X}) = 0.7\text{”} \quad (3.5)$$

aceasta nu înseamnă neapărat că:

$$P(a \text{ nu absolvi} \mid \text{nota “A” la cursul X}) = 0.3, \quad (3.6)$$

întrucât există mulți alți factori ce pot încă împiedica absolvirea.

În încercarea de a depăși aceste anomalii, Pearl a propus *modelul ecuațiilor structurale* pentru a trata sistemele cauzale [Pea01a], și în care noțiunea de *intervenție* este un element cheie pentru a distinge între o relație cauzală și orice alt fel de dependență: într-o relație cauză –efect modificări ale părinților rezultă în modificări ale copiilor și ale nici unui alt nod. Rețelele cauzale au fost folosite în diagnoza medicală în sisteme precum **CASNET** (pentru diagnosticul glaucomului [Lon01]), **Heart Disease Program** [LongNai92] .

3.2.1.3.Sistemul de diagnoză medicală CASNET. Sistemul CASNET, prezentat în continuare, [Weis74] a constituit o sursă importantă de inspirație pentru abordarea originală a acestei lucrări (în special în partea de reprezentare a cunoștințelor).

Reprezentarea cunoștințelor. Dacă domeniul medical abordat este suficient de bine înțeles încât permite reprezentarea detaliată a mecanismelor fiziologice ce stau în spatele declanșării simptomelor, este de bun simț să renunțăm la o reprezentare bazată doar pe asocieri boli-simptome (cum folosesc de exemplu sistemele de diagnoză medicală - clasice deja- PIP sau INTERNIST). Rețeaua cauzală a sistemului CASNET cuprinde stări de disfuncționalitate (diferite de bolile propriu zise) și teste (informații externe ce definesc probabilitatea de existență a stărilor ascunse). Un statut particular îl au stările inițiale (primare din punct de vedere etiologic) și cele finale (care nu au consecințe disfuncționale). Relațiile cauzale între noduri sunt legături de diferite ponderi nod cauză-nod efect, legături interpretate drept frecvența cu care primul nod îl cauzează pe al doilea. Nu sunt permise ciclurile în rețea. Rețeaua de stări de disfuncționalitate este un model simplificat al bolii, cu rolul de a ghida procesul de diagnoză. Nodurile pot avea statutul de confirmate sau infirmate prin teste.

Mecanismul de inferență. Inferența este probabilistă, bazându-se, în principiu, pe un proces de propagare a încrederilor/ neîncrederilor în adevărul nodurilor pe baza testelor de care sunt legate, creând căi acceptabile (adică fără noduri negate) în rețea. Aceste căi constituie potențiale explicații pentru diagnosticul final, și focalizează atenția pe regiunile de interes ale rețelei. Planul de testare este definit prin intermediul structurii rețelei- se testează, la fiecare pas, acele noduri “necunoscute” aflate pe căile aflate în construcție.

Practic, pentru fiecare nod se calculează două măsuri probabilistice: *ponderea* (estimarea verosimilității sale pe baza tăriei legăturilor cauzale cu nodurile confirmate/ infirmate cele mai apropiate-înrudite-) și *starea* (estimarea verosimilității pe baza testelor direct relevante pentru nod). Astfel, de câte ori se obține rezultatul unui test nou, starea fiecărui nod legat de testul respectiv este reactualizată: dacă rezultatul testului are încredere mai mică decât valoarea de încredere a nodului, nu modificăm nimic, dacă rezultatul este mai mare, valoarea nodului devine egală cu încrederea testului, dacă sunt egale dar de semne opuse, se raportează utilizatorului prezența unei contradicții.

Dezavantaje:

- Un prim dezavantaj al CASNET este moștenit de la rețelele bayesiene: după cum am subliniat mai sus, pentru rețelele multiplu conectate (cu mai mult de două căi între două noduri) inferența probabilistă e NP-dificilă. O problemă legată de acest aspect vine din faptul că se lucrează teoretic cu toată rețeaua simultan, o consecință nedorită fiind și că multe noduri cu relevanță foarte mică sunt implicate în calcul. Aceasta afectează nu doar eficiența, dar are rezultate nedorite asupra rezultatului final al calculului (ponderi mai mari decât trebuie pentru anumite diagnostice, diagnostice care apar deși ar trebui să lipsească etc).
- O altă problemă importantă în CASNET este felul în care se tratează agregarea rezultatelor testelor legate de un singur nod. Dacă trebuie redat faptul că două teste susțin un anumit nod doar când apar conjugate, atunci un test nou este definit pentru a reprezenta cele două teste împreună. Autorul argumentează că e suficient, în această situație, să considerăm tăria maximă a celor două teste ca rezultat al agregării, ceea ce, după părerea noastră, este discutabil în cazul general. În plus, abordarea este inaplicabilă acelor situații (destul de frecvente în diagnoză) când trebuie modelat faptul că o ipoteză e susținută doar de prezența unui “număr consistent” (criteriu vag) de teste din cele legate de ea.
- Maniera de tratare a contradicțiilor constituie, deasemenea, un dezavantaj. Modul de calcul al scorului unui nod prin adăugarea și scăderea de cantități în funcție de confirmarea/ infirmarea unor teste poate duce la rezultate ambigue, greu de interpretat și chiar greșite. Cea mai bună dovadă în acest sens este faptul că, dacă prin scăderi și adunări repetate se junge la scor 0, sistemul raportează contradicția utilizatorului, nefiind definit un mecanism automat de tratare. În plus, stabilirea pragului T ridică multe semne de întrebare: este clar că acest test ar trebui să fie dependent de nodul asociat, de numărul de teste confirmatoare/ infirmatoare etc., dar detalii clare despre aceste dependențe deocamdată nu avem. De aceea considerăm că o abordare categorică a problemei contradicțiilor constituie o soluție mai bună decât metoda probabilistă.

O îmbunătățire notabilă a sistemului CASNET a fost făcută în sistemul CHECK [TorCon89], dar și în sistemul original DiaMed.

3.2.2. Sisteme de inferență fuzzy

Sistemele de inferență fuzzy reprezintă un formalism potrivit pentru folosirea unui model furnizat de un expert uman, construind o schemă simplificată a unui model complex. Specificul lor principal stă în formalismul de reprezentare a incertitudinii. După cum a

fost subliniat în Capitolul II, medicina este un domeniu în care aceste sisteme își găsesc în mod natural aplicativitatea, datorită multiplelor surse de incertitudine, imprecizie și subiectivitate.

Reguli fuzzy Mamdani de tipul “dacă-atunci” sunt expresii de tipul prezentat mai jos, ce captează regulile empirice folosite de experți umani în condiții de imprecizie și incertitudine; fiecare regulă poate fi privită ca o descriere locală (a unei părți limitate) a sistemului:

“Dacă A atunci B”, (unde A și B reprezintă mulțimi fuzzy). (Exemplu: “Dacă presiunea este ridicată, atunci volumul este mic.”)

$$R_1: \text{DACĂ } x_1 \text{ este } A_{11} \text{ ȘI...ȘI } x_m \text{ este } A_{m1} \text{ ATUNCI } y \text{ este } B_1$$

$$R_2: \text{DACĂ } x_1 \text{ este } A_{12} \text{ ȘI...ȘI } x_m \text{ este } A_{m2} \text{ ATUNCI } y \text{ este } B_2$$

...

$$R_n: \text{DACĂ } x_1 \text{ este } A_{1n} \text{ ȘI...ȘI } x_m \text{ este } A_{mn} \text{ ATUNCI } y \text{ este } B_n$$

$$\text{Fapt: } x_1 \text{ este } x_{10}, x_2 \text{ este } x_{20}, \dots, x_m \text{ este } x_{m0},$$

Concluzie: Y este B

Iată un exemplu de schemă de raționament fuzzy pentru cazul sistemelor MISO (Multiple Input Single Output) [Ful95], pentru sistemul reprezentat mai sus. Concluzia se calculează ca $\text{Agg}(\text{Fapt} \circ R_1, \dots, \text{Fapt} \circ R_n)$, obținându-se:

$$B = \text{Agg}(x_{10} \times \dots \times x_{m0} \circ R_1, \dots, x_{10} \times \dots \times x_{m0} \circ R_n). \quad (3.7)$$

Ieșirea fuzzy a acestei baze de cunoștințe se calculează astfel:

- Nivelul de procesare al regulii R_i este $A_{1i}(x_{10}) \times \dots \times A_{mi}(x_{m0})$;
- Ieșirea fiecărei reguli R_i este mulțimea fuzzy $B_i'(w) = A_{1i}(x_{10}) \times \dots \times A_{mi}(x_{m0}) \rightarrow B_i(w)$, $\forall w \in W$, (W fiind mulțimea de discurs a variabilei Y);
- Ieșirea globală B a sistemului este: $B(w) = \text{Agg}\{B_1', \dots, B_n'\}$, $\forall w \in W$.

(S-a folosit metoda CRF (Combine Rules First)).

Operatorii care stau la baza agregării pot fi atât t-norme cât și t-conorme, după cum agregarea are o semnificație conjunctivă sau disjunctivă. Opțiuni diverse există și pentru produsul cartezian sau implicație (spre exemplu, t-norma produs).

Se observă că în modelul propus mai sus, funcția de apartenență ce definește mulțimea fuzzy-consecvent este o valoare lingvistică pentru una și aceeași variabilă (y). Dacă dorim o valoare crisp apelăm la defuzzificare.

3.2.3. Modele simbolice

Cele mai cunoscute și utilizate structuri preponderent simbolice sunt arborii de decizie și sistemele expert. Aceste formalisme au fost printre primele utilizate în raționamentul simbolic și constituie o etapă intermediară între clasificarea metrică și cea non-metrică.

Cele două structuri menționate folosesc de obicei cunoștințe de suprafață, adică euristici ce rezumă cunoștințele unui expert uman. Spre deosebire de sistemele ce folosesc cunoștințe profunde, nu există lanțuri cauzale, legături tip-cauză efect de la o regulă la alta [GiaRil94], astfel că inferența este mai rapidă dar și explicațiile mai primitive. Sunt de preferat atunci când elementele lanțului cauzal profund nu sunt accesibile pentru testare/observare. Componentele principale ale acestor sisteme sunt baza de cunoștințe și mecanismul de inferență.

MYCIN[GiaRil94] Modelul factorilor de certitudine (*încredere*) ce stă la baza acestui sistem expert de diagnosticare a infecțiilor hematologice a fost propus de Shortliffe și inspirat de teoria confirmării a lui Carnap. Motivația sa a plecat de la faptul că modelul bayesian nu este foarte potrivit pentru domeniul medical unde numărul testelor accesibile este limitat din diverse motive, și, în plus, rezultatele se obțin secvențial, în timp, unele cu întârziere. În plus, sunt prea multe probabilități condiționale de cunoscut apriori, și apare și problema menționată în discuția cauzalității într-o rețea bayesiană (3.4- 3.6).

În încercarea de a depăși modelul bayesian, Shortliffe propune o măsură care combină într-un singur număr (=factor de încredere) încrederea și neîncrederea într-o ipoteză condiționată de prezența unei anumite evidențe:

$$CF(H,E)=MB(H,E)-MD(H,E), (3.8)$$

unde MB=măsură a încrederii și MD-măsură a neîncrederii [GiaRil94]. Factorii de încredere astfel obținuți sunt folosiți la ierarhizarea ipotezelor de diagnostic.

Agregarea evidențelor din antecedentul unei reguli se face utilizând logica fuzzy. Încrederea finală într-o ipoteză se face prin combinarea după un model propriu a regulilor al căror consecvent se referă la ipoteza respectivă. Una dintre problemele mari ale acestui sistem de combinare a evidențelor este că interacțiuni nedorite apar între regulile bazei de cunoștințe dacă aceasta nu este atent construită.

S-a demonstrat că teoria factorilor de încredere nu este decât o aproximare a raționamentului probabilist, și succesul MYCIN se datorează de fapt simplității teoriei domeniului (lanțuri inferențiale scurte + ipoteze simple), teoretic existând probleme cu modelul propus.

În încercarea de a compara sistemele expert și modelele probabiliste, observăm că sistemele expert sunt mai puțin precise față de modelele pur probabiliste (bayesiene), dar mai transparente, prin faptul că folosesc pentru clasificare, în principal, informație simbolică (reguli, liste de atribute). În ceea ce privește diferențele între sistemele logice pe bază de reguli și sistemele probabiliste autorii menționează în [RusNor02]:

- Sistemele logice pe bază de reguli au proprietatea de *localitate*, (rezultatul aplicării unei reguli nu depinde de alte reguli) spre deosebire de sistemele probabiliste unde trebuie luată în considerare toată evidența disponibilă. (Rețelele bayesiene, spre exemplu, sunt totuși *local structurate*, în sensul că un nod interacționează numai cu un grup restrâns de noduri în calcularea probabilităților condiționale -arcele din graful ce modelează rețeaua reprezintă un mijloc de ghidare a raționamentului;
- Sistemele logice sunt în general *funcționale relativ la valoarea de adevăr* (adevărul unei propoziții complexe poate fi calculat pe baza componentelor), în timp ce

probabilitățile nu se pot combina funcțional decât în ipoteza de independență (adesea nerealistă).

3.3. Diagnoza pe bază de model

Introducem în continuare câteva dintre aspectele de bază ale diagnozei abordată cu metode logico-simbolice, pe bază de model cauzal de profunzime. Soluțiile sunt exacte, transparente (bune facilități de explicare), dar aici apare în prim plan problema eficienței.

3.3.1. Abducția și tehnicile bazate pe consistență

Abducția reprezintă paradigma generală a procesului de diagnoză. Termenul de "abducție" a fost introdus de Peirce în anii 1800 și definit drept "procesul de formare a unei ipoteze explanatorii plecând de la o mulțime de observații".

Structura generală a unui raționament abductiv

D=observații adevărate, H explică D, nu există H' mai bun decât H care explică D

H este presupus adevărat

Printre aplicațiile cele mai importante ale abducției se numără raționamentul în prezența informațiilor incomplete, generarea explicațiilor cauzale în diagnoza defectelor, simularea rețelelor bayesiene.

Definiția abducției în termeni logici. Fie T teoria domeniului, D - formula ce trebuie explicată. O explicație pentru D în T este o formulă E care îndeplinește condițiile:

- $T \cup E$ este consistent;
- $T \cup E \vdash D$;
- E este formată numai din predicate asumabile și e cea mai simplă care satisface 1 și 2.

De nonmonotonicitatea abducției ne putem da seama urmărind exemplul următor ([Dup94]). Fie $p_1 \rightarrow q, \dots, p_n \rightarrow q$, unde p_i, q sunt propoziții din teoria T a domeniului, și implicația materială este folosită pentru a modela relații cauză-efect (pentru un comentariu legat de cât este de potrivită această utilizare a se vedea Yeov Shoham-1987). Raționamentul abductiv spune, în acest exemplu, că pentru a explica q, trebuie să asumăm (să considerăm adevărat) cel puțin un p_i -ceea ce nu este corect decât dacă am presupus că avem cunoștințe complete în T despre q. Abducția este așadar o formă de raționament de tip invalidabil, întrucât depinde de cunoștințele (posibil incomplete) disponibile la un moment dat.

Avantajele sistemelor bazate pe model (i.e cunoștințe profunde ce surprind structurile componente ale sistemului și legăturile cauzale între acestea) sunt abilitatea de a rezolva probleme necunoscute expertului uman și facilitatea de a explica concluziile procesului de raționament, -care devine însă foarte complex pe un astfel de model declarativ. Există două tipuri de formalizări logice ale diagnozei pe bază de model. Formalizarea abductivă este preferată atunci când dispunem de un model "de defect" (model al comportamentului anormal) al sistemului, și acest model este complet-în acest context, prin diagnostic se înțelege o mulțime minimală de ipoteze de anormalitate care "acoperă" (implică)

observațiile (conform observației de mai sus, în condiții de completitudine, abducția se reduce la deducție)([Kon92]).

Dacă însă dispunem de un model al comportamentului corect, se folosește diagnoza pe bază de consistență: în cadrul acesteia, diagnostic înseamnă o mulțime minimală de componente ale sistemului asumate defecte, astfel încât comportamentul corect al celorlalte componente este consistent cu observațiile. Noțiunea de explicație diferă de la o abordare la alta: dacă în cadrul diagnozei bazate pe consistență o explicație pentru o manifestare m este tot ceea ce nu susține $\neg m$, în abducție explicația trebuie să susțină direct m . (În abducție este necesar $i \rightarrow o$, în consistență este necesar ca $i \wedge o$ să nu fie contradictorie.) Avantajul abordării pe bază de consistență stă în validitatea logică, indiferent de completitudinea modelului. Dezavantajele sunt legate de imposibilitatea de a explica rezultatul și de a identifica exact defectul. În [ConTor92] autorii propun o viziune unificată a celor două direcții, o problemă de diagnostic fiind privită ca o problemă de abducție cu constrângeri de consistență.

Pentru abducție s-au folosit atât euristici cât și tehnici de demonstrare directă ([Kon92, McIlr98]), de exemplu *tehnicele rezolutive de găsire a consecințelor unei teorii*. Această abordare nu respectă însă semantica unei probleme de diagnoză. Ideea folosită este de a rescrie relația $\Sigma \wedge E \vdash O$ (Σ fiind teoria domeniului, E explicația, O observațiile), prin aplicarea contrapozitivei, drept $\Sigma \wedge \neg O \vdash \neg E$. Versiunea duală (indirectă) a abducției, ce se obține astfel este următoarea.

Definiție.(Abducția indirectă) E este o explicație pentru O dacă:

- $E \subseteq H$;
- $\Sigma \cup \neg O \vdash \neg E$;
- Din Σ nu se deduce $\neg E$, unde $\neg E$ reprezintă conjuncția literalilor negați.

O idee folositoare ce decurge din versiunea duală este că o explicație pentru O se poate genera căutând teoremele lui $\Sigma \cup \neg O$, care nu sunt teoreme doar ale lui Σ .

Revenind însă la problema noastră de diagnoză, lipsa anumitor observații nu înseamnă obligatoriu lipsa unui anume diagnostic (acesta poate fi prezent prin alte manifestări ale sale). Un dezavantaj în plus îl constituie faptul că rezoluția este completă pentru respingere (i.e. completă pentru găsirea demonstrațiilor), dar nu este completă pentru găsirea consecințelor (nu e deductiv completă). De fapt, teoretic există metode (versiuni ale rezoluției) complete, dar care sunt extrem de ineficiente din cauza unui spațiu de căutare mult prea mare, în timp ce metodele eficiente sunt incomplete [Kon92].

3.3.2. Metode de diagnoză specifice sistemelor dinamice

În contextul sistemelor dinamice devine evidentă necesitatea integrării timpului în modelul de diagnoză. Metodele de diagnoză a sistemelor depind atât de tipul sistemului cât și de tipurile posibile de defecte ce îl pot afecta.

O schemă robustă FDI (FDI=Fault Detection and Isolation) trebuie să asigure atât sensitivitate satisfăcătoare la defecte cât și invarianța la zgomot și incertitudinile de modelare [PattChe97] (aceste două cerințe aflându-se în conflict).

“*Diagnoza pe bază de model* se definește ca determinarea defectelor unui sistem prin compararea măsurătorilor disponibile la un anumit moment cu informațiile a priori date de modelul analitic/matematic al sistemului și generarea și analiza de cantități reziduale. Un reziduu este un indicator de defect ce reflectă condiția degradată a sistemului.” [Patt94].

Din definiție, se observă că un element central în diagnoza pe bază de model este generarea de reziduuri [Nyb99]. Stadiile principale ale diagnozei defectelor folosind reziduuri sunt reprezentate în Figura 3.4 (conform [Fran96]). Problema principală a diagnozei pe bază de model stă în faptul că erorile de modelare și perturbațiile afectează sistemul. Incertitudinile de modelare afectează abilitatea sistemului de a recunoaște defecte incipiente, ducând la alarme false sau defecte nedetectate. Trebuie, astfel, realizat un compromis între sensibilitatea la erori și robustețea la zgomot.

Așadar, diagnoza sistemelor dinamice constă în două etape principale distincte: detectarea defectelor și diagnoza defectelor (Figura 3.4). Detectarea defectelor constă în generarea euristică sau analitică a simptomelor (bazate pe reziduuri).

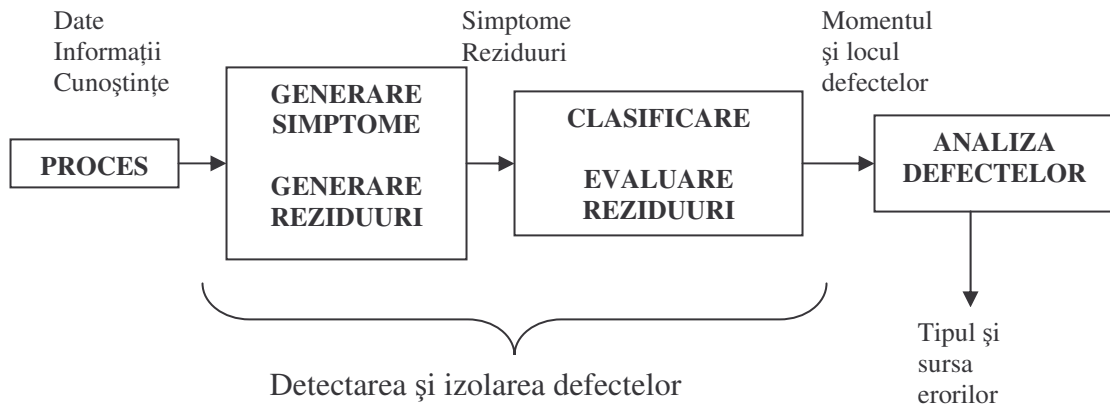


Figura 3.4.[Fran96].Reprezentarea schematică a procedurii de diagnoză a defectelor

Diagnoza defectelor determină tipul, locația, momentul apariției și amplitudinea defectului, prin observarea simptomelor euristice sau analitice [Ise97]. În cadrul acestei etape, dacă nu sunt disponibile relații cauzale profunde între defecte și simptome, se folosesc metode de clasificare ce mapează vectorul de simptome într-un vector de defecte (de exemplu, clasificare statistică sau geometrică, rețele neurale, clusterizare fuzzy). Dacă dimpotrivă relațiile respective sunt accesibile, (spre exemplu, sub forma unor relații cauzale), se pot aplica strategii de raționament pentru diagnoză (înlănțuire înainte/înapoi, raționament aproximativ probabilistic pentru rețele bayesiene și probabilități condiționale sau raționament posibilist cu ajutorul logicii fuzzy [Ise97]). Capitolul IX prezintă o abordare originală a acestei etape (de diagnoză a defectelor sistemelor dinamice), prin adaptarea sistemului hibrid DiaMed.

3.4. Concluzii și contribuții

Capitolul propune o privire de ansamblu asupra unei mari părți a metodelor de diagnoză cunoscute. O contribuție importantă a sa constă în încercarea de clasificare și sistematizare pe care o propune, pentru a îngloba într-un cadru coerent abordări distincte ale aceleiași probleme. Sunt confruntate, în principal, diagnoza pe bază de model și diagnoza prin metode statistice, fiecare dependentă de accesul la un anumit tip de informație despre sistemul studiat. Prezentarea este ilustrată de exemple clasice de implementare din literatură, și oferă o imagine clară despre formalismul de bază al fiecărei metode, cu avantajele și dezavantajele sale. Scopul principal a fost acela de a pune bazele unei justificări a utilității combinării celor două paradigme într-un model hibrid, justificare ce se va finaliza în capitolul următor.

IV. Sisteme inteligente hibride de diagnoză medicală

4.1. Tehnici inteligente hibride

Hibridizarea urmărește compensarea dezavantajelor și îmbunătățirea performanțelor prin cooperarea tehnicilor inteligente în cadrul rezolvării aceleiași probleme. Există patru tipuri de sisteme hibride [PalNea01]:

1. **Sisteme fuzionale:** modul de reprezentare și prelucrare a informațiilor caracteristic unei tehnici se contopește în modul de reprezentare și prelucrare al altei tehnici (de exemplu, sistemele cu rețele neurale pentru reprezentarea și procesarea informațiilor simbolice caracteristice sistemelor simbolice bazate pe cunoștințe sau sistemelor fuzzy).
2. **Sisteme de transformare:** transformă o formă de reprezentare a cunoștințelor în altă formă (spre exemplu, sistemele de extragere de reguli simbolice din rețelele neurale, reguli ce sunt apoi folosite de un sistem simbolic).
3. **Sisteme combinate:** păstrează identitatea fiecărei metodologii inteligente și presupun utilizarea modulară a două sau mai multe tehnici inteligente pentru a rezolva o problemă.
4. **Sisteme asociative:** combină mai multe din tipurile de hibridizare de mai sus.

Hibridizare neuro-simbolică. Integrarea neuro-simbolică își găsește justificarea în primul rând la nivel cognitiv, ideea fiind de a identifica în inteligența umană operațiile de nivel înalt (legate de partea de raționament și care sunt cel mai bine modelate de o abordare simbolică), și operațiile de nivel inferior (legate de percepție și mai natural de abordat într-un cadru neuronal). Întrucât această hibridizare reconciliază știința experților cu estimările statistice din date, este foarte potrivită aplicațiilor complexe ale lumii reale. Am ales pentru exemplificarea acestei paradigme sistemul hibrid KBANN (Knowledge-Based Artificial Neural Networks) utilizat în biomedicină [San99], și rețelele neurale

comitet, care ni s-au părut o generalizare adecvată și promițătoare a acestuia, prin prisma aplicațiilor de diagnoză medicală și nu numai.

TEHNICA INTELIGENTĂ	AVANTAJE	DEZAVANTAJE
Sisteme simbolice bazate pe cunoștințe	<ul style="list-style-type: none"> • Sunt transparente (furnizează explicații clare ale deciziilor luate); • Separă cunoștințele de sistemul ce le exploatează; • Înglobează cu ușurință cunoștințe structurate, de calitate și precise, prin reprezentarea declarativă. 	<ul style="list-style-type: none"> • Dificultăți la achiziționarea de cunoștințe și la adaptare; • Număr mare de reguli; • Raționament lent; • Dificultăți în operarea cu informații imprecise și incomplete; • Performanțele nu se îmbunătățesc semnificativ la completarea cunoștințelor; • Nu generalizează.
Rețele neurale	<ul style="list-style-type: none"> • Prelucrare paralelă; • Răspuns rapid; • Învățare, adaptare rapidă; • Capacitate de generalizare și procesare a informațiilor incomplete. 	<ul style="list-style-type: none"> • Incapacitatea de a explica rezultatul; • Riscul blocării în minimele locale.
Sisteme fuzzy	<ul style="list-style-type: none"> • Modelează imprecizia și incertitudinea; • Aproximează funcții nelineare; • Implementare ieftină și comportament robust; • Nu necesită un model foarte precis; • Viteză de inferență ridicată; 	<ul style="list-style-type: none"> • Probleme cu învățarea și adaptarea; • Probleme în definirea termenilor de apartenență, a numărului de reguli, a parametrilor consecințelor regulilor;
Algoritmi genetici	<ul style="list-style-type: none"> • Nu necesită un model matematic (sunt "orbi": fac puține presupuneri asupra domeniului problemei); • Nu au probleme cu minimele locale (în principal datorită faptului că explorează spațiul de căutare în mai multe puncte simultan, prin intermediul mai multor populații de soluții potențiale); • Explorează inteligent spațiul de căutare; • Potrivii pentru spațiile de căutare mari; • Au arii largi de aplicabilitate deoarece operează asupra unor codificări ale variabilelor de decizie; • Adaptare lentă. 	<ul style="list-style-type: none"> • Lenți; • Dificultăți în construirea operatorilor potriviți.

Tabel 4.1. Comparație a tehnicilor inteligente

Hibridizare neuro-fuzzy. Rețelele neurale și logica fuzzy se completează adesea în chip fericit în cadrul sistemelor inteligente: dacă rețelele neurale pot prelucra cantități mari de date primite de la senzori (în probleme de conducere sau recunoaștere a formelor), logica fuzzy prelucrează aceste date de nivel inferior într-un cadru structurat de nivel înalt. Există în principal două direcții importante în acest tip de hibridizare [DumiBuiu00]:

- Fuzificarea arhitecturilor convenționale de rețele neuronale (Secțiunea 4.2);
- Folosirea rețelelor neurale ca unelte în cadrul modelelor fuzzy (de exemplu, în furnizarea unor aproximații de calitate ale funcțiilor de apartenență).

Hibridizare geno-fuzzy. Combinarea celor două tehnici se face în baza anumitor asemănări și deosebiri. Între primele, menționăm faptul că nici una din tehnici nu necesită un model precis, utilizând o reprezentare distribuită a cunoștințelor (în populațiile de cromozomi sau, respectiv, în mulțimile și regulile fuzzy), sunt tolerante la defecte (ca urmare a reprezentării distribuite) și pot modela incertitudinea. Însă în ceea ce privește manevrarea cunoștințelor distribuite, dacă algoritmi genetici efectuează o procesare paralelă și *probabilistă*, logica fuzzy are un mecanism de inferență *posibilist*. În plus, dacă algoritmi genetici sunt prin natura lor adaptivi, sistemele de inferență fuzzy clasice nu au această proprietate, ele fiind în schimb mai avantajoase din punctul de vedere al vitezei de inferență [DumiBuiu00].

Între exemplele acestui tip de hibridizare menționăm modelul propus de Karr [Karr91], în care un algoritm genetic este folosit pentru învățarea funcțiilor de apartenență și numărului de reguli unui sistem fuzzy de tip Takagi-Sugeno. Funcțiile de apartenență sunt corespunzător parametrizate și codificate în cromozomii algoritmului, iar evaluarea populațiilor de soluții se face dependent de aplicația avută în vedere.

Hibridizare neuro-genetică. Această hibridizare pleacă de la ideea că este benefic ca evoluția și învățarea să poată lucra împreună sinergetic. Există două tipuri de abordări distincte: combinațiile suport (în care tehnicile sunt folosite secvențial) și combinațiile colaborative (unde tehnicile sunt rulate simultan). În general, rețelele neurale sunt asistate de către algoritmul genetic în selectarea caracteristicilor (intrărilor), a topologiei sau a regulii de învățare folosite.

Am justificat în secțiunea dedicată algoritmilor genetici în diagnoza medicală de ce nu este oportună utilizarea lor în aplicațiile acestui domeniu. Ne vom ocupa în continuare de exemple de hibridizare neuro-simbolică și neuro-fuzzy, care au fost aplicate deja sau care am considerat că ar fi potrivite diagnozei medicale.

Tehnicile inteligente și logica. Un tip aparte de hibridizare îl constituie utilizarea unei metode inteligente (simbolică, fuzzy, neurală, genetică) în sprijinul unui proces de raționament bazat pe logică (formală sau informală). Un exemplu notabil în acest sens îl constituie sistemul CHECK (sprijin simbolic cu reguli pentru logica formală), care a fost și punctul de plecare al genului de hibridizare combinativă folosit de sistemul original DiaMed (sprijin fuzzy pentru logica informală) (Secțiunea 4.4).

4.2. Tehnici inteligente hibride în diagnoza medicală

4.2.1. Hibridizare neuro-simbolică

4.2.1.1. Rețele neurale bazate pe cunoștințe. Rețelele neurale bazate pe cunoștințe (KBANN: Knowledge-Based Artificial Neural Networks) sunt o metodologie hibridă neuro-simbolică fuzională ce poate procesa mulțimi de date complexe, de dimensiuni reduse, și cu o repartizare neuniformă, în scopul clasificării. Modulul simbolic cuprinde teoria domeniului sub forma unei mulțimi de reguli structurată ierarhic, iar modulul

conexionist asociază fiecărui concept din teoria domeniului câte un nod al rețelei neurale, prin traducerea structurii cunoștințelor din reguli în topologia rețelei.

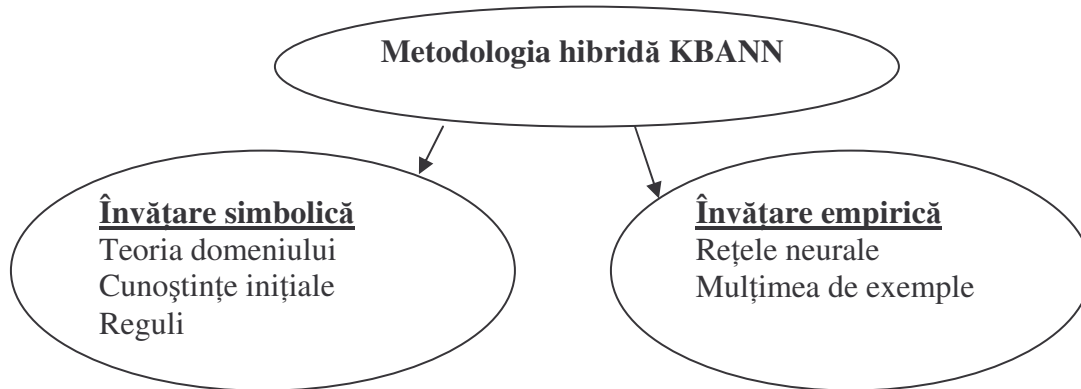


Figura 4.1. Cele două componente ale metodologiei KBANN

Sistemul a fost aplicat în clasificarea de țesuturi, cu rezultate satisfăcătoare în depistarea cancerului mamar.

4.2.1.2. Rețele neurale comitet. Rețelele comitet sunt sisteme hibride fuzionale ce cuprind rețele neurale organizate după modelul unui arbore de decizie, și conțin rețele neurale cu învățare supervizată organizate după principiul “divide et impera”. Învățarea este distribuită între experți (de obicei rețele neurale de tip perceptron) care partiționează spațiul de intrare. Mașina comitet este o combinație de experți, structurată după criterii ce vor deveni clare în continuare. Ca și perceptronul multistrat, rețelele comitet sunt aproximatori universali. Cuprind două mari subclase de arhitecturi: modelele statice și cele dinamice.

Modelele dinamice presupun că “integrarea cunoștințelor asimilate de experți se face sub acțiunea semnalului de intrare” și cuprind *mixtura de experți* și *mixtura de experți ierarhică*, exemple ce ilustrează foarte bine și proprietatea de modularitate. (O rețea neurală este *modulară* atunci când calculul efectuat de rețea poate fi descompus în două sau mai multe module / subsisteme ce operează pe intrări diferite fără să comunice între ele [Hay98].)

Modularizarea rețelelor neurale este o metodă de inserare a cunoștințelor explicite într-o rețea neurală, prin descompunerea unei probleme complexe în subprobleme mai simple. Avantajele modularizării se reflectă în creșterea vitezei de învățare, a abilităților de generalizare și reprezentare/interpretare, și proprietății de a satisface condițiile impuse de limitările hardware [NeaPal01]. Un avantaj în plus al rețelelor modulare este faptul că acestea pot fi o bază naturală de reprezentare a sistemelor de inferență fuzzy, constituind astfel un pas înainte către structurile hibride. Spre exemplu, un sistem de reguli fuzzy de tip Sugeno poate fi descris ca un anumit tip de rețea modulară: fiecărei reguli îi corespunde o rețea expert, iar rețeaua poartă definește contribuția fiecărei reguli la ieșirea finală.

4.2.2.Hibridizare neuro-fuzzy

Flexibilitatea și capacitățile de adaptare/ învățare/generalizare din exemple ale rețelelor neurale și avantajele reprezentării structurate și ușor interpretabile a cunoștințelor incerte din sistemele fuzzy de inferență pot fi combinate în cadrul unui formalism hibrid: calculul neuro-fuzzy, care și-a găsit până acum aplicații atât în problemele medicale de clasificare cât și în diagnoza tehnică.

Sistemele neuro-fuzzy se pot folosi ca generatoare de reguli și vin astfel de multe ori în completarea sistemelor ce folosesc cunoștințe furnizate de experți umani: acolo unde cunoștințele expertului uman sunt incomplete sau cunoștințele unor experți diferiți sunt contradictorii, modelul se completează folosind exemple clasificate pentru crearea regulilor. Prezentăm în continuare câteva exemple reprezentative de arhitecturi hibride și aplicațiile lor, care să ilustreze aceste afirmații, și care în plus ar fi aplicabile în diagnoza medicală.

4.2.2.1.ANFIS (Adaptive Neural Fuzzy Inference Systems). ANFIS reprezintă o arhitectură hibridă fuzională, de sisteme de inferență fuzzy extinse cu abilitățile de învățare ale rețelelor neurale pentru învățarea parametrilor ce definesc mulțimile fuzzy din antecedentul/ consecventul regulii, bazate pe rețele adaptive [Jan92].

Rețelele adaptive reprezintă un superset al rețelelor neurale feedforward cu învățare supervizată. O parte din noduri sunt adaptive, în sensul că ieșirea lor depinde de parametri ce sunt asociați cu fiecare din aceste noduri. Mecanismul de învățare (gradient descendent + chain rule) determină cum trebuie modificați acești parametri pentru a minimiza o măsură dată a erorii, și a fost introdus de Werbos în 1970.

Orice sistem de inferență fuzzy poate fi modelat printr-un sistem ANFIS, dovedindu-se echivalent funcțional, dar bucurându-se în plus de abilitățile de învățare specifice rețelei. Autorii arată în aceeași lucrare [Jan92] că sistemele de inferență fuzzy sunt echivalente, din punctul de vedere al funcției calculate, cu rețelele neurale de tip RBF (rețele cu funcții bază radiale), ce pot fi privite ca un caz special al ANFIS, dacă asociem fiecare regulă fuzzy DACĂ-ATUNCI cu un câmp receptor al rețelei RBF. (Această asociere este posibilă datorită proprietății de procesare locală a informației în rețelele RBF).

4.2.2.2.Clasificatorul neuro-fuzzy NFC. NFC este o arhitectură fuzională, concepută special pentru clasificare de forme [NFC=neuro-fuzzy classifier]. Este mai eficient decât clasificatorii neurali întrucât, pe de o parte, permite codificarea transparentă de cunoștințe apriori în parametrii rețelei și determină astfel învățarea în spațiul parametrilor să înceapă dintr-un punct inițial bun (aproape de optim). Pe de altă parte, parametrii obținuți după învățare se pot transforma ușor în cunoștințe structurate de forma regulilor IF-THEN.

4.2.2.3.Perceptronul Fuzzy Multistrat. Perceptronul fuzzy multistrat este un perceptron multistrat extins cu abilități de procesare de intrări-ieșiri fuzzy [Mit00], ce folosește hibridizare asociativă (fuzională și de transformare). Modelul fuzzy MLP

include mulțimile vagi la nivelul reprezentării intrărilor și ieșirilor, și conține un singur strat ascuns. Poate lucra atât cu intrări numerice cât și lingvistice. După ce rețeaua este antrenată cu backpropagation și ajustată (“pruned”) cu metoda de degradare a ponderilor (“weight decay”), se pot extrage reguli din ponderile rețelei, urmărind prin backtracking căile de pondere ridicată, de la stratul de ieșire spre cel de intrare, și asociind o regulă cu fiecare submulțime de noduri atașată (ce definește căile respective.)[Mit00] Arhitectura a fost folosită cu succes de către autori în extragerea de reguli pentru diagnosticarea afecțiunilor hepatice.

4.2.2.4. Neuronii fuzzy logici. Hibridizarea poate merge însă și mai departe, în interiorul rețelei, la nivelul structurii neuronilor (hibridizare fuzională). O arhitectură interesantă de rețea neurală fuzzy, bazată pe *neuronii fuzzy logici* a fost propusă de Witold Pedrycz [Ped03]. Aceasta conține mai multe tipuri de neuroni: de agregare (neuroni de tip ȘI sau SAU, a căror ieșire se calculează aplicând o t-normă sau, respectiv, o s-normă), neuroni logici de referință, ce folosesc la procesarea predicatelor logice de tipul: *similar, inclus în, domină*, și neuroni ce reprezintă modificatori lingvistici de tipul *foarte, puțin, mai mult sau mai puțin* (cu efect de diluare sau accentuare). Arhitectura și ponderile se pot antrena folosind algoritmi genetici. În final, rețeaua poate fi transformată într-o formulă logică transparentă, bine-structurată.

O posibilă aplicație a acestei arhitecturi se poate obține în diagnoza pe bază de funcții criteriu [Mun03], neuronii fuzzy logici servind la modelarea naturală a operatorilor și mulțimilor fuzzy ce apar în construcția funcțiilor respective. Această idee poate constitui o direcție de studiu interesantă, în cadrul tehnicilor de integrare a cunoștințelor simbolice în structuri neurale. Cunoștințele simbolice integrate pot preveni blocarea în minime locale și îmbunătățesc timpul de învățare al rețelei, fiind totodată utile în faza de explicare a rezultatelor furnizate de rețea. În plus, structura de rețea neurală ar ajuta inferența prin avantajele procesării paralele și robusteții la forme incomplete și distorsionate. Există totuși limitări ale capacității unei rețele neurale de a simula un raționament nemonoton, provenite tocmai din paralelismul rețelei și din lipsa abilităților de interacțiune cu utilizatorul (rețeaua nu poate pune întrebări care să ghideze raționamentul). Rămâne o problemă deschisă dacă și cum ar putea fi inserată interactivitatea și nonmonotonicitatea într-o structură hibridă neuro-fuzzy.

4.3. Un exemplu de hibridizare combinativă în diagnoza medicală

4.3.1. Sistemul CHECK

Problema diagnozei pe bază de model este în general NP-completă. Se pot însă folosi hibridizări combinate, în care primul nivel al sistemului de diagnoză se ocupă de focalizarea raționamentului, apelând la tehnici asociative de raționament. Reducerea spațiului de căutare în cazul diagnozei pe bază de model poate fi abordată și numeric, folosind măsuri de probabilitate/posibilitate ([TorCon89, Mun03]). Un exemplu de astfel de sistem hibrid este propus în CHECK. Sistemul se încadrează în tipul de hibridizare în care logica formală este asistată de o tehnică inteligentă (simbolică în cazul de față), și constituie o alternativă mult îmbunătățită a sistemului CASNET.

În [TorCon89] ni se propune o organizare hibridă (CHECK) (=ce folosește atât cunoștințe de profunzime cât și raționament asociativ), interesantă și originală a cunoștințelor pe 3 nivele: nivelul descrierii datelor (1), nivelul euristic (2)(folosit pentru inferență, include cunoștințe de suprafață), nivelul modelului cauzal (3) (folosit pentru explicații-include cunoștințe profunde). Sistemul este unul dintre cele mai complete și aprofundate sisteme de diagnoză și a fost folosit cu rezultate bune în diagnosticarea afecțiunilor ficatului. Pe toate cele trei nivele, reprezentarea cunoștințelor folosește cadre, cu sloturi specifice. La nivelul descrierii datelor există patru tipuri de informații (structurale, de control, prototipice, de interpretare). Nivelul euristic cuprinde cunoștințe structurale (de exemplu, o relație de ierarhizare între diagnostice -cazuri de bază/cazuri specializate=specifice), cunoștințe prototipice (manifestări necesare și suplimentare pentru un diagnostic), cunoștințe de control (reguli de activare și validare, ipoteze asociate și alternative unei ipoteze date, specializări default).

Descrierile prototipice ale unui diagnostic sunt folosite în evaluarea gradului de compatibilitate al datelor observate cu profilul unei anumite boli, și în asocierea unui rang de plauzibilitate fiecărei ipoteze de diagnostic. Evaluarea evidențelor se face corectând evidența observată cu măsura relevanței acesteia într-un anumit context și se face după formula:

$$E(H)=(((e(N) +_u e(S)) \circ_f [1-e(ER)]) +_u e(CR)) \quad (4.1)$$

unde:

$+_u$: $e_1 +_u e_2 = e_1 + (1-e_1)*e_1*e_2$; (varianta "unfair" a legii de combinare a evidențelor Bernoulli folosită în Mycin: primul membru-care corespunde în (4.1) manifestărilor necesare -este privilegiat, acordându-i-se o pondere mai mare);

\circ_f : $e_1 \circ_f e_2 =e_1*e_2$ (adaptat semanticii regulilor de excludere: gradul de evidență global nu trebuie să descrească dacă evidența din regulile de excludere este nulă);

$e(N)$: evidența din manifestările necesare; $e(S)$: evidența din manifestările suplimentare; $e(ER)$ =evidența din regulile de excludere; $e(CR)$ =evidența din regulile de confirmare.

Această evaluare a evidențelor este folosită de regulile de activare pentru instanțierea cadrului corespunzător unei ipoteze de diagnostic și trecerea sa în agenda de lucru. Regulile de validare sunt apoi folosite pentru confirmarea /excluderea instanței unui cadru (i.e. a unei ipoteze). Algoritmul de diagnoză corespunde unei căutări în lărgime (în paralel) la nivelul euristic, ceea ce face posibil și diagnosticul diferențial.

Nivelul cauzal cuprinde cunoștințele de profunzime și este folosit pentru confirmarea/infirmitatea unei ipoteze generate la nivelul euristic, generarea de ipoteze alternative sau analiza datelor "neașteptate"-poate fi folosit și pentru interogare. Pentru a modela raționamentul în această subcomponentă, se folosește o abordare calitativă bazată pe logica nemonotonă, care transformă grafurile asociate rețelei într-o mulțime de formule logice pe baza cărora se poate raționa calitativ. În această abordare, raționamentul logic nemonoton pe o rețea cauzală folosește *principiul rezoluției extins* pentru a găsi sursa unei inconsistențe. (Dacă o manifestare a unei stări lipsește, se aplică abducția indirectă pentru a găsi explicația acestei observații inconsistente). Dezavantajele principale ale abordării sunt cele enunțate în Secțiunea 3.3.1. în paragraful dedicat versiunii duale a abducției.

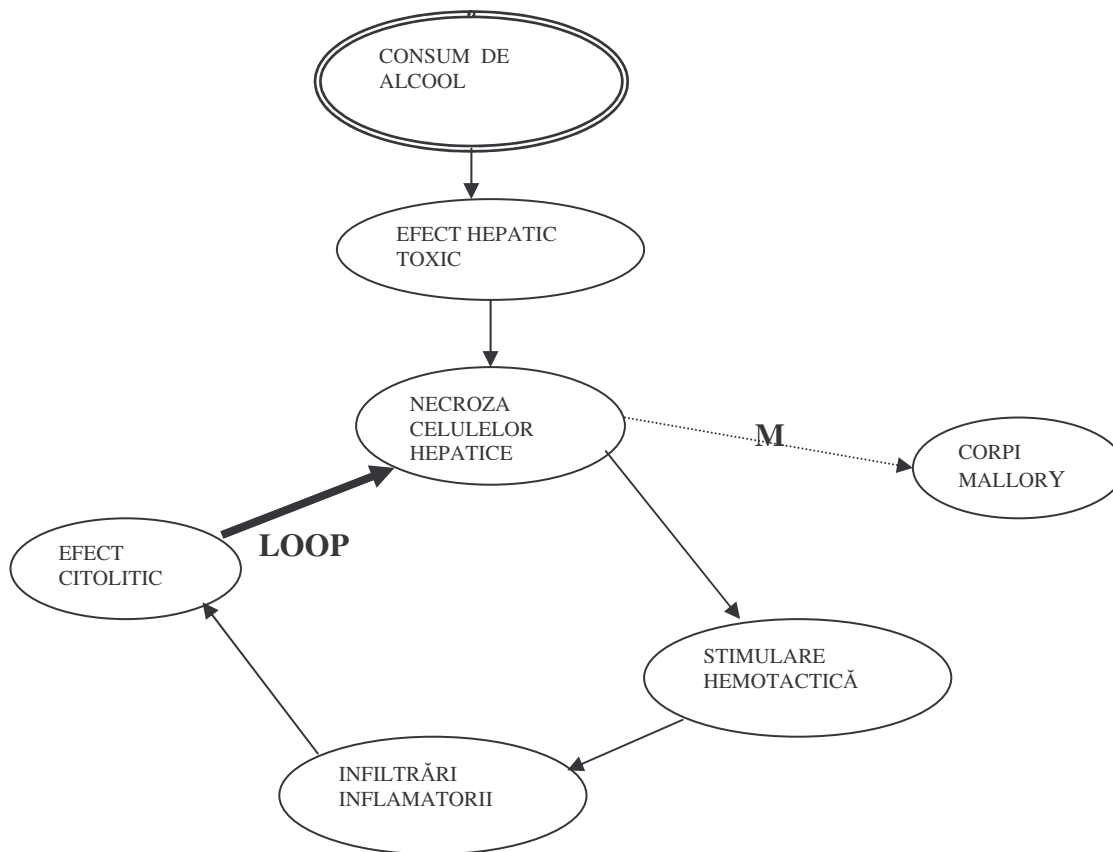


Figura 4.2. Rețea cauzală în CHECK

4.3.2 Comparație între sistemele CHECK și CASNET

Sistemul CHECK reușește să depășească toate cele trei mari dezavantaje ale sistemului CASNET enunțate în Secțiunea 3.2.1.3. Pe de o parte, prin activarea pe bază de reguli a anumitor ipoteze, raționamentul se concentrează pe un submodel din modelul medical total. Acestui submodel i se aplică, abia în faza de generare de explicații, raționament nemonoton, care abordează net superior contradicțiile. În plus, detalierea superioară a modelului și definirea explicită a mai multor tipuri de relații (de confirmare sau infirmare) va determina scăderea numărului de contradicții generate la pasul de explicare a concluziilor.

Totuși, încă nu putem modela în CHECK situația că o ipoteză e confirmată doar dacă *majoritatea* (criteriu vag) testelor sale sunt confirmate. Această problemă va fi rezolvată în DiaMed.

4.4.Sistemul hibrid DiaMed

În continuare, structura lucrării ține cont de obiectivul pe care ni l-am propus, și anume construirea unui sistem hibrid original de tip combinativ, structurat pe două nivele, pe

care l-am numit DiaMed (**Diagnoză Medicală**). Ne-am oprit asupra hibridizării combinate pentru a evita două mari dezavantaje ale hibridizării neuro-fuzzy sau neuro-simbolice, și anume: creșterea exponențială a dimensiunii problemei cu creșterea numărului de variabile (“curse of dimensionality”), și dificultatea de a se plia pe dinamismul unei probleme de diagnoză medicală.

Arhitectura DiaMed seamănă cu cea din sistemul CHECK, dar metodele efective de realizare sunt complet diferite. Primul nivel implementează selecția ipotezelor cu o metodă eficientă din categoria raționamentului asociativ, apelând în acest scop la funcții de decizie fuzzy pentru definirea funcțiilor de discriminare între clase. Avantajul major al acestor funcții de decizie față de legea combinării evidențelor din CHECK stă în faptul că pot exprima cu acuratețe criterii vagi de mare diversitate (de exemplu, *majoritatea, cel puțin x din n , un număr semnificativ din etc.*)

Al doilea nivel utilizează modelul causal de profunzime restrâns la contextul ipotezelor selectate la primul pas, pentru a discrimina și rafina diagnosticul final, și, eventual, pentru înlăturarea contradicțiilor generate la primul nivel. Reprezentarea modularizată are față de CASNET avantajul de a restrânge contextul, așa cum făceau și regulile de activare a ipotezelor (trigger) în CHECK. În plus, acest pas ia în considerare un model cât mai precis și complet, în care situațiile excepționale să poată fi reprezentate eficient și natural, și schema de raționament să se încadreze în specificul nemonoton al unui raționament de diagnoză. În acest scop, am apelat în cadrul acestui al doilea nivel la metode logico-simbolice din domeniul sistemelor de argumentare și la algoritmi CSP pentru rafinarea și explicarea rezultatelor. Avantajul principal față de CHECK al acestei abordări a nonmonotonicității la nivelul rafinării ipotezelor în DiaMed stă în faptul că folosește o metodă tratabilă, eficientă, spre deosebire de abordarea logică a abducției indirecte ce determină sursa eventualelor inconsistențe în CHECK.

4.5. Concluzii și contribuții

Capitolul IV a realizat mai mult decât o simplă trecere în revistă a sistemelor hibride de diagnoză medicală de succes de până astăzi. Întrucât am dorit ca în acest capitol să încadrăm deja în contextul actual sistemul original DiaMed propus de lucrarea prezentă, cu avantajele sale, am simțit nevoia detalierii anumitor formalisme, pentru ca secțiunile ulterioare să permită o mai bună comparație. Al doilea obiectiv a fost deschiderea de noi perspective asupra aplicabilității anumitor tehnici în diagnoza medicală în special. Secțiunile următoare vor analiza în detaliu sistemul original DiaMed, cu cele două nivele ale sale: selectarea ipotezelor și rafinarea mulțimii de diagnostice.

V. Selectarea ipotezelor de diagnostic prin decizie fuzzy în DiaMed

Operația de selectare a ipotezelor folosește, de obicei, tehnici eficiente dar mai puțin precise și aproape deloc transparente. Metodele cele mai populare de inteligență artificială ce folosesc cunoștințe superficiale pentru un raționament bazat pe asocieri sunt rețelele neurale [HayS98], sistemele bazate pe reguli [GiaRil94], anumite modele fuzzy [DubPra85]. Toate aceste tehnici modelează expertiza umană pe baza unor perechi intrări-ieșiri: corelația statistică fiind conceptul-cheie ce stă în spatele lor, indiferent de

formalizare. (Pentru o sinteză a abordărilor posibile ale diagnozei medicale, statistice sau pe bază de model vezi [Lon01]).

Secțiunea curentă prezintă contribuțiile originale pentru faza de selectare a ipotezelor în cadrul diagnosticării, contribuții ce cuprind un model original bazat pe funcții de decizie fuzzy [Mun03, Mun05]. Abordarea este comparată cu tehnica cea mai apropiată, și anume sistemele de inferență fuzzy. Puterea de discriminare a funcțiilor astfel construite este evaluată folosind indexul statistic C, rezultatele fiind foarte bune.

5.1. Sisteme de inferență fuzzy în diagnoza medicală

O problemă de diagnoză medicală se poate formaliza natural ca un proces de clasificare cu o mulțime de N clase (diagnostice) [CasFanMen03]:

$$\Delta = \{d_1, \dots, d_N\} \quad (5.1)$$

Mulțimea de simptome pe baza căreia se face clasificarea se poate scrie ca un vector K-dimensional $o = (o_1, \dots, o_K)$, iar diagnosticarea se face pe baza unei funcții clasificator:

$$D: A \subseteq \mathfrak{R}^K \rightarrow \Delta, \quad (5.2)$$

unde A se poate defini printr-un hiperinterval:

$$A = \prod_{j=1}^K \text{dom}_j, \quad o_j \in \text{dom}_j = [l_j, l_j]$$

Baza de reguli fuzzy ce stă în spatele funcției de clasificare D conține de obicei reguli de forma:

$$\text{IF } o \text{ is } G_r \text{ THEN } \tilde{D}(x) \text{ is } d_1(v_{r1}), \dots, d_N(v_{rN}), \quad (5.3)$$

unde G_r este o relație fuzzy K-dimensională.

Gradul de apartenență al vectorului de măsurători o în G_r definește gradul de activare al regulii r (și poate fi interpretat ca măsură a similarității între o și vectorul prototip al mulțimii G_r). Regulile de mai sus definesc funcția de clasificare :

$$\tilde{D}: A \rightarrow [0, 1]^M \quad (5.4)$$

adică $\tilde{D}(o) = (v_1, \dots, v_N)$, unde fiecare v_j reprezintă gradul în care este prezent diagnosticul d_j la un pacient cu simptome date de vectorul o , grad ce poate avea diferite semantici [KunSte99].

Gradul de apartenență al lui o la diagnosticul d_j calculat pe baza a R reguli de tipul (5.3) se calculează cu formula:

$$v_j = \frac{\sum_{r=1}^R G_r(o) v_{rj}}{\sum_{r=1}^R G_r(o)} \quad (5.5)$$

(unde $G_r(o)$ este gradul de apartenență al vectorului o la relația fuzzy G_r).

Decizia finală poate alege clasa cu grad maxim de realizare drept diagnostic, sau poate păstra în lista ipotezelor posibile toate diagnosticele al căror grad depășește un prag dat.

5.2. Decizie fuzzy în diagnoza medicală

Fie $\Delta = \{d_1, \dots, d_N\}$ un context restrâns de boli (clase/ipoteze de diagnostic) luate în considerare, și $M = \{m_1, \dots, m_K\}$ un set complet de manifestări ce le caracterizează. Considerăm deasemenea că Δ este o mulțime completă de cauze pentru M . Matricea $WEIGHTS = (w_{ij})_{j=1..K}^{i=1..N}$ conține indicii de relevanță ai simptomelor în cadrul definițiilor claselor, calculați statistic sau furnizați de un expert. Originalitatea abordării prezente constă în definirea claselor (bolilor) prin transformarea unui model de decizie fuzzy, după cum urmează:

- Fiecare simptom m_j este definit de o funcție fuzzy $M_j : dom_j \rightarrow [0,1]$, (dom_j reprezintă domeniul funcției M_j , și poate fi discret sau continuu, -dependent de simptom), și $A = \prod_{j=1}^K dom_j$ este mulțimea tuturor observațiilor posibile (Π este produsul cartezian; observațiile sunt luate în considerare pe domeniul tuturor simptomelor posibile, deși doar o mică parte sunt folosite efectiv în practică, întrucât restul sunt 0- nu au fost observate încă sau nu sunt prezente la sistemul sub observație);
- Observațiile sunt reprezentate de câte un punct K -dimensional $o = (M_1(o), \dots, M_K(o))$; -acest vector se poate modifica pe parcurs, pe măsură ce noi informații sunt obținute prin testare;
- Diagnosticele se definesc prin intermediul funcțiilor de decizie fuzzy: $D_1^{w_1}, \dots, D_N^{w_N}$, după cum urmează:

$$D_i^{w_i} = h_i(M_{i_1}(o), \dots, M_{i_{n_i}}(o)) \quad (5.6), \text{ unde:}$$

- $Simptome(d_i) = \{m_{i_1}, \dots, m_{i_{n_i}}\}$, și $i_1, \dots, i_{n_i} \in \{1, \dots, K\}$ reprezintă indicii simptomelor relevante pentru clasa de diagnostic d_i ;
- w_i este vectorul ponderilor simptomelor în cadrul definiției clasei d_i ;
- h_i este o funcție de agregare definită cu ajutorul unor operatori fuzzy ce modelează modul de raționare al unui expert uman.

Detaliem în continuare modul de calcul. Fie “ \wedge ”, “ \vee ” o T-normă și, respectiv, o S-normă[SouKay02]. Studiile desfășurate în cadrul logicii fuzzy și teoriei deciziei au demonstrat ca agregarea conjunctivă și disjunctivă sunt insuficiente pentru a modela corect comportamentul avut de experții umani în luarea unei decizii. Multe situații practice sugerează mai degrabă un mod de gândire compensator: proprietățile bune ale unor criterii compensează pentru proprietățile nesatisfăcătoare ale altora (diagnoza medicală este un exemplu foarte potrivit, întrucât observarea câtorva simptome dintr-o

mare diversitate este suficientă pentru a diagnostica o anumită boală). Definim mai jos tehnicile specifice de agregare compensatorie pe care le vom folosi.

Definiție 5.1 [SouKay02]. Un operator ordonat ponderat de medie (*ordered weighted averaging (OWA)*) a m variabile este o funcție $W : \mathfrak{R}^m \rightarrow \mathfrak{R}$ cu un vector asociat $w^T = (w_1, \dots, w_m)$, unde ponderile w_i satisfac:

$$\sum_{i=1}^m w_i = 1, \quad w_i \in [0,1], i = \{1, \dots, m\} \quad (5.7)$$

astfel încât:

$$W_w(b_1, \dots, b_m) = \sum_{i=1}^m w_i b_i, \quad (5.8)$$

folosind convenția $b_1 \leq \dots \leq b_m$. (Re-ordonarea este o proprietate fundamentală.) ■

Să presupunem că avem nevoie să exprimăm faptul că x criterii de diagnostic dintr-un număr dat trebuie îndeplinite, unde x poate fi o valoare crisp sau un criteriu vag de tipul “majoritatea”, “câteva” etc. Aceasta se poate obține prin alegerea unui vector de ponderi cu toate pozițiile nule, în afara a exact x poziții. Dacă, spre exemplu, $w^T = (0 \ 0 \ 0 \ 0.25 \ 0.25)$, W_w calculat cu Ecuația 5.8 va depăși pragul de 0.5 doar dacă cel puțin 2 din cele 5 criterii (modelate, după cum am precizat, de funcțiile fuzzy M_j), sunt îndeplinite în grad maxim (=1). Pentru a simplifica expunerea, vom considera de acum înainte simptomele prezente sau absente (adică, $\text{dom}_j = \{0,1\}$ - studiul poate fi cu ușurință extins la cazul general).

Operatorii OWA sunt cazuri speciale de integrale fuzzy, acestea din urmă fiind mai potrivite pentru calcularea scorului claselor în situațiile în care “mai multe” nu este specificat exact, și unde criteriile de diagnostic nu sunt precis definite (în diagnoza medicală ne vom referi la aceste boli drept boli de “tip B”, spre deosebire de bolile de “tip A” ce dispun de criterii ca cel din paragraful precedent).

Definiție 5.2[SouKay02] Fie $C = \{c_1, \dots, c_m\}$ mulțimea criteriilor de decizie. Fie $P(C)$ mulțimea părților lui C . O *măsură fuzzy* pe C este o funcție $g: P(C) \rightarrow [0,1]$ care satisface:

1. $g(\emptyset) = 0, g(C) = 1$;
2. $A \subset B \subset C \Rightarrow g(A) \leq g(B)$. (5.9) ■

Definiție 5.3.[SouKay02] Fiind dată o măsură fuzzy g pe C , *integrala fuzzy Sugeno* a funcției $p: C \rightarrow [0,1]$ în raport cu g este definită de:

$$S_g(p(c_1), \dots, p(c_m)) = \bigvee_{i=1}^m (p(c_{(i)}) \wedge g(A_{(i)})). \quad (5.10)$$

În ecuația (4.1.5) indexul inferior “._(i)” indică faptul că indicii au fost permutați astfel încât $0 \leq p(c_{(1)}) \leq \dots \leq p(c_{(m)}) \leq 1$, și $A_{(i)} = \{c_{(i)}, \dots, c_{(m)}\}$. ■

Pentru bolile de tip B, fie C_B mulțimea simptomelor relevante, $A \subseteq C_B$ și $g(A)$ ponderea normalizată a submulțimii A (i.e. suma ponderilor simptomelor din A împărțită la suma tuturor ponderilor simptomelor din C_B). Considerăm $p(m_j)$ valoarea criteriului

(simptomului) m_j la un anumit pacient (sistem) p . Această interpretare ne permite să dăm următoarea definiție.

Definiție 5.4. Fie $d_i \in \Delta$, p un pacient sub observație. Scorul bolii d_i de tip B la pacientul dat p , asociat vectorului de observații o este definit ca:

$$\text{Scor}(p, d_i) = S_g(M_{i_1}(o), \dots, M_{i_{n_i}}(o)) \quad (5.11)$$

(unde g este definit folosind w_i , iar măsura fuzzy a unei mulțimi de simptome este suma normalizată a ponderilor lor, și S_g este integrala Sugeno definită mai sus). ■

Pentru bolile de tip A , trebuie să folosim o abordare ușor diferită pentru a include ponderile în model. Funcția de decizie are aici o nouă formă (vezi Ecuația 5.13), și trebuie ca sensibilitatea ei față de criterii să fie în concordanță cu ponderea fiecărui criteriu. Aceasta se poate obține prin cerința ca funcția de decizie să satisfacă proprietăți suplimentare.

Definiție 5.5[SouKay02] O funcție generator a unui operator de agregare a mulțimilor fuzzy este o funcție $f: [0,1] \rightarrow \mathfrak{R} \subseteq [-\infty, \infty]$ continuă și strict monotonă. Funcția generator se spune că este un generator crescător (generator descrescător) dacă este monoton crescătoare (descrescătoare) în intervalul $[0,1]$. ■

Propoziție. [SouKay02] Un operator de agregare continuu $v(a,b)$ pentru $a, b \in [0,1]$ poate fi reprezentat prin intermediul funcției sale generator f ca:

$$v(a,b) = f^{-1}(k(f(a)+f(b))), \text{ cu } k > 0,$$

dacă operatorul respectiv este o t -normă archimedeană, o t -conormă archimedeană, un operator de medie generalizat (și câteva altele). ■

Deci reprezentarea pe bază de funcție generator a unei funcții de decizie D pe bază de t -norme/ t -conorme archimedee este:

$$D(a,b) = f^{-1}(f(a) + f(b)), \text{ (} f \text{ este funcția generator),}$$

și se poate extinde într-o funcție de decizie ponderată prin transformarea spațiului de decizie astfel:

$$D^w(a,b) = f^{-1}(w_a f(a) + w_b f(b)), w = (w_a, w_b); \quad (5.12) \text{ unde } w \text{ este vectorul de ponderi.}$$

Așadar, dacă $d_i \in \Delta$ este o boală de tip A și simptomele lui d_i sunt împărțite în n_{d_i} clase $C_1, \dots, C_{n_{d_i}}$, criteriul de diagnostic este definit de: $d_i \equiv h_i(W_1 C_1, \dots, W_{n_d} C_{n_d})$, unde W_i este un operator OWA pentru "cel puțin x din $|C_k|$ " definit ca mai sus, și h_i este funcția de agregare construită pe baza unei ierarhii de t -norme și t -conorme.

Exemplu 5.1. Reumatismul Articular Acut (RAA) este o boală de tip A . Tabloul clinic, ponderile simptomelor (definite de un expert, $\in [0,1]$) și criteriile de diagnostic sunt prezentate mai jos.

Criteriile Jones pentru diagnoza RAA [PopIon99]

1. Manifestări majore (C_1)

- Cardită 0.7

- Poliartrită 0.9
 - Coree 0.7
 - Eritem marginal 0.6
 - Noduli subcutanați 0.6
2. Manifestări minore (C_2)
- Febră 0.4
 - Artralgie 0.6
 - Factor reumatoid seric anormal 0.8
 - Proteina C reactivă crescută 0.8
3. Infecție streptococică (C_3)
- Titru crescut de ASLO 0.7
 - Streptococ din grupa A în faringe 0.6
 - Scarlatină 0.5

Criteriu de diagnostic: prezența a 2 criterii majore, sau a unui criteriu major și 2 minore sugerează o mare probabilitate pentru RAA, dacă a fost demonstrată infecția streptococică.

Criteriul general de diagnostic se traduce ca:

$$D_{AAR} = (W^{11} C_1 \wedge W^3 C_3) \vee (W^{12} C_1 \wedge W^2 C_2 \wedge W^3 C_3) \quad (5.13),$$

unde W^k sunt operatori OWA folosiți în definirea criteriului de diagnostic și bazați pe următoarele ponderi, respectiv:

$$w^{11} = (0 \ 0 \ 0 \ 0.25 \ 0.25), \text{ (cel puțin 2 din 5)}$$

$$w^{12} = (0 \ 0 \ 0 \ 0 \ 0.5), \text{ (cel puțin 1 din 5)}$$

$$w^2 = (0 \ 0 \ 0.25 \ 0.25), \text{ (cel puțin 2 din 4)}$$

$$w^3 = (0 \ 0 \ 0.5) \text{ (cel puțin 1 din 3),}$$

astfel încât, dacă $C_k = \{m_1^k, \dots, m_{n_k}^k\}$ este mulțimea manifestărilor din clasa C_k , putem scrie:

$$W^k C_k(o) = W^k (M_1^k(o), \dots, M_{n_k}^k(o)) = \sum_{j=1}^{n_k} w_j^k M_j^k(o) \quad (5.14),$$

$M_j^k(o)$ fiind valoarea observată a simptomului m_j^k (0 sau 1, după cum am convenit) la pacientul dat p , o fiind vectorul de observații. Ponderile de mai sus nu trebuie confundate cu ponderile simptomelor **observate**:

$$W_{AAR} = (W_{AAR}^1, W_{AAR}^2, W_{AAR}^3) \quad (5.15),$$

partiționate după cele 3 clase de simptome (w_{AAR}^k este vectorul de ponderi pentru simptomele observate ce aparțin clasei k).

Am ales să lucrăm cu perechea de norme duale $T(a,b)=ab$ și $S(a,b)= a+b-ab$, astfel încât prin înlocuirea operatorului “ \vee ” în 5.13, putem scrie:

$$D_{AAR}=1-[1-(W^{11}C_1 \wedge W^3C_3)]^{w_{13}}*[1-(W^{12}C_1 \wedge W^2C_2 \wedge W^3C_3)]^{w_{123}} \quad (5.16),$$

unde $w_{13}=w(C_1) + w(C_3)$; $w_{123}=w(C_1) + w(C_2) + w(C_3)$, și $w(C_i) = w_i$ este suma ponderilor simptomelor observate din clasa C_i , normalizată cu suma ponderilor tuturor simptomelor observate (am omis o din $W^k C_k(o)$ pentru a simplifica notația).

Să presupunem că avem, într-o notație simplificată, $o=\{\text{coree, artralgie, factor reumatoid seric anormal, streptococ din grupa A în faringe}\}$ - datele unui pacient suspect de RAA.

$$\text{Atunci } D_{AAR}=1-[1-((W^{11}C_1)^{w_1} * (W^3C_3)^{w_3})]^{w_{13}}*[1-((W^{12}C_1)^{w_1} * (W^2C_2)^{w_2} * (W^3C_3)^{w_3})]^{w_{123}}; \quad (5.17)$$

$$w_1=0.26, \quad w_2=0.5, \quad w_3=0.22;$$

$$W^{11}C_1=0.25, \quad W^{12}C_1=0.5, \quad W^2C_2=0.5, \quad W^3C_3=0.5.$$

Rezultatul final este $D_{AAR}(o)=0.62$, ceea ce înseamnă că RAA va fi într-adevăr selectată ca ipoteză de diagnostic și trecută printr-un proces de discriminare.

5.3.Evaluarea capacității de discriminare a unui test de diagnoză

În evaluarea sistemului de asistare a deciziilor medicale construit de noi ne vom folosi de indexul C (echivalentul ariei de sub curba ROC-Receiver Operator Characteristic). Acesta măsoară capacitatea de discriminare a sistemului, adică de a distinge între cazuri ce fac parte din clase diferite.

Presupunem că avem acces la rezultatul real al clasificării (“gold standard”) – ce poate fi 0 (pacientul este sănătos) sau 1 (pacientul suferă de boala D). Ieșirea sistemului este un număr real între 0 și 1. Practic, se alege un prag, de exemplu 0.5 și clasificarea propriu-zisă se face prin compararea cu acest prag (dacă scorul obținut este > 0.5 , se consideră că este bolnav). Aria de sub curba ROC este o estimare a puterii de discriminare a testului (este media sensibilității asupra tuturor specificităților). Dacă valoarea ariei este aproape de 0.5, testul este inutil, calitatea sa fiind cu atât mai bună cu cât aria se apropie de 1 (pentru 1 suntem în situația unei discriminări perfecte).

5.4.Clasificare prin decizie fuzzy

Am ales pentru exemplificare o pereche de boli care pune de obicei proleme de recunoaștere (*ciroza și insuficiența cardiacă*) întrucât există situații când insuficiența cardiacă se manifestă prin simptome asemănătoare cirozei și este prin urmare confundată cu aceasta. Am comparat indexul C obținut prin clasificarea cu o funcție de decizie fuzzy construită după modelul descris în [Mun05], pentru perechi diferite de norme duale (utilizate în calculul integralei fuzzy). Indicele de calibrare din ultima coloană măsoară sensibilitatea calculului (este o măsură simplă de genul sumă de diferențe, care stă la baza

majorității metodelor de calibrare- i.e. aducere a scorului bolii cât mai aproape de probabilitatea reală).

S-NORMA	T-NORMA	INDEX C	INDICE DE CALIBRARE
max	min	0.9856	546.6659
a+b-ab	ab	0.9856	1186.5055
min(a+b, 1)	max(a+b-1, 0)	0.9856	1632.4998

Tabel 5.1. Indexul C și calibrarea pentru testul “ciroză”

În urma acestui gen de experimente ne-am oprit asupra perechii (max, min) pentru sistemul final. Presupunem că superioritatea acestei perechi de norme a provenit din faptul că sunt mai puțin sensibile la ponderile simptomelor, care probabil nu au fost setate foarte corect. Au fost evidente avantajele abordării fuzzy: integrarea naturală a cunoștințelor expert în model.

S-NORMA	T-NORMA	INDEX C	INDICE DE CALIBRARE
max	min	0.956	392.1201
a+b-ab	ab	0.9088	908.2604
min(a+b, 1)	max(a+b-1, 0)	0.9064	1243.3226

Tabel 5.2. Indexul C și calibrarea pentru testul “insuficiență cardiacă congestivă”

Principalul dezavantaj pe care l-am întâlnit aici reprezintă o problemă încă deschisă în comunitatea științifică: lipsa unui indice riguros fundamentat științific de evaluare a puterii de discriminare în diagnosticul cu răspuns (posibil) multiplu. Dacă ar fi să abordăm problema în totalitate, nu doar pentru o anumită boală, cum ar trebui modificat/adaptat indexul C? Cum s-ar redefini însăși noțiunea de discriminare în noul context? La aceste întrebări deocamdată nu există un răspuns clar.

5.5. Concluzii și contribuții

Cheia reprezentării folosite mai sus este să privim simptomele drept criterii fuzzy ce definesc un diagnostic, în măsura în care sunt îndeplinite. Gradele de potrivire $D_i^{w_i}(o)$ între vectorul de observații și criteriul complex ce definește diagnosticul d_i , conduc la o ierarhizare a ipotezelor de diagnostic, pe baza evidențelor particulare ale unui caz dat. Mulțimea finală a diagnosticelor se determină prin aplicarea unui prag de semnificație, care ar trebui să fie altul pentru fiecare boală în parte (pentru a evita dezavantajele variabilității sumei totale de ponderi de la o boală la alta).

Considerăm că modelul de decizie fuzzy folosit de noi în DiaMed pentru selecția ipotezelor de diagnostic constituie o alternativă viabilă și mai eficientă a abordării fuzzy prezentate în secțiunea anterioară. Cu ajutorul său, fiecare bază de reguli fuzzy corespunzătoare unei anumite variabile de ieșire y se poate rescrie ca o funcție de decizie fuzzy după modelul descris. Comparând cu abordarea din Secțiunea 5.1, în loc să agregăm scorul v_j al unei boli din R reguli, îl putem calcula direct pe baza funcției de decizie asociată bolii respective.

Astfel, dacă avem mai multe variabile de ieșire- (m ieșiri), selectarea ipotezelor de defect va avea nevoie doar de m funcții de decizie fuzzy. În plus, aceste funcții sunt formate cu operatori de agregare fuzzy mai complecși, mai naturali, mai nuanțați și mai ușor de interpretat și citit (folosind, spre exemplu, operatorii compensatori, integralele fuzzy etc.), decât o bază formată în exclusivitate cu operatorul AND.

Un mare avantaj al funcțiilor de decizie fuzzy descrise mai sus este dat și de faptul că sunt mai ușor de construit decât bazele de inferență fuzzy, în condițiile în care există cunoștințe expert suficient de clare și de precise. Pentru aceasta însă este necesar să avem un model corespunzător adaptat: nu vom mai considera valoarea “mică” sau “mare” a unui anumit element drept un defect în sine, ci este necesar să reprezentăm valoarea elementului ca o funcție de alte elemente din sistem, de care depinde direct sau indirect (vezi Figura 8.1).

În final, în abordarea noastră, în loc să agregăm grade de realizare diferite ale valorilor fuzzy (mare, mic etc) pentru o singură variabilă de ieșire, obținem direct un număr pe care îl considerăm mare sau mic nu atât raportat la o scară absolută, cât mai mult raportat la un anumit context (în relație cu valorile obținute de celelate variabile de ieșire)

În plus, avantajul major al funcțiilor de decizie definite mai sus față de legea combinării evidențelor din CHECK, stă în faptul că pot exprima cu acuratețe criterii vagi de mare diversitate (de exemplu, *majoritatea, cel puțin x din n , un număr semnificativ din etc.*)

Așadar, principalele avantaje ale abordării constau în simplitatea și eficiența modului de calcul (ce ia în considerare și specificitatea fiecărui simptom în definițiile bolilor), și în focalizarea rapidă a căutării spre direcții relevante. Dezavantajele ce rezultă din lipsa facilităților explicative și din ignorarea deliberată a interacțiunilor posibile între ipoteze vor fi înlăturate de metodele prezentate în Capitolele VI, VIII.

VI.Sisteme de argumentare directă și algoritmi de satisfacere a constrângerilor în modelarea raționamentului nemonoton

În capitolul prezent, aprofundăm tehnicile de raționament nemonoton ce vor constitui suportul teoretic pentru modelarea nivelului de discriminare al sistemului hibrid DiaMed. Sunt introduse sistemele de argumentare directă, cu definirea unor noțiuni elementare, și cu sublinierea avantajelor și motivațiilor care au susținut dezvoltarea acestor sisteme (ce aparțin logicii informale). Este prezentată semantica argumentativă a admisibilității, pe care o folosim în DiaMed cu rolul unui mecanism de menținere a consistenței (aceasta din urmă constituind una din noțiunile necesare în definirea abducției). Pentru calculul

efectiv al acestei semantici am optat pentru algoritmi specifici problemelor de satisfacere a constrângerilor (CSP), plecând de la faptul că admisibilitatea se poate rescrie în termenii verificării satisfiabilității unei formule logice, pentru care este natural să folosim algoritmi menționați.

Secțiunea discută, deasemenea, versiunea dinamică a problemelor CSP (DCSP), mai potrivită unei probleme al cărei context se modifică dependent de teste, cum este diagnoza medicală. Ca implementare efectivă pentru o problemă de tip DCSP, am ales algoritmul backtracking dinamic al lui Ginsberg, întrucât este una din alternativele cele mai eficiente și mai adecvate. Secțiunea prezintă și o versiune de adaptare efectivă a backtracking-ului dinamic pentru DCSP [VerSchi], adaptare pe care o vom modifica la rândul nostru în cadrul nivelului de discriminare între diagnostice din DiaMed.

În secțiunea prezentă, aprofundăm tehnicile de raționament nemonoton ce vor constitui suportul teoretic pentru modelarea nivelului de discriminare al sistemului hibrid DiaMed. Sunt introduse sistemele de argumentare directă, cu definirea unor noțiuni elementare, și cu sublinierea avantajelor și motivațiilor care au susținut dezvoltarea acestor sisteme (ce aparțin logicii informale). Este prezentată semantica argumentativă a admisibilității, pe care o folosim în DiaMed cu rolul unui mecanism de menținere a consistenței (aceasta din urmă constituind una din noțiunile necesare în definirea abducției). Pentru calculul efectiv al acestei semantici am optat pentru algoritmi specifici problemelor de satisfacere a constrângerilor (CSP), plecând de la faptul că admisibilitatea se poate rescrie în termenii verificării satisfiabilității unei formule logice, pentru care este natural să folosim algoritmi menționați.

Secțiunea discută, deasemenea, versiunea dinamică a problemelor CSP (DCSP), mai potrivită unei probleme al cărei context se modifică dependent de teste, cum este diagnoza medicală. Ca implementare efectivă pentru o problemă de tip DCSP, am ales algoritmul backtracking dinamic al lui Ginsberg, întrucât este una din alternativele cele mai eficiente și mai adecvate. Secțiunea prezintă și o versiune de adaptare efectivă a backtracking-ului dinamic pentru DCSP [VerSchi], adaptare pe care o vom modifica la rândul nostru în cadrul nivelului de discriminare între diagnostice din DiaMed.

6.1. Sisteme de argumentare directă

Logica de ordinul întâi, deși reprezintă un formalism puternic de reprezentare, și-a demonstrat limitările în privința multor probleme de Inteligență Artificială. Aceste limitări își au originea în două cauze principale: monotonicitatea și capacitatea redusă de a reprezenta incertitudinea, de care este legat și inconvenientul semidecidabilității.

Definiție. Un mecanism de inferență este *monoton*, dacă pentru o teorie dată Δ avem:

$$T(\Delta \cup S) \supseteq T(\Delta), \quad (6.1)$$

oricare ar fi setul de propoziții S ($T(\Delta)$ este mulțimea de propoziții derivabile din Δ).

(O propoziție adevărată într-o teorie rămâne adevărată după adăugarea de noi propoziții la teoria respectivă).

Inferența nemonotonă este opusul celei monotone (întrucât cunoștințe noi pot contrazice ipoteze vechi), și este mijlocul de modelare a incertitudinii cu mijloace ale logicii (formale sau informale) [Cla90]. Conceptul a devenit necesar în momentul în care s-a înțeles că incertitudinea probabilistă nu este suficientă pentru reprezentarea problemelor lumii reale. Spre exemplu:

“Cele mai multe păsări zboară.”

este o propoziție probabilistă, bazată pe statistici calculate asupra unor populații de păsări. Se poate ca un sistem probabilist incert să rămână monoton, după modelul rețelelor bayesiene. Dacă însă vom dori să considerăm exemplul:

“Păsările zboară, dacă nu se poate demonstra contrariul.”

avem deja o afirmație pur logică, ce necesită o reprezentare distinctă a incertitudinii pe care o reprezintă.

Non-monotonicitatea poate avea expresii diferite în cadrul diverselor formalizări (unele dintre ele mai expresive, altele -mai eficiente), și a fost abordată în programarea logică (de exemplu, completarea predicatelor, programarea logică generală, programarea logică disjunctivă, semanticile stabilă și admisibilă -[BarGel94], [Dix55], [PerApAI93]), sau în circumscripția lui McCarthy [Cart80] (o formă mai generală de completare). Exemplele enumerate fac parte din “sistemele de argumentare indirectă” [Kon88]. Noțiunea de **argument** apare în cadrul lor implicit, în sensul general de “demonstrație –formală sau informală- a adevărului unei formule pe baza adevărului premizelor”. Există însă sisteme în care argumentele sunt obiecte explicite ale formalismului – așa numitele “sisteme de argumentare directă” [Kon88]. În cazul din urmă, non-monotonicitatea se reduce la o relație de atac între structuri monotone- argumentele ([BDKT97], [Pol02], [Vre92]). Vom justifica în continuare preferința noastră pentru acestea.

Tehnicile de completare a teoriei permit deducerea de ipoteze globale pornind de la ceea ce nu a putut fi dedus în cadrul teoriei. Pentru a reprezenta ipotezele implicite (o propoziție este implicit validă dacă nu există evidențe împotriva ei), se folosesc reprezentări mai complexe, -cum ar fi logica nemonotonă a lui McDermott și Doyle ([DerDoy80][Der82]), sau logica implicită a lui Reiter [Rei80]. În conformitate cu aceste formalisme, o concluzie este acceptată dacă nu introduce inconsistențe la nivelul global al teoriei. În afara faptului că testarea consistenței este doar semi-decidabilă și foarte costisitoare computațional, abordările enumerate suferă de lipsa unui mecanism determinist de selectare a unei extensii particulare (=o mulțime consistentă de concluzii), atunci când se generează mai multe. Rezolvarea unui conflict (atunci când există argumente pro și contra pentru aceeași concluzie) este, așadar, o sarcină dificilă pentru "sistemele de argumentare indirectă" [Kon88]. Sistemele de argumentare directă, în schimb, dispun de o reprezentare explicită a argumentelor, care reduce rezolvarea conflictelor la o relație de înfrângere între aceste structuri.

Semantica sistemelor de argumentare directă înlocuiește teoria modelelor a lui Tarski (semantica teoretică pe bază de model) substituind “adevărat” cu “justificat”: o concluzie este justificată dacă există un argument neînfrânt ce o susține în contextul evidențelor actuale. Această înlocuire a teoriei modelelor se bazează pe concluziile mai multor cercetători (Pollock, Loui, Vreeswijk), că raționamentul nemonoton are nevoie de

o semantică diferită. După cum observau autorii în [PraVre], teoria modelelor a fost folosită în mod tradițional pentru a defini înțelesul limbajelor logice pornind de la *realitate*: înțelesul unor simboluri logice se definește plecând de la cum arată lumea, dacă o expresie cu aceste simboluri este adevărată, și definește consecința logică cercetând ce altceva este adevărat, pornind de la premisele date. Prin contrast, atunci când discutăm despre “adevărat în general, dacă nu sunt îndeplinite anumite excepții”, această clasificare normal/exceptional cuprinde deja condensată în ea, implicit, o parte de demonstrație (de raționament). Noțiunile de atac și înfrângere, centrale în raționamentul invalidabil, nu sunt ‘propoziționale’ întrucât înțelesul lor nu este dat de corespondența unei propoziții cu lumea [PraVre], ci de relațiile între propoziții, stabilite dependent de un context dinamic de evidențe. S-a ajuns astfel la semantica argumentativă amintită.

Un avantaj în plus din punctul de vedere al lucrării de față stă în faptul că genul de raționament de diagnoză de care avem nevoie presupune revizuirea iterativă a încrederilor, pentru care putem folosi natural formalismul sistemelor de argumentare directă (conceput special pentru simularea cât mai fidelă a raționamentului uman în fața unor situații complexe). Acest formalism este, în plus, adaptabil interactivității necesare pentru a lua în calcul testele efectuate secvențial, generarea de ipoteze alternând cu obținerea de noi informații.

Definiție[Vre92]. Un *sistem de argumentare abstract* A este un triplet (L, R, \leq) , unde L este un limbaj, R este o mulțime de reguli de inferență, și “ \leq ” este o relație de ordine reflexivă și tranzitivă pe mulțimea argumentelor.

Definiție[GovMah00]. Un *argument* pentru literalul p este un arbore de demonstrare (folosind derivare monotonă) pentru p , construit pe baza regulilor de inferență din R .

Cele mai utilizate tehnici de rezolvare a conflictelor sunt: "principiul specificității" (Toulmin: “informația mai specifică suprascrie informația mai generală” [Tou58], metode dependente de domeniu, metode numerice (acestea din urmă asignează un număr –spre exemplu, tăria celei mai slabe legături- fiecărui argument [Pol02])).

Definiție[Vre92]. O *mulțime de bază* este o submulțime finită compatibilă a lui L .

O mulțime de bază conține “informație ireductibilă” –ea constituie punctul de plecare într-un raționament forward, și punctul final al raționamentului backward.

Partea interesantă a sistemelor directe de argumentare, din punctul de vedere al diagnozei, este faptul că abducția poate fi modelată mai eficient în cadrul lor decât într-un cadru ipotetico-deductiv. Noțiunea de explicație este înlocuită natural cu cea de argument, anularea ipotezelor cu cea de rezolvare a conflictelor între argumente, și observațiile inițiale pot fi considerate drept o mulțime de bază. Vom utiliza în lucrarea de față un cadru abductiv bazat pe argumente atât pentru fundamentarea și discriminarea ipotezelor de diagnostic, cât și pentru menținerea coerenței concluziilor finale.

Un cadru argumentativ pentru generarea de explicații a fost deja dezvoltat sub denumirea WOZ (Wizard of OZ) [ShaMus99], plecând de la structura argumentelor propusă de Toulmin [Tou58]. Componenta de generare a explicațiilor a fost integrată într-un sistem multiagent de decizie, cu aplicații în recomandările terapeutice, rezultatul final numindu-se ATHENA DSS [Sha00] – un instrument puternic de asistare a personalului medical în deciziile sale, și singurul ce folosește această metodologie până acum, din cunoștințele

noastre. Abordarea curentă duce ideea un pas mai departe: structura argumentelor a lui Toulmin nu este suficient de expresivă pentru a surprinde întreaga complexitate a raționamentului de diagnoză (spre exemplu, nu putem reprezenta excepțiile la excepții), astfel că vom propune în continuare o schemă argumentativă mai complexă. În plus, scopul principal al autorilor sistemului ATHENA DSS este implementarea bazată pe agenți, nu detaliile raționamentului nemonoton, acestea din urmă fiind tratate într-o manieră simplificată.

Introducem în continuare cadrele de lucru argumentative ce au constituit o sursă de inspirație pentru abordarea proprie.

Sistemul de argumentație al lui Dung. Dung a fost unul din pionierii acestui domeniu. El a propus un sistem de argumentare ce abstractizează structura internă a unui argument. Majoritatea abordărilor raționamentului nemonoton din inteligența artificială și programarea logică pot fi privite drept cazuri particulare ale acestui sistem. Cadrul său conține o noțiune primitivă de argument și o relație binară de înfrângere:

Definiție.[Dun95]

1. Un *cadru de argumentare* AF (“Argumentation Framework”) este o pereche

$$AF = \langle AR, atacuri \rangle, \quad (6.2)$$

unde AR este o mulțime de argumente, și $atacuri$ este o relație binară pe AR : $atacuri \subseteq AR \times AR$;

2. Un AF este *finit* dacă fiecare argument din AR este înfrânt de către cel mult un număr finit de argumente;
3. O mulțime S de argumente este *fără conflicte* dacă nu există argumentele A, B în S astfel încât A atacă B .

Noțiunea de *acceptabilitate* este centrală în argumentare, întrucât definește de fapt relația argumentării cu raționamentul nemonoton. Nu există deocamdată o metodă de calcul universală a acestei noțiuni.

Definiție([Dun95]). Un argument A este *acceptabil* relativ la o mulțime S de argumente ddacă orice argument care înfrânge A este înfrânt de un argument din S .

Definiție([Dun95]). O mulțime de argumente S fără conflicte este *admisibilă* ddacă fiecare argument din S este acceptabil în raport cu S .

Semantica admisibilă (credulă) a fost introdusă de Dung pentru programarea logică și este de preferat din punctul de vedere al teoriei argumentației întrucât îmbunătățește mult semantica stabilă (sceptică-i.e. atacă tot ce nu conține), care nu poate surprinde înțelesul intuitiv al unor cadre de argumentare, și al cărei caracter *global* o face intratabilă în general. Prin contrast, extensii admisibile există pentru orice cadru de argumentare, indiferent de structura de atac asociată, și, în plus, admisibilitatea este mai tratabilă computațional întrucât este *locală*: se construiește mai întâi o mulțime de ipoteze care implică monoton propoziția pe care dorim să o justificăm, apoi se augmentează această mulțime cu “apărări” împotriva oricărui atac posibil. Pe baza acestor observații, propunem o definiție originală și naturală a unei mulțimi de diagnostic multiplu.

Definiție. Un *diagnostic multiplu* este o mulțime admisibilă de ipoteze care acoperă toate observațiile, și este minimală cu această proprietate. (Vom folosi această definiție în algoritmul din Capitolul VIII.)

Exemplu. Fie 3 argumente $A=\{\text{leucemie}\}$, $B=\{\text{nivel PT,PTT normal}\}$, $C=\{\text{medicamente care afectează nivelul PT,PTT}\}$. Relația de atac este dată de graful: $A\leftarrow B\leftarrow C$. Atunci A este acceptabil în raport cu C, C fiind o apărare a lui A față de atacul B. Deci $\{A,C\}$ este o mulțime admisibilă, și în acest caz "leucemie" este un diagnostic posibil în contextul definit de C.

Definiție[PraVre]. O *asignare de stare* unei mulțimi X de argumente ordonate printr-o relație binară de înfrângere este asignarea fiecărui argument fie a stării "in", fie a stării "out", astfel încât să fie satisfacute condițiile:

1. Un argument este "in" dacă toate argumentele care îl anulează sunt "out";
2. Un argument este "out" dacă este anulat de un argument "in".

Remarcă. Atât în abordarea credulă, cât și în cea sceptică, ciclurile impare de înfrângere nu au asignare de stare.

6.2.Algoritmi DCSP

Constrângerile sunt o metodă naturală de reprezentare a cunoștințelor în foarte multe domenii, după cum observa autorul în [MigI01]. Aplicațiile metodelor CSP se întind de la recunoașterea imaginilor sau probleme de diagnoză în inteligența artificială [Wal75], până la programarea logică (unde înlocuirea algoritmului de unificare al lui Robinson cu metodele de rezolvare a constrângerilor au dat naștere la Programarea Logică pe bază de Constrângeri –CLP(Constraint Logic Programming)- [JafLas87], cu toate derivatele sale ulterioare), și până la rezolvarea de probleme combinatoriale NP-dificile în domeniile cercetărilor operaționale și matematicii discrete. Implicarea recentă a tehnicilor inteligente (rețele neurale Hopfield în care punctele de stabilitate corespund unor soluții ale problemelor bazate pe constrângeri discrete [LeeTam], algoritmi genetici pentru focalizarea căutării soluțiilor, în care constrângerile se reprezintă ca funcții de penalizare, reprezentarea fuzzy a constrângerilor pentru relaxarea problemelor fără soluții și obținerea de soluții parțiale [MigI01]ș.a.) reprezintă o garanție a faptului că domeniul este încă în plină dezvoltare și promite mai mult decât a reușit deja să ofere, în special în probleme în care alte metode au eșuat, și ariile sale de aplicabilitate continuă să se extindă.

6.2.1.Backtracking dinamic pentru CSP

Algoritmii "clasici" folosiți în abordarea problemelor de satisfacere a constrângerilor se împart în două categorii: *algoritmi de preprocesare*, în care constrângerile sunt folosite activ pentru a reduce spațiul de căutare prin restrângerea prealabilă a domeniului variabilelor (algoritmii de arc-consistență cu variantele de îmbunătățire a nivelului de consistență prin considerarea simultană a unui grup de constrângeri- Path-consistency), și *algoritmi de rezolvare* efectivă prin căutare, în care constrângerile sunt folosite pasiv,

pentru testarea soluțiilor generate deja (aceștia din urmă sunt o paradigmă a tehnicilor “generează și testează”).

Una dintre metodele cele mai folosite de căutare sistematică este tehnica Backtracking, - în principiu o căutare în adâncime în graful soluțiilor. Unul dintre marile sale dezavantaje stă în faptul că depistarea unei inconsistențe și revenirea la variabilele asignate anterior pentru a le modifica se face “muncitorește”, fără a lua în considerare care este variabila care a fost efectiv implicată în inconsistența respectivă. Metoda de Backtracking controlat prin dependențe (“dependency-directed backtracking”[StaSus77]) înlătură acest neajuns prin revenirea direct la sursa de conflict. Aceasta pastrează însă în continuare un mare neajuns: la momentul revenirii “se sare” peste soluția construită de la punctul de eroare până la pasul curent, pierzându-se astfel o asignare parțială ce ar putea fi utilă până în final. Ginsberg [Gin96] a propus astfel o nouă îmbunătățire – backtracking dinamic, ce structurează în mod dinamic căutarea: pentru fiecare valoare eliminată a unei variabile se reține o listă de “nogoods” – ca în TMS-, adică variabilele ale căror valori curente sunt în conflict cu valoarea eliminată. Aceste liste se numesc “explicații eliminatorii”, și felul în care sunt efectiv folosite în algoritm determină salvarea de informații “nogood” numai pentru valorile curente ale variabilelor asignate; cele care depind de asignări depășite sunt șterse, îmbunătățind astfel semnificativ performanțele algoritmului:

Propoziție[Gin96]. Spațiul necesar pentru backjumping este $o(i^2v)$, unde $i=|I|$, este numărul variabilelor problemei și v este numărul de valori pentru variabila cu cea mai mare mulțime de valori V_i .

Rezultatul pentru backjumping rămâne valabil și pentru backtracking dinamic întrucât spațiul rămâne limitat de structura explicațiilor eliminatorii.

6.2.2.CSP pentru sisteme de argumentare

În[BesDou04] sunt prezentate trei metode de calcul a extensiilor admisibile. Ne-am oprit asupra celei bazate pe verificarea satisfiabilității, întrucât, după cum remarcă și autorii, aceasta poate fi un punct de plecare în abordarea sistemelor de argumentare folosind tehnici de rezolvare a constrângerilor, iar noi dorim să adaptăm un algoritm DCSP la calculul acestor extensii.

Propoziție[BesDou04]. Fie (A,R) un cadru de argumentare. O mulțime $S \subseteq A$ este admisibilă ddacă formula:

$$\bigwedge_{a \in S} (a \wedge (\bigwedge_{b:(b,a) \in R} (\neg b \wedge (\bigvee_{c:(c,b) \in R} c)))) \wedge \bigwedge_{a \notin S} \neg a; \quad (6.3)$$

este satisfiabilă.

(Ultimul termen exprimă condiția de maximalitate, și putem renunța la el dacă astfel ne apropiem mai bine de cerința problemei particulare pe care o avem în vedere).

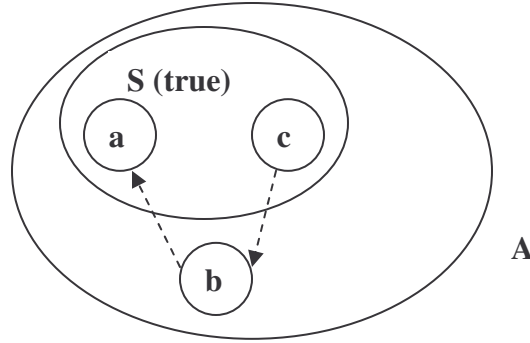


Figura 6.1. Reprezentarea grafică a relației 6.3. (săgețile simbolizează atacuri)

Dacă formula (6.3) este satisfiabilă, înseamnă că există o soluție a ei în cadrul sistemului. Algoritmii CSP pot calcula exact aceste soluții. Concret, dacă relațiile de atac ar fi definite drept constrângeri asupra variabilelor implicate în atac, un algoritm CSP poate calcula extensiile posibile ale unui sistem de argumentare într-o manieră eficientă și sistematică. O mulțime admisibilă de argumente se obține prin filtrarea finală a soluțiilor cu relația (6.3), care reprezintă tot o constrângere, astfel că algoritmul CSP calculează în final exact mulțimile admisibile de argumente, ce vor fi interpretate ca soluții posibile ale problemei de diagnoză: argumentele “in” fiind cele pentru care s-a asigat valoarea de adevăr *true*.

6.2.3. Algoritmi dinamici de satisfacere a constrângerilor

Se știe că este esențială capacitatea sistemelor de inteligență artificială de a raționa pe baza unui mediu dinamic. Problemele clasice de satisfacere a constrângerilor nu sunt potrivite pentru probleme a căror structură se modifică dinamic (de exemplu, ca urmare a apariției unor cerințe neprevăzute din partea utilizatorilor, sau prin apariția de noi date ce trebuie încadrate în sistemul de constrângeri). Aceste situații pot fi tratate corect numai dacă se consideră că setul de constrângeri ce definește soluția problemei se poate modifica dinamic (prin adăugare sau eliminare). S-a ajuns astfel la tehnica DCSP, cu variantele sale de reprezentare și rezolvare [MigI01]. Ne vom opri pe scurt la varianta bazată pe activități, pe care o vom utiliza în cadrul aplicației.

Definiție[MigI01]. O problemă de Satisfacere a Constrângerilor Dinamică bazată pe Activități (aDCSP- activity-based Dynamic Constraint Satisfaction) constă dintr-o problemă CSP clasică și un mecanism ce descrie cum sunt adăugate (șterse) variabilele din problemă, pe baza valorilor altor variabile (Context) (acest mecanism se poate reprezenta, la rândul său, tot prin constrângeri).

Astfel, o problemă aDCSP conține două tipuri de constrângeri: constrângeri de *compatibilitate* (cele clasice), și constrângeri de *activitate*, care specifică în ce condiții o anumită variabilă și domeniul său asociat devin relevante în construirea soluțiilor. În principal, constrângerile de activitate iau una din formele:

$$c_1 \in C_a : x_i = d_{i_a}, x_j = d_{j_b}, \dots \rightarrow \text{activ}(x_k); \quad (6.4)$$

$$c_2 \in C_a : x_i = d_{i_a}, x_j = d_{j_b}, \dots \rightarrow \text{inactiv}(x_k); \quad (6.5)$$

$$c_3 \in C_a : \text{activ}(x_i), \text{activ}(x_j), \dots \rightarrow \text{activ}(x_k); \quad (6.6)$$

$$c_4 \in C_a : \text{activ}(x_i), \text{activ}(x_j), \dots \rightarrow \text{inactiv}(x_k). \quad (6.7)$$

Un caz particular de modificare a structurii problemei este situația când aceasta se schimbă nu ca urmare a unor intervenții externe, ci chiar ca rezultat al deciziilor luate în cursul rezolvării[MigI01]. Astfel, în abordarea noastră, mecanismul dinamic ce modifică structura problemei este legat de testarea dinamică de simptome, testare ce reactualizează continuu contextul problemei.

6.3. Concluzii și contribuții

Capitolul curent a detaliat bazele teoretice ale abordării originale din această lucrare. Am prezentat sistemele de argumentare directă ca formalism modern de reprezentare, adecvat abordării abducției, și pe care îl vom folosi la rândul nostru în cadrul sistemului DiaMed. Introducerea sistemelor de argumentare pentru modelarea nonmonotonicității ne este necesară întrucât modelul medical va conține și relații asimetrice de atac -ilustrate de situația concretă a bolilor mascate, descrisă în Capitolul II (o relație de atac exclusiv simetrică se poate reduce la o problemă de consistență). Am introdus semantica specific argumentativă a admisibilității întrucât o vom utiliza atât în definirea originală a unui diagnostic multiplu, cât și pentru a modela menținerea consistenței pe parcursul raționamentului. Paragraful cheie al capitolului este cel din subsecțiunea 6.2.2, în care se vede cum se reduce calculul semanticii admisibile la o problemă de satisfiabilitate, pentru a cărei implementare algoritmi specifici problemelor de satisfacere a constrângerilor (CSP) sunt o opțiune naturală. Între versiunile posibile ale acestor algoritmi, ne-am oprit asupra backtracking-ului dinamic pentru DCSP [VerSchi] pentru eficiența și adecvarea sa la specificul unei probleme de diagnoză medicală.

VII. Relația abducției cu sistemele de argumentare, tehnicile CSP și sistemele TMS

Abducția, ca schemă generală de raționament a fost abordată procedural, până în prezent, în cadrul unor formalisme cum ar fi programarea logică, tehnicile de rezolvare a constrângerilor (CSP) și sistemele de menținere a consistenței (TMS).

În programarea logică, spre exemplu, s-a arătat că semantica admisibilă (la care ne-am oprit și în lucrarea de față) reprezintă o opțiune de abordare a abducției mai potrivită decât semantica stabilă. Argumentul principal este că procedurile de calcul abductiv dezvoltate în programarea logică nu sunt valide în general, dacă se utilizează alte semantici (mai precis- cea stabilă pentru programele logice normale sau cea stabilă generalizată pentru programarea logică abductivă), dar devin valide odată cu utilizarea semanticii admisibile [EshKo88, EshKo89, KKoT98]. Sistemele de argumentare reprezintă, însă, o opțiune de reprezentare a abducției mai puternică decât programarea logică. Demonstrația s-a făcut arătând că procedurile de calcul abductiv menționate mai sus se pot traduce natural și valid în formalismul sistemelor de argumentare directă [KaManDun94, KKoT98].

Sistemele TMS (TMS [McAll80], ATMS [deKleer86]) sunt, deasemenea, o alternativă procedurală în abordarea raționamentului abductiv. Scopul general al acestor sisteme este de a manevra dependențe logice într-un cadru nemonoton. Motivația principală care a determinat crearea acestui formalism a stat în necesitatea de a trata procedural menținerea adevărului și revizuirea încrederilor. Un aspect demn de menționat al ATMS prin prisma abordării din această lucrare stă în faptul că ATMS și CSP pot fi de fapt privite drept cazuri particulare ale unui cadru unic- sistemul de management a clauzelor al lui deKleer [ReiKle87].

Comparând TMS cu Programarea Logică Abductivă (ALP), se constată că un dezavantaj al sistemelor TMS față de ALP rezultă din avantajul constrângerilor de integritate “clasice” față de cele de tip “nogood”: există multe mulțimi de ipoteze ce se pot ataca via aceeași constrângere de integritate, mulțimi ce în TMS ar fi fost reținute explicit; în schimb, o constrângere “clasică” este reprezentanta unei mulțimi largi de posibile atacuri (care nu e nevoie să fie reținute explicit) -atacuri între tot atâtea mulțimi posibile ce ar conține variabilele din constrângere.

În ceea ce privește avantajele TMS față de programarea logică, un argument principal vine din ideea de a folosi **structuri de dependență** în raționament și rezolvare de probleme, structuri de dependență utilizate atât pentru a face sistemele de raționament mai **incrementale** (adică pentru a păstra maximum din soluția construită până la un anumit moment) în timpul retragerii încrederilor și backtracking-ului, cât și pentru a facilita mai bine **explicațiile** (care sunt mai rudimentare în programarea logică). În plus, prin exploatarea acestor dependențe s-a evitat nu doar backtracking-ul inutil, dar și redescoperirea contradicțiilor sau a inferențelor (memorate în modulul TMS). Structurile de dependență menționate sunt descoperite (construite) dinamic pe parcursul calculului de către modulul de rezolvare (PS- Problem Solver), folosind logica de ordinul întâi, și apoi memorate. În plus, după cum s-a mai menționat, TMS este un modul folosit de sistemele deductive pentru a menține relațiile logice între încrederile manipulate [McAll80]; aceste relații sunt baza modificării incrementale a structurii de încredere când se modifică premisele, oferind un mecanism de “context” mai flexibil decât în programarea logică. Relațiile memorate între încredere pot fi deasemenea folosite pentru a găsi direct sursa contradicțiilor, îmbunătățind eficiența backtracking-ului printr-o schemă de backtracking controlat prin dependențe (DDB: Dependency Directed Backtracking).

7.1. Calculul abducției folosind TMS. Legătura între CSP și abducție

O abordare riguroasă a legăturii între TMS și abducție (și CSP-abducție) găsim la de Kleer [deKleer89b], în sistemul general CMS (Clause Management System- sistem de management a clauzelor). CMS sunt utile pentru raționamentul abductiv în special și pentru eficientizarea căutării în general.

Definiție. O clauză S este un suport pentru C în raport cu Σ ddacă

$$\Sigma \models S \quad (\Sigma \cup \{\neg S\} \text{ este satisfiabilă});$$

$$\Sigma \models S \cup C \quad (\Leftrightarrow \Sigma \models \neg S \supset C \text{ sau } \neg S \rightarrow C \equiv S \cup C)$$

Σ este un set de clauze trimise către CMS de către Reasoner; C = query.

Rolul CMS-ului este de a găsi toate clauzele suport minimale pentru C în raport cu Σ . Minimalitatea este importantă pentru Reasoner din două motive: 1) Abducție; 2) Căutare eficientă. CMS-ul este un fel de “cache” inteligent care poate fi exploatat să organizeze și să controleze căutarea, astfel că mare parte din calculul Reasoner-ului poate fi evitat (înainte de a face un calcul, acesta verifică dacă acesta a mai fost realizat anterior). Înainte de a alege o valoare pentru o anumită variabilă, Reasoner-ul interoghează întâi CMS pentru a vedea dacă variabila e determinată de alegerile curente; dacă nu e determinată- se alege o valoare ce poate fi adăugată consistent la mulțimea curentă de opțiuni.

Definiție. *Prim implicant* al unei mulțimi Σ de clauze este o clauză (disjuncție de literal) C :

1. $\Sigma \models C$;
2. Nu există C' submulțime proprie a lui C astfel încât $\Sigma \models C'$.

Caracterizarea CMS. Fie $\Sigma =$ mulțime de clauze; $C = \{1\}$ o clauză unitară; Atunci S este suport clauzal minimal pentru C în raport cu Σ ddacă

$$\text{Există un prim implicant } \Pi \text{ al lui } \Sigma \text{ astfel încât } 1 \in \Pi, S = \Pi - \{1\}$$

Partea interesantă a CMS stă în faptul că atât ATMS cât și CSP sunt cazuri speciale ale sale [ReiKle87]. CMS-ul este o generalizare ATMS: în ATMS, CMS este restricționat la clauze Horn (=justificările) și clauze negative (inconsistențele de tip “nogood”), iar C sunt întotdeauna literal.

Caracterizarea ATMS. Fie J o mulțime de justificări trimise către ATMS de către Reasoner, $\{\alpha\}$ o interogare (query) ($\alpha =$ simbol propozițional, ipoteză sau nu).

Atunci răspunsurile la această interogare sunt date de:

$$\{A_1 \wedge \dots \wedge A_k \mid k \geq 0, \text{ și } \neg A_1 \vee \dots \vee \neg A_k \vee \alpha \text{ este un prim implicant al lui } J\} \quad (7.1)$$

(abducție: ipoteză= explicație (ipoteza explicativă)).

Caracterizarea CSP. O ipoteză poate fi un simbol propozițional ce face parte dintr-o soluție propusă pentru CSP (o valoare pentru o variabilă din CSP).

7.2.CSP în calculul semanticii admisibile din argumentație

Capitolul VI a introdus bazele teoretice ale utilizării algoritmilor CSP în calculul semanticii admisibile. Am arătat acolo cum putem folosi backtracking dinamic pentru DCSP pentru a aproxima semantica admisibilă (o variantă net superioară [vezi rezultatul de complexitate backtracking dinamic] algoritmilor exponențiali bazați pe calculul părților unei mulțimi care au fost dezvoltati până acum pentru această semantică- [Doutre02], în afară de avantajul modelării dinamice a spațiului de căutare).

Am văzut, în plus, în secțiunile precedente, că în comparație cu ATMS sau LTMS, ținând cont de criteriul dimensiunii bazelor de inconsistențe (*nogoods*), metoda backtracking dinamic are un avantaj evident întrucât reține doar inconsistențele asociate cu valorile actuale ale variabilelor (celelate sunt eliminate), fiind orientat spre descoperirea unei singure soluții (poate fi însă extins cu ușurință, dacă specificul problemei o cere). În

concluzie, nu numai că algoritmi DCSP sunt o abordare procedurală mai eficientă decât TMS pentru abducție, dar aceștia sunt în plus adecvați pentru ceea ce avem nevoie în aplicația noastră, adică:

- implementarea admisibilității;
- lucrul simultan cu mai multe soluții alternative (pentru a fi mai ușor comparate în vederea diagnosticului diferențial);
- incrementalitate;
- adaptare la un mediu dinamic de evidențe;
- eficienți în general.

7.3. Avantaje ale argumentelor asupra TMS

Dorim să subliniem încă un aspect în ceea ce privește avantajele abordării structurilor cauzale localizate pentru reprezentarea argumentelor (ca cele din abordarea pe care o propunem), față de abordarea TMS. Este vorba despre o suită de lucrări ale lui de Kleer & comp [deKleer90], în care autorii arată cum poate beneficia un TMS de localitatea din reprezentarea cunoștințelor pentru a-și controla și restrânge inferențele. Autorii pleacă de la ideea că mulți rezolvitori AI sunt inerent locali: fiecare parte componentă a problemei are un model de comportament fixat; și o mare parte a raționamentului poate fi văzută ca propagare (când se inferează un semnal nou, modelele componentelor implicate sunt consultate pentru noi inferențe posibile), TMS fiind potrivit pentru această propagare. Se știe în plus că Propagarea Booleană a Constrângerilor (BCP - Boolean Constraint Propagation) este eficientă dar incompletă pe CNF, deci TMS-urile sunt incomplete logic când sunt astfel folosite (petele oarbe -“blind spots” – apar deoarece localitatea prezentă în modelul original se pierde în TMS). Pe de altă parte, satisfiabilitatea propozitională este NP-completă, dar costul completitudinii logice poate fi redus mult prin exploatarea localității. Astfel, local, în cadrul fiecărui modul, TMS este complet logic dacă se folosește BCP pe formule logice (BCP fiind completă pe formule logice, dar mai costisitoare).

Se știe că prin transformarea formulilor logice în CNF se pierde informația legată de localitate:

$$\text{din } \neg x \vee y \vee z \text{ și } x \vee y \vee z \text{ și } y=F, \text{ BCP nu deduce } z=T.$$

Propunerea făcută de către de Kleer este ca mulțimea de formule (i.e. combinații de clauze) ce reprezintă un model local să fie transmisă TMS ca un singur modul, și TMS-ul să folosească o metodă de inferență completă local pe module. Extrapolând, localitatea este exploatată în modelul nostru, prin faptul că DCSP lucrează la un moment dat pe contextul asociat doar cu ipotezele de diagnostic selectate, context conținut într-un model cauzal local.

În concluzie, atât abordările CSP, cât și sistemele TMS reprezintă două formalisme înrudite pentru calculul abducției, fiecare cu avantajele și dezavantajele sale și am aratat de ce am ales algoritmi specifici CSP în favoarea TMS (ca implementare pentru o problemă cu constrângeri), și de ce am ales sistemele de argumentare atât în favoarea sistemelor TMS, cât și în favoarea programării logice.

7.4. Concluzii și contribuții

Am văzut în capitolul de față că atât abordările CSP, cât și sistemele TMS reprezintă două formalisme înrudite pentru calculul abducției, și am prezentat avantajele și dezavantajele lor și domeniul de aplicabilitate. Am justificat, deasemenea, de ce am ales algoritmi specifici CSP în favoarea TMS (ca implementare pentru o problemă cu constrângeri), și de ce am ales sistemele de argumentare în favoarea sistemelor TMS. Prima parte a capitolului s-a dorit o justificare a faptului că sistemele de argumentare directă reprezintă un formalism cel puțin la fel de bun ca programarea logică în problema abducției (și, prin urmare, merită atenția cercetătorilor în domeniu), și a arătat, în plus, meritul semanticii admisibile față de cea stabilă prin prisma programării logice.

O contribuție importantă a capitolului stă în încercarea de sistematizare a ideilor ce stau la baza formalismelor enumerate mai sus, cu scopul de a descoperi mecanismele elementare ce apropie între ele abordări distincte, dar alăturate pe criteriul aplicabilității la aceeași problemă.

Ni s-a părut interesant să relevăm legătura între CSP (tehnica pe care am ales-o și noi pentru implementare) și abducție în general, și să detaliem, măcar parțial, unele mecanisme ale sistemelor TMS. Acestea din urmă au fost alese atât pentru că reprezintă o modalitate notabilă de abordare a raționamentului nemonoton, cât și pentru faptul că am considerat util să aducem în prim plan, pentru comparație, o alternativă de exploatare a conceptului de “localitate”, întâlnit atât în rețelele cauzale cât și în sistemele de argumentare directă.

Ideile prezentate ne-au servit, în final, la accentuarea avantajelor utilizării de algoritmi CSP pentru semantica admisibilă din argumentație.

VIII. Argumentarea deciziei de diagnostic în DiaMed

Capitolul de față descrie în detaliu algoritmul asociat cu nivelul de discriminare și explicare al sistemului hibrid DiaMed [MunDum06a], algoritm care este o îmbunătățire a versiunii prezentate în [VerSchi], astfel încât dinamica constrângerilor să fie determinată practic de rezultatele testelor medicale.

8.1. Reprezentarea cunoștințelor

Modelul pe care îl propunem conține asocieri cauzale clase-caracteristici reprezentate cu ajutorul următoarelor componente.

Clasele de diagnostic (de exemplu bolile, în cazul medical) sunt modelate în abordarea noastră sub forma unor *rețele cauzale* de tip special, cu mai multe tipuri de noduri și arce, și care rezumă modelul cauzal profund al modelului. Astfel, există trei tipuri de noduri:

- noduri rădăcină, asociate claselor (ipotezelor de diagnostic), corespund unor cauze profunde, primare, ale manifestărilor observate (redate cu roșu, cele cu roz sunt folosite pentru reprezentarea complicațiilor);

- noduri corespunzătoare manifestărilor (simptomelor) de profunzime (inaccesibile sau accesibile cu dificultate, prin teste invazive, costisitoare, consumatoare de timp);
- noduri ce corespund unor simptome superficiale, direct/ ușor observabile.

Nodurile asociate simptomelor (observate sau nu) sunt de două tipuri: **necesare** sau **suplimentare**. Cele colorate în albastru reprezintă manifestările profunde, mai greu accesibile, iar cele în verde sunt manifestările superficiale, accesibile prin observații directe asupra pacientului. Infirmarea la testare a unui simptom necesar atrage după sine eliminarea completă a ipotezei de diagnostic corespunzătoare. Arcele ce leagă nodurile în cadrul rețelei cauzale sunt de asemenea de mai multe tipuri:

1. Implicații necesare: prezența cauzei atrage întotdeauna după sine apariția consecinței;
2. Implicații posibile: cauza *poate* atrage după sine apariția consecinței, dar nu este obligatoriu; (observăm că această incertitudine își are de fapt originea într-o incompletitudine a modelului: există anumite elemente/condiții suplimentare care influențează validitatea implicației dar care nu au fost modelate explicit);
3. Atacuri bi- sau unidirecționale: modelează, în principiu, elemente ce nu pot apare simultan în cadrul aceluiași sistem .

Fiecare clasă de diagnostic este definită de câte o astfel de rețea cauzală, ce conține toate elementele posibile relaționate cu clasa respectivă, organizate pe nivele din ce în ce mai superficiale, și deci mai accesibile observației directe asupra sistemului. Nodurile intermediare între rădăcină și frunze sunt, de obicei, fie inaccesibile, fie greu accesibile (prin teste costisitoare, de lungă durată și/sau invazive), și gradul de accesibilitate crește pe măsură ce ne apropiem de frunze (Figura 8.1)

Definiție. Se numește *argument* asociat unei anumite clase o instanță a rețelei cauzale ce definește clasa. O instanță a unei rețele cauzale este o submulțime a nodurilor sale ce conține cel puțin o manifestare observată (restul fiind presupuse adevărate).

Propunem în continuare o definiție originală și naturală a unui diagnostic multiplu.

Definiție. Un *diagnostic multiplu* (i.e. o mulțime nevidă de boli posibile la un pacient dat) este o mulțime admisibilă¹ de ipoteze² care acoperă toate observațiile, și este minimală cu această proprietate.

Exemplu. Fie 3 argumente $A=\{\text{leucemie}\}$, $B=\{\text{nivel PT,PTT normal}\}$, $C=\{\text{medicamente care afectează nivelul PT,PTT}\}$. Relația de atac este dată de graful: $A \leftarrow B \leftarrow C$. Atunci A este acceptabil în raport cu C, C fiind o apărare a lui A față de atacul B. Deci $\{A,C\}$ este o mulțime admisibilă, și în acest caz “leucemie” este un diagnostic posibil în contextul definit de C.

¹ A se vedea definiția din Secțiunea 6.1.

² considerăm drept ipoteză orice boală activă, ce poate fi asimilată cu argumentul ce o susține.

Observații.

- Dung a observat că relațiile de atac între argumente în raționamentul implicit depind doar de asumptiile (ipotezele) pe care se bazează acele argumente, și a propus o definiție a argumentului ca fiind o deducție ale cărei premise sunt toate asumptiile; în plus, argumentul a atacă argumentul b dacă a atacă o asumptie pe care b se bazează. (Un argument atacă o asumptie α când concluzia lui a este contrariul *not* α al lui α).

- Un argument corespunde configurației particulare de simptome prezentă în cadrul cazului observat, prin care clasa se manifestă, și care poate varia de la pacient la pacient.

- Merită să comparăm în acest punct anumite aspecte pentru a elimina eventualele confuzii: dacă rețelele bayesiene și cauzale din Capitolul III sunt structuri suport ce sintetizează pași de inferență statistică, și inferența se realizează exclusiv pe baza acestor structuri, după anumiți algoritmi probabiliști, rețelele cauzale din abordarea noastră sunt un mod de reprezentare compactă a unor module din modelul medical, inferența realizându-se nemonoton cu ajutorul relației de atac. Un argument poate fi privit aici drept o instanță a unei clase a modelului, instanță ce este adăugată/ ștersă dinamic din contextul curent ținând cont de atacuri.

Definiție. *Graful de atac abstractizat* este, simplu, graful de atac luat în considerare între argumentele ce conțin nodurile ce se atacă (presupunem că nu există noduri ce se atacă în cadrul aceluiași argument).

Atacurile folosite concret în modelul nostru sunt în principiu de trei tipuri. În primul rând se atacă două cauze alternative pentru același efect (cum se vede și în Figura 8.2.), în ideea că doar una dintre ele a provocat efectul respectiv și este nefolositor să le considerăm pe ambele “in”. Situațiile care ar infirma acest model sunt probabil foarte rare, dacă există. În cadrul celor 30 de boli alese pentru exemplificare și al exemplelor rulate noi nu am întâlnit o astfel de infirmare.

Al doilea tip de atac este descris de următoarea situație: dacă o boală b se poate manifesta sub forma unei boli a , astfel că deseori b se confundă cu a , atunci considerăm că b atacă a . Ideea este că inițial se presupune a , dar teste suplimentare infirmă a și confirmă b . Acest atac sintetizează de fapt experiența practică a experților asupra unei anumite ordini temporale posibile de apariție a ipotezelor în cadrul raționamentului. Spre exemplu, insuficiența cardiacă atacă ciroza, deoarece uneori insuficiența se confundă cu ciroza.

În sfârșit, al treilea tip de atac este definit de inconsistența logică: dacă în insuficiența cardiacă este necesar ca presiunea și volumul cardiac să fie afectate, am considerat că “presiune și volum cardiac normal” atacă direct (prin abstractizare) insuficiența cardiacă. Există însă excepții care complică această situație; spre exemplu, în hipotiroidie avem insuficiența cardiacă fără ca presiunea sau volumul să fie afectate (mecanismul de producere fiind diferit: hipocalcemia) așa că am considerat că hipotiroidia atacă direct “presiune volum cardiac normal”, în contextul “insuficiență cardiacă”. Altfel spus, hipotiroidia este o apărare a insuficienței cardiace în fața “presiune/volum cardiac normal” (vezi și graful de atac din Figura 8.5). De fapt, în această excepție, avem tot atac de tipul “cauză alternativă”, așa că nu l-am tratat separat.

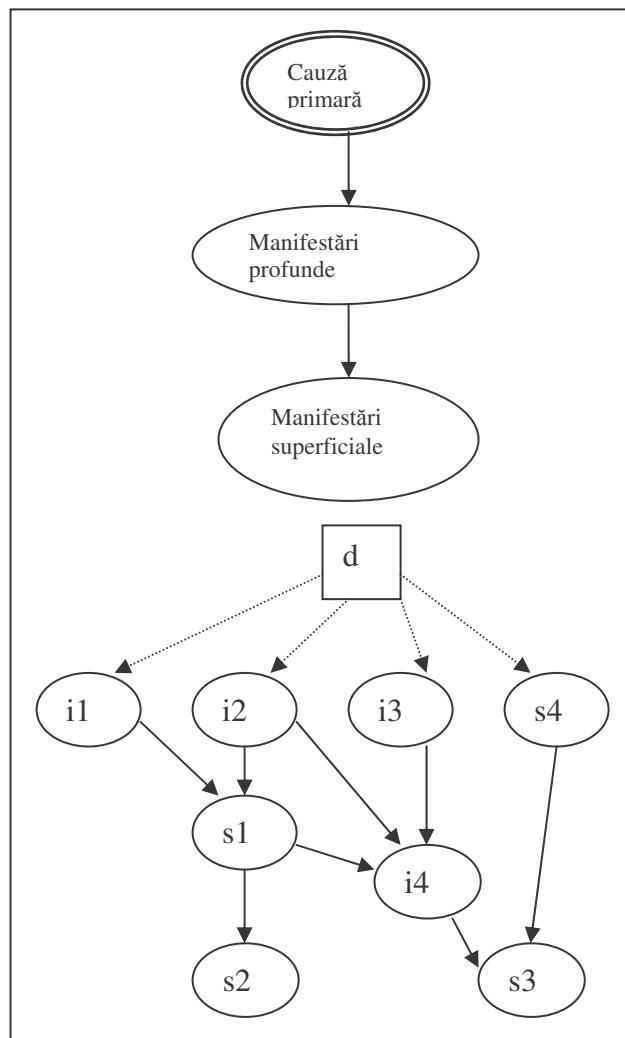


Figura 8.1. Structura generală a unei rețele cauzale și un exemplu de subrețea (d-boală, i-noduri intermediare inaccesibile, s -simptome testabile)

8.2. Modelarea problemei în cadrul DCSP

Notatii.

- Ipoteze de diagnostic: d_1, \dots, d_N (N clase (boli) în total);
- Rețelele cauzale asociate ipotezelor de diagnostic: C_1, \dots, C_N ;
- Argumente: A_1, \dots, A_L (L instanțieri posibile ale rețelelor cauzale);
- $I_p = \{d_1, \dots, d_t\}$ -mulțimea de ipoteze selectate, eventual ordonate descrescător după ponderea primită la selecție;
- $Manifestări_Confirmate = \{mc_1, \dots, mc_k\}$ mulțimea manifestărilor confirmate (true);

- $Manifestări_Infirimate = \{mi_1, \dots, mi_p\}$ mulțimea manifestărilor infirmate (false);

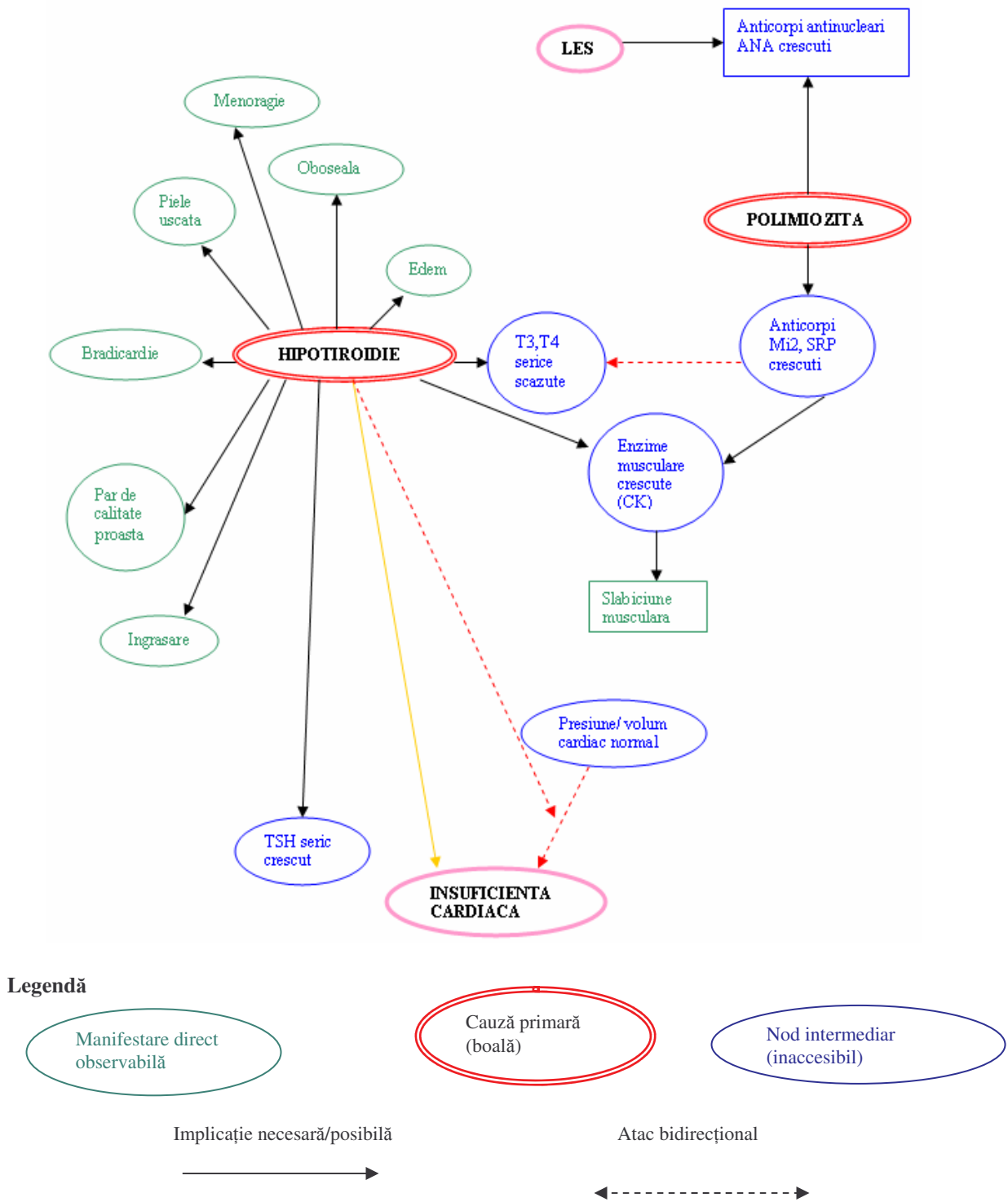


Figura 8.2. Exemplu de dependențe cauzale

- $Context = Manifestări_Confirmate \cup Manifestări_Infirimate$ -mulțimea totală de simptome confirmate sau infirmate prin testare;
- G_a - graful de atac abstractizat restricționat la mulțimea I_p ;
- Ne vom folosi de o partiționare a mulțimii I_p în: $Ipozeze_infirimate$, $Ipozeze_considerate$, $Ipozeze_noi$.

Așadar, pornind de la $Manifestări_confirmate$, se face selecția prin decizie fuzzy (faza abductivă a generării de ipoteze) și se ajunge la mulțimea de ipoteze active: $\{d_1, \dots, d_t\}$, cu rețelele cauzale asociate C_1, \dots, C_t . Într-un prim pas, se calculează tot ceea ce este legat de manifestările confirmate/ infirmate prin relații deterministe (de exemplu, un nod atacat al cărui atac este *true* și atacul nu e defensibil, va fi sigur *fals*). Nodurile rămase din rețelele activate netestate încă formează mulțimea de variabile active $Var_Active = \{V_1, \dots, V_p\}$ pe care se aplică algoritmul backtracking dinamic pentru CSP dinamice [VerSchi] într-o formă adaptată (faza de testare a consistenței). Variabilele active sunt asociate de obicei cu noduri defensibile: au statutul “in” (presupus adevărat) sau “out” (presupus fals) la un moment dat, dar pot fi reinstaurate/ contrazise pe parcurs. Constrângerile active ale algoritmului CSP - $Constr_Active$ - sunt cele în care este implicată cel puțin o variabilă activă. După cum s-a definit, un diagnostic multiplu este o soluție minimală- adică o mulțime admisibilă, complet acoperitoare și minimală. (Practic, în implementare am simplificat lucrurile prin introducerea în Var_Active direct a ipotezelor de diagnostic, acolo unde a fost posibil).

Evidența nodurilor testate se ține cu ajutorul mulțimilor $Manifestări_confirmate$ (conține ipoteze de diagnostic sau simptome ce le susțin) – utilizate pentru explicația finală și $Manifestări_infirimate$ - utilizate pentru a nu reactiva prin selecție ipoteze/ noduri înfrânte deja.

Toate cele trei tipuri de implicații ce pot apare într-o rețea cauzală sunt reprezentate drept constrângeri binare (reprezentare inspirată din BCP- Boolean Constraint Propagation), astfel:

- Implicații necesare: $a \rightarrow b : C_{ab} = \{00, 11, 01\}$;
- Atacuri: $a \vdash b : C_{atac} = \{10, 01, 00\}$;
- Implicațiile posibile nu restrâng cu nimic domeniul posibil de valori pentru perechea de variabile implicate, astfel că acesta va fi $\{00, 01, 10, 11\}$.

Fie:

$Cons(v)$ = constrângerile în care apare variabila v ;

$Var(c)$ = variabilele ce apar în constrângerea c .

Mulțimea constrângerilor curente $Constr_Active$ se definește natural:

$$Constr_Active = \bigcup_{v \in Var_Active} Cons(v). \quad (8.1)$$

Definiție. Se numește *soluție* a problemei o asignare completă și consistentă de valori de adevăr pentru toate variabilele active (activate prin selectarea anumitor ipoteze), și care acoperă toate manifestările confirmate. O soluție cu un număr minim de noduri-ipoteze se numește soluție *minimală*. Aceasta corespunde definiției diagnosticului multiplu de mai sus.

Mai multe soluții (nu neapărat minimale)- se pot ierarhiza după *gradul de determinare*, un indice original ce folosește varianta “unfair” a legii combinării evidențelor a lui Bernoulli pentru a agrega măsuri ale nodurilor necesare, respectiv suplimentare confirmate, raportat la cele asumate (presupuse adevărate).

8.3.Algoritm

1.Se ordonează *Var_Active* pornind de la variabilele cele mai constrânse (aceasta reprezintă o euristică binecunoscută pentru eficientizarea algoritmilor CSP). Întrucât atacul este constrângerea cea mai restrictivă, se poziționează pe primele locuri variabilele implicate în atacuri;

2.Am adaptat algoritmul de backtracking dinamic [VerSchi] pentru mulțimea *Var_Active* [Verfaillie, Schiex], cu constrângerile asociate- *Constr_Active*, pentru calculul diagnosticului, după cum urmează. (Noduri *In* sau *Out*, =noduri defensibile (ce pot fi ulterior atacate)).

Inițializări

Fie $V_1 = \emptyset$ mulțimea de variabile asignate, $V_2 = Var_Active$ mulțimea variabilelor neasignate; se inițializează *Domeniile* variabilelor cu mulțimea $\{in, out\}$.

dbt(csp)

$V =$ mulțimea de variabile a problemei CSP;

return **dbt-variabile**(\emptyset, V).

backward-checking(V,v,val)

-verifică valoarea curentă *val* a variabilei *v* față de valorile variabilelor din V (= V_1 curent reunit, eventual, cu variabilele fixate din asignarea inadmisibilă ce se încearcă a fi extinsă) și întoarce prima constrângere încălcată (*c*) sau *succes* (dacă nu există o astfel de constrângere);

-concret, se verifică dacă valoarea *val* este admisibilă în contextul actual astfel:

1. dacă *val*=“in”, *val* este admisibilă dacă nu atacă (e atacată de) un nod asignat deja ca fiind “in”;
2. dacă se încearcă atribuirea valorii *val*=“out” nodului curent *n* și dacă el atacă un nod *m* “in”, trebuie să existe un nod “in” care atacă nodul *n*;

crează-explicație-eliminatorie(v,val,V)

-reține mulțimea de variabile V (o lista de variabile cu valorile lor asociate) ca o explicație pentru eliminarea valorii *val* din domeniul variabilei *v*;

sterge-explicatii-eliminatorii(v,V)

-pentru toate variabilele din V șterge explicațiile eliminatorii care conțin variabila *v*-trebuie reactualizat domeniul pentru $V_4 \cup V_2, V_4$ de la pasul (*);

dbt-valoare(V1,v,val)

-verifică dacă în contextul definit de V_1 (variabile asignate deja), valoarea val pentru variabila v este admisibilă:

$v \leftarrow val$;

$c = \text{backward-checking}(V_1, v)$;

if ($c = \text{succes}$) then return *succes*;

else { $V_3 \leftarrow$ mulțimea de variabile din constrângerea c ;
crează-explicație-eliminatorie(v, val, V_3);
return *failure*; }

dbt-variabile (V_1, V_2)

// V_1 este mulțimea variabilelor asignate, V_2 – a celor neasignate;

if ($V_2 = \emptyset$) then { if (V_2 este o acoperire completă și admisibilă a simptomelor prezente)

afișează soluția;

Soluții = Soluții \cup Soluția_curentă;

- aici se propune utilizatorului o ierarhizare a soluțiilor în funcție de gradul de determinare, însoțite de explicațiile corespunzătoare (argumentele asociate fiecărei ipoteze); utilizatorul are în plus posibilitatea de a opri algoritmul în acest punct;

if (V_2 inadmisibilă) se reține într-o listă asignarea curentă pentru a putea fi eventual extinsă ulterior; variabilele sale vor fi fixate, rolul său fiind limitat la verificarea admisibilității;

else { găsit = false;

$v \leftarrow$ o variabilă din V_2 ;

$d =$ domeniul curent al variabilei v ;

$val \leftarrow$ o valoare din d ;

if (**dbt-valoare**(V_1, v, val)) then

{ găsit = true; $v \leftarrow val$; Asignare.add(v, val);

$V_1 = V_1 \cup \{v\}$; $V_2 = V_2 - \{v\}$;

dbt-variabile($V_1 \cup \{v\}, V_2 - \{v\}$); }

if (găsit = false) then **dbt-bt-variabilă**(V_1, V_2);

(nu mai am valori pentru v , trebuie să deasinez un conflict al său)

//end else

dbt-bt-variabilă (V_1, V_2, v)

$V_3 \leftarrow$ mulțimea de conflicte a lui v (= uniunea explicațiilor eliminatorii ale valorilor eliminate ale lui v);

if ($V_3 = \emptyset$) then { //nu există soluții pentru variabilele curente;

date-noi = testează();

-se reia pasul de selecție cu mulțimea reactualizată de teste (cele vechi + cele nou testate în cadrul pasului de discriminare curent):

creaza_Contexte (date);

AC = constrângerile ce definesc noul context de ipoteze;

RC = constrângerile asociate cu contextele în care un nod necesar a fost atacat de nodurile corespunzătoare simptomelor nou testate;

Reactualizare constrângeri pe baza testării:

Nodurile accesibile se testează și:

- Se șterg din *Var_Active*;

- Dacă sunt confirmate și înfrâng un nod necesar al unei ipoteze, toate constrângerile asociate ipotezei respective se șterg (se șterge din *Var_Active* și nodula asociat ipotezei însăși);
- Dacă sunt infirmate și sunt noduri necesare pentru o ipoteză, această ipoteză se șterge din *Var_Active* și constrângerile asociate ei se șterg din *Constr_Active*.

(Am definit astfel în mod natural și un plan de testare original semi-automat pentru procesul de diagnoză).

ddbt(V₁,V₂, AC, RC); }

else { (V₃ ≠ ∅): (*)

Se revine și se deasignează ultima variabilă din V₃ (adică ultima ipoteză făcută care e în conflict cu v, variabila pentru care nu se mai găsesc soluții).

v' ← ultima variabilă ∈ V₃ din V₁;

val' ← valoarea curentă a lui v';

V₄ ← mulțimea variabilelor de după v' din V₁;

deasignează(v');

crează-explicație-eliminatorie(v',val', V₃-{v'});

șterge-explicații-eliminatorii(v', V₄ ∪ V₂); (*)

dbt-variabile (V₁-{v'}, V₂ ∪ {v'})

8.4.Exemplu

Să presupunem că am observat inițial următoarele simptome (vezi planșele I și VI): Manifestări_confirmare={anorexie, aritmii, ascită, dispnee, edem, oboseală, slăbiciune musculară, anemie de boală cronică}. La o primă rulare, sistemul selectează prin decizie fuzzy următoarele ipoteze (boli), cu scor mai mare sau egal decât pragul ales (0.2):

- Angină pectorală 0.51 (BI1)
- Ciroză 0.22 (BH1)
- Hipotiroidie 0.25 (BE3)
- Infarct miocardic 0.26 (BI2)
- Insuficiență cardiacă 0.29 (BI3)
- Miocardită 0.70 (BI4)
- Pericardită 0.34 (BI5)

(Am trecut în paranteză codificarea prescurtată a bolii folosită în implementarea practică). În lista de ipoteze mai este selectat automat de către algoritm un nod cu statut special: “presiune/ volum cardiac normal” (SI32). Statutul special al nodului este dat de faptul că acesta apare în relațiile de atac în care sunt implicate ipotezele selectate (fără să fie simptom al vreuneia din boli). (Figura 8.3 prezintă graful de atac asociat contextului selectat).

Prin algoritmul CSP se obțin trei soluții admisibile minimal acoperitoare, și anume:

- Hipotiroidie, insuficiență cardiacă, miocardită (BE3, BI3, BI4)
- Hipotiroidie, insuficiență cardiacă, infarct miocardic (BE3, BI3, BI2)
- Hipotiroidie, insuficiență cardiacă, angină pectorală (BI1, BI3, BE3)

Presupunem că utilizatorul (medicul) nu este mulțumit de rezultat și vrea să completeze investigațiile cu noi teste. Aici el are posibilitatea de a completa în program simptome nou observate și poate infirma simptome necesare ale bolilor active la acest moment pentru a mai restrânge, eventual, contextul. În acest scop, o listă de simptome necesare posibile în contextul dat este construită automat de către program, și acesta este singurul indiciu automatizat care să ajute testarea. Simptomele necesare au acest statut preferențial față de cele suplimentare pentru că doar prin infirmarea lor se poate obține o restrângere a spațiului de lucru (Figura 8.3).

În urma noilor observații, Manifestări_infirmate={"dureri piept ce cedează la nitroglicerină", "dureri piept ce nu cedează la nitroglicerină", "ritm de galop", "frecătură pericardică"}, ceea ce exclude din lista de ipoteze posibile nodurile {BI1, BI2, BI4, BI5}-adică boli pentru care anumite manifestări necesare lipsesc la pacientul observat. (A se vedea și Anexele 1-3). Manifestările infirmate susțin faptul "*presiune/volum cardiac normal*", ce apără ipoteza „ciroză”. Se descoperă în plus, la testare, prezența de “anticorpi Mi2, SRP crescuți”. Acest simptom va adăuga în lista de ipoteze posibile “Polimiozita” (BREUM3), ce a obținut un scor de 0.45. Graful de atac a devenit cel din Figura 8.5.

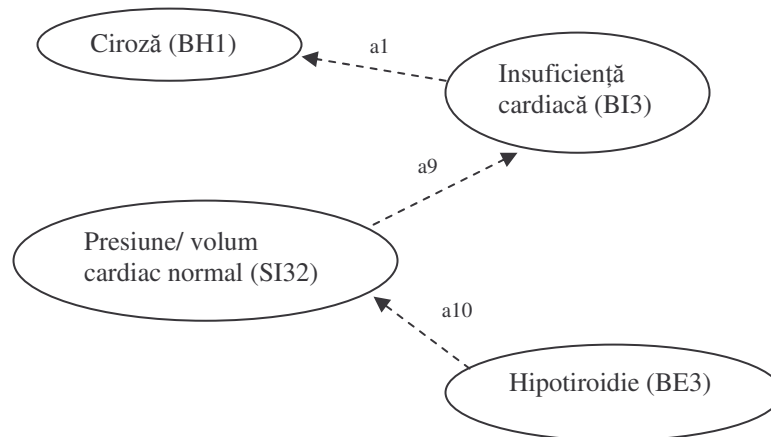


Figura 8.3.Graful de atac pentru primul context selectat

La reluarea algoritmului, se încearcă extinderea asignărilor inadmisibile de la pasul precedent cu apărări. Astfel, algoritmul preia, spre exemplu, asignarea inadmisibilă {"ciroza" "in", "presiune/ volum cardiac normal" "in"} și obține în noul context o asignare admisibilă dar neacoperitoare: {"ciroza" "in", "presiune/ volum cardiac normal" "in", "polimiozita" "in"}, adică nodul BREUM3 este o apărare a mulțimii {SI32, BH1}, față de posibilele {BI3, BE3} activate de evidențe.

Concluzia este că am reușit să simulăm dinamica spațiului de căutare în funcție de datele noi și infirmările ipotezelor anterioare foarte aproape de modul de raționament al unui expert uman (acesta are totuși ultimul cuvânt în decizia de diagnostic).

În secțiunea "Soluții posibile" apar combinații admisibile de boli complet acoperitoare ale simptomelor, iar în secțiunea "Soluții parțiale admisibile" sunt combinații admisibile neacoperitoare dar cu scor ridicat. Acest scor a fost folosit în cadrul algoritmului ca o euristică foarte utilă în eliminarea combinațiilor mai puțin probabile, ceea ce a redus

timpul de calcul de câteva ori (euristica abandonează asignările de scor scăzut, scorul fiind proporțional cu ponderea simptomelor acoperite în cadrul bolilor asignate ca “in”).

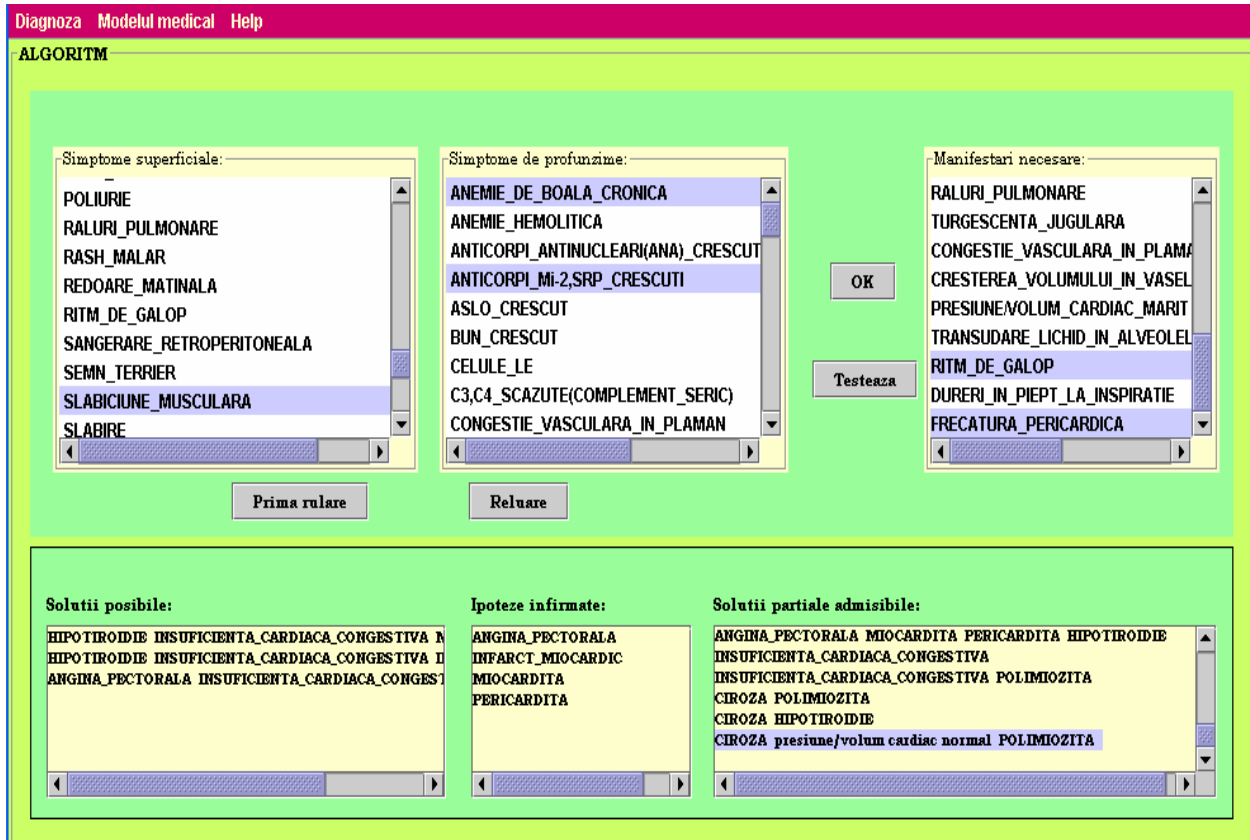


Figura 8.4. Exemplu (Anexa 1, planșele I, VI)

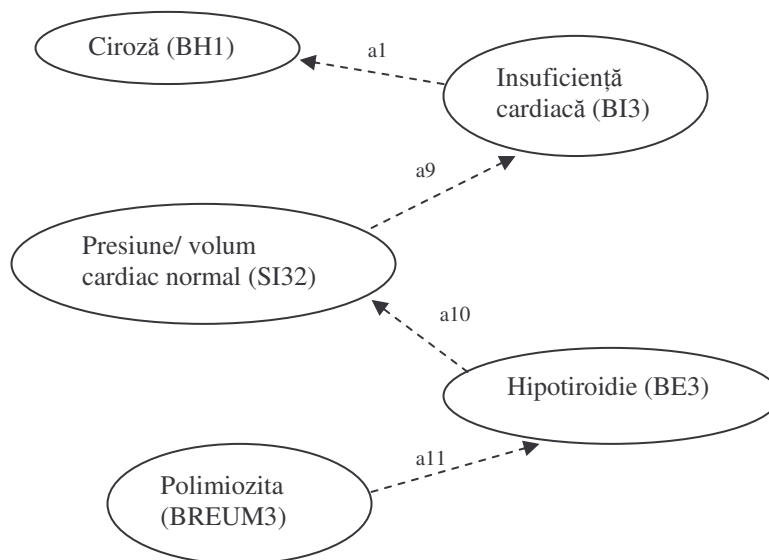


Figura 8.5. Graful de atac pentru al doilea context selectat

8.5. Concluzii și contribuții

Abordarea DCSP din algoritmul de mai sus reprezintă o traducere eficientă a mecanismului de remodelare dinamică a contextului de lucru pe baza evidențelor aduse de procesul de testare, permițând lucrul pe secțiuni limitate ale modelului. Adăugarea și ștergerea de ipoteze ca urmare a selecției reluate după re-completarea testelor stă la baza constrângerilor de activitate, adăugând/ștergând din variabilele problemei, pe baza unor reguli dependente de domeniu. Constrângerile de activitate asociate problemei aDCSP de activare/inactivare a variabilelor sunt conținute implicit în funcțiile de decizie fuzzy și instanțele activate ale rețelelor cauzale (i.e argumentele).

Backtracking dinamic pentru DCSP este o metodă incrementală, ce pastrează subsoluția validă la tranziția spre alte porțiuni ale modelului. În plus, ni se oferă flexibilitate: ipotezele sunt generate chiar atunci când nu avem informații suficiente, dar avem posibilitatea să reconsiderăm aceste ipoteze dacă suntem confrunțați cu necesitatea păstrării coerenței în fața a noi evidențe;

Prin algoritmul de mai sus s-a arătat, deasemenea, cum poate fi folosit DCSP pentru calculul (aproximativ al) semanticii admisibile; mai precis, CSP este o tehnică validă de determinare a admisibilității unei mulțimi de argumente, iar versiunea dinamică permite determinarea contextuală a acestei proprietăți ce este, în mod natural, dependentă de context; avem așadar o metodă constructivă și eficientă de calcul aproximativ a semanticii admisibile.

O altă contribuție originală este dată de definiția naturală și originală a diagnosticului multiplu folosind semantica admisibilă a argumentelor, și de indicele original de calcul a gradului de determinare al unui argument. S-a propus pentru argumente o definiție originală, adaptată domeniului studiat, care structurează informația, grupând bolile pe nuclee posibile de interacțiune.

În câteva cuvinte, mecanismul de ștergere și adăugare de constrângeri folosit surprinde cu fidelitate, într-o manieră sistematică și eficientă în același timp, mecanismul de retragere și adăugare de încrederi propriu raționamentului nemonoton. Avantajul principal față de sistemul CHECK al acestei abordări a nonmonotonicității la nivelul rafinării ipotezelor în DiaMed, stă în faptul că folosește o metodă tratabilă, eficientă, spre deosebire de abordarea logică a abducției indirecte ce determină sursa eventualelor inconsistențe în CHECK.

Am prezentat, așadar, o abordare originală, pe bază de logica argumentării și algoritmi DCSP a raționamentului nemonoton în general (și care poate fi utilă în discriminarea și rafinarea mulțimii de ipoteze de diagnostic în special). Așa cum observau autorii în [BreDix97], “cercetările ulterioare în domeniul raționamentului nemonoton trebuie să pună mai mult accentul pe aspectele computaționale”, întrucât acest domeniu “va avea un impact de durată asupra domeniului Inteligenței Artificiale în general, numai dacă va pune la dispoziție instrumente utile oamenilor în rezolvarea problemelor lumii reale”. Acest citat subliniază importanța și semnificația algoritmului prezentat în Secțiunea 8.3. În plus, am făcut legătura în această lucrare între semantica admisibilă și diagnosticele multiple, justificând apropierea la nivel intuitiv a celor două noțiuni, și am arătat cum semantica admisibilă se poate calcula prin algoritmi CSP. Am arătat, mai mult, cum varianta dinamică a acestor algoritmi se poate adapta pentru a pune la dispoziție un

instrument deopotrivă eficient și sistematic pentru raționamentul nemonoton. Prezentarea, deși poate fi utilizată pentru diagnoză în general, s-a orientat spre specificul diagnozei medicale, pe care am avut-o în vedere în cadrul sistemului DiaMed.

IX. Aplicații ale algoritmului DiaMed în diagnoza sistemelor dinamice

9.1. Decizie fuzzy și sisteme de inferență fuzzy

Ne-am propus în capitolul curent să adaptăm abordarea DiaMed pentru diagnoza sistemelor dinamice și să o comparăm cu alte abordări înrudite ale acestei probleme [MunDum06c]. Termenii de referință ai acestei comparații sunt sistemele de inferență fuzzy (în forma propusă de Frank în [Fran96]), și algoritmul lui Koscielny [KoSedZac99] de ajustare dinamică a bazei de reguli fuzzy pentru diagnoză, folosit pentru eficientizarea calculului în cazul sistemelor complexe, de dimensiuni mari.

Am stabilit că diagnoza sistemelor dinamice presupune două etape principale, distincte: detectarea defectelor și diagnoza defectelor. Ne vom concentra în continuare numai asupra etapei de diagnoză a defectelor, întrucât aceasta este etapa în care abordarea DiaMed își poate dovedi aplicabilitatea și utilitatea.

a) Sisteme de inferență fuzzy în modelarea sistemelor și diagnoza defectelor

Subsecțiunea prezentă ilustrează cum se poate utiliza logica fuzzy în motorul de inferențe decizional pentru diagnoza defectelor.

Un model analitic precis și fidel al unui sistem dinamic va conduce la inconveniența ca orice eroare de modelare să afecteze performanțele schemei de detectare și izolare a defectelor (FDI), în special în cazul sistemelor nelineare și incerte (adică majoritatea sistemelor reale) [Cal01]. Această problemă poate fi evitată prin aplicarea regulilor și logicii fuzzy, înlocuind astfel un model exact prin descrierea comportamentului sistemului sub forma unor relații de tipul *if-then*. Logica fuzzy are în plus avantajul reprezentării naturale a informațiilor simbolice, lucru de foarte mare ajutor în multe situații.

Frank [Fran96] propune o abordare bazată pe logica fuzzy pentru evaluarea reziduurilor, pentru a izola defectele. Fie $R = \{r_1, r_2, \dots, r_m\}$ mulțimea de reziduuri. Fiecare reziduu r_i , ($i=1, \dots, m$), este descris prin intermediul mulțimilor fuzzy $\{r_{i1}, r_{i2}, \dots, r_{is}\}$, ale căror funcții de apartenență se pot identifica fie prin cunoștințele unui expert, fie prin învățare cu rețele neurale. Relațiile cauzale între reziduuri și defecte se pot exprima prin reguli *dacă-atunci* de forma:

$$DACĂ (efect = r_{ip}) \text{ȘI} \dots \text{ȘI} (efect = r_{jq}) ATUNCI (cauza = defectul k) \quad (9.1)$$

Ieșirea clasificatorului fuzzy este vectorul de defecte F . Procesul de inferență fuzzy asociază fiecărei componente F_l , $l=1, \dots, n$, o valoare între 0 și 1 ce indică gradul în care sistemul monitorizat e normal (componenta corespunzătoare fiind F_0) sau e afectat de către defectul l , $l=1, \dots, n$. Dacă se presupune că sistemul poate fi afectat doar de un defect la un moment dat, vectorul de defecte va conține doar o componentă cu o valoare mai

mare decât un prag dat, corespunzătoare defectului ce afectează sistemul. Dacă se lucrează cu ipoteza defectelor multiple, atunci toate componentele vectorului de clasificare ce depășesc pragul sunt indicatori ai defectelor apărute în sistem. Avantajul folosirii acestui clasificator fuzzy rezidă în faptul că regulile fuzzy oferă detalii asupra mapării reziduurilor într-o anumită stare de defect. Dezavantajul apare în cazul sistemelor complexe, întrucât atunci numărul de reguli devine foarte mare. O soluție a acestei probleme este propusă în [KoSedZac99]. Menționăm, deasemenea, că în abordarea lui Frank este necesară câte o regulă pentru fiecare valoare anormală a aceleiași variabile. Acest dezavantaj este îmbunătățit semnificativ de metoda din secțiunea următoare (practic o adaptare a etapei de selecție a ipotezelor din DiaMed).

b) Decizie fuzzy pentru selectarea simptomelor

Modelul de decizie fuzzy folosit în DiaMed poate fi adaptat în diagnoza sistemelor dinamice fără modificări notabile, după cum se observă în continuare.

Cheia reprezentării din DiaMed este să definim defectele $F = \{f_1, \dots, f_N\}$ cu ajutorul unor funcții de decizie ponderate $(F_1^{w_1}, \dots, F_N^{w_N})$ [Mun05]:

$$F_i^{w_i} = h_i(R_{i_1}(o), \dots, R_{i_{n_i}}(o)) \quad (9.2), \text{ unde:}$$

- $Simptome(f_i) = \{r_{i_1}, \dots, r_{i_{n_i}}\}$, și $i_1, \dots, i_{n_i} \in \{1, \dots, K\}$ sunt indicii reziduurilor relevante pentru defectul f_i ;
- $R_j : dom_j \rightarrow [0,1]$, este funcția fuzzy ce definește reziduul r_j (dom_j este domeniul funcției R_j),
- w_i este vectorul ponderilor simptomelor în definiția defectului f_i ;
- h_i este o funcție de agregare bazată pe operatori fuzzy ce exprimă modul de raționare al unui expert uman; și
- $o = (R_1(o), \dots, R_K(o))$ este punctul K-dimensional al observațiilor pentru sistemul monitorizat.

Ideea ce stă la baza diagnozei este aceeași cu cea prezentată în Capitolul V: se calculează indicii $F_i^{w_i}(o)$ care măsoară gradul de potrivire al observațiilor cu criteriul complex ce descrie defectul f_i , se compară cu un prag, și apoi se ierarhizează ipotezele selectate pe baza indicilor. Reluăm într-o notație modificată definiția unui astfel de indice din Capitolul V.

Definiție. Fie $f_i \in F$, și s sistemul monitorizat. *Scorul* defectului f_i la sistemul s este:

$$Scor(s, f_i) = S_g(M_{i_1}(o), \dots, M_{i_{n_i}}(o)) \quad (9.3)$$

(semnificația notațiilor a rămas aceeași). ■

Se moștenește astfel avantajul focalizării rapide a căutării pe direcțiile relevante, printr-o tehnică de calcul deopotrivă simplă și eficientă, dar care ignoră interacțiunile posibile între ipotezele de diagnostic și nu este foarte transparentă, dezavantaj depășit de metoda descrisă în Secțiunea 9.3.

9.2. Studiu comparativ a două abordări pentru sistemele complexe

a) Un algoritm dinamic pentru un sistem complex

Autorii lucrării [KoSedZac99] propun o metodă de a evita explozia exponențială a spațiului de reprezentare din abordarea Frank. Rezumăm în continuare ideile lor.

Definiție [KoSedZac99]. Un sistem de izolare a defectelor (FIS: Fault Isolation System) este un cvadruplu $\langle F, R, V, \varphi \rangle$, unde:

- F este mulțimea de stări $F = \{f_0, f_1, \dots, f_k\}$, f_0 este starea normală și restul sunt stările de defect;
- $R = \{r_1, \dots, r_J\}$ este mulțimea reziduurilor (simptomele);
- $V = \bigcup_{r_j \in R} V_j$, unde $V_j = \{v_{ji}\}$, ($i \in I_j$), este variabila lingvistică ce descrie reziduuul r_j , cu valorile sale fuzzy;
- $\varphi(f_k, r_j) = V_{kj}$ reprezintă mulțimea valorilor lingvistice ale reziduuului r_j în cadrul defectului f_k (pot exista mai multe reguli pentru același defect).

Dacă luăm ca valori fuzzy posibile ($N+$, P , $N-$ negativ $-$, pozitiv = normal, negativ $+$) regula corespunzătoare defectului f_k ar putea fi, spre exemplu, de forma:

$$DAC\dot{A} \ (v_I = N+) \ \dot{s}i... \ \dot{s}i \ (v_J = P) \ \dot{s}i... \ \dot{s}i \ (v_J = N-) \ ATUNCI \ defect \ f_k \quad (9.4)$$

unde v_j sunt valorile fuzzy cu grad de realizare maxim pentru fiecare reziduu r_j .

Gradul de activare al fiecărei reguli este calculat ca un indice ce măsoară apropierea între un reziduu r_j și valorile sale obținute pentru un anumit defect f_k , diagnosticul final fiind format din defectele cu grad de activare maxim. Pentru diagnoza sistemelor complexe, abordarea este foarte greu de aplicat practic, din cauza dimensiunii mulțimii defectelor și reziduurilor. Koscielny propune restrângerea dinamică a mulțimii de reziduuri cu care se lucrează efectiv la o submulțime R^* , în funcție de o mulțime de defecte posibile la un moment dat F^* [KoSedZac99].

b) Rafinarea deciziei de diagnostic în DiaMed și diagnoza sistemelor complexe

Al doilea nivel al sistemului hibrid DiaMed (folosit pentru discriminare și explicare) reprezintă o alternativă a abordării [KoSedZac99] în evitarea exploziei exponențiale a spațiului de reprezentare. Deși necesită un model mai detaliat și mai bine structurat al sistemului observat, are avantajul unor mai bune facilități de explicare a deciziei.

Abordarea DiaMed este mai potrivită pentru procesele tehnice pentru care se cunosc, măcar parțial, relații de bază între defecte și simptome, și aceste relații se pot reprezenta sub forma unor relații cauzale de tipul *defect* \rightarrow *evenimente* \rightarrow *simptome*. Este suficient să reprezentăm adecvat sistemul (după schema din Figura 8.1), pentru a putea implementa diagnoza cu ajutorul procedurii descrise de Secțiunea 8.3.

Am subliniat la momentul potrivit faptul că algoritmul propus oferă o schemă dinamică și eficientă de abordare a raționamentului nemonoton, cu explicațiile deciziilor înglobate în structura argumentelor. Avantajul său principal față de abordarea lui Koscielny a fost subliniat deja: acesta constă în nivelul superior de transparență a deciziei de diagnostic, bazată pe un model mai detaliat al sistemului observat. În plus, structura modelului pe care lucrăm permite o schemă mai inteligentă de restricționare dinamică la un sub-model

posibil din spațiul total de cautare, iar admisibilitatea este o alegere mai naturală pentru reprezentarea diagnosticelor în situații reale.

9.3. Concluzii și contribuții

Am confruntat în capitolul curent modelul hibrid DiaMed cu problema diagnozei sistemelor dinamice. Am folosit sistemele de inferență fuzzy ca termen de referință al comparației, întrucât ele reprezintă alternativa de diagnostic cea mai apropiată de decizia fuzzy. Dezavantajul DiaMed (necesitatea unui model mai complet al structurii și interacțiunilor din sistem) este echilibrat de avantajele obținute: atât precizie și explicații superioare, cât și o reprezentare mai naturală, mai nuanțată și mai ușor de interpretat. În plus, eficiența are de câștigat din faptul că modelăm efectiv mai puține variabile în sistemul de diagnoză (spre deosebire de sistemele de inferență fuzzy).

Contribuția capitolului stă în faptul că propune sistemul DiaMed ca pe o alternativă de modelare, care, atunci când dispunem de informațiile corespunzătoare asupra sistemului, poate fi o modalitate eficientă de abordare a sistemelor complexe, cu multe variabile. Am arătat, prin prezentarea algoritmului lui Koscielny, că ideea de a restructura dinamic un spațiu de căutare complex a mai fost folosită în automatică, și am argumentat teoretic faptul că DiaMed prezintă anumite avantaje (enunțate mai sus), față de algoritmul respectiv.

Atât avantajele cât și dezavantajele rămân deschise studiului practic prin implementarea pe sisteme cât mai variate și prin compararea cu alte modalități de lucru.

X. Concluzii și direcții de dezvoltare

*“-În ce direcție trebuie să merg acum?”
„-Depinde unde vrei să ajungi.”*

(Lewis Carroll: “Alice în Țara Minunilor”)

Lucrarea prezentă și-a propus o analiză a problemei diagnozei în general și a diagnozei medicale în special. Originalitatea sa stă în contribuții ce ating atât partea de modelare și raționament pentru diagnoză, cât și unele aspecte de sistematizare și comparare a abordărilor de până în prezent.

Capitolul II ne-a introdus în specificul diagnozei medicale, cu relevarea principalelor surse de dificultate și eroare și oferind o imagine detaliată a procesului de diagnoză în sine. Acest specific reprezintă justificarea teoretică a necesității abordărilor complexe din capitolele următoare. Capitolele III și IV au propus o sistematizare comparativă a tehnicilor de diagnoză de până în prezent, introducând sistemul original DiaMed în contextul general și prezentând, spre comparație, două sisteme clasice de diagnoză medicală înrudite cu el: CASNET și CHECK (DiaMed fiind o continuare posibilă a ideilor din aceste sisteme). O contribuție importantă a celor două capitole stă în faptul că acestea prezintă unele idei de tratare posibilă a diagnozei ce nu au fost încă exploatate și

pot constitui punctul de plecare al unor studii viitoare. Capitolele justifică, deasemenea, utilitatea hibridizării combinate în DiaMed.

Am văzut, apoi, în Capitolul V, cum putem modela diagnosticele medicale prin intermediul unor funcții de decizie fuzzy (folosite drept funcții de discriminare pentru nivelul de selecție de ipoteze din DiaMed), și care sunt avantajele acestei modelări, comparativ cu sistemele de inferență fuzzy. Deși funcțiile de decizie în sine nu sunt o contribuție personală, felul în care le-am adaptat pentru descrierea bolilor (defectelor) este o idee originală.

Bazându-se pe nonmonotonicitatea raționamentului de diagnoză și pe necesitatea unor relații asimetrice de atac (necesitate exemplificată în Capitolul II), Capitolul VI introduce sistemele de argumentare directă ca formalism modern de reprezentare, și propune o definiție originală a diagnosticului multiplu, apelând la semantica admisibilă (specific argumentativă). Capitolul prezintă și bazele implementării eficiente a calculului acestei semantici folosind backtracking dinamic pentru algoritmi de satisfacere a constrângerilor dinamici (DCSP). O contribuție a acestui capitol este dată de traducerea mecanismului dinamic de testare din diagnoză într-o problemă de rezolvare a constrângerilor dinamică.

Capitolul VIII adaptează un algoritm eficient pentru problemele dinamice de satisfacere a constrângerilor (DCSP) la diagnoza medicală, definindu-i un model de ștergere/adăugare de constrângeri bazat pe rezultatele testelor medicale, obținute în cursul investigării unui anumit pacient, și implementează efectiv acest algoritm în sistemul DiaMed. În plus, am folosit un model original de reprezentare a cunoștințelor în medicină, model bazat tot pe argumente.

Capitolul IX arată că tehnologia de modelare din DiaMed nu este restricționată la probleme de tip medical și se poate adapta sistemelor dinamice, în măsura în care există un model adecvat pentru defectele acestora, aducând unele avantaje în cazul sistemelor complexe.

Ca o concluzie finală, credem că lucrarea noastră constituie o pledoarie pentru sistemele de argumentare directă și pentru aplicabilitatea lor la problemele dificile ale lumii reale.

Am subliniat deja realizările și avantajele sistemului original DiaMed propus de lucrarea noastră. Ca întotdeauna însă, se poate merge și mai departe. Încă nu am răspuns la o întrebare importantă: cum anume este corect să fie măsurată puterea de discriminare a unui test cu răspuns multiplu. Deasemenea, am lăsat o parte importantă a testării la decizia utilizatorului (nu am automatizat decât parțial acest proces), și ar mai fi cu siguranță multe de adăugat pe marginea testării, care poate constitui, singură, subiectul unui tratat. Deasemenea, modul de reprezentare poate fi îmbunătățit, iar modelul medical necesită completări serioase din partea unei echipe de specialiști, pentru a face posibilă o evaluare în funcție de un set semnificativ de date reale. Ar mai merita studiat în continuare impactul tehnicilor inteligente asupra căutării regulilor aplicabile în inferența propozițională în general.

Așa cum argumentam în Capitolul II, diagnoza medicală este una dintre cele mai dificile probleme de raționament uman. A automatiza diagnoza medicală înseamnă, practic, a automatiza gândirea umană. Ceea ce probabil că nu se va realiza niciodată pe deplin. Și poate că, de fapt, nici nu este de dorit.

Bibliografie selectivă

[BarGel94]Baral C., Gelfond M. (1994), "Logic Programming and Knowledge Representation", *Journal of Logic Programming* 19,20, :73-148;

[BDKT97]Bondarenko A., Dung P.M., Kowalski R.A.,Toni F. (1997), "An abstract , argumentation-theoretic approach to default reasoning", *Artificial Intelligence*, Vol.93, Issue 1-2, :63-101;

[BesDou04]Besnard P., Doutre S. (2004): "Checking the acceptability of a set of arguments", *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning*, Canada, :59-64;

[BreDix97]Brewka, Dix, Konolige (1997) "Nonmonotonic Reasoning: an overview", *CLSI Lecture Notes 73*, Stanford, California;

[Byl91]Bylander, D.Allemang, M.Tanner, J.R.Josephson (1991): "The Computational Complexity of Abduction",*Artificial Intelligence*, Vol.49, :25-60;

[Cal01] Calado J. & all (2001). "Soft Computing Approaches to Fault Diagnosis for Dynamic Systems", *European Journal of Control* 7:248-286;

[Cart80]McCarthy, J.,(1980), "Circumscription - a Form of Non-Monotonic Reasoning", *Artificial Intelligence* 13, North-Holland, :27-39;

[CasFanMen03] Castellano G., Fanelli A., Mencar C. (2003): "A Fuzzy Clustering Approach for Mining Diagnostic Rules", *Proceedings of 2003 IEEE International Conference on Systems, Man and Cybernetics*, vol1, :2007-2012, Washington D.C., USA;

[Cla90]Clark P. (1990), "Nonmonotonic Reasoning, Argumentation and Machine Learning", *Technical Report TIMLG-38*, Turing Institute, Glasgow, UK;

[ConTor92]L.Console, P.Torasso (1992): "A Spectrum of Logical Definitions of Model-based Diagnosis", in W.Hamscher et al.(eds.) *Readings in Model-based Diagnosis*, Morgan Kaufman Publ., San Mateo,CA;

[deKleer86]de Kleer J.(1986), "An assumption-based truth maintenance system", *Artificial Intelligence* 28, :127-162;

[deKleer89b]deKleer J. (1989), "Propositional Inference in CSP and ATMS techniques", SSL Paper P89-00023;

[deKleer90]deKleer J. (1990), "Exploiting locality in a TMS", *Proceedings of AAAI-91*, :264-271, Anaheim, CA.

- [DerDoy80]McDermott,D., Doyle,J.(1980), "Non-Monotonic Logic I", *Artificial Intelligence 13*, North-Holland, :41-72;
- [Der82]McDermott,D.(1982), "Non-Monotonic Logic II: Nonmonotonic Modal Theories", *JACM 29*, :33-57;
- [Dix55]Dix J. (1955), "A Classification Theory of Semantics of Normal Logic Programs: I.Strong Properties", *Fundamenta Informaticae XXII(3)*, :227-255;
- [Doutre02]Doutre S.(2002), "Autour de la semantique preferee des systemes d'argumentation", These en informatique, Universite Toulouse III-Paul Sabatier;
- [DubPra85]Dubois D., Prade H.(1985) : "*Theorie des possibilites: applications a la representation des connaissances en informatique*", Barcelon NY Paris Masson;
- [DudHarSto00]R.Duda , P.Hart , D.Stork (2000): "Pattern Classification", John Wiley Interscience;
- [DumiBuiu00] Dumitrache I., Buiu C., (2000): „*Algoritmi Genetici – Principii Fundamentale și Aplicații în Automatică*“, Ed Mediamira, Cluj-Napoca;
- [DumMihRos98]I. Dumitrache, I.R.Mihu, K.Rosu(1998): "Hybrid Decision Making System for Medical Diagnosis", *Proceedings of the Third International Conference on Neural Networks and Expert Systems in Medicine and Healthcare*, World Scientific Publishing;
- [Dun95]Dung.P.M. (1995) "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-persons games", *Artificial Intelligence 77*:321-357,
- [Dup94]D.T.Dupre (1994): "Characterizing and Mechanizing Abductive Reasoning", Ph.D. Thesis, Department of Computer Science, University of Torino;
- [EshKo88]Eshghi K., Kowalski R.(1988), „Abduction through deduction“, *Technical Report Department of Computing*, Imperial College, London;
- [EshKo89]Eshghi K., Kowalski R.(1989), „Abduction compared with negation by failure“, *Proceedings 6th International Conference on Logic Programming*, MIT Press, Lisbon, :234-255;
- [Fran96]P.M.Frank (1996): "Analytical and Qualitative Model-based Fault Diagnosis ", *European Journal of Control*, 2 (1), :6-28;
- [Ful95] Fuller R., (1995). "*Neural Fuzzy Systems*", Abo Akademi's tryckeri, Abo, ESF Series A: 443;

[GiaRil94]J.Giarratano, G. Riley (1994): "Expert Systems: Principles & Programming", PWS Pub.Co, Boston;

[Gin96]Ginsberg M., Crawford J., Etherington D., "Dynamic Backtracking", University of Oregon, *Technical Report 8/1/1996*;

[GovMah00]Governatori G., Maher M.J., Antoniou G., Billington D. (2000), "Argumentation Semantics for Defeasible Logics", in Mizoguchi, R. and Slaney J., editors, *PRICAI 2000: Topics in Artificial Intelligence*, Vol. 1886 of LNAI, Berlin Springer Verlag, :27-37;

[Hay98]S.Haykin (1998): "Neural Networks: A Comprehensive Approach", Prentice Hall;

[HuaHen96]K. Huang, M.Henrion (1996): "Efficient Search-Based Inference for Noisy-OR Belief Networks: TopEpsilon", *12th Conference on Uncertainty in Artificial Intelligence, Portland, OR*, 325-331;

[Karr91] Karr C., (1991): "Design of an adaptive fuzzy logic controller using a genetic algorithm", in *Proceedings of the 4th International Conference on Genetic Algorithms*, Belew and Brooker eds.;

[KunSte99] Kuncheva L., Steimann F. (1999): „Fuzzy Diagnosis“, *Artificial Intelligence in Medicine*, 16(2):121-128;

[Ise97] Isermann R., (1997). "Supervision, Fault Detection and Fault Diagnosis Methods-an Introduction", *Control Engineering Practice*, vol. 5, no. 5 : 639-652;

[JafLas87]Jaffar J., Lassez J-L.,(1987) "Constraint Logic Programming", Proc. 14th ACM Symposium on Principles of Programming Languages, Munich;

[Jan92]R.Jang (1992): "Neuro-Fuzzy Modeling: Architecture, Analysis and Applications", PhD. thesis , University of California at Berkeley;

[KaManDun94]Kakas A., Mancarella P., Dung P.M.(1994), „The acceptability semantics for logic programs“, *Proceedings 11th International Conference on Logic Programming*, MIT Press, Santa Margherita Ligure, Italy, :504-519;

[KKoT98]Kakas A., Kowalski R., Toni F., (1998), "The role of abduction in logic programming", *Handbook of logic in Artificial Intelligence and Logic Programming 5*, Oxford University Press, :235-324;

[Kon88]Konolige, K.,(1988), "Defeasible argumentation in reasoning about events" , in Z.W.Ras and L.Saitta, editors, *Methods for Intelligent Systems*,3, Elsevier NY, pp 380-390;

- [Kon92]K.Konolige (1992): “Abductive Theories in Artificial Intelligence” in Brewka, G. (ed) *Principles of Knowledge Representation*, CSLI Publications, Stanford University;
- [KoSedZac99] Koscielny, J. M., Sedziak, D., and Zackroczynsky, K. (1999). “Fuzzy-logic fault isolation in large-scale systems”, *International Journal of Applied Mathematics and Computer Science* **9** (3): 637-652.
- [LeeTam]Lee J., Tam V., (1995): “A framework for integrating artificial neural networks and logic programming”, *International Journal of Artificial Intelligence Tools* **4**, :3-32;
- [Lon01]W. Long (2001): “Medical Informatics: reasoning methods”, *Artificial Intelligence in Medicine*, 2371-2387;
- [McAll80]McAllester D.(1980), “An outlook on truth maintenance”, *Artificial Intelligence Laboratory*, AIM-551, MIT, Cambridge, MA;
- [McIlr98]S.McIlraith (1998): “Logic-based Abductive Inference”, Knowledge Systems Laboratory, *Technical Report KSL-98-19*;
- [Mich96]Michalewicz Z. (1996), “Genetic Algorithms + Data Structures= Evolutionary Programs”, Springer, Berlin;
- [MichFog04]Michalewicz Z., Fogel D.(2004), “How to solve it: modern heuristics”, Springer, Berlin;
- [MigI01]Miguel I., “*Dynamic Flexible Constraint Satisfaction and its Application to AI Planning*”, PhD Thesis, University of Edinburgh, 2001;
- [Mit00]S. Mitra (2000): “Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework”, *IEEE Transactions on Neural Networks*, vol. 11-3;
- [Mun03]Munteanu, S.(2003), "Medical diagnosis modeled in terms of fuzzy decision", *Proc. Of 14th International Conference on Control Systems and Computer Science*, Bucharest;**
- [Mun04]Munteanu S. (2004), “Argumentation-based Conflict Resolution for Discriminating Diagnostic Hypotheses”, *The 12th International Symposium on Modeling, Simulation and System’s Identification*, Galati, :218-225;**
- [Mun05]Munteanu, S.(2005), “A Hybrid Model for Diagnosing Multiple Disorders”, *International Journal of Hybrid Intelligent Systems*, Vol 2. No.1 IOSPress, :35-55;**
- [MunDum06a]Munteanu S., Dumitrache I. (2006) “A DCSP-based nonmonotonic framework for medical diagnosis”, submitted to *Artificial Intelligence in Medicine*;**

[MunDum06b]Munteanu S., Dumitrache I. (2006) “The Intelligence of a Hybrid System: DiaMed”, *Technical Report, Polytechnica University, Bucharest- to appear in Control Engineering and Applied Informatics*;

[MunDum06c]Munteanu S., Dumitrache I. (2006) “Applying DiaMed Hybrid Algorithm to the Diagnosis of Dynamic Systems”, *Technical Report, Polytechnica University, Bucharest -to appear in the Scientific Bulletin of Politehnica University, Bucharest*;

[NeaPal01]D. Neagu, V. Palade (2001): “*Sisteme Inteligente Hibride*”, Ed. MatrixRom, București;

[Nyb99]M.Nyberg (1999): “Model Based Fault Diagnosis”, PhD. thesis, Linkoping University, Sweden;

[PalNea01]Palade V., Neagu C.(2001): “*Sisteme inteligente hibride*”, MatrixRom, București;

[Par94]Paredis J.(1994), “Co-evolutionary constraint satisfaction”, *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*”, : 46-55;

[Patt94]R.Patton (1994): “Robust Model Based Fault Diagnosis: the State of the Art”, *IFAC SAFEPROCESS '94*, Helsinki, Finland;

[PattChe97]R.J. Patton , Chen J. (1997) “Observer-based Fault Detection and Isolation: Robustness and Applications”, *Control Eng. Practice*, vol. 5, no.5, pp 671-682;

[Pea01a]J.Pearl (2001)a: “Causal Inference in the Health Sciences : A Conceptual Introduction”, *TR R-282*, Cognitive Systems Lab. UCLA;

[Pea01b]J. Pearl (2001)b: “Bayesianism and causality or why I am only a half bayesian”,in D. Corfield, J. Williamson (eds.), *Foundations of Bayesianism*, Kluwer Applied Logic Series;

[Ped04]W.Pedrycz (2004): “Logic-Oriented Fuzzy Neural Networks”, *IJHIS* vol 1.(1):3-11;

[PerApAl93]Pereira, L.M., Aparicio, J.N., Alferes, J.J.(1993), "Non-monotonic Reasoning with Logic Programming", *Journal of Logic Programming*, Special Issue on Nonmonotonic Reasoning 17 (2,3&4);

[Poll92]J.Pollock (1992) : “How to Reason Defeasibly”, *Artificial Intelligence* 57, pp1-42;

[Poll02]Pollock, J. (2002), "Defeasible Reasoning with Variable Degrees of Justification", *Artificial Intelligence* 133, :233-282;

- [PopIon99]Popescu E., Ionescu R., (1999) "Rheumatology", Technical Press, Bucharest;
- [PraVre]H.Prakken,G.Vreeswijk:"Logics for Defeasible Argumentation" (2002), Gabbay D., Guenther F., eds. "*Handbook of philosophical logic*", vol. 4, Kluwer Academic Publishers, :218-319;
- [Rei80]Reiter, R. (1980), "A Logic for Default Reasoning", *Artificial Intelligence 13*, North-Holland, :81-132;
- [ReiKle87]Reiter R., de Kleer J. (1987): "Foundations of Assumption-based TMS", *National Conference on AI*, Seattle, WA, :183-189;
- [Res88]Restian A.(1988), "Diagnosticul medical", Ed. Dacia, Cluj;
- [Rim83]Rimniceanu R.(1983), "Aspecte si probleme ale medicinei contemporane", Ed. Medicala, Bucuresti;
- [RusNor02]Russell S., Norvig P.(2002), "Artificial Intelligence: A Modern Approach", Prentice Hall;
- [San99]Sanchez M.M. (1999), "A Neurosymbolic Approach to the Classification of Scarce and Complex Data", PhD Thesis, University of Sussex;
- [Sha00]Shankar R.D & al.(2000), "Implementing Clinical Practice Guidelines while Taking Account of Evidence", *AMIA Annual Symposium*, Los Angeles, CA, :303-304;
- [ShaMus99]Shankar R.D, Musen M.(1999) ,"Justification of Automated Decision-Making: Medical Explanations as Medical Arguments", *Proc AMIA Symposium*, Washington DC, :395-399;
- [SouKay02]Sousa, J., Kaymak, U. (2002) :"*Fuzzy Decision Making in Modeling and Control*", World Scientific Pub Co;
- [StaSus77]Stallman R., Sussman G., (1977) "Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis", *Artificial Intelligence*, 9(2):135-196;
- [TaSu85] Takagi T., Sugeno M. (1985) "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Transactions on Systems, Man and Cybernetics*, vol SMC-15, no.1, January-February 1985;
- [TeoCheGal89] H.N.Teodorescu, M.Chelaru, D.Galea & al.(1989) "*Fuzzy Systems and Applications*", I.P.Iași;

[TeoCheKan01] H.N.Teodorescu, M.Chelaru, A. Kandel & al. (2001) „Fuzzy Methods in tremor assessment, prediction and rehabilitation”, *Artificial Intelligence in Medicine*, 21(1-3), :107-130;

[TorCon89]P.Torasso, L.Console (1989): “A multilevel architecture for diagnostic problem-solving”, *Computational Intelligence*,1 :101-112.

[Tou58]Toulmin, S.E. (1958), *"The Uses of Argument"*, University Press, Cambridge, UK;

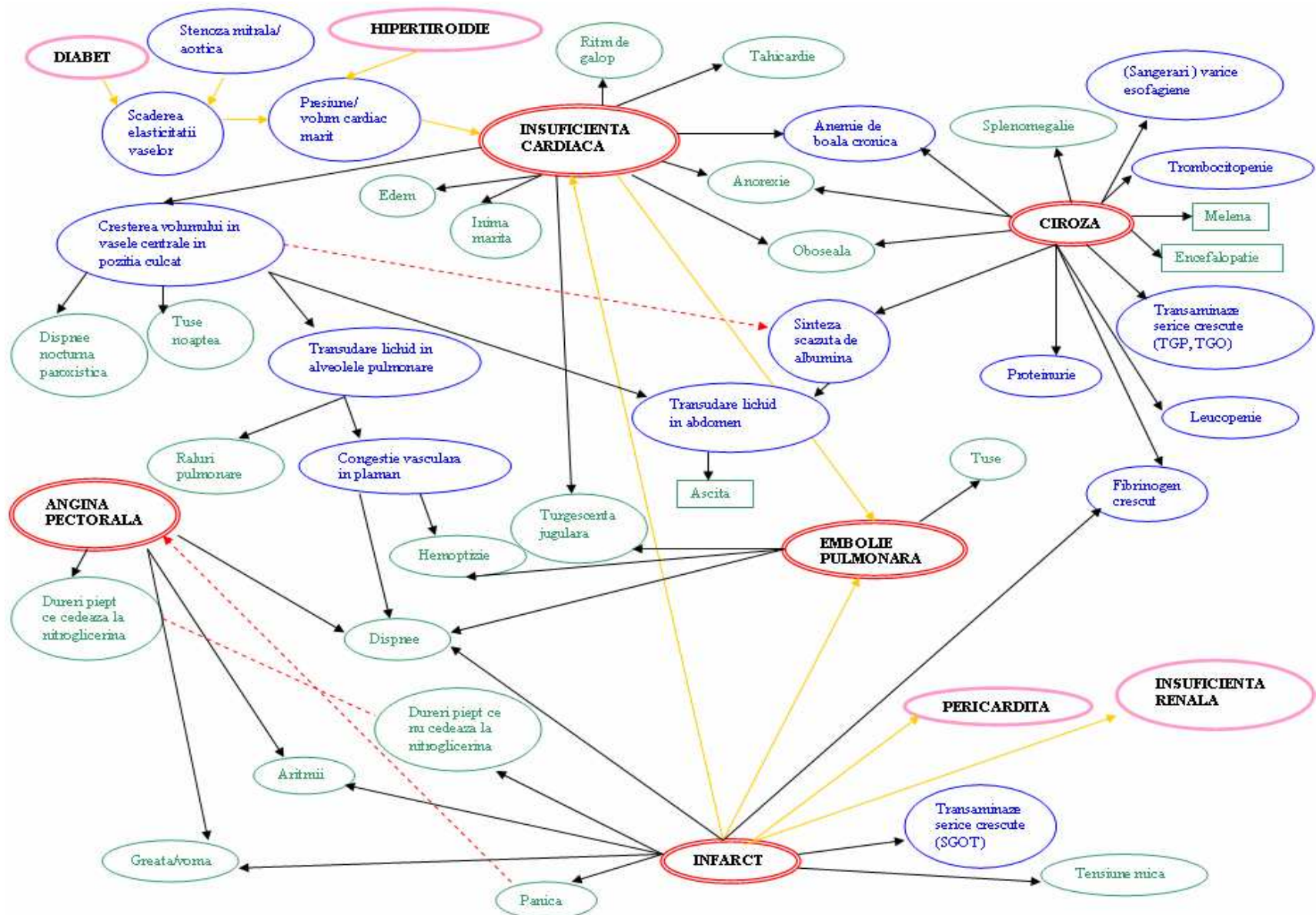
[VerSchi]Verfaillie G., Schiex T. (1994), “Dynamic Backtracking for Dynamic Constraint Satisfaction Problems”, *Proceedings of the ECAI '94 Workshop on Constraint Satisfaction Issues Raised by Practical Applications*, Amsterdam, the Netherlands;

[Wal75]Waltz D. (1975), “Understanding line drawings of scenes with shadows”, in *"The Psychology of Computer Vision"*, :19-91. McGrawHill;

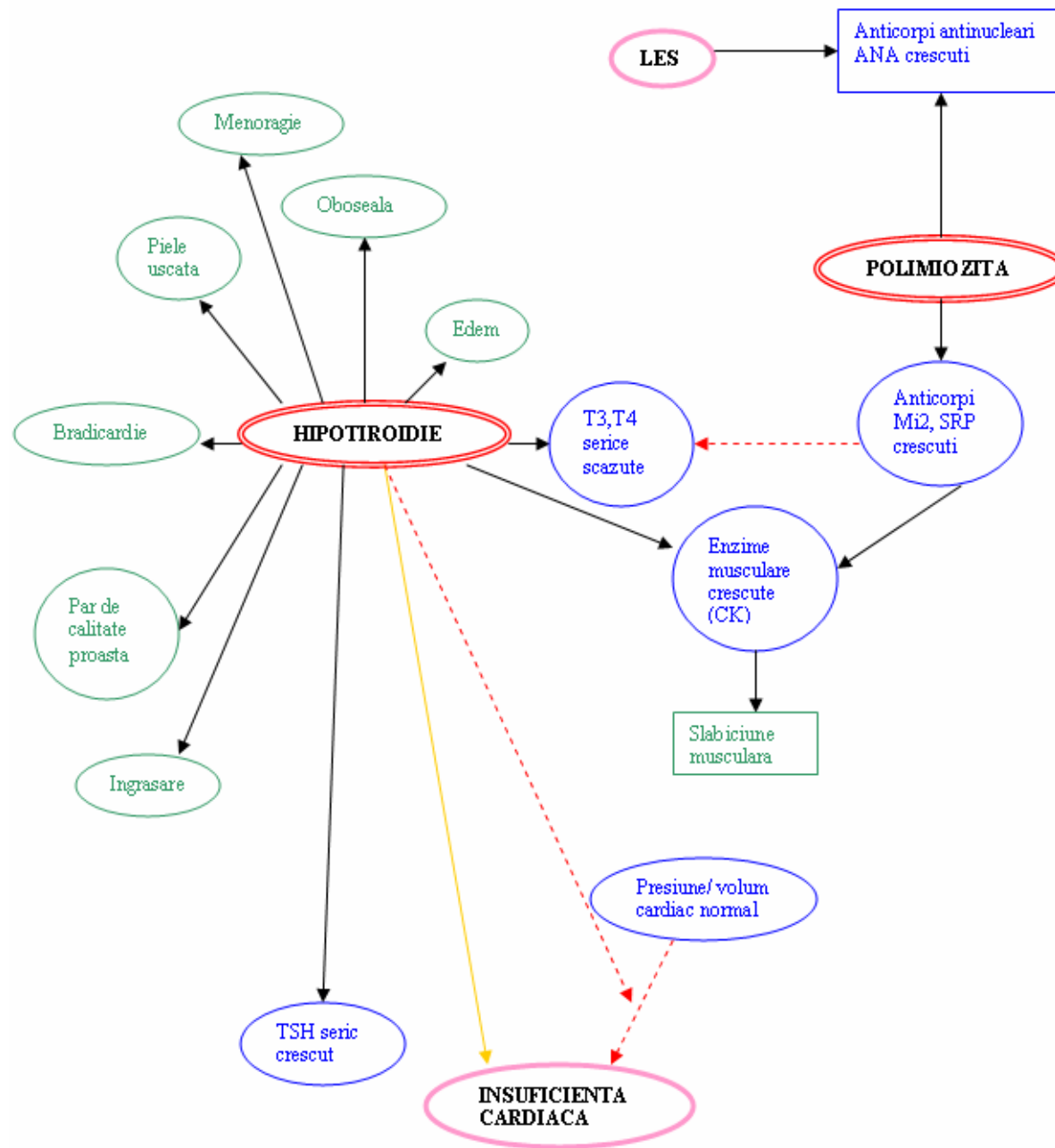
[Weis74] Weiss S. (1974), “A System for Model-Based Computer Aided Diagnosis and Therapy”, PhD Thesis, Department of Computer Science, Rutgers University, CBM-TR-27;

[Zad73] Zadeh L., (1973) “Outline of a new approach to the analysis of complex systems and decision processes”, *IEEE Transactions on Systems, Man and Cybernetics*, 3, :28-44.

Anexa1. Modelul medical



PLANȘA I



PLANȘA VI