Computational aspects of planning using fuzzy inference algorithms

Conf.dr. Vasile MAZILESCU
Catedra de Informatică Economică, Universitatea „Dunărea de Jos”, Galați

In this paper is investigated the usefulness of temporal projection and the decidability of a fuzzy inference algorithm, for a planning problem. We focus this way on the main qualitative aspects, used in agent synthesis.
Keywords: *intelligent agents, planning, decidability, fuzzy inference algorithm.*

1 Introducere

Planificarea este un task de raționament, în vederea determinării unei secvențe de operatori care permite atingerea scopului, pornind de la o stare inițială dată. Pot fi formulate probleme de planificare diferite prin restricționarea tipului de operatori, impunând o serie de limitări a numărului de precondiții şi postcondiții (mulțimi finite de elemente care pot fi variabile sau constante) şi prin utilizarea anumitor operații logice în structura operatorilor. O problemă de planificare (**PP**) se poate defini astfel: dându-se o stare inițială, o stare dorită şi o mulțime de acțiuni posibile, să se determine o mulțime de acțiuni (parțial sau total ordonată), care aplicată stării inițiale conduce sistemul în starea dorită. Planificarea este o problemă dificilă sau poate deveni o problemă mai simplă în cazul în care se aplică anumite restricții. Validarea unei **PP** se reduce la verificarea pentru un plan, a unei stări inițiale şi a unei stări finale date, că toate acțiunile menționate în plan pot fi executate cu succes, adică toate precondițiile sunt îndeplinite, iar acțiunile menționate în plan conduc la starea dorită. Pentru astfel de restricții, **PP** poate fi polinomială sau NP-completă. Se demonstrează în [2] că **PP** este nedecidabilă, evidențiindu-se astfel dificultatea problemei de planificare în general, dar nu se demonstrează ce proprietăți trebuie îndeplinite pentru reducerea complexității. Complexitatea computațională a planificării este investigată permanent, dată fiind importanța practică a ei. Bylander analizează în [1] problema generală privind deciderea existenței unei soluții pentru un task de planificare în contextul sistemului STRIPS şi demonstrează că această problemă generală este PSPACE-completă.

Printre dificultățile de concepție ale unui sistem simbolic de inteligență artificială de tip planificator, enumerăm: 1) complexitatea computațională a proiecției temporale; 2) cele referitoare la decidabilitatea într-o anumită logică temporală a unei formule scop [5].

Lucrarea de față este alcătuită după cum urmează: secțiunea 2 precizează principalele aspecte ale raționamentului temporal prin prisma validării unui plan, precum şi prin prisma coerenței sistemelor cu evenimente [1,2]. În scopul susținerii ideilor prezentate anterior, secțiunea 3 prezintă un Algoritm de Inferență Fuzzy (**AIF**) elaborat de autor şi folosit efectiv în structura unui agent de planificare, fiind analizată decidabilitatea sa. Secțiunea 4 constituie concluziile lucrării şi punctează câteva aspecte calitative, ca rezultat al implementării unui agent specializat, notat **AgP**.

2. Coerența raționamentului temporal

Un tip de raționament temporal folosit în sinteza agenților inteligenți pentru **PP**, îl reprezintă evaluarea consecințelor unei mulțimi de evenimente. Această problemă este legată şi de alte aspecte ale raționamentului temporal, cum ar fi planificarea şi validarea unui plan. Întrucât planificarea se derulează incremental, interesează nu numai validarea planului dar şi determinarea cauzelor pentru care planificarea nu este încă validă.

Pentru formalizarea evaluării consecințelor unei mulțimi de evenimente vom evidenția noțiunile de structură cauzală, mulțime de evenimente parțial ordonată şi sistem de evenimente.

Definiția 1. O *structură cauzală* se defineşte prin tripletul **SC** = {P,E,R}, unde:

i) $P=\{p_1,\ldots,p_n\}$ este o mulțime de atomi propoziționali, numite şi condiții;

ii) **E**$=\{e_1,\ldots,e_n\}$ este o mulțime de tipuri de evenimente;

iii) $R=\{r_1,\ldots,r_n\}$ este o mulțime de reguli cauzale de forma $r_i=<e_i,\varphi_i,\alpha_i,\delta_i>$ unde e_i este tipul evenimentului de activare, $\varphi_i \subseteq P$ este o mulțime de precondiții, $\alpha_i \subseteq P$ este o listă de adăugare, iar $\delta_i \subseteq P$ este o listă de ștergere.

Considerând o structură cauzală **SC** = {P,E,R}, se poate defini o mulțime de evenimente parțial ordonată peste **SC**, notată $MEPO_{SC}$, unde $MEPO_{SC} = <E_{SC},\triangle>$. Cele două elemente din structura lui $MEPO_{SC}$ au următoarea semnificație: $\mathbf{E_{SC}}=\{e_1,\ldots,e_p\}$ reprezintă mulțimea de evenimente actuale cărora li se asociază funcția *tip*: $E_{SC}\rightarrow E$, iar relația $\triangle$ este o relație de ordine parțială strictă peste mulțimea $\mathbf{E_{SC}}$, adică este tranzitivă și nereflexivă. $MEPO_{SC}$ denotă mulțimi posibile de secvențe de evenimente care satisfac relația de ordine parțială $\triangle$.

Definiția 2. O secvență parțială de evenimente de lungime m peste $MEPO_{SC}$, notată $MEPO_{\mathbf{m}}$, se definește prin $\boldsymbol{f}=<f_1,\ldots,f_m>$ astfel încât:

i) $\{f_1,\ldots,f_m\}\subseteq \mathbf{E_{SC}}$,

ii) $f_i \neq f_j$ dacă $i\neq j$ și

iii) pentru fiecare pereche de evenimente f_i,f_j din $\boldsymbol{f}$, dacă $f_i \triangle f_j$, atunci $i<j$.

Dacă secvența de evenimente este $|MEPO_{\mathbf{m}}|$, atunci ea se numește o secvență completă de evenimente peste mulțimea de evenimente parțial ordonate $MEPO_{SC}$.

Mulțimea tuturor secvențelor de evenimente complete peste $MEPO_{SC}$ se notează cu MSEC.

O secvență parțială de evenimente $\boldsymbol{f}$ poate fi extinsă la o secvență de evenimente $\boldsymbol{g}$ dacă $|\boldsymbol{f}|<|\boldsymbol{g}|$ și pentru oricare f_i, f_j cu $i<j$ există $g_k = f_i$ și $g_l = f_j$ astfel încât $k<l$. Pentru o secvență de evenimente de lungime m, cu $k < m$, secvența de lungime k se numește secvență inițială notată $\boldsymbol{f} / f_k$. În mod similar, se notează secvența de evenimente până la evenimentul k, cu $\boldsymbol{f} \setminus f_k$.

Fiecare eveniment permite transformarea unei submulțimi de stări din P în altă submulțime de stări. Fie $S\subseteq P$ o stare și e un eveniment. În aceste condiții, regula cauzală r este aplicabilă în starea S dacă și numai dacă $r = (tip(e), \varphi, \alpha,\delta)$ și în plus $\varphi\subseteq S$.

Dându-se evenimentul e și starea S, *apl*(S,e) denotă mulțimea tuturor regulilor aplicabile în starea S pentru evenimentul e. Un eveniment e se numește admisibil în starea S dacă și numai dacă *apl*(S,e) $\neq\emptyset$. Se poate defini ce se înțelege prin rezultatul acțiunii unei secvențe de evenimente asupra unui sistem aflat în starea S, utilizând funcția ***Rezultă***. Ea se construiește recursiv pentru stări și secvențe de evenimente cu valori într-o mulțime de stări, astfel:

Rezultă (S,( ))=S,

Rezultă(S,(f;g))=***Rezultă***(S,f) $\{\delta(r) \mid r\in$ *apl* (***Rezultă***(S,f),g)$\} \cup \{\alpha(r) \mid r\in$ *apl*(***Rezultă*** (S,f),g).

Este posibil ca două reguli să fie aplicabile în situația în care o regulă adaugă un atom și cealaltă îl șterge. Deși este o situație nedorită, definiția permite acest lucru.

Definiția 3. O secvență de evenimente $\boldsymbol{f} = (f_1,\ldots,f_n)$ se numește admisibilă în starea S, dacă și numai dacă fiecare eveniment f_i, $1\leq i\leq m$, este admisibil în starea ***Rezultă***(S, $\boldsymbol{f} \setminus f_i$). Mulțimea tuturor secvențelor de evenimente complete (MSEC) care sunt admisibile în S se notează prin MSECA. Dacă MSEC=MSECA atunci structura cauzală este *coerentă* relativ la S.

Se pot analiza consecințele unei mulțimi MEPO pentru o anumită stare inițială, introducând noțiunea de sistem de evenimente Θ, care este o pereche $(MEPO_{SC},I)$, $MEPO_{SC}$ este o mulțime de evenimente parțial ordonată peste structura cauzală **SC**, iar $I\subseteq P$ reprezintă starea inițială a sistemului cu evenimente.

Definiția 4. Problema evaluării consecințelor unei mulțimi de evenimente este de a determina dacă o anumită condiție g este îndeplinită în mod *posibil* sau *necesar*, după o anumită secvență de evenimente din sistemul Θ.

Acest lucru se poate rescrie sub forma $g\in Poss^{+}(e,\Theta)$ și respectiv $g\in Nec^{+}(e,\Theta)$. Se consideră și problema determinării mulțimilor de condiții care au loc în mod posibil sau necesar, înainte de un anume eveniment, notată prin $g\in Poss^{-}(e,\Theta)$ și $g\in Nec^{-}(e,\Theta)$ respectiv. Pentru un sistem de evenimente Θ, un eveniment e și o condiție g:

$g\in Poss^{+}(e, \Theta) \Leftrightarrow (\exists)\boldsymbol{f}\in MSEC : g\in$ ***Rezultă***(I, $\boldsymbol{f}$/e)

$g\in Nec^{+}(e, \Theta) \Leftrightarrow (\forall)\boldsymbol{f}\in MSEC : g\in$ ***Rezultă*** (I, $\boldsymbol{f}$/e)

$g\in Poss^{-}(e, \Theta) \Leftrightarrow (\exists)\boldsymbol{f}\in MSEC : g\in$ ***Rezultă*** (I, $\boldsymbol{f} \setminus$ e)

$g\in Nec^{-}(e, \Theta) \Leftrightarrow (\forall)\boldsymbol{f}\in MSEC : g\in$ ***Rezultă*** (I, $\boldsymbol{f} \setminus$ e)

În felul acesta există patru probleme de evaluare a consecințelor unei mulțimi de evenimente în locul uneia. Din punct de vedere computațional, Nec^{+} și Nec^{-} sunt echivalente (după transformări polinomiale), proprietate care are loc și pentru $Poss^{+}$ și $Poss^{-}$. Dându-se o mulțime de condiții S şi o secvenţă de evenimente ***f***, ***Rezultă***(S,***f***) poate fi calculată într-un timp polinomial prin interpretarea definiţiei funcţiei ***Rezultă*** în mod procedural. Întrucât mulţimea MSEC poate conţine mai multe secvenţe în mod exponenţial, nu este evident că se poate decide în timp polinomial faptul că $g \in Poss^+$ sau $g \in Nec^+$. Într-un caz general problema evaluării consecinţelor unei mulţimi de evenimente este foarte dificilă, iar pentru cazul $g \in Poss^+$ problema de decizie este NP-completă iar pentru $g \in Nec^+$ problema este co-NP-completă, chiar în prezenţa anumitor restricţii severe (de tipul α sau δ să fie vide pentru toate regulile, sau cerând să existe o singură regulă cauzală pentru fiecare tip de eveniment).

Unul din motivele importante de analiză a problemei de evaluare a consecinţelor unei mulţimi de evenimente îl constituie faptul că ea reprezintă elementul central în planificarea neliniară, context în care evenimentele se numesc acţiuni, iar mulţimile de evenimente parţial ordonate se numesc planuri neliniare sau simplu planuri.

Definiţia 5. Un task de planificare TP este dat prin (SC,I,G), unde SC=(P,E,R) este o structură cauzală definită ca mai sus, $I \subseteq P$ şi $G \subseteq P$ reprezintă starea iniţială şi respectiv starea scop.

În aceste condiţii, un plan Δ_{SC} rezolvă task-ul de planificare dacă şi numai dacă: **i**) planul în mod necesar atinge scopul, adică $G \subseteq$ ***Rezultă***(I, ***f***) pentru toate $f \in MSEC(\Delta_{SC})$; **ii**) planul este coerent, adică $MSECA(\Delta_{SC},I)=MSEC(\Delta_{SC})$. Se admit numai planurile în care toate acţiunile sunt admisibile (adică se garantează faptul că precondiţiile lor sunt satisfăcute). Există abordări care nu impun ca planurile valide să fie coerente şi definesc faptul că acţiunile neadmisibile nu au nici un efect.

Definiţia 6. Problema de planificare este de a decide dacă există o soluţie pentru un task de planificare, sau de a determina această soluţie. O instanţă a problemei de existenţă a unui plan o reprezintă task-ul de planificare iar întrebarea care apare este dacă există un plan care rezolvă acest task.

Pentru task-urile de planificare simple se poate decide în timp polinomial dacă există o soluţie. Identificarea problemelor de planificare polinomială reprezintă un aspect important în înţelegerea exigenţelor computaţionale efective pentru raţionamentul temporal, care apare şi în sistemele de planificare. O structură cauzală este independentă dacă şi numai dacă pentru orice tip de eveniment logic $e \in E$, există numai o singură regulă care poate fi activată din cauza evenimentului e.

Un sistem de evenimente ($MEPO_{SC}$,I) este:

i) *independent* dacă şi numai dacă structura cauzală este independentă;

ii) *aproape simplu* dacă şi numai dacă este independent şi pentru fiecare regulă condiţională $r=<e,\varphi,\alpha,\delta>$, mulţimile α şi δ sunt singletoane şi $\delta \subseteq \varphi$;

iii) *simplu* dacă şi numai dacă este independent, **I** este un singleton şi pentru fiecare regulă condiţională $r=<e,\varphi,\alpha,\delta>$ mulţimile φ, α şi δ sunt singletoane, cu $\varphi=\delta$.

În aceste condiţii, o observaţie importantă se referă la faptul că sursele de complexitate (conjuncţii multiple de motive $|\varphi|>1$, disjuncţii sub forma mai multor reguli cauzale pentru un singur eveniment, structuri de cunoştinţe care nu sunt utilizate optim) nu sunt responsabile pentru dificultatea problemei de evaluare a consecinţelor unei mulţimi de evenimente. Singura sursă de complexitate pare să fie ordonarea parţială a evenimentelor. Aceste rezultate sunt oarecum surprinzătoare, întrucât se poate suspecta faptul că planificarea şi validarea unui plan sunt simple în prezenţa unor restricţii impuse asupra structurii sistemului de evenimente.

Similar cu sistemele simple cu evenimente se definesc şi task-urile de planificare simple, adică:

i) există numai o singură regulă cauzală asociată cu fiecare tip de eveniment;

ii) pentru toate regulile cauzale $|\varphi|=|\alpha|=|\delta|=1$, $\varphi=\delta$ şi

iii) $|\mathbf{I}|=1$. Pentru task-urile de planificare simple se poate decide în timp polinomial dacă există o soluţie.

Se ridică în mod natural întrebarea de ce evaluarea consecinţelor unei mulţimi de evenimente, care reprezintă problema de bază în validarea unui plan, este mult mai dificilă decât planificarea însăşi în anumite cazuri. O explicaţie se bazează pe faptul că un planificator poate să-şi creeze o structură complicată pe care nu o exploatează optim.

Complexitatea teoretică nu apare niciodată în practică. În acest caz toate soluţiile pentru task-urile simple de planificare sunt liniare, adică sunt secvenţe de evenimente pentru care problema de evaluare a consecinţelor unei mulţimi de evenimente se poate rezolva satisfăcător. Caracteristica importantă a acestor probleme este că ele nu sunt definite prin restricţii locale asupra regulilor cauzale, restricţiile pot fi exprimate în formalismul de reprezentare a structurilor cauzale utilizate şi prezintă mai mult interes practic.

Există o relaţie importantă între problema evaluării consecinţelor unei mulţimi de evenimente şi problema validării unui plan. O instanţă a problemei de validare a unui plan este dată prin task-ul de planificare TP şi un plan Δ_{SC} care trebuie să rezolve task-ul de planificare. În cazul structurilor cauzale fără restricţii, problema validării unui plan este NP-dificilă. Din punct de vedere a complexităţii, problema validării uni plan şi problema evaluării consecinţelor unei mulţimi de evenimente sunt echivalente, iar din punct de vedere conceptual evaluarea consecinţelor unei mulţimi de evenimente apare ca o subproblemă a validării. Problema deciderii dacă un plan este soluţia unui task de planificare cu structură cauzală independentă este o problemă polinomială.

Un aspect surprinzător al acestui rezultat este faptul că pare mai uşor de rezolvat global o problemă decât prin descompunerea ei în subprobleme de evaluare a consecinţelor unor evenimente cu rezolvarea acestora izolat. Se evidenţiază astfel o anumită sinergie în soluţionarea problemelor care impun o coerenţă în cazul sistemelor cu evenimente şi care ne permit rezolvarea prin verificarea anumitor condiţii sintactice simple.

Există cel puţin două aspecte în analiza şi importanţa coerenţei unui sistem cu evenimente. În primul rând noţiunile de validare şi evaluare a consecinţelor unor evenimente sunt destul de strâns legate între ele. În timp ce validarea unui plan consideră că structurile de evenimente ce conţin exact un eveniment care nu este admisibil într-o posibilă secvenţă de evenimente sunt invalide, evaluarea consecinţelor unor evenimente poate da rezultate chiar în prezenţa unor evenimente care nu sunt admisibile. Coerenţa structurilor cauzale joacă un rol deosebit de important în analiza task-rilor de raţionament temporal. Pentru cazul unor structuri cauzale mai generale, coerenţa structurilor de evenimente poate fi înlocuită cu alte forme de coerenţă.

3. Decidabilitatea unui algoritm de inferenţă fuzzy

Fie următorul algoritm de inferenţă fuzzy (**AIF**), înglobat în structura unui agent de planificare (**AgP**), conform lucrării [3]:

Algoritmul AIF

Iniţializări: k=0, $\Pi_k[i]=0$, $N_k[i]=0$, $\Theta_k[i]=0$, $K_k[i]=0$, $\Lambda_k=0$, $W_k=0$, $ra_k^i=0$, $a_k[i,j]$

Etapa 1. Filtrarea se realizează folosind structura compilată de cunoştinţe fuzzy, în raport cu apariţia unui anumit tip de eveniment ca intrare **AgP**. La sfârşit rezultă mulţimea de conflicte MC_k.

Pentru r = 1 la n_r ***execută***

Dacă (r∈ MC_k) *atunci* $ra_k^r = 1$

Dacă (există exact o regula r' astfel încât $ra_k^{r'} = 1$) *atunci* execută regula r'

Dacă (nu există o regulă r' astfel încât $ra_k^{r'} = 1$) *atunci Stop* {I/C=∅ sau apel modul control)

Etapa 2. Această etapă foloseşte diferite strategii care însă nu sunt toate active pentru un anume model de cunoştinţe pentru **PP** sau pentru o situaţie dată.

Pentru r = 1 *la* n_r *execută* {selecţie bazată pe refracţie}

Dacă ($ra_k^r = 1$) *atunci*

Dacă ($\Lambda_k[r]$ =1) *atunci* $ra_k^r = 0$

Dacă (există exact o regulă r' astfel încât $ra_k^{r'}$=1) *atunci* execută regula r'

Dacă (nu există nici o regulă r' asfel încât $ra_k^{r'}$=1) *atunci stop* {**AgP** nu poate răspunde la situaţia curentă şi se apelează modulul de control}

W = - ∞ {selecţie bazată pe vechime}

Pentru j=1 *la* 2 *execută* { Caută regulile cu valoarea vechimii cea mai mică }

Pentru r = 1 *la* n_r *execută*

Dacă ($ra_k^{r'}$ = 1) *atunci*

Dacă – $W_k[r]$ < W *atunci* $ra_k^r = 0$

altfel W = $-W_k[r]$

Dacă (există exact o regulă r' astfel încât $ra_k^{r'}$ = 1) *atunci* execută regula r'

d = 0, k dat { selecţie bazată pe gradul de diferenţă dintre reguli. Variabila d trebuie să conţină numărul de părţi de antecedent din structura regulilor care reprezintă componenta statică din acest vector, precum şi elemente cu caracter dinamic, adică cele care se schimbă în timpul funcționării **AIF**. Această componentă este alcătuită din valorile parametrilor ce gestionează imprecizia cunoştinţelor, adică Π_k [i], N_k [i], Θ_k[i], K_k[i]}

Pentru j = 1 *la* 2 *execută* {Caută regulile cu gradul de diferență cel mai ridicat}

Pentru r = 1 *la* n_r *execută*

Dacă (ra_k^r = 1) *atunci*

Dacă (d_r < d) *atunci* ra_k^r = 0 *altfel* d=d_r

Dacă (există exact o regulă r' astfel încât $ra_k^{r'}$ = 1) *atunci* execută regula r'

p=0 {selecție bazată pe prioritatea regulilor}

Pentru j = 1 *la* 2 *execută* { Caută regulile cu prioritatea cea mai ridicată. Pentru două sau mai multe reguli aflate în MC_k, se poate folosi ca metodă de rezolvare a conflictelor fie prima regulă (respectiv ultima), fie se utilizează o prioritate dinamică utilizând drept regulă prioritară regula care are N maxim, θ minim, K maxim}

Pentru r = 1 *la* n_r *execută*

Dacă (ra_k^r = 1) *atunci*

Dacă (p_k^r < p) *atunci* ra_k^r = 0 *altfel* p = p_k^r

{selecție arbitrară} Fie r' o regulă arbitrară de la 1 la n_r cu $ra_k^{r'}$=1; execută regula r'

Etapa 3. $e' = \{r', e_k^i\}$ {evidențiem astfel tipul evenimentului la nivelul **AgP**} $(\mathbf{x}_{k+1}^{b}, \mathbf{x}_{k+1}^{int}) = f^{\mathbf{AgP}}(\mathbf{x}_k^{b}, \mathbf{x}_{k+1}^{int})$ {ca rezultat al apariției evenimentului e' se actualizează starea **AgP**, adică componenta de stare factuală şi componenta $\mathbf{x}_{k+1}^{int}$}

Λ_{k+1}(r')=1 {Elimină din mulțimea de conflicte regulile, utilizând refracția}

Pentru r = 1 *la* n_r *execută*

Dacă (r ∈ MC_k) *atunci* $W_{k+1}^r = W_k^r + 1$ {Incrementează vechimea regulii din MC_k}

Pentru r = 1 *la* n_r *execută*

Dacă (a[r,r'] = 1) *atunci* Λ_{k+1}(r) =0 ; W_{k+1}^r = 0 {Permite regulilor care sunt afectate de activarea regulii r' să fie considerate în MC_k şi resetează vechimea acestor reguli}

Sfârşit

Problema decidabilității se defineşte în cadrul **AgP** astfel: dându-se un proces (problemă) P, o formulă g∈G şi un **AgP**, se poate decide efectiv dacă **AgP** face validă formula g. În vederea soluționării acestei probleme de decidabilitate se construieşte o familie de relații de echivalență notate $\approx_{n,\mathbf{AgP}}$ pe mulțimea şirurilor **AgP**-admisibile, astfel încât relația de echivalență $\approx_{n,\mathbf{AgP}}$ implică satisfacerea aceloraşi formule de rang (notat **r**(g)) cel mult n. De notat este faptul că $\alpha \approx_{0,\mathbf{AgP}} \beta$ dacă şi numai dacă α şi β au acelaşi prim simbol, $\alpha \approx_{1,\mathbf{AgP}} \beta$ dacă şi numai dacă α şi β au acelaşi prim simbol şi acelaşi al doilea simbol, iar starea curentă a **AgP** şi a procesului sunt aceleaşi când se aplică α şi β.

Dacă m<n şi $\alpha \approx_{n,\mathbf{AgP}} \beta$, atunci $\alpha \approx_{m,\mathbf{AgP}} \beta$. Fiecare clasă de echivalență $\alpha \approx_{m,\mathbf{AgP}} \beta$ va conține mai multe clase de echivalență $\alpha \approx_{n,\mathbf{AgP}} \beta$.

Decidabilitatea se bazează pe o serie de observații prezentate în cele ce urmează. Fie g o formulă astfel încât **r**(g) ≤n şi fie α şi β şiruri **AgP**-admisibile astfel încât $\alpha \approx_{n,\mathbf{AgP}} \beta$. Atunci (**AgP**,α) ⊨ g dacă şi numai dacă (**AgP**,β) ⊨ g. Un şir $\alpha \in \mathbf{X}^N$ poate să prezinte caracteristici de fracție periodică mixtă cu n_2 lungimea părții periodice şi n_1 lungimea părții neperiodice.

Şirul α este determinat de perechea (ν,τ) în care ν denotă comportarea tranzitorie, iar τ semnifică comportarea în timpul unei perioade. În aceste condiții se poate decide efectiv, dându-se o formulă g, un **AgP** şi o pereche finită de şiruri (ν,τ) care determină şirul $\alpha \in \mathbf{X}^N$, dacă (**AgP**,α) ⊨ g. Trebuie de observat că satisfacerea formulei g se poate demonstra pentru $g \in X \cup X^{\mathbf{AgP}}$ sau pentru og, □g, ◊g (o, □, ◊ operatori modali) [3].

Pentru un **AgP** cu g formulă scop, **AgP** face formula g validă dacă (**AgP**,α) ⊨ g pentru orice şir α **AgP**-admisibil. Există însă în general $2^{\aleph_0}$ şiruri **AgP**-admisibile, iar periodice un număr numărabil. Problema este de a decide dacă un agent **AgP** face formula g validă neluând în considerare toate şirurile **AgP**-admisibile, ci de a considera o mulțime finită de astfel de şiruri cu părțile periodice şi neperiodice mărginite.

Pentru cazul **AgP** care înglobează baze de cunoştințe ferme, cu n∈**N** şi orice şir α **AgP**-admisibil, există o fracție periodică mixtă β cu proprietatea că $\alpha \approx_{n,\mathbf{AgP}} \beta$. În plus, dându-se n (adică $|X^{\mathbf{AgP}}|$ şi |X|) se pot calcula efectiv n_1 şi n_2. Pentru o formulă fuzzy g dată, putem sintetiza un agent **AgP** care să permită ca formula g să fie validă în timp polinomial sau nu, aşa cum rezultă din lucrările [3, 4].

4. Concluzii

Conceperea şi testarea algoritmului **AIF** de tipul celui implementat în structura unui **AgP**, presupune existența unor metode științifice de achiziție a cunoștințelor, care este un proces eterogen, dificil și consumator de timp. Din acest motiv, conceperea unui **AgP** care să înglobeze experiența de planificare se face iterativ, într-un timp destul de mare. Plecând de la aceste observații, aspectele calitative de bază pe care le putem sintetiza, sunt după cum urmează:

1. Adaptarea și agregarea prin analogie a unor tehnici de IA (teoria posibilităților, logici simbolice, sisteme expert) cu modele specifice de timp real (planificare, sisteme cu evenimente logice discrete, elemente de analiză calitativă), în vederea sintezei unui **AgP** bazat pe cunoștințe fuzzy, cât mai aproape de modalitatea naturală de percepție și acțiune a decidentului uman.
2. Prezentarea prin analogie a sistemelor de planificare convențională în raport cu cele de IA, pornind de la proprietatea de predictibilitate, proprietate fundamentală pe care **AgP** conceput trebuie să o îndeplinească. A fost în acest sens delimitată și efectiv utilizată în [4] noțiunea de predictibilitate microscopică (adoptându-se pentru etapa de filtraj tehnica compilării) și cea de predictibilitate macroscopică, prin specificarea, proiectarea și realizarea **AgP**, ca sistem cu evenimente discrete logice [3]. Acest punct reprezintă de fapt eforturile depuse în vederea justificării aspectelor de timp care pot apare într-un sistem multiagent de timp real.
3. Utilizarea tehnicii de compilare a cunoștințelor în vederea îmbunătățirii etapei de filtraj, în sensul algoritmului *Rete*, pentru cazul cunoștințelor factorizate fuzzy.
4. Elaborarea formalismului de reprezentare a cunoștințelor, specific. Alegerea între probabilități și posibilități sau mulțimi fuzzy nu este ușoară, întrucât pentru spații finite probabilitățile ar putea avea o flexibilitate mai mare din punct de vedere a puterii de reprezentare, dar o complexitate computațională crescută. Din aceste considerente, în cadrul **AIF** am ales ca măsuri ale impreciziei măsurile de posibilitate Π și N. Pentru legarea variabilelor fuzzy este necesară rezolvarea compunerii substituțiilor fuzzy, care depinde de rezultatul evaluării compatibilității mulțimilor fuzzy implicate în antecedentul unor metareguli.
5. Alegerea pragurilor pentru Π și N trebuie să fie consistentă cu schema de inferență aleasă și are o importanță deosebită în toate etapele de procesare a impreciziei în cadrul **AgP** (filtrarea și unificarea fuzzy, rezolvarea conflictelor, similaritatea stărilor).
6. Conceperea unei strategii de planificare. Semnificația acestei strategii este deosebită, întrucât ea reprezintă într-o formă aplicativă, specificațiile de timp real ale viitorului model lingvistic de planificare. În cadrul strategiei de conducere sunt sintetizate proprietățile temporale modale care pot fi înglobate în structura **AgP**. În felul acesta, modelul de planificare nu este unic și depinde esențial de strategia aleasă (este deci subiectivă).

Bibliografie

1. Bylander T., 1994 – *The computational complexity of propositional STRIPS planning*, Artificial Intelligence, n° 69, pp. 165-204
2. Lassaigne R., Rougemont M., 1996 – *Logique et complexité*, Hermes, Paris
3. Mazilescu V., 2001 – *The Management of Fuzzy Knowledge in Planning Systems*, The Fifth International Symposium of Economic Informatics, Bucharest, 10-13 May, Academy of Economic Studies, Faculty of Economics, Cybernetics, Statistics and Informatics, p. 967-977
4. Mazilescu V., 2004 - *Sinteza unui Agent Inteligent de Timp Real folosind G2*, Revista de Informatică Economică, Nr. 1
5. Nebel B., Bäckström C.,1994 - *On the Computational Complexity of Temporal Projection, Planning and Plan Validation*, Artificial Intelligence 66, pp. 125-160