

УДК 681.5

Д.М. Обідін¹, О.В. Барабаш², Р.В. Хращевський¹¹Кіровоградська льотна академія Національного авіаційного університету, Кіровоград²Національний авіаційний університет, Київ

АЛГОРИТМИ ЛОГІЧНОГО ВИВЕДЕННЯ ЗНАТЬ В РОЗПОДІЛЕНІЙ БАЗІ ЗНАТЬ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ

Запропоновані чотири основні алгоритми логічного виведення в розподіленій базі знань САУ, які доведені до необхідного ступеня деталізації і можуть бути реалізовані існуючими інструментальними засобами логічного програмування.

Ключові слова: алгоритм, розподілена база знань, нечіткі моделі.

Вступ

Постановка завдання аналізу. Основою для аналізу і розробки моделей баз знань в розподіленій базі знань (РБЗ) інтелектуалізованої системи автоматичного управління (САУ) літального апарату є логічна структура. Необхідно вирішити питання про те, як формально визначити логічну структуру РБЗ САУ, які математичні засоби доцільно для цього використовувати. Логічна структура РБЗ САУ повинна розглядатися як структура ієрархічна, з точним визначенням рівнів і підлеглості інтелектуальних модулів бази знань.

Це положення обумовлюється тим, що структура системи управління сучасного ЛА має складну ієрархію, що визначає різні типи інформаційних зв'язків в проектованій системі [1]. При цьому, як правило, спостерігається сувора підпорядкованість систем виконавчого рівня вищим рівням управління.

Аналіз останніх досліджень і публікацій. Питанню дослідження ситуацій в системах автоматичного управління літальним апаратом приділяється багато уваги [1 – 3]. Значний внесок у дослідження зазначеного питання зробили такі вчені як В.В. Солодовніков, А.А. Красовський, В.А. Бесекерський, В.А. Боднер, А.А. Воронов, А.Е. Асланян, Ю.П. Зайченко, А.М. Воронін, О.В. Харченко, Ю.К. Зіатдінов, В.І. Гостев та багато інших.

В [4] розроблено достатньо ефективні алгоритми верифікації різноманітних баз знань (БЗ), хоча через складність моделей можливість реалізації таких алгоритмів є вкрай сумнівною. В інших роботах розглядається можливість спрощення БЗ шляхом трансляції їх у матричну форму для подальшої верифікації, при цьому методика перевірок не пропонується. Крім того усі зазначені підходи вимагають необхідності наявності людини (групи експертів) у процесі виведення знань, що досить складно забезпечити для динамічних систем (наприклад, на літаку у польоті). Тому загальна проблема щодо створення компактних та ефективних методів та алгоритмів виведення знань з БЗ інтелектуалізованої САУ літального апарату залишається невирішеною.

Метою даної статті є розробка алгоритмів логічного виведення знань в архітектурі розподіленої нечіткої семантичної мережі інтелектуалізованої системи автоматичного управління.

Основна частина

Спираючись на основні положення системного аналізу [5], введемо поняття дерева інтелектуальних модулів (ДІМ), під яким розумітимемо зв'язний неорієнтований граф без циклів, вершинами якого є інтелектуальні модулі РБЗ САУ (що позначаються далі як $IM_1, \dots, IM_{ij}, \dots, IM_{ijk}, \dots$), а дуги сполучають інтелектуальний модуль (ІМ) з іншими ІМ вищого або нижчого рівнів, так що ІМ нижчого рівня структурно входить в один з ІМ вищого рівня.

У дереві ІМ можуть бути виділені три типи вершин: Початкова вершина IM_0 , яка не має вхідних дуг, відповідає центральному ядру РБЗ САУ. Проміжні вершини – це вершини, що мають строго одну вхідну дугу і n вихідних дуг. Термінальні вершини – це вершини, які мають строго одну вхідну дугу і не мають вихідних дуг, що відповідає ІМ найнижчого рівня ієрархії (рівень датчиків та інтерфейсів).

Множину вершин ДІМ – $\{V\}$ можна розбити на три підмножини: Множина початкової вершини V_0 , що складається з одного елемента IM_0 ; множина проміжних вершин $\{V_p\}$, множина термінальних вершин $\{V_t\}$, так що $\{V\} = \{V_0\} \cup \{V_p\} \cup \{V_t\}$.

Позначимо зв'язність вершин по входу в ДІМ як ρ_+ , а зв'язність вершин по виходу як ρ_- . Для всіх вершин, окрім IM_0 , зв'язність вершин по входу дорівнює одиниці:

$$\forall V_i ((V_i \in \{V\}) \wedge (i \neq 0)) \rightarrow (\rho_+(V_i) = 1).$$

Для множин проміжних і термінальних вершин справедливі наступні формули в численні предикатів:

$$\forall V_i ((\rho_+(V_i) = 1) \wedge (\rho_-(V_i) \neq 0) \rightarrow (V_i \in \{V_p\}));$$

$$\forall V_i ((\rho_+(V_i) = 1) \wedge (\rho_-(V_i) = 0) \rightarrow (V_i \in \{V_t\})).$$

Кількість вершин в ДІМ визначається рівністю

$$N_V = N_p + N_t + 1,$$

де N_p – кількість проміжних вершин; N_t – кількість термінальних вершин.

Ці співвідношення можуть бути використані на етапі декомпозиції РБЗ САУ для перевірки коректності побудови ДІМ. У лінійному вигляді ДІМ можна відобразити списком за рівнями ієрархії:

$$\begin{aligned} &IM_0 (IM_1, IM_2, \dots, IM_n); \\ &IM_1 (IM_{11}, IM_{12}, \dots, IM_{1n_1}); \\ &IM_n (IM_{n1}, IM_{n2}, \dots, IM_{nn_n}); \\ &IM_{11} (IM_{111}, IM_{112}, \dots, IM_{11n_1}). \end{aligned} \quad (1)$$

Формально ДІМ визначається структурою вигляду $T_{IM} = (I, \Omega)$, де $I = \{IM\}$ – множина ІМ; $\Omega: I \times I \rightarrow P$, де P – множина ребер, що відповідає матриці інцидентності.

З кожним інтелектуальним компонентом РБЗ САУ зв'язуватимемо його формальну об'єктну систему, таким чином, ДІМ буде пов'язаним елементом між логічною і об'єктною структурами РБЗ САУ.

До складу даного інтелектуального модуля IM_i входять підлеглі йому інтелектуальні модулі IM_{ij} , що мають продукційний інтелект. Тоді з кожною вершиною ДІМ буде зв'язаний список вигляду $IM_i (IM_{i1}, IM_{i2}, \dots, IM_{in_i})$.

З урахуванням структури ІМ структурно-логічна модель РБЗ САУ визначається таким чином:

$$T_{IM} = (I, P, \Omega, A_p), \quad (2)$$

де $I = \{IM\}$ – множина ІМ; P – множина зв'язків між ІМ; Ω – відображення, яке задає структуру перевірочних зв'язків; A_p – множина процедур верифікації РБЗ САУ.

Розробка представленої вище логічної структури РБЗ САУ у вигляді дерева інтелектуальних модулів дозволяє поставити питання про структуру і зміст методу пошуку рішення на такій базі знань.

Розробка загальної парадигми методу пошуку рішення на дереві інтелектуальних модулів

Вибір методу пошуку рішень і реалізації механізмів виведення визначається цілим рядом чинників: специфікою області застосування, розмірами простору пошуку, рівнем визначеності і надійності знань і даних, динамікою змін, що відбуваються в наочній області, моделями представлення знань, обчислювальними ресурсами, які можуть бути застосовані в інтелектуальній системі. Рівень визначеності і надійності знань у випадку застосування процедур динамічної децентралізованої верифікації обумовлює необхідність застосування нечітких підходів щодо розробки алгоритмів логічного виведення.

Класичні підходи розробки алгоритмів виведення передбачають розробку деякої аксіоматизованої формальної системи Φ над множиною інформаційних об'єктів (термів), що викликає необхідність застосування дедуктивних методів пошуку рішень.

У дедуктивних системах проблема, що підлягає вирішенню формулюється у вигляді тверджень деякої формальної мови, а мета – у вигляді твердження, справедливості якого необхідно довести або спростувати на підставі аксіом і правил виведення формальної системи. Проблема пошуку рішення зводиться до доведення теореми, що встановлює логічний вихід цільової формули з формул-посилок. Дедуктивні методи успішно використовуються при побудові глобальних баз знань [7]. Такі глобальні БЗ містять десятки і сотні тисяч логічних формул і успішно функціонують. Для випадку дедуктивної теорії з розділеними (чіткими) базами знань відомий алгоритм виведення МР (Message-Passing) [4], який полягає в наступному.

Початкові дані: множина модулів бази знань $\{BF_i\}_{i \leq n}$, які складають опис предметної області (теорію); граф перетинів $G = (V, E, W)$, що описує взаємозв'язок між базами; Q – цільова формула з сигнатурою $J(Q) \subseteq J\{BF_k\}$.

Крок 1. Нехай $\text{dist}(i, j) (i, j \in V)$ – довжина найкоротшого шляху між i та j у графі G . Нехай $i \angle j$ тільки якщо $\text{dist}(i, k) < \text{dist}(i, j)$. Тут \angle є відношення строгого часткового порядку на множині V .

Крок 2. Паралельно виконується послідовний пошук для кожного модуля бази $BF_i, i \leq n$.

Крок 3. Для кожної пари $(i, j) \in E$ такої, що $i \angle j$, якщо вдається довести $BF_j \vdash \varphi$ і $J(\varphi) \subseteq J(W(i, j))$, то формула φ додається до BF_i . Змістовно це означає рух від вершин в порядку зменшення відстані $\text{dist}(i, j)$ до вершини BF_k додаванням в простір пошуку формул φ .

Крок 4. Якщо вдається довести формулу Q на базі BF_k то видається повідомлення “Мета досягнута”, інакше пошук завершується безрезультатно.

Алгоритм МР не дає можливості здійснити виведення цільової формули Q , такої, що $J(Q) \not\subseteq J\{BF_k\}$, тобто сигнатура цільової формули не належить цілком сигнатурі БЗ одного блоку. Для РБЗ САУ пошук таких формул цілком можливий, оскільки бази знань модулів РБЗ зберігають знання певного рівня ієрархії і нові знання (в даному випадку складна цільова формула Q) можуть виникати в процесі сумісного виведення по декількох БЗ різних модулів. Тому, для подолання цієї невідповідності, пропонується наступне удосконалення (алгоритм МРQ), що розвиває і удосконалює алгоритм МР, стосовно розподілених баз знань РБЗ САУ з урахуванням елементів нечіткості в окремих модулях РБЗ.

Позначимо як T_{IM_i} – піддерево дерева інтелектуальних модулів, що починається з IM_i , I_i – індексну послідовність IM_i , що однозначно визначає положення IM_i в ДІМ, I_d – останню цифру індексної послідовності, $|$ – операцію відсікання в індексній послідовності, I_t – поточну індексну послідовність. Сигнатура піддерева T_{IM_i} розуміється як об'єднання

$$J(T_{IM_i}) = \bigcup (J(BF_{ij}) | IM_{ij} \in T_{IM_i}).$$

Початкові дані: дерево інтелектуальних модулів T_{IM_0} відповідної РБЗ САУ, що включає множину баз знань $R = \{BF_i\}_{i \leq n}$ де n – число ІМ в ДІМ; Q – цільова формула з сигнатурою $J(Q)$.

Крок 1. Визначається початкова точка процесу логічного виведення. Піддерево T_{IM_i} вибираємо по найближчому ІМ для нечіткого терму, з якого починається процес логічного виведення. Встановлюємо $I_i, I_t := I_i$.

Крок 2. Проводиться порівняння $J(Q)$ і $J(T_{IM_i})$. Якщо $J(Q) \not\subseteq J(T_{IM_i})$, то виведення Q в піддереві T_{IM_i} неможливе. Перехід до кроку 6, інакше до кроку 3.

Крок 3. Для всіх $IM_{ij} \in T_{IM_i}$ виконується порівняння $J(Q)$ і $J(BF_{ij})$. Якщо існує j , такий що $J(Q) \subseteq J(BF_{ij})$ то застосовується алгоритм МР для модуля БЗ піддерева T_{IM_i} . Інакше – до кроку 4.

Крок 4. Для модуля бази знань піддерева T_{IM_i} будується граф перетинів G , далі він перетворюється в дерево G_T відповідно до процедури BREAK-CYCLES ($G = (V, E, W)$) [4].

Крок 5. Виконується логічне виведення в графі G_T за одним з алгоритмів для нечіткого виведення. Якщо вдається вивести Q , то видається повідомлення “Мета досягнута”, йти до кроку 7, інакше до кроку 6.

Крок 6. Перевіряємо рівність $I_t=0$, тобто чи не досягнутий нульовий рівень ієрархії в ДІМ, якщо ні, то $I_t := I_t | I_d$. Перехід до кроку 2. При $I_t=0$ виведення завершується безрезультатно.

Крок 7. Завершення алгоритму МРQ.

Для кожної пари $(i, j) \in E$, такої що $i < j$ якщо виводиться $BF_j | \varphi$ і $J(\varphi) \subseteq J(W(i, j))$, то φ додається до BF_i . Відбувається рух процесу логічного виведення від термінальних вершин T_{IM_i} в порядку зменшення відстані $dist(i, j)$ до вершини дерева T_{IM_i} з додаванням в простір пошуку формул φ .

Нечітке термінальне низхідне неаддитивне виведення

Нечітке термінальне низхідне неаддитивне виведення (НТННВ) – це логічне виведення, здійснюване від заданого IM_i , з послідовним обходом всіх

гілок піддерева IM_i без об'єднання БЗ між сусідніми гілками, з подальшими поверненнями у разі недосягнення мети, до тих пір, поки не будуть обійдені всі гілки піддерева IM_i . У разі НТННВ необхідно враховувати області дії аксіом і зміст баз фактів, тому схема виведення набуває вигляду

$$\begin{aligned} AG, ANG, BF_0 &\Rightarrow AG, ANG, AL_1, Con(BF_0, BF_1) \Rightarrow \\ AG, ANG, AL_1^R, AL_{11}, Con(BF_{11}, Con(BF_0, BF_1)) &\Rightarrow \dots \\ &\Rightarrow AG, ANG, AL_1^R, AL_{11}^R, \dots, AL_{ij}^R, AL_{ij1}^R, \\ &Con(BF_{ij}(Con(\dots)\dots)), \end{aligned} \quad (3)$$

де AL_{ij}^R – поширювані вниз локальні аксіоми, Con – операція конкатенації баз фактів. З урахуванням інтерполяційної теореми визначаємо операцію конкатенації таким чином.

Нехай BF – поточний стан бази знань процесу логічного виведення, що включає результати попередніх конкатенацій. Якщо процес логічного виведення досяг інтелектуального модуля i з базою даних BF_i то здійснюємо виведення в базі $BF \cup BF_i$ всіх логічних формул, таких що

$$J(\varphi_i) \subseteq J(W(i, j)),$$

де j – наступний в ланцюжку виведення (тобто за низхідною ієрархією для НТННВ) інтелектуальний компонент. Новий зміст BF вважаємо рівним $BF := BF \cup \{\varphi\}$, якщо $\{\varphi\} \neq \emptyset$, то поточний стан BF не змінюється:

$$\begin{aligned} Con(BF(t), BF_i) &= \\ = \left\{ \begin{aligned} BF(t+1) &= BF \cup \{\varphi\}, \{\varphi | J(\varphi_i) \subseteq J(W(i, j))\} \\ BF(t+1) &= BF(t), \text{ якщо } \{\varphi\} = \emptyset \end{aligned} \right\}. \end{aligned} \quad (4)$$

Оскільки при здійсненні логічного виведення по ДІМ необхідне виконання повернень, то вводимо операцію відсікання баз фактів:

$$Del(BF(t), BF_i) = BF(t-1), \quad (5)$$

де t – номер поточного кроку послідовності виведення.

НТНН-виведення здійснюється за таким алгоритмом. Початкові дані: піддерево ДІМ IM_i , мета виведення в попередній нормальній формі (ПНФ) (рис. 1).

Крок 1. Установка початкових параметрів.

Тут N_T – номер поточної вершини піддерева IM_i , j – приріст номера поточної вершини, BF – сукупність баз фактів, $ТП_P$ – сукупність таблиць предикативних символів, V – індекс вершини в піддереві, α – ознака здійснення логічного виведення.

Крок 2. Аналіз мети: Якщо мета не містить предикативних символів, що не зустрічаються в поточній сукупності баз фактів, то перехід до кроку 4. Якщо мета містить невідомі предикативні символи, то перевіряється умова: чи є поточна вершина ДІМ термінальною? Якщо так, то до кроку 5. Якщо ні, то до кроку 3.

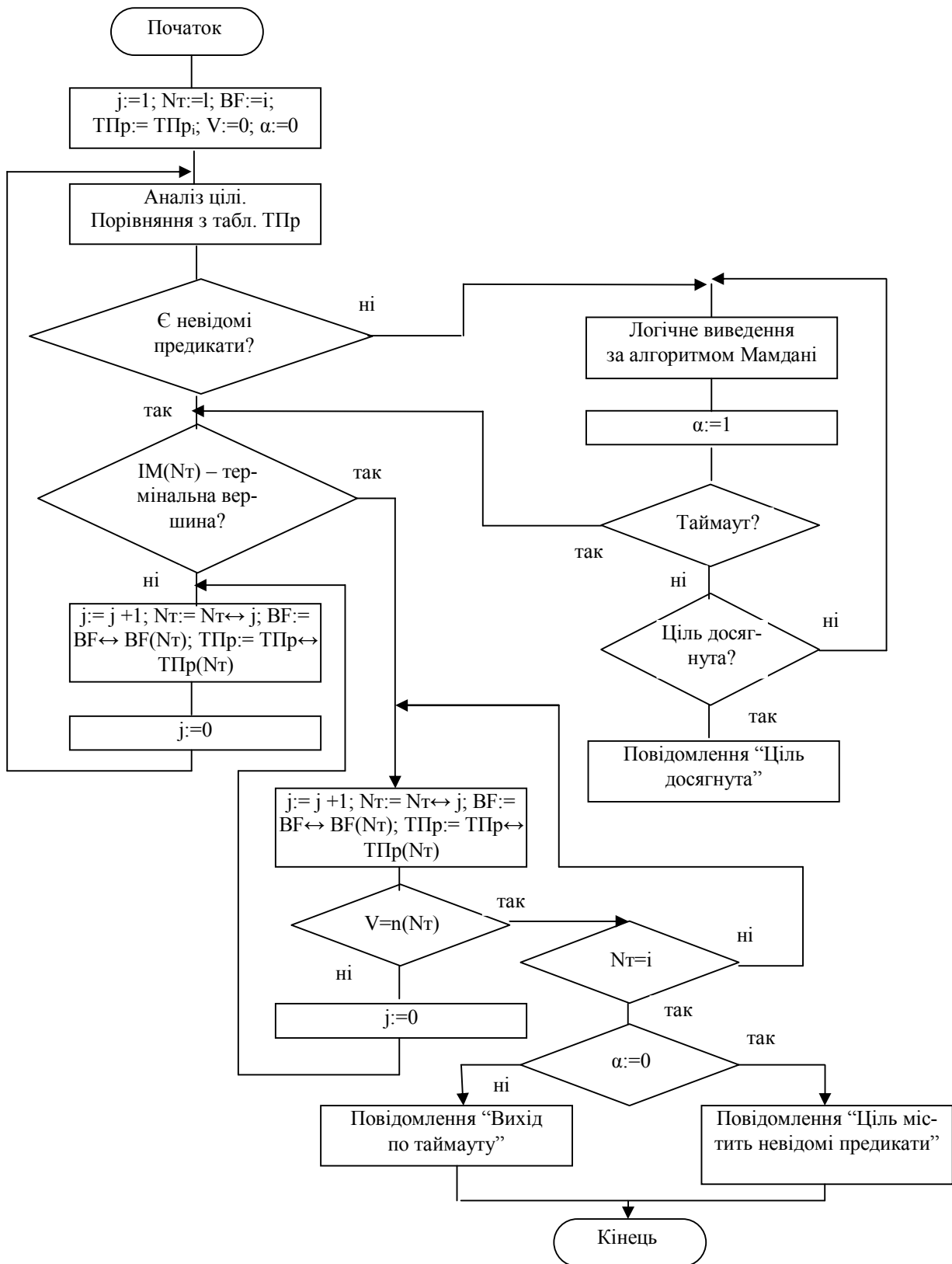


Рис. 1. Блок-схема алгоритму НТНН-виведення

Крок 3. Подовження низхідного ланцюга для подальшого аналізу мети. Поточний номер вершини нарощується: на одну цифру, значення якої визначається параметром j . Тут \leftrightarrow позначає операцію конкатенації. Визначаються база знань і база фактів чергового ІМ. Далі приріст j скидається в 0 для по-

дальшого можливого переходу до найлівоїшої нижчої вершини. Повернення до кроку 2.

Крок 4. Логічне виведення в сукупності баз фактів BF . Ознака α встановлюється одиницею. Виведення закінчується двома способами: або мета досягнута; тоді виводиться результат і перехід на Кінець,

або мета не досягається за встановлений час, тоді відбувається вихід по таймауту.

Крок 5. Повернення від термінальної вершини. В цьому випадку пройдено ланцюг вершин від i до деякої термінальної вершини з поточним номером N_T і в цього ланцюга залишилися невідомі предикативні символи. Тому потрібно перейти до аналізу найближчої правої вершини. Для цього відсікаються $TP_r(N_T)$ і $BF(N_T)$ і остання цифра номера поточної вершини N_T , що дозволяє піднятися до найближчої вищої вершини. Тут Pc – остання цифра в параметрі V , вона дозволяє запам'ятати індекс останньої пройденної вершини в піддереві. Якщо значення параметра V дорівнює максимальному індексу (тобто зв'язності по виходу) поточної вершини, то це означає, що піддереву від вершини пройдено і необхідно робити ще один крок вгору, до наступної вершини. При цьому виконується перевірка на досягнення початкової вершини піддерева IM_i . Якщо вона не досягнута, то повторюється крок 5, якщо досягнута, то перехід до кроку 7. Якщо значення параметра V не дорівнює максимальному індексу поточної вершини, це означає, що потрібно переміститися до наступної правої вершини у даному піддереві. Перехід до кроку 6.

Крок 6. Установка нового значення приросту номера поточної вершини. Перехід до кроку 3.

Крок 7. Аналіз ситуації при поверненні до IM_i . Якщо ознака логічного виведення була встановлена, то це говорить про те, що формула мети коректна, але ні на одній гілці піддерева IM_i цю мету вивести не вдалося при встановленому часі таймауту. Якщо ж ознака логічного виведення не була встановлена, то це означає, що у формулі мети присутні предикати, які або взагалі відсутні в таблиці предикатів піддерева, або знаходяться на різних гілках цього піддерева, і отже, НТНН-виведення не може дати результату.

Кінець алгоритму. НТНН-виведення буде глобальним виводом по всьому ДІМ у випадку, якщо як початкова вершина вибирається IM_0 .

Нечітке адитивне ієрархічне виведення

Нечітке адитивне ієрархічне виведення (НАІ-виведення) – це логічне виведення, здійснюване від заданого IM_i з послідовним обходом всіх гілок піддерева IM_i по рівнях низхідної ієрархії з об'єднанням БЗ і таблиць предикативних символів. Алгоритм АІ-виведення складається з двох основних частин. Спочатку виконується аналіз піддерева IM_i з метою розподілу його вершин за рівнями ієрархії, щоб отримати масив такого вигляду (табл. 1).

Побудувавши такий масив, ми упорядкуємо вершини піддерева IM_i по їх відстані від початкової вершини і визначимо максимальну глибину дерева G . В другій частині алгоритму проводиться послідовне нарощування баз фактів і таблиць предикативних символів по рівнях ієрархії.

Таблиця 1

Масив вершин за рівнями ієрархії

Рівень	Номер вершини
0	I
1	i_1, i_2, \dots, i_n
2	iI_1, iI_2, \dots, iI_n
...	...
G	

Крок 1. Установка початкових параметрів.

Параметри, що мають однакові позначення з алгоритмом НТНН-виведення, мають той же зміст, тому їх призначення не повторюється. Вводиться двовимірний масив $D(l,m)$ змінної довжини, в якому зберігатиметься структура піддерева IM_i по рівнях ієрархії (тобто по величині відстані в дугах від початкової вершини) IM_i . Ця відстань встановлюватиметься в параметрі l , в масиві $D(l,m)$ – це номер рядка.

Алгоритм НАІ-виведення. Початкові дані: піддереву ДІМ IM_i , мета виведення в ПНФ.

Параметр m указуватиме положення елемента в рядку, елемент $D(l,m)$ зберігатиме номер m -ї вершини в l -му рівні ієрархії. Масив $\varphi(l)$ зберігатиме поточне значення індекса t для даного рівня l , після проходження всього рівня l елемент $\varphi(l)$ міститиме кількість вершин даного рівня ієрархії.

Параметр G служить для збереження максимального рівня, досягнутого для даного дерева IM_i для того, щоб знати, коли закінчувати обхід рівнів.

Крок 2. Перевірка початкової вершини. В граничному випадку можливо, що IM_i є термінальною вершиною; тому немає сенсу в обході дерева і слід відразу перейти до кроку 7.

Крок 3. Подовження низхідного ланцюга. Нарощується поточний номер вершини аналогічно кроку 3 алгоритму НТНН-виведення. Разом з тим нарощується значення відстані l і до масиву $D(l,m)$ записується номер чергової досягнутої вершини відповідно до її рівня ієрархії. Проводиться стеження за відстанню l , якщо досягнуто максимуму, то це значення зберігається в змінній G .

Крок 4. Перевірка поточної вершини. Якщо поточна вершина не є термінальною, то слід спуститися на наступний нижчий рівень ієрархії, для чого переміщення скидається в 0 і здійснюється перехід до кроку 3. Якщо поточна вершина є термінальною, то перехід до кроку 5.

Крок 5. Повернення від термінальної вершини. Крок в основному аналогічний кроку 5 алгоритму НТНН-виведення, але при цьому декрементується значення відстані l і нарощується індекс рядка $\varphi(l)$. Якщо досягнута початкова вершина дерева IM_i , то перехід до кроку 6, якщо ж вона не досягнута, то повторюється крок 5. Якщо значення параметра V не дорівнює максимальному індексу поточної вершини, то повернення до кроку 3.

Крок 6. Початок логічного адитивного виведення. Установка початкових значень масиву вершин $D(l,m)$.

Крок 7. Аналіз мети. Якщо мета не містить предикативних символів, які не зустрічаються в поточній сукупності баз фактів, то перехід до кроку 8. Якщо мета містить такі символи, то перевіряється, чи не досягнутий кінець рядка в масиві $D(l,m)$. Якщо він досягнутий, то перехід до кроку 9.

Якщо не досягнутий, то переміщаємося до чергової вершини в поточному рядку масиву $D(l,m)$ і нарощуємо сукупності баз значень і таблиць предикативних символів і повертаємося до кроку 7.

Крок 8. Логічне виведення в сукупності баз фактів BF. Виведення здійснюється до тих пір, поки не буде вичерпано час таймауту або не буде досягнута мета виведення. Якщо спрацює ознака таймауту, то виконується повернення в крок 7 для продовження руху по даному рівню ієрархії.

Крок 9. Перехід до наступного рівня ієрархії. Інкрементується параметр l і виконується порівняння з максимальним рівнем G . Якщо G не досягнутий, то повернення до кроку 7, інакше – до наступного кроку.

Крок 10. Завершення виведення. Аналізується ознака здійснення логічного виведення і формуються відповідні результати.

Кінець алгоритму.

НАІ-виведення буде глобальним виводом у разі обрання у якості початкової вершини IM_0 .

Нечітке адитивне узагальнене виведення

Нечітке адитивне узагальнене виведення (НАТ-виведення) – це логічне виведення, здійснюваний від заданого з об'єднанням всіх БЗ і всіх таблиць предикативних символів у даному піддереві.

Початкові дані: піддерево ДІМ IM_i , мета виведення в ПНФ. Алгоритм АТ-виведення. Використовувані позначення відповідають алгоритмам НТНН-виведення і АІ-виведення, кроки НАТ-виведення в основному аналогічні попереднім блок-схемам, тому детальний словесний опис не потрібний. У разі $i = 0$ АТ-виведення здійснюватиметься по всій РБЗ САУ.

Нечітке локальне виведення

Нечітке локальне виведення (НЛ-виведення) – це логічне виведення, здійснюваний в межах даного IM_i . Початкові дані: інтелектуальний компонент, мета виведення в ПНФ.

Висновки

Таким чином, запропоновано чотири основні алгоритми логічного виведення знань в РБЗ САУ, які

доведені до необхідного ступеня деталізації і можуть бути реалізовані існуючими інструментальними засобами логічного програмування.

Запропоновані алгоритми виведення на дереві інтелектуальних модулів являють собою удосконалення відомого алгоритму МР, який відрізняється принципом поділу формул розподіленої бази знань, що забезпечує можливість здійснення логічного виведення складних цільових формул, сигнатура яких не належить повністю сигнатурі бази знань одного інтелектуального модуля.

Такий підхід дозволяє реалізувати схему дедуктивного виведення, як одну з найбільш ефективних для систем управління реального часу та розробити комплекс часткових алгоритмів: нечіткого термінального низхідного неадитивного виведення; нечіткого адитивного ієрархічного виведення; нечіткого адитивного узагальненого виведення; нечіткого локального виведення. Дані алгоритми враховують ієрархічну організацію бази знань САУ літального апарата та її нечіткість, визначену на основі процедур верифікації, що дозволяє здійснювати виведення рішень на основі інформації різнорідних джерел в умовах динамічної зміни цілей управління.

Список літератури

1. Балонин Н.А. *Новый курс управления движением* / Н.А. Балонин. – СПб.: Изд-во С.-Петерб. ун-та, 2000. – 160 с.
2. Асланян А.Э. *Системы автоматического управления полетом летательных аппаратов* / А.Э. Асланян. – К.: КВВАИУ, 1984. – 436 с.
3. Гостев В.И. *Нечеткие регуляторы в системах автоматического управления* / В.И. Гостев. – К.: Радіоаматор, 2008. – 372 с.
4. Вагин В.Н. *Достоверный и правдоподобный вывод в интеллектуальных системах* / В.Н. Вагин, Е.Ю. Головина, А.А. Загорянская, М.В. Фомина. – М.: Физматлит, 2004. – 704 с.
5. *Сложные технические и эргатические системы: методы исследования: моногр.* / А.Н. Воронин, Ю.К. Зиятдинов, А.В. Харченко, В.В. Остаевский. – Х.: Факт, 1997. – 204 с.
6. *Модели и методы принятия решений в нечетких условиях: моногр.* / М.З. Згуровский, Ю.П. Зайченко. – К.: Наук. думка, 2011. – 280 с.
7. Гаврилова Т.А. *Базы знаний интеллектуальных систем* / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000. – 384 с.

Надійшла до редколегії 22.08.2012

Рецензент: д-р техн. наук, проф. Ю.В. Кравченко, Національний університет оборони України, Київ.

АЛГОРИТМЫ ЛОГИЧЕСКОГО ВЫВЕДЕНИЯ ЗНАНИЙ В РАСПРЕДЕЛЕННОЙ БАЗЕ ЗНАНИЙ ИНТЕЛЛЕКТУАЛИЗИРОВАННОЙ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Д.Н. Обидин, О.В. Барабаш, Р.В. Хращевский

Предложены четыре основные алгоритмы логического вывода в распределенной базе знаний САУ, которые доведены до необходимой степени детализации и могут быть реализованы существующими инструментальными средствами логического программирования.

Ключевые слова: алгоритм, распределенная база знаний, нечеткие модели.

**ALGORITHMS FOR INFERENCE OF KNOWLEDGE IN A DISTRIBUTED KNOWLEDGE BASE INTELLECTUALIZED
AUTOMATIC CONTROL SYSTEM**

D.M. Obidin, O.V. Barabash, R.V. Khrateshevskiy

Are four basic inference algorithms in a distributed knowledge base ACS, which brought to the level of detail required and can be implemented by existing tools of logic programming.

Keywords: *algorithm, a distributed knowledge base, fuzzy models.*