А. А. Яровий, І. Р. Арсенюк, В. І. Месюра

ЕКСПЕРТНІ СИСТЕМИ

Частина 2

Міністерство освіти і науки України
Вінницький національний технічний університет

А. А. Яровий, І. Р. Арсенюк, В. І. Месюра

ЕКСПЕРТНІ СИСТЕМИ
Частина 2

Навчальний посібник

Вінниця
ВНТУ
2017

УДК 004.891:004.832
Я76

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 7 від 22.12.2016 р.)

Рецензенти:
Л. І. Тимченко, доктор технічних наук, професор
О. В. Бісікало, доктор технічних наук, професор
А. М. Пєтух, доктор технічних наук, професор

Яровий, А. А.

Я76 Експертні системи. Частина 2 : навчальний посібник / Яровий А. А., Арсенюк І. Р., Месюра В. І. – Вінниця : ВНТУ, 2017. – 105 с.

В другій частині посібника розглядаються основні методи та моделі пошуку рішень в експертних системах, що базуються на дедуктивному та індуктивному логічному виведенні; наведено особливості логічного виведення в умовах ненадійних або неповних знань. Розглянуто питання щодо застосування при побудові моделей логічного виведення теорії ймовірностей, теорії Демпстера-Шефера, методики Перла та баєсових мереж довіри.

Посібник розроблений відповідно до плану кафедри комп'ютерних наук і призначений для студентів, магістрантів, аспірантів, а також для всіх бажаючих ознайомитися з основами теорії експертних систем.

УДК 004.891:004.832

ЗМІСТ

1 МЕТОДИ ПОШУКУ РІШЕНЬ В ЕКСПЕРТНИХ СИСТЕМАХ. ПОНЯТТЯ ЛОГІЧНОГО ВИВЕДЕННЯ ТА ЙОГО ОСНОВНІ МОДЕЛІ

Під логічним виведенням розуміють процес одержання висновків з передумов. Існують різні моделі логічного виведення. Спочатку розглянемо поняття дедуктивного виведення в логіці числення предикатів, що базується на застосуванні правил виведення до множини вихідних істинних тверджень. Теорія числення предикатів дозволяє одержувати правильні висновки, опираючись на загальнозначущі твердження й обґрунтовані правила логічного виведення.

1.1 Модель дедуктивного виведення на основі логіки числення предикатів

1.1.1 Постановка задачі дедуктивного виведення

Розглянемо модель дедуктивного виведення на основі подання задач у вигляді теорем. Нехай задача описується мовою числення предикатів. Якщо позначити через Φ_0 множину вихідних тверджень та через F_g цільове твердження, то задача, подана у вигляді доведення теореми, формально може бути записана у вигляді [1]:

$$\Phi_0 \Longrightarrow F_g. \tag{1.1}$$

Тобто, необхідно довести, що вираз F_g логічно слідує з множини виразів Φ_0.

Вираз F_g логічно слідує з Φ_0, якщо кожна інтерпретація, що задовольняє Φ_0, задовольняє і F_g. Коли множина Φ_0 подана виразами $F_1, F_2, \ldots, F_n$, то задачею дедуктивного логічного виведення у численні предикатів є доведення загальної значимості виразу [1]:

$$F_1 \wedge F_2 \wedge \ldots \wedge F_n \rightarrow F_g\,. \tag{1.2}$$

В процесі логічного виведення часто використовують метод доведення від зворотного, тобто встановлюють не загально значимість наведеного вище виразу, а нездійсненність виразу [1]:

$$\overline{F_1 \wedge F_2 \wedge \ldots \wedge F_n \rightarrow F_g}$$

або

$$F_1 \wedge F_2 \wedge \ldots \wedge F_n \rightarrow \overline{F_g}. \tag{1.3}$$

Тобто, доводять, що об'єднання $\Phi_0 \cup \bar{F}_g$ нездійсненне. Таке доведення може бути істотно простіше прямого виведення. Процес встановлення нездійсненності деякої множини виразів називають *спростуванням* [1].

Розроблено ефективні методи автоматизації дедуктивного виведення. Історія їхнього створення бере свій початок від основних робіт Ж. Ербрана, котрий запропонував механічну процедуру дедуктивного виведення (1930). В 1965 році Дж. Робінсон розробив принцип резолюції, що дозволяє автоматизувати процес спростування і є теоретичною базою для побудови багатьох систем автоматичного доведення теорем. Саме ці два підходи розглянемо далі при описі дедуктивного логічного виведення.

Стандартизація предикатних виразів

Для здійснення дедуктивного виведення методом Ербрана або на основі принципу резолюції, всі вирази множини $\Phi_0 \cup \bar{F}_g$ повинні бути подані у вигляді диз'юнкцій літералів.

Літералом називають елементарний вираз або його заперечення. Диз'юнкція літералів називається реченням або *клозом* (від англ. clause). Наприклад, вираз $P(x) \vee Q(x,y) \vee \bar{R}(x,y)$ є реченням. Таким чином, вирази множини $\Phi_0 \cup \bar{F}_g$ повинні бути записані у вигляді речень, тому що речення містить тільки знаки операцій диз'юнкції і заперечення, причому заперечення повинне поширюватися не більше ніж на одну предикатну літеру. Тому, для приведення виразу до вигляду речення необхідно виключити з нього всі інші операції і скоротити область дії операції заперечення [1].

Зведення виразів множини $\Phi_0 \cup \bar{F}_g$ до вигляду речень виконують у процесі зазначених нижче тотожних перетворень.

Виключення знаків імплікації і еквівалентності. Для цього використовують рівності:

$$(P \to Q) = \bar{P} \vee Q, \tag{1.4}$$

$$(P \leftrightarrow Q) = (\bar{P} \vee Q) \wedge (P \vee \bar{Q}). \tag{1.5}$$

Зменшення області дії операції заперечення. При цьому використовують закони де Моргана:

$$\overline{(P \to Q)} = (\bar{P} \vee \bar{Q}),\ (P \to Q) = \bar{P} \vee \bar{Q}, \tag{1.6}$$

$$\overline{\forall x P(x)} = \exists x \bar{P}(x),\ \ \overline{\exists x P(x)} = \forall x \bar{P}(x). \tag{1.7}$$

Наприклад,

$$\overline{\forall x (P(x) \wedge (Q(x) \vee R(x)))} = \exists x (\bar{P}(x) \wedge (\bar{Q}(x) \vee \bar{R}(x))). \tag{1.8}$$

Стандартизація змінних. У цьому випадку виконують перейменування змінних, пов'язаних кванторами спільності або існування. Змінні перейменовуються так, щоб кожен квантор зв'язував окрему змінну, яка не зустрічається в інших кванторах. Наприклад:

$$\forall x P(x) \vee \exists x Q(x) = \forall x P(x) \vee \exists z Q(z). \tag{1.9}$$

Виключення кванторів існування. У найпростішому випадку це відповідає правилу екзистенціальної конкретизації, що дозволяє перейти від $\exists x P(x)$ до $P(a)$, де a – константа.

Якщо квантор існування знаходиться в області дії квантора спільності, задача стає складнішою. Розглянемо, наприклад, вираз $\forall x \exists y (Q(y) \wedge R(x,y))$, котрий може означати, що для кожного x існує такий конкретний y, при якому правильно $Q(y)$ та $R(x, y)$. У цій інтерпретації значення змінної y залежить від значення змінної x. Така залежність подається функцією $y=f(x)$, яку називають функцією Сколема. Функція Сколема дозволяє виключити квантор існування і переписати розглянутий вираз у вигляді:

$$\forall x \left(Q\left(f(x)\right) \wedge R\left(x, f(x)\right)\right). \tag{1.10}$$

Якщо квантор існування знаходиться в області дії двох і більше кванторів спільності, то функція Сколема буде залежати від двох і більше аргументів. Для позначення функції Сколема не повинні застосовуватися функціональні літери, котрі вже є присутні у виразах.

Переміщення кванторів спільності. Усі квантори спільності записуються на початку виразу. Наприклад, вираз $\forall x (P(x) \vee \forall y \bar{Q}(y))$ перетвориться до вигляду $\forall x \forall y (P(x) \vee \bar{Q}(y))$.

Приведення до кон'юнктивного нормального вигляду. Для цього використовується дистрибутивний закон:

$$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C) \tag{1.11}$$

Наприклад, вираз $(A \wedge B) \vee (B \wedge C)$ приводиться до кон'юнктивного нормального вигляду таким чином:

$$\begin{aligned}(A \wedge B) \vee (B \wedge C) &= [A \vee (B \wedge C)] \wedge [B \vee (B \vee C)] = \\ &= (A \vee B) \wedge (A \vee C) \wedge B \wedge \ (B \vee C).\end{aligned} \tag{1.12}$$

Таким чином, будь-який вираз з множини $\Phi_0 \cup \bar{F}_g$ записується у вигляді $\forall x_1 \ldots \forall x_n (K_1 \wedge K_2 \wedge \ldots \wedge K_n)$. Тут кожен член K_i є реченням (диз'юнкт), тобто має вигляд $(L_1 \vee L_2 \vee \ldots \vee \ldots L_k)$, де (L_i) – літерал.

Виключення кванторів спільності. Оскільки усі змінні у виразах пов'язані кванторами спільності, а порядок встановлення кванторів спіль-

ності не має значення, то їх можна не вказувати в явному вигляді, тобто виключити.

Виключення кон'юнкцій. Будь-яка інтерпретація задовольняє вираз $(K_1 \wedge K_2 \wedge \ldots \wedge K_n)$ у тому випадку, коли вона задовольняє вирази $(K_1, K_2, \ldots, K_n)$. Тому кожний вираз, приведений до кон'юнктивного нормального вигляду, може бути замінений множиною виразів

$$\{K_1, K_2, \ldots, K_n\}. \tag{1.13}$$

Розглянемо приклад перетворення виразу до множини речень. Нехай цей вираз

$$\forall x\{P(x) \to \overline{\{\forall y[Q(x,y) \to R(y)]\}}. \tag{1.14}$$

Виключимо знаки імплікації:

$$\forall x\{\bar{P}(x) \vee \overline{\{\forall y[\bar{Q}(x,y) \vee R(y)]}. \tag{1.15}$$

Зменшимо область дії операції заперечення:

$$\forall x\{\bar{P}(x) \vee \{\exists y[Q(x,y) \vee \bar{R}(y)]\}. \tag{1.16}$$

Виключимо квантор існування:

$$\forall x\{\bar{P}(x) \vee [Q(x,f(x)) \vee \bar{R}(f(x))]. \tag{1.17}$$

Приведемо вираз до кон'юнктивного нормального вигляду:

$$\forall x\{(\bar{P}(x) \vee Q(x,f(x)) \wedge (\bar{P}(x) \vee \bar{R}(f(x)))\}. \tag{1.18}$$

Виключивши квантор спільності і кон'юнкцію, отримаємо множину речення:

$$\{\bar{P}(x) \vee Q(x,f(x)), \bar{P}(x) \vee \bar{R}(f(x))\}. \tag{1.19}$$

1.1.2 Метод Ербрана

Як зазначалося вище, доведення теореми $\Phi_0 \Longrightarrow F_g$ полягає в тому, щоб показати, що розширену множину виразів, утворену шляхом об'єднання множини аксіом Φ_0 і заперечення цільового твердження F_g, виконати неможливо. Позначимо розширену множину виразів, приведених попередньо до вигляду речень (диз'юнктів), через S. Множина диз'юнктів S нездійсненна тоді, коли не існує інтерпретації, що її задовольняє. Для того, щоб це з'ясувати, необхідно перерахувати всі інтерпретації на всіх можли-

вих областях. Очевидно, що розглянути всі інтерпретації на всіх областях неможливо. Але все ж розв'язок задачі може бути знайдено. Він ґрунтується на понятті універсуму Ербрана. *Універсум Ербрана* являє собою таку область інтерпретації H, що якщо не існує інтерпретації множини речень S у цій області, то її не існує взагалі [1]. Універсум Ербрана будується з константних термів і визначається в такий спосіб:

а) множина усіх предметних констант, які зустрічаються в S, належить H; якщо в S немає констант, то в H включається довільна константа a;

б) множина усіх функцій, що зустрічаються в S і визначених на множині термів з H, належить H.

Продемонструємо побудову універсуму Ербрана на прикладі. Нехай

$$S = \{P(x, f(x, a)) \vee Q(x, a) \vee P(x, b)\}. \qquad (1.20)$$

У множину S входять предметні константи a і b, функціональна літера f. В цьому випадку область H являє собою нескінченну зліченну множину:

$$H = \{a, b, f(a, a), f(a, b), f(b, a), f(b, b), f(a, f(a, a)), f(a, f(a, b)), \ldots\},$$

яку формують послідовно у вигляді рівнів:

$$H_0 = \{a, b\};$$
$$H_1 = \{a, b, f(a, a), f(a, b), f(b, a), f(b, b)\};$$
$$H_2 = \{a, b, f(a, a), \ldots, f(b, b), f(a, f(a, a)), f(a, f(a, b)), \ldots\}, \ldots$$

Тут H_0, H_1, H_2 і т. д. – рівні універсуму Ербрана і $H=H_\infty$.

Диз'юнкт, який не містить змінних, називається *фундаментальним*. Його формують заміною предметних змінних константними термами. Якщо терми є елементами універсуму Ербрана, то фундаментальний диз'юнкт називають *фундаментальним прикладом диз'юнкта*. Множина усіх фундаментальних атомів вигляду $P(t_1, t_2, \ldots t_n)$, де P – предикатні вирази, що зустрічаються в S, а $t_1, t_2, \ldots, t_n$ – елементи універсуму Ербрана, називається *ербрановою базою (ербрановим базисом)* для S [1]. Наприклад, ербрановою базою для розглянутого вище прикладу є множина:

$$A = \{\bar{P}(a, f(a, a)), Q(a, a), R(a, b), \bar{P}(b, f(b, a)), \ldots\}. \qquad (1.21)$$

Задача інтерпретації множини S в області H завершується, коли з кожним атомарним виразом ербранівської бази пов'язане відповідне значення істинності.

Введені визначення дозволяють сформулювати *теорему Ербрана:* множина диз'юнктів S нездійсненна тоді і тільки тоді, коли існує кінцева нездійсненна підмножина фундаментальних прикладів диз'юнктів у S. Ця

теорема дозволяє формально побудувати процедуру спростування. Для встановлення нездійсненності множини диз'юнктів S необхідно:

1) утворити множину S_1, S_2, ..., S_i... S_n фундаментальних прикладів диз'юнктів для кожного рівня i множини H;

2) послідовно перевіряти їх на хибність.

Відповідно до теореми Ербрана, якщо S нездійсненне, то процедура знайде такий рівень i, що S_i буде помилковим.

На вказаному принципі працювали перші програми доведення теорем. Основним недоліком процедури Ербрана є експонентний ріст множини фундаментальних прикладів S_i при збільшенні рівня i. Тому на практиці вдається за допомогою процедури Ербрана доводити тільки прості теореми.

1.1.3 Принцип резолюції

Принцип резолюції є процедурою виведення, за допомогою якої породжуються нові диз'юнкти з S. Якщо в ході застосування цієї процедури виводиться пусте речення, то S нездійсненне.

Правило виведення, що лежить в основі принципу резолюції, власне кажучи, являє собою розширення правила modus ponens і правила силогізму. Ці правила виведення можна записати у такому вигляді:

$$\frac{A, \bar{A} \vee B}{B} \text{(modus ponens)}, \quad \frac{\bar{A} \vee B, \bar{B} \vee C}{\bar{A} \vee C} \text{ (силогізм)}.$$

У кожному з цих правил є диз'юнкти-посилання, що містять *додаткові (контрарні) пари літер* (A і $\bar{A}$, B і $\bar{B}$). Можна помітити, що наслідок правил формально отримується викреслюванням у диз'юнктах-посиланнях контрарних пар літер і об'єднанням частин диз'юнктів, що залишилися.

Розглянуті диз'юнкти-посилання містили одну чи дві літери. Дж. Робінсон поширив зазначений прийом на випадок довільних диз'юнктів з будь-якою кількістю літер і сформулював *правило резолюції:* якщо будь-які два диз'юнкта D_1 і D_2 містять додаткову пару літер, наприклад L і $\bar{L}$, то, викреслюючи їх, формуємо новий диз'юнкт із тих частин *диз'юнктів* D_1 і D_2, що залишилися. Знову сформований диз'юнкт називається *резольвентою* (наслідком) вихідних диз'юнктів. Резольвента, отримана з двох диз'юнктів, є логічним наслідком цих диз'юнктів. Правило резолюції дозволяє одержувати резольвенти множини диз'юнктів S. Якщо в процесі виведення резольвент отримуємо два однолітеральних диз'юнкта, які утворюють контрарну пару, то резольвентою цих двох диз'юнктів буде порожній диз'юнкт, що позначається символом *NIL* [1].

Об'єднання вихідної множини диз'юнктів з множиною всіх резольвент, що можуть бути утворені з диз'юнктів, які входять у S, будемо позначати $R(S)$. Застосовуючи принцип резолюції до $R(S)$, отримаємо $R(R(S))=R_2(S)$. У загальному випадку $R_{n+1}(S)=R(R_n(S))$. При цьому $R_0(S)=S$. Помітно, що

якщо множина S нездійсненна, то $R_i(S)$ при будь-яких $i \geq 1$ також нездійсненне, і навпаки, якщо нездійсненне $R_i(S)$, то нездійсненна і S. Ця властивість називається *повнотою принципу резолюцій.* Вона встановлюється такою теоремою, доведеною Робінсоном: якщо S – довільна кінцева множина диз'юнктів, то S нездійсненне тоді і тільки тоді, коли $R_i(S)$ містить порожній диз'юнкт. Ця теорема дозволяє побудувати процедуру спростування, яка послідовно будує для вихідної множини S множини $R_1(S)$, $R_2(S)$, ..., $R_i(S)$ і перевіряє, чи містить множина $R_i(S)$ порожній диз'юнкт [1].

Виведення за допомогою резолюції наочно можна проілюструвати за допомогою *графа спростування.* Вершинами графа є вихідні диз'юнкти і резольвенти, які отримано в процесі виведення. Диз'юнкти вихідної множини S являють собою вершини графа, які розташовані в його верхній частині. Якщо два диз'юнкта, що знаходяться в будь-яких вершинах графа, утворять резольвенту, то вершина, що відповідає резольвенті, зображується нижче цих диз'юнктів і з'єднується з ними ребрами [1].

Як приклад, на рис. 1.1 побудовано граф спростування множини вихідних диз'юнктів:

$$S = \{A, \bar{A} \vee B, \bar{B} \vee C, \bar{C} \vee D, \bar{D}\}. \qquad (1.22)$$

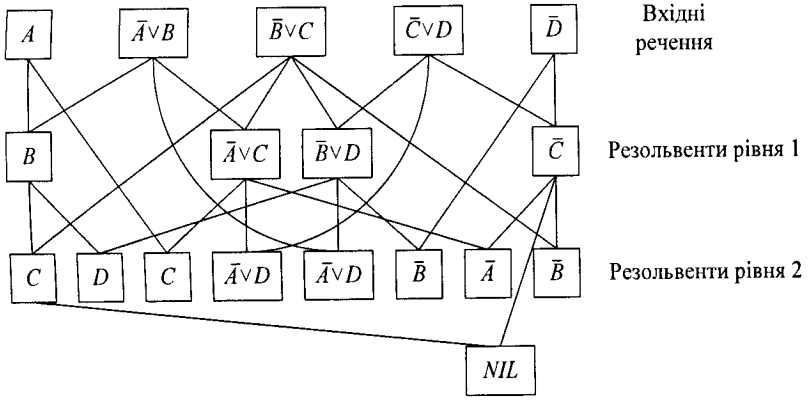

Рисунок 1.1 – Граф спростування

У логіці висловлювань знаходження контрарних пар не викликає труднощів. В логіці предикатів – складніше. Наприклад, якщо є диз'юнкти $P(x,f(x))$, $\bar{P}(a,y) \vee Q(z)$, $\bar{Q}(b)$, то резольвента може бути отримана тільки після виконання таких підстановок: a замість x, $f(x)$ замість y, b замість z. Потім за правилом резолюції отримується порожнє речення.

Підстановкою називається кінцева множина[1]:

$$a = \{t_1/x_1, t_2/x_2, \dots, t_n/x_n\}, \tag{1.23}$$

де x_i – змінна;

t_i – терм, відмінний від x_i, $1 \le i \le n$.

Застосування підстановки a до деякого виразу F означає, що усі входження змінної x_i замінюються на терм t_i. Вираз, отриманий в результаті підстановки, позначають F_a і називають *прикладом F*. Для попереднього прикладу підстановку можна записати у вигляді $\alpha = \{a/x, f(x)/y, b/z\}$. Тоді приклад виразу $P(x,f(x))$ буде мати вигляд $P(x,f(x))_a$=-$P(a, f(a))$.

Композицією двох підстановок α і β називається підстановка $\alpha\beta$, яку отримуємо за допомогою застосування підстановки β до термів підстановки α з додаванням із β усіх пар t_i/x_i, які містять змінні, що відсутні в α. Наприклад, якщо α=$\{f(x)/z\}$, β=$\{a/x, b/y, c/z\}$, то композиція підстановок буде відповідати множині $\alpha\beta$=$\{f(a)/z, a/x, b/y\}$ [1].

Підстановки виконуються з метою приведення літералів до деякого загального уніфікованого вигляду, щоб потім виключити утворені контрарні пари літералів. Підстановка a називається *уніфікатором* для множини літералів $\{L_1, L_2, \dots, L_k\}$, якщо L_{1a}=L_{2a}=...=L_{ka} *[1]*.

Наприклад, літерали $\{P(y,f(x)), P(a,z)\}$ за допомогою підстановки α_1=$\{a/y, f(x)/z\}$ перетворяться до вигляду $P(a, f(x))$. Отже, α_1 є уніфікатором вказаної множини літералів. Якщо ввести підстановку α_2=$\{b/x\}$, то можна переконатися в тому, що композиція підстановок $\alpha = \alpha_1\alpha_2$ також уніфікує розглянуту множину літералів і перетворить їх до вигляду $P(a, f(b))$. Тому підстановка α – теж уніфікатор. Однак, підстановка α отримується з підстановки α_1 за допомогою α_2.

Уніфікатор, що дозволяє одержати всі інші уніфікатори за допомогою підходящих підстановок, називається *найбільш загальним уніфікатором* (НЗУ). Для того, щоб одержати резольвенту, необхідно знайти НЗУ [1].

Розглянемо алгоритм побудови НЗУ на прикладі множини літералів:

$$\{P(g(x), x, f(x)),\ P(y, a, x)\}. \tag{1.24}$$

Якщо два літерали мають спільну змінну, то необхідно її замінити новою змінною в одному з них. Замінивши змінну x на z в другому літералі, одержимо $P(x,a,z)$. Далі розглядаємо зліва направо кожен з літералів і знаходимо позиції, в яких виявляється незбігання термів. У цьому випадку не збігаються терми $g(x)$ і y. Ці терми утворять множину неузгодженості $\{g(x), y\}$, де y – змінна. Тому виконується підстановка α_1=$\{g(x)/y\}$, що перетворить множину вихідних літералів до вигляду $\{(P(g(x)), x, f(x)), P(g(x)), a, z)\}$. Продовжуючи процес співставлення термів, виявляємо розбіжність x та α, які формують множину неузгодженості

{x,a}, що відповідає підстановці α_2=*{a/x}*. Застосувавши α_2 до множини літералів, які уніфікуються, одержимо *{P(g(a)),a,f(a),P(g(a)),a,z}*. Нарешті, знайшовши розбіжність літералів *f(a)* і *z*, виконуємо підстановку α_3=*{f(a)/z}*. Таким чином, НЗУ являє собою композицію підстановок $\alpha_3 = \alpha_1\alpha_2\alpha_3$.

Алгоритм пошуку НЗУ для двох літералів L_1 та L_2, які не містять спільних змінних, можна подати у вигляді процедури [2]:

Procedure Unify (*L1, L2*);
Begin
a:=[ ]; {порожня підстановка}
Q1:=L1;Q2:=L2;
While True Do
Begin
Побудувати множину неузгодженості *D* для ланцюжків символів, що утворять Q_1 і Q_2;
If *D*=[ ] Then Повернення (α);
t_1=*first(D)*;t_2=*last(D)*;
If var(t_1) and *non_contain*(t_1,t_2) Then α_1:={t_2/t_1}
Else If var(t_2) and *non_contain*(t_2,t_1) Then α_1={t_1/t_2}
Else Вихід ('Невдача');
$Q1:=Q1_{\alpha 1}$;
$Q2:=Q2_{\alpha 1}$;
$\alpha := \alpha\alpha_1$
End
End.

У цій процедурі вважається, що множина неузгодженості *D* складається з двох термів t_1 і t_2, тобто *D*={t_1,t_2}. Тоді виклик функції *first(D)* відповідає виділенню t_1 – першого елемента множини, а виклик функції *last(D)* – видаленню t_2. Функція *var(t)* повертає значення *true*, якщо аргумент *t* – змінна. Функція *non_contain(x,y)* – предикат, що повертає значення *true*, якщо *x* не міститься в *y*. Якщо *x* входить в *y*, то уніфікація нездійсненна. Наприклад, не можна уніфікувати *x* і *f(x)*.

Узагальнюючи сказане, відзначимо, що в логіці предикатів знаходження резольвенти двох пропозицій (диз'юнктів) зводиться до таких дій:

1) змінні в реченнях перейменовуються так, щоб вони не збігались;

2) знаходиться підстановка (НЗУ), при якій літерал одного речення стає доповнюючим для якого-небудь літерала іншого речення;

3) доповнюючі літерали викреслюються;

4) якщо отримані однакові літерали, то усі вони, за винятком одного в якому-небудь реченні, викреслюються;

5) диз'юнкція тих літералів, що залишилися, кожного з речень і є резольвентою.

На рис. 1.2 зображено дерево спростування нездійсненної множини диз'юнктів $S=\{R(x)\vee Q(x), \bar{Q}(f(z)), \bar{R}(w)\vee P(b), \bar{P}(y)\}$, що є виразами числення предикатів. На дереві показані диз'юнкти, які отримуються в результаті відповідних підстановок.

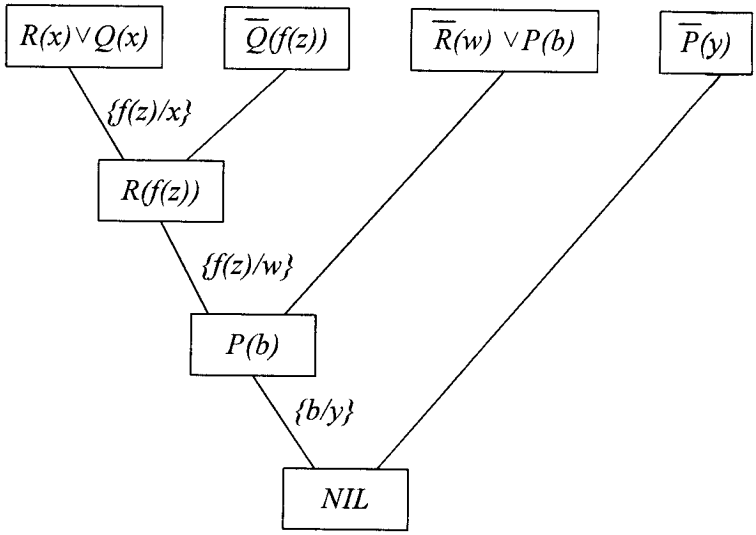

Рисунок 1.2 – Дерево спростування

У процесі уніфікації можуть отримуватись фактори диз'юнктів. Якщо декілька літералів диз'юнкта D мають НЗУ a, то D_a називають *фактором.* Наприклад, нехай $D=Q(x)\vee Q(f(y))\vee \bar{P}(x)$. Тоді НЗУ $a=\{f(y)/x\}$ і фактор $D_a=Q(f(y))\vee \bar{P}(f(y))$. Фактори є окремими випадками, що дозволяють зменшити довжину диз'юнктів. Принцип резолюції припускає використання не тільки резольвент, але і факторів.

Загалом, процес спростування не проходить настільки ефективно. У багатьох випадках множини $R_1(S)$, $R_2(S)$, ..., $R_i(S)$ дуже швидко розростаються, оскільки безпосереднє застосування принципу резолюцій відповідає "сліпому" пошуку.

Існує велика кількість модифікацій принципу резолюцій, спрямованих на знаходження ефективних стратегій пошуку найкращих диз'юнктів для бінарної резолюції. Далі коротко розглянуто лише деякі із стратегій пошуку, що знайшли найбільш широке застосування.

Стратегії пошуку

Для вирішення проблеми "комбінаторного вибуху", що виникає при безпосередньому застосуванні принципу резолюцій, необхідно здійснювати цілеспрямований підбір речень, що беруть участь у процесі уніфікації. Вибір повинен виконуватися так, щоб порожнє речення досягалось якомога швидше. Звичайно, на кожному етапі спростування вибір залежить від поточної ситуації. Однак, є і загальні стратегії пошуку, що, незалежно від контексту задачі, дозволяють скоротити кількість резолюцій. Нижче розглянемо такі чотири стратегії: стратегію унітарності, стратегію опорної множини, стратегію вихідних даних, лінійну стратегію.

Стратегія унітарності. Відповідно до цієї стратегії, перевага віддається тим бінарним резолюціям, де одне з речень містить єдиний літерал. Така стратегія гарантує зменшення довжини результуючих речень. Наприклад, наслідком одно-літерального речення P та речення $\bar{P} \vee \bar{Q} \vee R$ буде $\bar{Q} \vee R$, яке коротше вихідного речення. Стратегія унітарності упорядковує вибір пар речень, спрощуючи наступні етапи доказу нездійсненності вихідної множини диз'юнктів.

Стратегія опорної множини. Більш ефективною може виявитися спроба виключити з розгляду відразу декілька речень на основі поняття опорної множини. Нездійсненну множину речень S можна розбити на дві підмножини S_1 і S_2. Підмножина S_1 складається з аксіом, а S_2 – із заперечень тих речень, які необхідно довести. Очевидно, що підмножина S_1 не містить протиріч. Тому знаходження резольвент речень з S_1 не допускається. Опорною називають множину речень, що складається з диз'юнктів, що входять у S_2, і резольвент тих пар речень, з яких хоча б одне належить опорній множині. Стратегія опорної множини допускає виконання тих резолюцій, у яких, принаймні, одна з пропозицій входить в опорну множину. Якщо опорна множина відносно невелика, то це значно скорочує простір пошуку. Перевага стратегії опорної множини також полягає в тому, що формоване дерево спростування легко інтерпретується людиною, тому що воно керується метою.

Якщо множина вихідних речень нездійсненна, то розглянута стратегія приводить до порожнього речення, тобто вона є повною. Стратегія опорної множини є спеціальним випадком більш загальної семантичної резолюції [3].

Лінійна резолюція. Спочатку знаходиться наслідок двох речень вихідної множини S. На наступних етапах у резолюції бере участь резольвента C_i, отримана на попередньому кроці (називається центральним диз'юнктом), і диз'юнкт B (називаний бічним), що є або одним з диз'юнктів вихідної множини S або центральним диз'юнктом C_j, що передує у висновку диз'юнкту C_i, тобто $j<i$. Лінійна стратегія є повною і одною з найбільш ефективних у реалізації.

Стратегія вхідних даних. Така стратегія відповідає лінійній резолюції з одним обмеженням: як бічні диз'юнкти обираються тільки диз'юнкти вихідної множини *S*. Дерево спростування, зображене на рис. 1.2, побудовано відповідно до стратегії вхідних даних.

Вхідна резолюція є повною тільки тоді, коли речення множини *S* є диз'юнктами Хорна.

Розглянуті стратегії пошуку можна комбінувати. Наприклад, стратегії унітарності й опорної множини. Крім цього, можна спрощувати множину вихідних речень *S*. Для цього виключаються тавтології, речення, що містять унікальні літерали, підвипадки [4]. Наприклад, якщо множина *S* містить диз'юнкт *P(x)*, то не має сенсу включати до складу *S* диз'юнкти вигляду *P(a)* чи $P(a) \vee Q(b)$, тому що вони являють собою підвипадки *P(x)*.

Розроблено велику кількість стратегій пошуку резольвент при доведенні теорем. Основний спосіб вирішення проблеми "комбінаторного вибуху" полягає у використанні семантики і впровадженні в правила виведення специфіки конкретної предметної області [3].

Застосування принципу резолюції

Застосування принципу резолюції припускає переведення вихідної постановки задачі на мову числення предикатів. При цьому розглядаються два випадки. У першому випадку потрібно з'ясувати, чи логічно слідує деякий вираз F_g з множини виразів Φ_0. У другому випадку, що зустрічається частіше, ставиться задача визначення значення аргументу *x*, за якого цей вираз F_g, що містить *x*, слідує з множини Φ_0. Іншими словами, у другому випадку потрібно спочатку встановити справедливість твердже-ння

$$\Phi_0 \Rightarrow \exists x F_g(x), \tag{1.25}$$

а потім знайти значення змінної *x*, за якого зазначене твердження виконується. З'ясуємо, як отримати з доведеної теореми відповідь на поставлене питання. Процес формування відповіді розглянемо на прикладах задач інформаційного пошуку, планування переміщення робота, автоматичного написання програм.

Інформаційний пошук

Нехай є база даних, що містить інформацію про хімічні сполуки. Наприклад, факти, що вказують, які сполуки вважаються оксидами, і який колір має та чи інша сполука. Допустимо, що серед багатьох фактів, що зберігаються в базі даних, існує твердження про те, що з'єднання *MgO* – оксид, і колір цього оксиду – білий. Ці факти можна подати мовою числення предикатів у вигляді двох одномісних предикатів: *oxide(MgO)* і *white(MgO)*. Тут як імена предикатів вибрані слова англійської мови *oxide* (оксид) і *white* (білий).

Нехай потрібно встановити, чи існує оксид білого кольору. Це питання мовою числення предикатів запишеться у вигляді:

$$F_g(x) = \exists x(oxide(x) \wedge white(x)). \quad (1.26)$$

Вираз $F_g(x)$, який потрібно довести, називають припущенням. Знайдемо заперечення припущення

$$F_g(x) = \exists x(\overline{oxide}(x) \wedge \overline{white}(x)) \quad (1.27)$$

і доведемо, що множина $S=\{oxide(MgO), white(MgO), \overline{oxide}(x) \vee \overline{white}(x)\}$ нездійсненна. Для цього побудуємо дерево спростування (рис. 1.3).

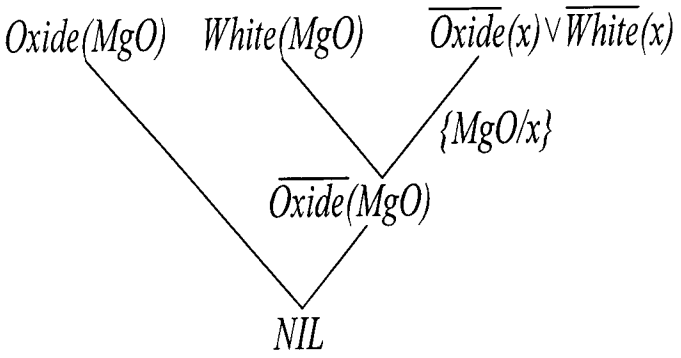

Рисунок 1.3 – Дерево спростування

Потім необхідно отримати те часткове значення змінної, яке належить до квантора існування, що і є відповіддю на поставлене питання. Для цього будують *модифіковане дерево доведення.* Процес його побудови полягає в такому:

1) до заперечення припущення дописують через диз'юнкцію саме припущення, тобто замість заперечення припущення записують тавтологію;

2) знаходять резольвенти тих же речень, що брали участь у побудові дерева спростування.

У корені модифікованого дерева доведення формується речення, яке і використовується як розгорнута відповідь на поставлене питання. Модифіковане дерево доведення зображено на рис. 1.4.

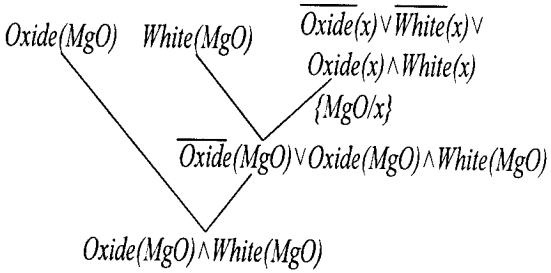

Рисунок 1.4 – Модифіковане дерево доведення

Інтерпретуючи речення, що знаходиться в корені модифікованого дерева доведення, встановлюємо, що існує оксид *MgO*, який має білий колір.

Розглянутий приклад демонструє відповідь на питання типу "так" чи "ні". Питання, що належать до типу "скільки", також можна сформулювати у вигляді доведення теорем. Нехай існує правило [5]: "якщо в $x \in m$ елементів типу y, а в $y \in n$ елементів типу z, то в $x \in m \times n$ елементів типу z". Це правило можна записати у вигляді виразу, використовуючи предикат *HP(a,b,c)*, який інтерпретується в такий спосіб: "*a*" містить *"c"* елементів типу "*b*". Тоді правило виражається у вигляді:

$$\forall x \forall y \{HP(x,y,m) \wedge HP(y,z,n) \rightarrow HP(x,z,times(m,n))\}, \qquad (1.28)$$

де функція *times(m,n)* означає добуток *m* на *n*. Припустимо, що під об'єктом *x* розуміється людина. Системі відомо, що людина має дві руки і що на кожній руці п'ять пальців. Потрібно з'ясувати, скільки пальців на руках у людини. Безумовно, з погляду природного інтелекту, це не питання. Однак відзначимо, що система повинна формально вивести відповідь на поставлене питання на основі відомих їй фактів і загального правила.

Факти, відомі системі, і питання можна записати у такому вигляді:

HP(man, hand, 2);
HP(hand, fingers, 5);
$\exists k(HP(man, fingers, k))$,

де *man* – людина, *hand* – рука, *fingers* – пальці.

Питання інтерпретується так: чи існує таке *k*, що справедливо *HP(man, fingers, k)*.

Побудуємо множину *S*. Для цього приведемо правило до вигляду речення

$$\overline{HP}(x,y,m) \vee \overline{HP}(y,z,n) \vee HP(x,z,times(m,n)) \qquad (1.29)$$

і знайдемо заперечення припущення

$$\overline{F_g}(k) = \forall k \overline{HP}(man, fingers, k). \qquad (1.30)$$

Тоді

$$S = \{HP(man, hand, 2),$$
$$HP(hand, fingers, 5),$$
$$\overline{HP}(x, y, m) \vee \overline{HP}(y, z, n) \vee HP(x, z, times(m, n)),$$
$$\overline{HP}(man, fingers, k)\}.$$

Модифіковане дерево доведення, побудоване за принципом вхідної резолюції, зображено на рис. 1.5. На першому кроці знаходиться наслідок речення, що являє собою правило, і заперечення цільового твердження. На другому і третьому кроках використовуються факти *HP (man, hand,2)*, *HP (hand, fingers,5)*. У корені дерева формується відповідь на поставлене питання *HP(man, fingers, times(2,5))*, тобто кількість пальців на руках у людини можна обчислити за допомогою виклику функції *times(2,5)*.

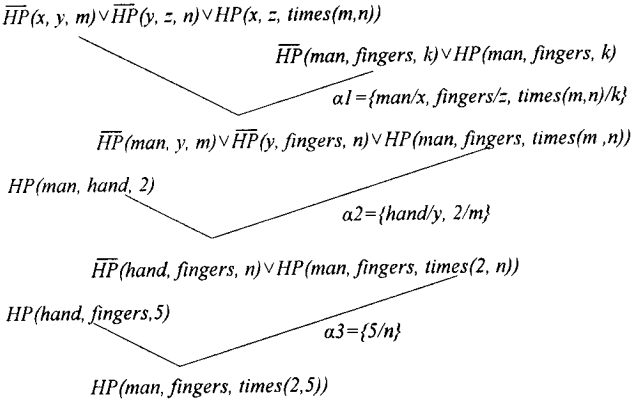

Рисунок 1.5 – Модифіковане дерево доведення

У розглянутих вище прикладах припущення містило квантор існування. У тих випадках, коли припущення містить квантор спільності, виникають додаткові труднощі. Справа в тому, що при запереченні квантора спільності він переходить у квантор існування, що у процесі стандартизації предикатних виразів виключається шляхом введення функцій Сколема. Труднощі полягають в інтерпретації цих функцій, якщо вони з'являються у відповідному твердженні. У [6] рекомендуються такі функції заміняти новими змінними. При доведенні ніяких підстановок в ці змінні не роблять.

Планування пересування робота

Ця задача формулюється як задача пошуку шляху на графі станів (рис. 1.6) [4, 5]. У вихідному стані задачі робот знаходиться в позиції "*a*". Необхідно знайти послідовність операторів, що забезпечують переміщення робота в позицію "*c*", котра відповідає вирішеному стану задачі. Для вирішення задачі необхідно встановити відповідність між описом задачі в просторі станів і описом її мовою числення предикатів.

Операторам простору станів ставляться у відповідність функції мови числення предикатів. При цьому перетворення стану s_1 у стан s_2 задається функцією $s_2=f(s_1)$. Для ділянки графа станів можна записати аксіому $P(s_1)\rightarrow Q(s_2)$ або $P(s_1)\rightarrow Q(f(s_1))$, де $P(s_1)$ – предикат, що описує властивість початкового стану ділянки графа; $Q(s_2)$ – предикат, що задає властивість кінцевого стану ділянки графа. Іншими словами, якщо стан s_1 має властивість P, то стан $s_2=f(s_1)$ має властивість Q.

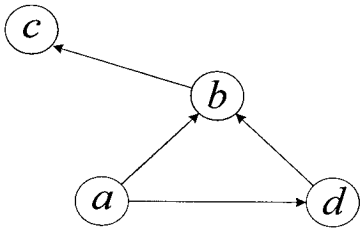

Рисунок 1.6 – Граф станів задачі переміщення робота

Введемо функцію *move(x,y,s)*, яка відповідає переміщенню робота з точки *x* в точку *y* в стані *s*. Для опису стану задачі будемо використовувати предикат *AT(x, s)*, який означає, що для стану задачі *s* робот знаходиться у точці *x*. У цьому випадку можна записати такі аксіоми:

$$AT(a, s_0),$$
$$\forall s_1(AT(a, s_1) \rightarrow AT(b, move(a, b, s_1))),$$
$$\forall s_2(AT(a, s_2) \rightarrow AT(d, move(a, d, s_2))),$$
$$\forall s_3(AT(b, s_3) \rightarrow AT(c, move(b, c, s_3))),$$
$$\forall s_4(AT(d, s_4) \rightarrow AT(b, move(d, b, s_4))).$$

Цільове твердження формулюється таким чином: чи існує стан задачі *s*, за якого робот буде знаходитися в точці "*c*". На мові числення предикатів це можна записати у вигляді:

$$\exists s AT(c, s). \qquad (1.31)$$

Дерево спростування для цієї задачі зображено на рис. 1.7. Результуюча функція, що забезпечує перехід робота з точки "*a* " у точку "*c*", виходить у результаті серії підстановок у змінну *s*:

$$s = move(b, c, move(a, b, s_0)). \tag{1.32}$$

Виконання дій відповідно до цієї функції полягає в тому, що робот спочатку повинен переміститися з точки *"a"* у точку *"b"*, а потім

$$\exists X append([a, b], [c], X), \tag{1.33}$$

яке на мові Пролог запишеться у вигляді:

$$?-append([a, b], [c], X). \tag{1.34}$$

Результатом виконання програми буде значення змінної X. Відповідне дерево спростування, яке реалізоване пролог-системою, зображено на рис. 1.7.

Значення змінної X визначається композицією підстановок:

$$\begin{aligned} &X \leftarrow [H|W] = [a|W], \\ &W \leftarrow [H'|W'] = [b|W'], \\ &W' \leftarrow L = [c], \end{aligned} \tag{1.35}$$

тобто

$$\mathrm{X} = [\mathrm{a} \mid [b \mid [c]\,]\,] = [\mathrm{a} \mid [b, c]\,] = [a, b, c]. \tag{1.36}$$

Звернемо увагу на той факт, що правила уніфікації, прийняті в мові програмування Пролог, складніші за розглянуті правила.

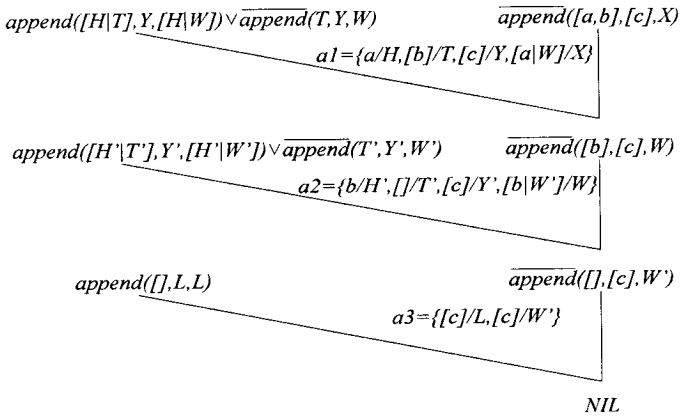

Рисунок 1.7 – Дерево спростування для відношення *append*

1.2 Модель індуктивного логічного виведення

Крім дедуктивного підходу, у системах штучного інтелекту широко використовують індуктивні схеми логічного виведення. Такі схеми дозволяють виводити узагальнення наявних часткових тверджень. Здатність до узагальнення є важливою функцією природного інтелекту і використовується для одержання нових знань. В ході навчання, завдяки узагальненню сукупності наявних фактів, формуються нові поняття і правила, що забезпечують віднесення об'єктів, подій, ситуацій до відповідних класів, завдяки цьому індуктивні схеми логічного виведення знаходять широке застосування при автоматизації процесу набуття знань.

Розглянемо основні поняття індуктивного виведення та процедури індуктивного формування понять.

1.2.1 Постановка задачі індуктивного формування понять

Індукцією називають умовивід, що являє собою знання про клас предметів, отримане в результаті дослідження окремих представників цього класу [7]. Якщо дедуктивне виведення будується за принципом руху думки від загального до часткового, то індуктивне виведення визначається як логічний перехід від часткового до загального, тобто як узагальнення.

Узагальнення тут розуміється як процес отримання знань, які пояснюють наявні факти. Метою узагальнення є формування понять.

Поняття – це узагальнена інформація про множину об'єктів (ситуацій, подій), поданих наборами значень ознак, яка:

а) відображає характерні для цієї множини логічні відношення між окремими значеннями ознак;

б) є достатньою для розпізнавання з допомогою деякого правила класифікації об'єктів, що належать множині, від об'єктів, що не належать їй [8].

Таким чином, поняття охоплює лише суттєві ознаки об'єктів, ситуацій, подій, що об'єднує їх в один клас, і не охоплює ті ознаки, що індивідуалізують їх.

Завдання індуктивного формування понять можна визначити таким чином. Дано множину фактів (спостережень, даних) F, що являють собою специфічні знання про деякі об'єкти (ситуації, події тощо); множину обмежень, які має задовольняти шукане поняття. Потрібно сформувати поняття H (опис, гіпотезу), з якого виведені всі факти, тобто $H \Rightarrow F$, де відношення $\Rightarrow$ розуміється ширше, ніж відношення логічного наслідування в численні предикатів.

Нехай D_i – множина значень i-ої ознаки, $i=1,2,...,n$; n – кількість ознак. Опису ознак поняття відповідають точки n – вимірного простору ознак $D=\{D_1 \times D_2 \times ... \times D_n\}$. Якщо кожному із значень ознак поставити у відповідність змінну, що приймає значення 1 або 0, залежно від того, спостерігається або не спостерігається це значення ознаки, то можна опи-

сувати поняття виразами булевої алгебри. *Кон'юнктивним* називається поняття, що описується кон'юнкцією значень ознак. Запис поняття у вигляді логічного виразу являє собою правило класифікації. Якщо логічний вираз, що являє собою поняття буде мати значення 1, то досліджуваний об'єкт (ситуація, подія) узагальнюється цим поняттям.

Тому задачу індуктивного формування понять на основі аналізу конкретних фактів часто розглядають як задачу відновлення правил.

Спостережувані факти називають *позитивними* по відношенню до деякого поняття, якщо вони описуються цим поняттям, інакше їх називають *негативними* фактами цього поняття.

Задачу індуктивного формування понять можна розглядати як задачу навчання. Виділяють навчання з вчителем (навчання за прикладом) і навчання без вчителя (навчання на основі спостереження).

У першому випадку сукупність фактів F має вигляд навчальної вибірки, яка є множиною прикладів об'єктів (ситуацій, подій) із заданою приналежністю до того чи іншого класу (поняття) K_1, K_2, ..., K_m. В основі навчання лежить впорядкований відбір кон'юнкцій значень ознак, що характеризують групи позитивних прикладів поняття і не визначають ні одного з негативних прикладів (контрприкладів). Потрібно побудувати правило, що дозволить визначити приналежність довільного об'єкта, ситуації, події, до заданого класу K_i.

У разі навчання без вчителя апріорний поділ фактів за класами відсутній. Потрібно за тими чи іншими критеріями виділити сукупність класів $\{K_i\}$ і побудувати для них вирішальні правила. Така задача складніша задачі навчання з вчителем. Методи формування понять за фактами без їх апріорного розбиття на класи розглядаються в кластерному аналізі.

Основна роль у процесі навчання відводиться індуктивному узагальненню. При цьому розрізняють повну і неповну індукцію [8]. Повною індукцією називають загальний висновок, який виводиться на основі вивчення всіх наявних фактів. Це можливо лише в тих випадках, коли кількість можливих фактів скінченна і невелика. Якщо висновки виводяться на основі вивчення лише частини фактів, то індукція називається неповною. Висновки, виконані з допомогою неповної індукції, є правдоподібними, оскільки з істинності часткового не обов'язково слідує істинність загального.

В основу індуктивного виведення покладено правило індуктивного узагальнення

$$\frac{F, H \Rightarrow F}{H}, \tag{1.37}$$

де F – множина відомих фактів; H – поняття (гіпотеза).

Зміст цього правила полягає в такому. Нехай є множина фактів F. Якщо ввести деяку гіпотезу H і показати, що з H виводимо будь-який факт

F, то гіпотеза H правильна. Особливістю правила індуктивного узагальнення (1.37) є те, що множина об'єктів, що описується гіпотезою H, ширша, ніж множина, яка відповідає фактам F. Це може призводити до того, що з гіпотези H будуть виводитися й інші факти. Отже, є ризик зробити помилку. Тому при виборі гіпотези H необхідно прямувати до мінімального узагальнення.

Також відзначимо, що задача індуктивного формування понять близька до задачі навчання розпізнавання образів. І в тому і в іншому випадку формується модель класу об'єктів, ситуацій, подій. Однак у випадку формування понять, отримана модель повинна забезпечувати не тільки розпізнавання, але і можливість генерації описів конкретних об'єктів, ситуацій тощо [7].

Це пов'язано з розумінням процесу формування понять як процесу виділення закономірностей, набуття нових знань, де повинні відображатися ознакові, структурні і логічні характеристики об'єктів, ситуацій, подій реального світу. Враховуючи вказану специфіку, розглянемо процедури індуктивного узагальнення, що застосовуються в підсистемах набуття знань.

1.2.2 Індуктивне формування понять на основі алгоритму Вінстона

У 1975 році П. Вінстоном була розроблена програма, яка формує описи структурних понять. Програмі послідовно пред'являються описи позитивних і негативних прикладів (контрприкладів). Як контрприклади використовуються приклади, в які спеціально внесені помилки. На рис. 1.8 зображено позитивні і негативні приклади для поняття "арка".

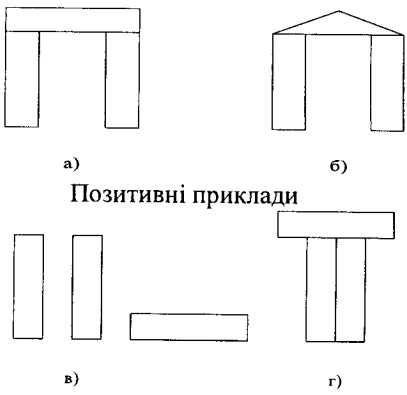

Рисунок 1.8 – Позитивні і негативні приклади поняття "арка"

Програма будує опис поняття у вигляді семантичної мережі, структура якої уточнюється по мірі пред'явлення прикладів. При цьому "сусідні" приклади повинні мінімально відрізнятися один від одного. Навчання програми виконується на основі операцій узагальнення та спеціалізації.

У разі пред'явлення програмі опису арки, що складається з трьох прямокутних блоків, формується семантична мережа, зображена на рис. 1.8, а. Наступний приклад арки, в якому верхній блок замінений пірамідою, описується семантичною мережею, зображеною на рис. 1.8, б.

Зіставляючи семантичні мережі, зображені на рис. 1.9, а і 1.9, б, програма виявляє відмінності між арками. Відмінність полягає в тому, що верхнім елементом першої арки є блок, а другої – піраміда. Використовуючи семантичну мережу, що відображає ієрархію зазначених "будівельних елементів" (рис. 1.10), програма узагальнює опис поняття "арка". При цьому замість поняття блок і піраміда використовується їх найближчий супертип – багатокутник, тобто виконується мінімальне узагальнення. В результаті формується узагальнений опис арки, що являє собою семантичну мережу, яка зображена на рис. 1.11.

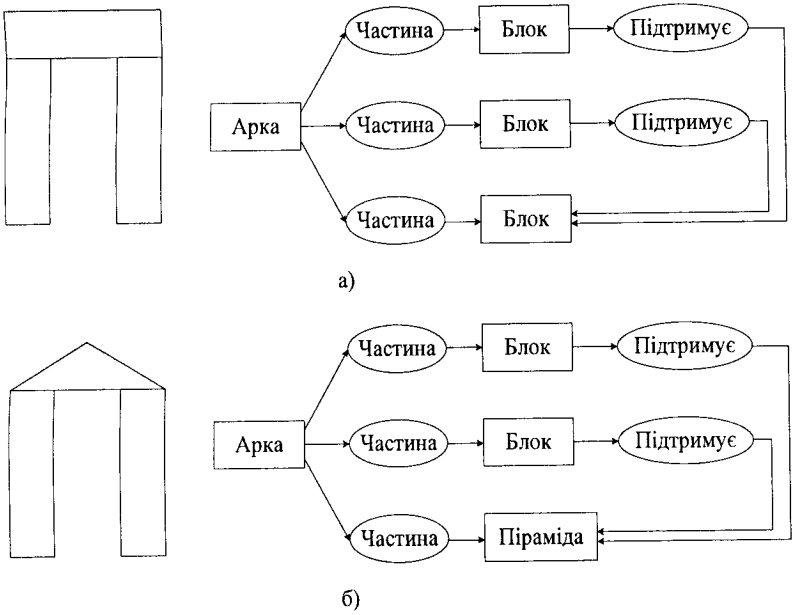

Рисунок 1.9 – Опис арки семантичною мережею

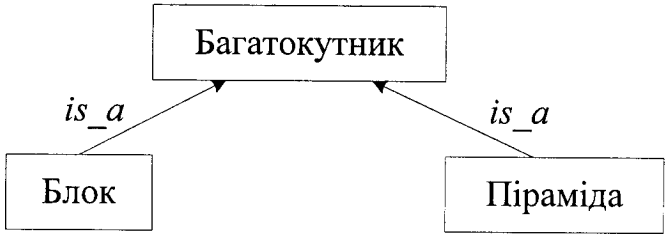

Рисунок 1.10 – Ієрархія "будівельних блоків"

Ця семантична мережа узагальнює позитивні приклади, зображені на рис. 1.9, а і 1.9, б. У разі пред'явлення негативних прикладів виконується обмеження спільності (спеціалізація) формованого поняття. Якщо пред'являється контрприклад, в якому підтримувальні блоки дотикаються, то формується семантична мережа контрприкладу, зображена на рис. 1.12. Мережа контрприкладу відрізняється наявністю відношення "дотикається".

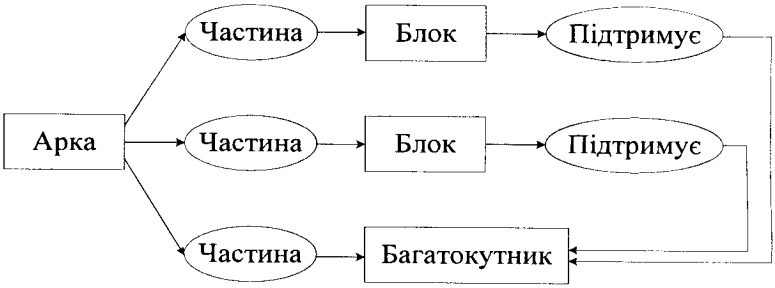

Рисунок 1.11– Узагальнений опис арки

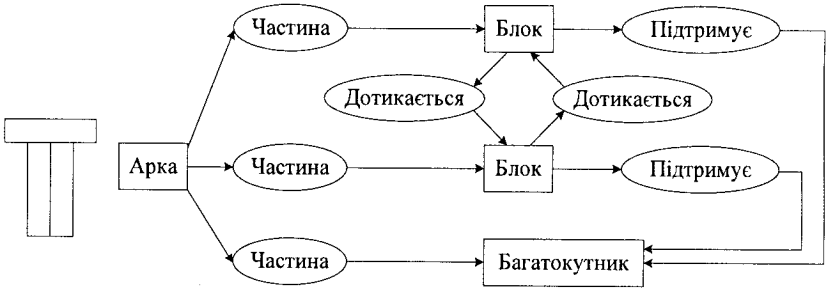

Рисунок 1.12 – Семантична мережа контрприкладу

Тому програма виконує спеціалізацію мережі, що зображена на рис. 1.12, додаючи відношення "не повинен дотикатися" (рис. 1.13). Метою спеціалізації є виключення з формованого опису поняття контрприкладу. Зазначимо, що контрприклади повинні бути максимально близькі до формованого опису цільового поняття. Розглянутий контрприклад поняття "арка" відрізняється від позитивного прикладу тільки одним додатковим відношенням. Це спрощує виконання операції спеціалізації.

Таким чином, в ході навчання виконуються дві основні операції: *узагальнення* і *спеціалізація*. Узагальнення полягає в заміні вузлів мережі більш загальними поняттями. Спеціалізація припускає додавання в мережу нових зв'язків. Якщо подати поняття, що формуються на основі операцій спеціалізації і узагальнення, у вигляді деяких станів процесу навчання, то можна помітити, що програма П. Вінстона виконує пошук в просторі понять. При цьому пошук управляється навчальними прикладами (даними) [9].

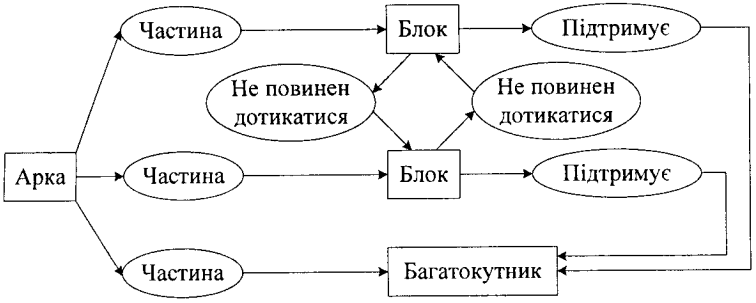

Рисунок 1.13– Спеціалізація опису

Оскільки в цьому випадку не передбачається повернення до попередніх станів, то алгоритм П. Вінстона чутливий до порядку, у якому подаються навчальні приклади. Приклади повинні подаватися в "якісному методичному порядку", який виключає в графі пошуку попадання у вершини "глухого кута". Крім цього, сусідні приклади і контрприклади повинні мінімально відрізнятися один від одного, що полегшує виконання операцій узагальнення та спеціалізації, а також спрощує процедуру зіставлення підграфів семантичної мережі.

Програма П. Вінстона була однією з перших програм, що реалізує ідеї індуктивного навчання. Вона позначила важливі моменти, властиві багатьом програмам машинного навчання:

– застосування операцій узагальнення і спеціалізація для формування простору понять;

– застосування методів пошуку, керованих даними (прикладами);

– залежність результатів навчання від якості навчальних даних.

1.2.3 Індуктивне формування понять на основі алгоритму Мітчелла

Алгоритм Т. М. Мітчелла, який називають алгоритмом виключення понять-кандидатів, аналогічно алгоритму П. Вінстона, призначений для формування одного поняття за пред'явленими прикладами. Алгоритм ґрунтується на двонаправленому пошуку в *просторі версій*, що являє собою множину всіх понять, які охоплюють позитивні приклади і не виконуються на негативних прикладах. В процесі виконання алгоритму пошуковий простір скорочується в напрямку від часткових понять до загальних, так і у зворотному напрямку. Якщо пред'являється позитивний приклад, то з пошукової області виключаються всі поняття-кандидати, які не охоплюють його. Якщо пред'являється, негативний приклад, то, навпаки, виключаються всі поняття-кандидати, що охоплюють його. Коли в пошуковій області залишається одне поняття, алгоритм припиняє роботу.

Термін "охоплююче поняття" пов'язаний з операцією узагальнення. При цьому узагальнення може виконуватися такими способами:

а) заміною константи на змінну, наприклад, висловлювання, задане атомарним виразом

колір(м'яч, синій)

узагальнюється предикатом колір (X, синій);

б) виключенням кон'юнктивних членів з виразів, наприклад, вираз

колір(X, синій)∧вага(X, великий)∧форма(X, прямокутна),

є окремим випадком виразу

колір(X, синій)∧форма(X, прямокутна);

в) додаванням диз'юнктивних елементів, наприклад, твердження

колір(X, синій)∧форма(X, прямокутна)

узагальнюється виразом

колір(X, синій)∧(форма(X, прямокутна)∨форма(X, трикутна));

г) узагальненням деякої властивості відповідно до ієрархії узагальнень, наприклад, предикати

форма(X, прямокутна)
форма(X, трикутна)

узагальнюються за допомогою виразу

форма(X, багатокутна)

для ієрархії узагальнень, що зображена на рис. 1.14.

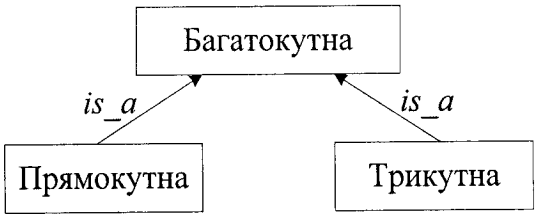

Рисунок 1.14 – Ієрархія узагальнень

Якщо поняття p є більш загальним, ніж поняття q, то p – *охоплююче поняття* для q.

В алгоритмі Мітчелла приклади подаються за допомогою атомарних виразів числення предикатів, і в ході пошуку будуються (відновлюються) продукційні правила, умови яких є кон'юнкціями зазначених атомів. Відновлені правила повинні забезпечувати правильну класифікацію запропонованих прикладів, тобто віднесення їх до того чи іншого класу (поняття) за мінімальної сукупності відмінних ознак.

Розглянемо приклад. Нехай є множина об'єктів, що характеризуються такими властивостями та їх можливими значеннями:

колір = {жовтий, зелений, червоний, синій};
форма = {коло, квадрат, трикутник};
розмір = {маленький, середній, великий}.

Кожен з об'єктів можна подати предикатом

об′єкт(Колір, Форма, Розмір),

якому відповідає простір понять, зображений на рис. 1.15. Потрібно на підставі запропонованих прикладів побудувати правило, що дозволяє виявляти коло червоного кольору.

Будемо використовувати такі позначення: G – множина найбільш загальних понять-кандидатів g, $g \in G$; S – множина найменш загальних (найбільш спеціалізованих) понять-кандидатів s, $s \in S$.

Алгоритм виключення понять-кандидатів формулюється таким чином:

1) створити множину G, включивши в неї найбільш загальне поняття простору пошуку;

2) створити множину S, включивши в неї перший позитивний приклад;

3) для кожного позитивного прикладу p виконати таке:

– виключити з G всі поняття-кандидати, які не охоплюють p;

– для кожної $s \in S$, якщо s не можна порівняти з p, замінити s більш загальним поняттям, яке можна порівняти з p, виконавши мінімально можливе узагальнення;

– виключити з s всі поняття-кандидати більш загальні, ніж будь-яке інше поняття, що входить в S;

– виключити з S всі поняття-кандидати, що охоплюють деякі поняття, які входять в G;

4) для кожного негативного прикладу n виконувати таке;

– виключити з S всі поняття-кандидати, які охоплюють n;

– для кожного $g \in G$, якщо g можна порівняти з n, замінити g менш загальним поняттям, яке не можна порівняти з n, виконавши мінімально можливу спеціалізацію;

– виключити з G всі поняття-кандидати менш загальні, ніж будь-яке інше поняття, що входить в G;

– виключити з G всі поняття-кандидати, які не охоплюють деякі поняття, що входять до S;

5) якщо $G=S$ і обидві множини є одноелементними, то знайдене поняття узгоджується з усіма даними, і пошук закінчується;

6) якщо G і S – порожні множини, то не існує поняття, яке охоплює всі позитивні приклади і не охоплює жодного негативного прикладу.

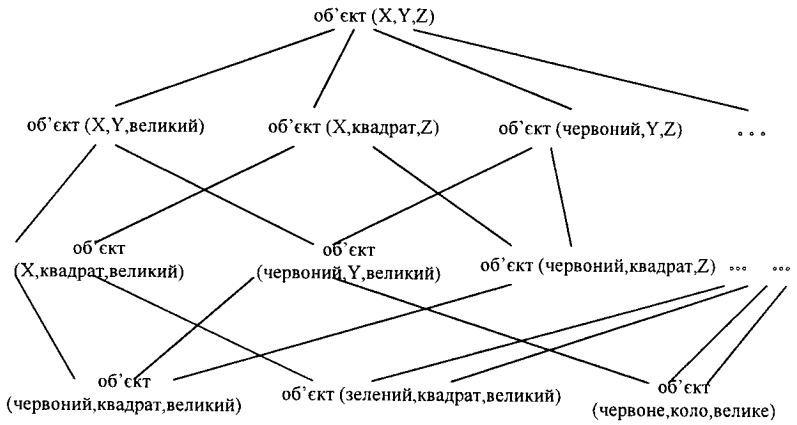

Рисунок 1.15 – Простір понять

Процес навчання поняттю "червоне коло" з допомогою розглянутого алгоритму подано в табл. 1.1.

Таблиця 1.1 – Навчання поняттю "червоне коло"

Позитивні *(p)* і негативні *(n)* приклади	Множини *G* і *S*
–	*G*: об'єкт (X, Y, Z). *S*: {}.
p: об'єкт (середній, коло, червоний)	*G*: об'єкт (X, Y, Z). *S*: об'єкт (середній, коло, червоний).
n: об'єкт (середній, квадрат, зелений)	*G*: об'єкт (X, Y, червоний), об'єкт (X, коло, Z). *S*: об'єкт (середній, коло, червоний).
p: об'єкт (великий, коло, червоний)	*G*: об'єкт (X, Y, червоний), об'єкт (X, коло, Z). *S*: об'єкт (X, коло, червоний).
n: об'єкт (середній, трикутник, червоний)	*G*: об'єкт (X, коло, червоний). *S*: об'єкт (X, коло, червоний).

Зазначимо, що множини *G* і *S*, які подані в табл. 1.1, містять лише остаточні результати, що відповідають третьому і четвертому пунктам розглянутого алгоритму. Так, у разі пред'явлення негативного прикладу

об'єкт(середній, квадрат, зелений)

множина *G* на першому етапі збільшується при виконанні операції спеціалізації за рахунок понять-кандидатів, які не можна порівняти з прикладом:

об'єкт (X, У, жовтий),
об'єкт (X, Y, червоний),
об'єкт (X, У, синій),
об'єкт (X, коло, Z),
об'єкт (X, трикутник, Z),
об'єкт (маленький, У, Z),
об'єкт (великий, У, Z).

Потім з множини *G* виключаються всі поняття-кандидати, які не охоплюють понять, що містяться в множині *S*. У підсумку в *G* залишаються два поняття

об'єкт (X, У, червоний),
об'єкт (X, коло, Z),

що відображено в табл. 1.1. Така властивість алгоритму пояснюється тим, що *S* – множина найменш загальних понять, які охоплюють позитивні приклади, тому будь-яке нове поняття, що включається в *G* в ході операції

спеціалізації і не може бути порівняним з будь-яким із понять, що входять у S, не може охоплювати позитивних прикладів і повинне бути виключене.

Має місце симетрична ситуація, коли виконується операція узагальнення S. Оскільки G – множина найбільш загальних понять, які не охоплюють жодного негативного прикладу, то будь-яке більш загальне поняття, що включається до S, можна буде порівняти з деяким негативним прикладом. Отже, воно повинне бути виключене.

Умовно пошук понять у просторі версій можна подати як на рис. 1.16 [9]. Тут знаком "+" відзначені позитивні навчальні приклади, а знаком "-" – негативні приклади.

Рисунок 1.16 – Навчання відповідно до алгоритму Мітчелла

Множина S містить в собі найменш загальні поняття, які охоплюють всі позитивні приклади. Множина G містить в собі найбільш загальні поняття, які не охоплюють жодного негативного прикладу. Будь-яке поняття більш загальне, ніж поняття G, буде охоплювати деякі негативні приклади; будь-яке поняття, менш загальне, ніж поняття з множини S, може не охоплювати деякі позитивні приклади. Тому в процесі навчання межа множини G звужується, щоб виключити поняття, що охоплюють негативні приклади, а межа множини S розширюється за рахунок включення понять, що охоплюють позитивні приклади. Процес спеціалізації понять, що входять в G, і узагальнення понять, що входять в S, закінчується успішно, коли множини G і S будуть містити одне і те ж поняття.

Таким чином, алгоритм Мітчелла належить до групи пошукових алгоритмів навчання. При цьому виконується пошук в ширину, що не ефективно, якщо множини G і S швидко розширюються. У цьому випадку можуть застосовуватися методи евристичного пошуку.

В алгоритмі Мітчелла передбачається, що кожен клас описується одним кон'юнктивним правилом, і надані приклади не спотворені шумом. На практиці часто необхідно відновлювати правила в умовах зашумлених даних. Алгоритм Мітчелла досить чутливий до наданих прикладів. Навіть один помилковий приклад може призвести до втрати збіжності.

1.2.4 Індуктивне формування понять на основі алгоритму Iterative Dichotomizer 3

Алгоритм ID3 (Iterative Dichotomizer 3) був запропонований Дж. Квінланом у 1983 р. Так само як і алгоритм Мітчелла, він забезпечує індуктивне формування понять за прикладами. Однак поняття в цьому випадку подаються за допомогою дерев рішень. Кожен нетермінальний вузол дерева рішень відповідає деякій ознаці класифікації, а гілки, що виходять з вузла, характеризують альтернативні значення ознаки. Термінальні вузли дерева рішень визначають клас (поняття), до якого належить пред'явлений приклад. На рис. 1.17 зображено фрагмент дерева рішень при визначенні ступеня ризику серцево-судинних захворювань.

Алгоритм ID3 забезпечує побудову найпростішого дерева рішень, яке охоплює всі пред'явлені приклади. Побудова дерева відповідно до алгоритму ID3 виконується в напрямку зверху вниз. У цьому випадку кожна ознака поділяє множину навчальних прикладів на підмножини, що містять приклади з однаковим значенням тієї чи іншої ознаки. Потім алгоритм рекурсивно застосовується до підмножин і т. д. Процес триває, доки пред'явлений приклад не буде віднесений до заданого класу.

У загальному випадку різний порядок аналізу ознак призводить до побудови різних дерев рішень. В алгоритмі ID3 для визначення порядку, в якому аналізуються ознаки, використовується інформаційний критерій. Сформулюємо алгоритм індуктивної побудови дерев рішень, а потім розглянемо критерій відбору ознак.

Наведемо алгоритм у вигляді рекурсивної функції на псевдомові [9]. При цьому будемо використовувати такі позначення: *MP* – множина прикладів; *SP* – список ознак класифікації; *V(P)* – множина значень ознаки *P*; *v* – значення ознаки *P*, $v \in V(P)$.

```
Function Induce_Tree(MP, SP);
Begin
If Всі елементи множини MP належать одному класу
  Then
    Повернути елемент дерева, забезпечивши його міткою класу
```

Else If список *SP* порожній Then

Повернути елемент дерева, подавши його у вигляді диз'юнкції всіх класів, що формують множину *MP*

Else Begin

Вибрати як корінь нового піддерева чергову ознаку *P*;

Видалити ознаку *P* зі списку *SP*;

For $v \in V(P)$ Do

Begin

Створити гілку дерева, забезпечивши її міткою *v*;

Сформувати з *MP* підмножину прикладів *MPv*, ознака *P* для яких має значення *v*;

Здійснити виклик Induce_Tree (*MPV, SP*) і приєднати піддерево, яке повертається, до гілки *v*

End

End

End.

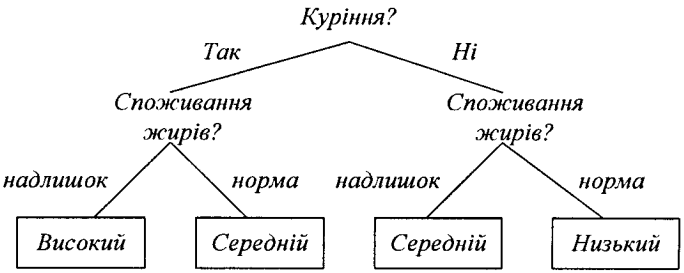

Рисунок 1.17 – Фрагмент дерева рішень

Розглянемо процес побудови дерева рішень, що зображено на рис. 1.17. Множину прикладів *MP* задано табл. 1.2.

Таблиця 1.2 – Приклади оцінок ризику серцево-судинних захворювань

№	Ризик	Куріння	Споживання жирів
1	Високий	так	надлишок
2	Середній	так	норма
3	Середній	ні	надлишок
4	Низький	ні	норма

При першому виклику функції Induce_Tree як корінь дерева вибирається ознака "куріння", яка характеризується двома значеннями "так" і "ні". Ці значення розбивають множину прикладів *MP* на дві підмножини MP_1 і

MP_2. Елементами першої підмножини є приклади 1 та 2. Елементами другої підмножини – приклади 3 і 4. Для кожної з підмножин MP_1 та MP_2 функція Induce_Tree викликається рекурсивно. При цьому в ході кожного з викликів відбувається подальше розбиття підмножин MP_1 і MP_2, але вже відносно ознаки споживання жирів. В цьому випадку утворюються підмножини, складені з елементів одного класу, і рекурсія повертає елементи дерева (рис. 1.17), позначені мітками відповідних класів. Елементи зв'язуються з відповідними гілками, позначені можливими значеннями ознаки "споживання жирів", тобто "норма" і "надлишок". У результаті формуються два піддерева, відповідні підмножинам MP_1 і MP_2. Вказані піддерева, в свою чергу, зв'язуються з гілками, позначеними можливими значеннями ознаки "куріння", і процес формування дерева рішень завершується.

Кожен виклик функції Induce_Tree формує чергове піддерево рішень, корінь якого являє собою ознаку P. Як критерій вибору чергової ознаки використовується кількість інформації, обумовлена включенням тієї чи іншої ознаки в дерево рішень.

Нехай є незалежні і несумісні повідомлення x_1, x_2, ..., x_n, ймовірності отримання яких рівні $p(x_1)$, $p(x_2)$, ..., $p(x_n)$. Тоді кількість інформації для всієї сукупності повідомлень можна оцінити з допомогою міри К. Шеннона:

$$I(x) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i). \qquad (1.38)$$

Якщо розглядати навчальні приклади як повідомлення, то за допомогою (1.38) можна оцінити кількість інформації, що міститься в навчальній вибірці і, отже, у дереві рішень, яке охоплює цю навчальну вибірку. Так, для навчальної вибірки, заданої табл. 1.2 і відповідної дереву рішень, що зображено на рис. 1.17, отримаємо p(високий) = 1/4; p(середній) = 2/4; p(низький) = 1/4, і

$$I(x) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} = -\frac{1}{2}\log_2\frac{1}{8} = -\frac{1}{2}\cdot(-3) = \frac{3}{2} = 1{,}5 \text{ біт}.$$

Кількість інформації, обумовлена включенням тієї чи іншої ознаки в дерево рішень, визначається як різниця між кількістю інформації, відповідним всьому дереву рішень, і кількістю інформації, відповідних піддерев, розташованих нижче вузла, який визначається вибраною ознакою класифікації. Кількість інформації, що міститься у всіх зазначених піддеревах, становить очікуваний обсяг інформації, який може бути отриманий після їх побудови.

Нехай є множина прикладів *MP*. Якщо на черговому кроці класифікації вибирається ознака *P*, яка характеризується *n* значеннями, то множина *MP* розбивається на *n* підмножин MP_1, MP_2, ..., MP_n. Тоді очікуваний об'єм інформації, пов'язаний з побудовою піддерев, відповідних підмножини MP_1, MP_2, ..., MP_m, буде дорівнювати

$$E(P) = \sum_{i=1}^{n} \frac{|MP_i|}{|MP|} I(MP_i), \tag{1.39}$$

де |MP|, $|MP_i|$ – відповідно потужності множин *MP* і MP_i;

$I(MP_i)$ – кількість інформації, відповідна піддереву множини MP_i.

Кількість інформації, що отримується при виборі ознаки *P*, визначиться з виразу

$$G(P) = I(MP) - E(P). \tag{1.40}$$

На кожному кроці виконання алгоритму ID3 вибирається ознака *P*, яка забезпечує максимальну кількість інформації *G(P)*.

Алгоритм ID3 можна розглядати як пошук, що відповідає алгоритму підйому вгору в просторі можливих дерев рішень. На кожному кроці пошуку аналізуються всі ознаки, які можуть використовуватися для розширення поточного дерева рішень. Вибирається та ознака, яка забезпечує отримання максимальної кількості інформації.

Розглянутий метод індуктивної побудови дерев рішень застосовується при вирішенні багатьох задач класифікації. Наприклад, алгоритм ID3 застосовувався для навчання при класифікації результатів шахових партій, при класифікації захворювань щитоподібної залози [10]. Це один з найбільш широко використовуваних методів індуктивного відновлення правил за прикладами, що забезпечує автоматичну побудову баз знань (БЗ) діагностичних експертних систем (ЕС).

1.3 Контрольні питання

1. В яких випадках неможливо використовувати дедуктивний метод та числення предикатів для логічного виведення?
2. Наведіть модель дедуктивного виведення на основі подання задач у вигляді теорем.
3. Що таке спростування? Наведіть приклад на основі попереднього питання.
4. Що таке літерал?
5. Що називають клозом (реченням)?
6. Що потрібно зробити для приведення виразу до вигляду речення?
7. Що таке універсум Ербрана?

8. Як визначається універсум Ербрана?

9. Що таке фундаментальний диз'юнкт?

10. Сформулюйте теорему Ербрана. На якому етапі формування логічного висновку її можна сформулювати?

11. Що потрібно виконати для встановлення нездійсненності множини диз'юнктів?

12. Назвіть основний недолік процедури Ербрана.

13. Що таке принцип резолюції?

14. Сформулюйте правило резолюції.

15. Що таке резольвента?

16. Що таке граф спростування?

17. Охарактеризуйте особливості застосування процедури підстановки у принципі резолюцій.

18. Що таке композиція двох підстановок?

19. Назвіть стратегії пошуку індуктивного навчання та опишіть їх.

20. Що таке модифіковане дерево пошуку і з якою метою його будують?

2 ЛОГІЧНЕ ВИВЕДЕННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ. ЕКСПЕРТНІ СИСТЕМИ І ТЕОРІЯ ЙМОВІРНОСТЕЙ

2.1 Логічне виведення в умовах ненадійних або неповних знань

Теорія числення предикатів дозволяє отримувати правильні висновки, опираючись на загальнозначущі твердження та обґрунтовані правила логічного виведення. Однак, у багатьох випадках потрібно здійснювати виведення в умовах, коли вихідні дані не є абсолютно точними і достовірними, а правила виведення мають евристичний характер і є ненадійними. У цьому випадку застосовують процедури, що забезпечують логічне виведення висновків з визначеним ступенем впевненості (процедури логічного виведення на ненадійних знаннях), а також процедури немонотонного логічного виведення, що допускають одержання в ході виведення висновків, які можуть логічно суперечити твердженням із бази знань.

Людина робить необхідні умовиводи в подібних умовах щодня: ставить медичні діагнози і призначає лікування, керуючись симптомами; з'ясовує причини поганої роботи двигуна за акустичним шумом; правильно розуміє уривки фраз природної мови; впізнає друзів за їх голосами і т. д. Знання, якими володіє людина у зазначених випадках, властива невизначеність. Ця невизначеність має різну природу. Вона може породжуватись неповнотою опису ситуації, ймовірнісним характером подій, за якими спостерігають, неточністю подання даних, багатозначністю слів природної мови, використанням евристичних правил виведення та інше. Найбільш важливі види невизначеності можна подати у вигляді дерева, зображеного на рис. 2.1 [11].

На першому рівні дерева зображені поняття, що являють собою якісну оцінку характеру невизначеності. Невизначеність може бути пов'язана або з неповнотою знань, або з їх неоднозначністю.

Неповнота знань виникає, коли зібрана не вся інформація, необхідна для побудови логічного виведення, висновку.

Неоднозначність означає, що істинність тих чи інших висловлювань не може бути встановлена з абсолютною достовірністю. Вона породжується або фізичними причинами (фізична невизначеність), або лінгвістичними (лінгвістична невизначеність).

Фізична невизначеність може бути пов'язана з випадковістю подій, ситуацій, станів об'єкта або неточністю поданих даних.

Лінгвістична невизначеність пов'язана з використанням природної мови для подання знань, що мають якісний характер, і виникає через множинність значень слів (полісемія) і сенсу фраз. Наприклад, "Двигун часто перегрівається" або "Петрик уже великий". У цих прикладах

неоднозначність зумовлена нечіткістю понять "часто" і "великий". Або, наприклад , "Він зустрів її на галявині з квітами". Як він її зустрів: з квітами чи без квітів? Такого роду невизначеності присутні в системах, на поведінку яких значною мірою впливають судження людини. При аналізі невизначеності сенсу фраз виділяють синтаксичну, семантичну і прагматичну невизначеності [11].

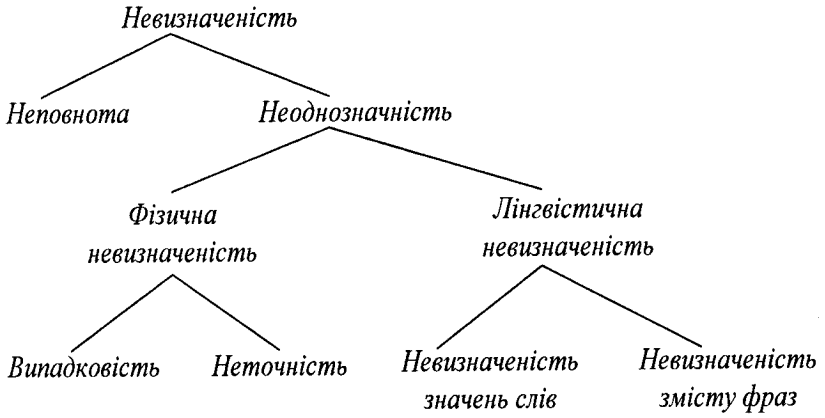

Рисунок 2.1 – Види невизначеності

Розглянута схема видів невизначеності певною мірою умовна. У реальних системах зазначені види невизначеності можуть накладатися одна на одну. Наприклад, фізична невизначеність може ускладнюватися лінгвістичною невизначеністю.

Часто зазначені вище види неоднозначності не виділяють в окремі групи, а розглядають в рамках одного терміну "ненадійні знання" [12]. Основоположним поняттям, що використовують при побудові моделей виведення на таких знаннях, є поняття достовірності. Достовірність висновків на основі ненадійних знань може бути визначена за допомогою різних підходів. Найбільш часто використовуються: ймовірнісна баєсова логіка, коефіцієнти впевненості, нечітка логіка, теорія Демпстера-Шефера [1].

Неповнота знань призводить до необхідності здійснення немонотонних висновків. Для формальної обробки знань, які характеризуються неповнотою, використовується логіка Рейтера, немонотонна логіка Мак-Дермотта і Дойла, системи підтримки значень істинності [1].

2.2 Загальна характеристика основних видів і джерел невизначеності

При вирішенні проблем ми часто зустрічаємося з безліччю джерел невизначеності використовуваної інформації, але в більшості випадків їх можна розділити на дві категорії: *недостатньо повне знання предметної області* і *недостатня інформація про конкретну ситуацію.*

Теорія предметної області, тобто наші знання про цю область, може бути неясною чи неповною: у ній можуть використовуватися недостатньо чітко сформульовані концепції чи недостатньо вивчені явища. Наприклад, у діагностиці психічних захворювань існують декілька теорій про походження і симптоматику шизофренії, які відрізняються одна від одної.

Невизначеність знань приводить до того, що правила впливу не завжди дають коректні результати навіть у простих випадках. Володіючи неповним знанням, ми не можемо впевнено передбачити, який ефект дасть та чи інша дія. Наприклад, при використанні нових препаратів терапія досить часто дає зовсім несподівані результати. І навіть коли ми маємо у своєму розпорядженні досить повну теорію предметної області, експерт може вирішити, що ефективніше використовувати не точні, а евристичні методи. Так, методика усунення несправності в електронному блоці шляхом заміни підозрілих вузлів виявляється значно більш ефективною, ніж скрупульозний аналіз ланцюгів у пошуку деталі, що вийшла з ладу.

Але крім неточних знань, невизначеність може бути внесена і неточними чи ненадійними даними про конкретну ситуацію. Будь-який сенсор має обмежену розподільну здатність, і аж ніяк не стовідсоткову надійність. При складанні звітів можуть бути допущені помилки або в звіти можуть потрапити недостовірні дані. На практиці далеко не завжди можна одержати повні відповіді на поставлені питання хоча і можна скористатися різного роду додатковою інформацією про пацієнта. Наприклад, за допомогою дорогих процедур або хірургічним шляхом. Такі методики використовуються вкрай рідко через високу вартість і ризикованість. Крім усього іншого, існує ще і фактор часу. Не завжди є можливість швидко одержати необхідні дані, коли ситуація потребує ухвалення термінового рішення. Якщо робота ядерного реактора викликає підозру, навряд чи хто-небудь буде чекати закінчення всього комплексу перевірок, перш ніж приймати рішення про його зупинку.

Підсумовуючи все сказане, відзначимо, що експерти користуються неточними методами з двох основних причин:

– точних методів не існує;

– точні методи існують, але не можуть бути застосовані на практиці через відсутність необхідного обсягу даних чи неможливості їхнього нагромадження з точки зору вартості або ризику через відсутність часу на збір необхідної інформації.

Більшість дослідників, що займаються проблемами штучного інтелекту, давно прийшли до єдиної думки, що неточні методи відіграють важливу роль у розробці експертних систем. Але багато суперечок викликає питання, які саме методи повинні використовуватися. До останнього часу багато вчених погоджувалися з твердженнями Мак-Карті і Хейєса про те, що теорія ймовірностей не є адекватним інструментом для вирішення задач подання невизначеності знань і даних [13]. Висувалися такі аргументи на користь цієї думки:

– теорія ймовірностей не дає відповіді на питання, як комбінувати ймовірності з кількісними даними;

– призначення ймовірності визначеним подіям потребує інформації, яку ми просто не маємо.

Інші дослідники додавали до цих аргументів власні:

– незрозуміло, як кількісно оцінювати такі поняття, що часто зустрічаються на практиці, як "у більшості випадків", "у рідких випадках", чи такі приблизні оцінки, як "старий" чи "високий";

– застосування теорії ймовірностей потребує "занадто багато чисел", що змушує інженерів давати точні оцінки тим параметрам, які вони не можуть оцінити;

– відновлення вірогідних оцінок обходиться дуже дорого, оскільки потребує великого обсягу обчислень.

Усі ці думки породили новий формальний апарат для роботи з невизначеностями, що одержав назву *нечітка логіка (fuzzy logic)* або *теорія функцій довіри (belief functions)*. Цей апарат широко використовується для вирішення задач штучного інтелекту й особливо при побудові експертних систем. Однак в останні роки захисники теорії ймовірностей почали досить ефективну контратаку, а тому будуть також подані основні концепції цієї теорії і її головних конкурентів.

2.3 Експертні системи і теорія ймовірностей

У цьому розділі будуть розглянуті ті аспекти теорії ймовірностей, що мають відношення до подання невизначеностей. Ми почнемо з поняття *умовної ймовірності* і зупинимося на тих причинах, через які ймовірнісний підхід критикується більшістю дослідників, що займаються експертними системами. Потім ми повернемося до коефіцієнтів впевненості, розглянемо їх докладніше і порівняємо отримані результати при використанні цього апарата й апарата теорії ймовірностей.

Умовна ймовірність

Умовна ймовірність події d при даному s – це ймовірність того, що подія d настане за умови, що наступила подія s. Наприклад, ймовірність того, що пацієнт дійсно страждає захворюванням d, якщо в нього (чи в неї) виявлений тільки симптом s.

У традиційній теорії ймовірностей для обчислення умовної ймовірності події d при даному s використовується така формула:

$$P(d \mid s) = \frac{P(d \wedge s)}{P(s)}. \qquad (2.1)$$

Як видно з (2.1), умовна ймовірність визначається в термінах сумісності подій. Вона являє собою відношення ймовірності збігу подій d і s до ймовірності появи події s. З формули (2.1) випливає, що

$$P(d \wedge s) = P(d \mid s)P(d). \qquad (2.2)$$

Якщо поділити обидві частини на $P(s)$ і підставити в праву частину (2.1), то одержимо правило Баєса в найпростішому вигляді:

$$P(d \mid s) = \frac{P(s \mid d)P(d)}{P(s)}. \qquad (2.3)$$

Це правило іноді називають *інверсною формулою для умовної ймовірності*. Воно дозволяє визначити ймовірність $P(d|s)$ появи події d за умови, що відбулася подія s через відому умовну ймовірність $P(s|d)$. В отриманому виразі $P(d)$ – апріорна ймовірність настання події d, а $P(d|s)$ – апостеріорна ймовірність, тобто ймовірність того, що подія d відбудеться, якщо відомо, що подія s здійснилася.

Для систем, заснованих на знаннях, формула (2.3) набагато зручніша, ніж формула (2.1), в чому ви зможете надалі переконатися.

Припустимо, що в пацієнта є деякий симптом захворювання, наприклад, біль у грудях. Бажано знати, яка ймовірність того, що цей симптом є наслідком визначеного захворювання, наприклад, інфаркту міокарда чи перикардиту (запалення каверн у легенях), чи чого-небудь менш серйозного, типу нетравлення шлунку. Для того, щоб обчислити ймовірність P *(інфаркт міокарда | біль у грудях)* за формулою (2.1), потрібно знати або оцінити яким-небудь способом, скільки людей у світі страждають таким захворюванням і скільки людей хворі інфарктом міокарда та скаржаться на біль у грудях (тобто мають такий же симптом). Як правило, така інформація відсутня, особливо остання, яка потрібна для обчислення ймовірності P *(інфаркт міокарда ∧ біль у грудях)*. Таким чином, визначення, дане формулою (2.1), у клінічній практиці не може бути використано.

Відзначена складність одержання потрібної інформації є причиною негативного відношення багатьох фахівців в галузі штучного інтелекту до ймовірнісного підходу взагалі [14]. Це негативне відношення підкріплювалося тим, що в більшості класичних робіт з теорії ймовірностей поняття ймовірності визначалось як *об'єктивна частотність* (частота появи при досить тривалих незалежних дослідах).

Однак існує думка, що ці базові припущення небезперечні з погляду практичних задач [15, 16]. Прихильники такого підходу дотримуються *суб'єктивістської* точки зору на визначення ймовірності, що дозволяє працювати з оцінками спільної появи подій, а не з дійсною частотою. Такий погляд на речі пов'язує ймовірність сукупності подій із суб'єктивною вірою в те, що подія дійсно настане.

Наприклад, лікар може не знати чи не мати можливості обчислити, яка частина пацієнтів, що скаржаться на біль у грудях, страждає інфарктом міокарда. Але на підставі власного досвіду він може оцінити у якої частини його пацієнтів, що страждають цим захворюванням, зустрічався такий симптом. Отже, він може оцінити значення ймовірності *P(біль у грудях | інфаркт міокарда)*. Суб'єктивний погляд на природу ймовірності тісно пов'язаний із правилом Баєса з такої причини. Припустимо, ми маємо у своєму розпорядженні досить достовірну оцінку ймовірності $P(s|d)$, де s означає симптом, а d – захворювання. Тоді за формулою (2.3) можна обчислити ймовірність $P(d|s)$. Оцінку ймовірності $P(d)$ можна взяти з медичної статистики, що публікується, а оцінити значення $P(s)$ лікар може на підставі власних спостережень.

Обчислення $P(d|s)$ не викликає ускладнень, коли мова йде про єдиний симптом. Тобто мається множина захворювань D і множина симптомів S, причому для кожного члена з D потрібно обчислити умовну ймовірність того, що в пацієнтів, які страждають цим захворюванням, спостерігався один визначений симптом з множини S. Проте, якщо в нескінченності D мається m членів, а в нескінченності S – n членів, буде потрібно обчислити $mn + m + n$ оцінок ймовірностей. Це аж ніяк не проста робота, якщо в системі медичної діагностики нараховується до 2000 видів захворювань і величезна кількість найрізноманітніших симптомів.

Але ситуація значно ускладнюється, якщо ми спробуємо включити в процес складання діагнозу не один симптом, а декілька.

У більш загальній формі правило Баєса має вигляд:

$$P(d \mid s_1 \wedge \ldots \wedge s_k) = \frac{P(s_1 \wedge \ldots \wedge s_k \mid d)P(d)}{P(s_1 \wedge \ldots \wedge s_k)}, \qquad (2.4)$$

і потребує обчислення $(mn)^k + m + n^k$ оцінок ймовірностей, що навіть при невеликому значенні k є дуже великим числом. Ці оцінки ймовірностей потрібні нам тому, що в загальному випадку для обчислення $P(s_1 \wedge \ldots \wedge s_k)$ потрібно попередньо обчислити добуток вигляду:

$$P(s_1 \mid s_2 \wedge \ldots \wedge s_k)P(s_2 \mid s_3 \wedge \ldots \wedge s_k)\ldots P(s_k). \qquad (2.5)$$

Однак, якщо припустити, що деякі симптоми незалежні один від одного, то обсяг обчислень істотно знижується. Незалежність будь-якої пари симптомів s_i і s_j означає, що

$$P(s_i) = P(s_i \mid s_j), \tag{2.6}$$

з чого випливає співвідношення

$$P(s_i \wedge s_j) = P(s_i)P(s_j). \tag{2.7}$$

Якщо всі симптоми незалежні, то обсяг обчислень буде таким, як і у випадку обліку при діагнозі єдиного симптому.

Але, навіть якщо це не так, у більшості випадків можна припустити наявність *умовної незалежності*. Це означає, що пари симптомів s_i і s_j є незалежними, оскільки в нашому розпорядженні є якісь додаткові свідчення на цей випадок чи фундаментальні знання E. Таким чином,

$$P(s_i \mid s_j, E) = P(s_i \mid E). \tag{2.8}$$

Наприклад, якщо в автомобілі немає пального і не працює освітлення, то можна сказати, що ці ознаки незалежні, оскільки наших знань у пристрої автомобіля цілком достатньо, щоб припустити, що між ними немає ніякого причинного зв'язку. Але якщо автомобіль не заводиться і не працює освітлення, то заявляти, що ці ознаки незалежні не можна, оскільки вони можуть бути наслідком однієї і тієї ж несправності акумуляторної батареї. Ступінь довіри до ознаки "не працює освітлення" тільки збільшиться, якщо виявиться, що до того ж і двигун не заводиться. Необхідність відслід-ковувати такого роду зв'язки в програмі і відповідно коректувати ступінь довіри до ознак значно збільшує обсяг обчислень у загальному випадку [17].

Таким чином, використання теорії ймовірностей ставить перед нами такі проблеми, які найкраще сформулювати як задачу вибору:

– або апріорі передбачається, що всі дані незалежні, і використовувати менш трудомісткі методи обчислень, за що прийдеться платити зниженням вірогідності результатів;

– або потрібно організувати відстеження залежності між використовуваними даними, кількісно оцінити цю залежність, реалізувати оперативне відновлення відповідної нормативної інформації, тобто ускладнити обчислення, але одержати більш достовірні результати.

2.4 Логічне виведення на основі баєсового підходу

Баєсовий метод

При баєсовому підході ступінь достовірності кожного з фактів бази знань оцінюється ймовірністю, яка приймає значення в діапазоні від 0 до 1. Ймовірності вихідних фактів визначають або методом статистичних випробувань, або опитуванням експертів. Ймовірності висновків визначають на основі правила Баєса для обчислення апостеріорної умовної

ймовірності $p(H|E)$ події (гіпотези) H за умови, що відбулася подія свідчення E

$$p(H|E) = \frac{p(E,H)}{p(E)}, \tag{2.9}$$

де $p(E)$ – безумовна (апріорна) ймовірність свідчення E;

$p(E,H)$ – спільна ймовірність свідчення E і гіпотези H.

Спільна ймовірність $p(E,H)$ дорівнює добутку безумовної ймовірності гіпотези $p(H)$ на умовну ймовірність того, що свідчення (факт) E має місце, якщо спостерігається гіпотеза H:

$$p(E,H) = p(H) \cdot p(E|H). \tag{2.10}$$

Звідси слідує, що апостеріорну ймовірність $p(H|E)$ можна обчислити за допомогою виразу

$$p(H|E) = \frac{p(H) \cdot p(E|H)}{p(E)}. \tag{2.11}$$

Уточнимо практичне застосування виразу (2.11) на простому прикладі. Нехай H позначає деяке захворювання, а E – симптом. Тоді апріорна ймовірність захворювання H може бути визначена за виразом

$$p(H) = N_H/N, \tag{2.12}$$

де N_H – кількість жителів деякого регіону, що мають захворювання H;

N – кількість всіх жителів регіону.

Ймовірність $p(E)$ визначається аналогічно

$$p(E) = N_E/N, \tag{2.13}$$

де N_E – кількість жителів, у яких спостерігається симптом E.

Зазвичай значення ймовірностей $p(H)$ і $p(E)$ з'ясовують у експертів.

Ймовірність $p(E|H)$ відповідає наявності симптому E у хворого з захворюванням H. Її значення також визначають методом опитування експертів.

Вираз (2.11) справедливий для випадку одного свідчення E і однієї гіпотези H. Вона дозволяє перераховувати значення ймовірності гіпотези H в тому випадку, коли виявлено свідчення E на її користь, тобто отримувати на основі апріорної ймовірності $p(H)$ значення $p(H|E)$ апостеріорної ймовірності. Якщо розглядається повна група несумісних гіпотез H_1, H_2, ..., H_n з апріорними ймовірностями $p(H_1)$, $p(H_2)$, ..., $p(H_n)$, то апостеріорну

ймовірність кожної з гіпотез при реалізації свідчення E обчислюють за допомогою виразу

$$p(H_i|E) = \frac{p(E|H_i)\cdot p(H_i))}{\sum_{k=1}^{n} p(E|H_k)\cdot p(H_k)}, \quad (2.14)$$

який називають теоремою гіпотез Баєса.

Вираз (2.14) дозволяє спростити обчислення ймовірності гіпотези H при реалізації свідчення E. Так, якщо розглядати дві несумісні гіпотези H і "H" ("не H"), то

$$p(H|E) = \frac{p(E|H)\cdot p(H)}{p(E|H)\cdot p(H)+p(E|H)\cdot(1-p(H))}, \quad (2.15)$$

де $p(E|H)$ – ймовірність свідчення E за умови, що гіпотеза H не спостерігається;

$1\text{-}p(H)=p(H)$ – апріорна ймовірність невиконання гіпотези H.

У виразі (2.15), на відміну від (2.11), відсутня апріорна ймовірність $p(E)$, яка є незручною з точки зору експертного оцінювання [18].

Розглянемо відношення

$$D_E = \frac{p(E|H)}{p(E|H)}, \quad (2.16)$$

яке називають *відношенням правдоподібності*. Воно характеризує відношення ймовірності отримання свідчення E за умови, що гіпотеза H правильна, до ймовірності отримання цього ж свідчення за умови, що гіпотеза H неправильна. Поділивши чисельник і знаменник (2.15) на $p(E|H)$, отримаємо

$$\frac{p(H|E)}{1-p(H|E)} = D_E \frac{p(H)}{1-p(H)}. \quad (2.17)$$

Відношення, записані у виразі (2.17), називають шансами. *Шанси* – це інша шкала для подання ймовірності. Так, відношенням

$$O(H) = \frac{p(H)}{1-p(H)} \quad (2.18)$$

подаються апріорні шанси на користь гіпотези H, а відношенням

$$O(H|E) = \frac{p(H|E)}{(1-p(H|E))} \quad (2.19)$$

подаються апостеріорні шанси. Наприклад, якщо $p(H)$=0,3, то $O(H)$=3/7, тобто 3 випадки "за" і 7 "проти".

З виразу (2.17) слідує, що

$$O(H|E) = D_E \cdot O(H). \tag{2.20}$$

Таким чином, апостеріорні шанси досить просто обчислюються через апріорні шанси. Для цього необхідно знати відношення правдоподібності свідчення E. Вважається, що використання шансів відповідно до виразу (2.20) для оцінювання правдоподібності гіпотез простіше, ніж безпосереднє обчислення ймовірностей за виразом (2.17). Це пояснюється двома причинами [18]:

1) для багатьох користувачів використання шкали шансів виглядає простіше;

2) шанси позначають цілими числами, а їх простіше вводити при відповідях на запитання системи.

Відзначимо, що можна отримати вираз, симетричний (2.20). Це дозволяє обчислювати апостеріорні шанси на користь гіпотези H, якщо завчасно відомо, що E хибне

$$O(H|-E) = N_E \cdot O(H), \tag{2.21}$$

де N_E – відношення правдоподібності $p(-E|H)/p(-E|-H)$.

Отже, шанси на користь деякої гіпотези H можна обчислювати, опираючись або на істинність свідчення E, або на його хибність. Такий підхід використовується в експертній системі PROSPECTOR, яка застосовується в геології. При цьому відношення правдоподібності D_E і N_E називають відповідно *фактором достатності* і *фактором необхідності*. Фактори D_E та N_E пов'язані один з одним простим співвідношенням

$$N_E = \frac{1 - D_E \cdot p(E|H)}{1 - p(E|H)}, \tag{2.22}$$

яке дозволяє встановити діапазон зміни їх значень.

Можна показати, що значення D_E належать діапазону $[1, \infty)$, а значення N_E – діапазону [0, 1]. З кожним правилом в системі PROSPECTOR пов'язують фактори D_E і N_E, значення яких можуть призначатися незалежно. Це іноді призводить до суперечностей. Щоб розв'язати такі протиріччя, фактор D_E використовується, коли свідчення E істинне, а фактор N_E – коли хибне.

На практиці гіпотеза H може підтверджуватися не одним свідченням, а кількома. Вираз (2.20) легко узагальнюється на випадок кількох свідчень. Наприклад, якщо реалізувалися два свідчення E_1 та E_2 на користь гіпотези H, то її апостеріорні шанси можна обчислити за виразом

$$O(H|E_1, E_2) = D_{E_2} D_{E_1} \cdot O(H) \tag{2.23}$$

або

$$O(H|E_1, E_2) = D_{E_1} \cdot O(H|E_1). \quad (2.24)$$

Оскільки, зазвичай, D_E>1, то отримання нових свідчень на користь гіпотези H дозволяє збільшити її правдоподібність.

В розглянутих вище виразах передбачалося, що свідчення E_1, E_2, ..., E_n повністю визначені, тобто підтверджуються з ймовірністю 1. Однак на практиці факти, що використовуються в системі, можуть підтверджуватися з меншою ймовірністю. Це може відбуватися через невизначені відповіді користувача, а також в результаті логічного виведення, коли висновки одних правил виступають в ролі фактів (свідчень) інших правил. У цьому випадку необхідно при обчисленні правдоподібності тієї чи іншої гіпотези враховувати ненадійність фактів. Нехай відомо (наприклад, з попередніх висновків), що свідчення E підтверджується з ймовірністю $p(E|E')$, де через E' позначено деяке свідчення, що підтверджує E. Тоді апостеріорну ймовірність гіпотези H можна обчислити за виразом [18, 19]:

$$p(H|E') = p(E|E') \cdot p(H|E) + (1 - p(E|E')) \cdot p(H|E). \quad (2.25)$$

З виразу слідує:

1) якщо свідчення E підтверджується з ймовірністю $p(E|E')$=1, то $p(H|E')=p(H|E)$;

2) якщо $p(E|E')$=0 (тобто свідчення E не підтверджується), то $p(H|E')=p(H|E)$.

Крім того, якщо $p(E|E')=p(E)$ (тобто свідчення підтверджується з апріорною ймовірністю), то значення ймовірності не повинно змінитися: $p(H|E')=p(H)$.

Таким чином, значення $p(H|E')$ лежать в діапазоні від $p(H|E)$ до $p(H)$, якщо $0 \le p(E|E') \le p(E)$ і в діапазоні від $p(H)$ до $p(H|E)$, якщо $p(E) \le p(E|E') \le 1$. Виконавши кусково-лінійну апроксимацію залежності $p(H|E')$ від $p(E|E')$, отримують вирази [19]:

$$p(H|E') = p(H|E) + \frac{p(H) - p(H|E)}{p(E)} \cdot p(E|E'), \quad (2.26)$$

якщо $0 \le p(E|E') \le p(E)$.

$$p(H|E') = p(H) + \frac{p(H|E) - p(H)}{1 - p(H)} \cdot \big(p(E|E') - p(E)\big), \quad (2.27)$$

якщо $p(E) \le p(E|E') \le 1$.

Ці вирази дозволяють виконати корекцію значення $p(H|E)$ залежно від ймовірності підтвердження свідчення E. В експертних системах "PROSPECTOR" та "Малая ЭС" при заповненні бази знань користувачам дозволяється замість ймовірності $p(E|E')$ задавати коефіцієнти (фактори)

впевненості, що лежать в діапазоні від -5 до +5. Надалі значення цих коефіцієнтів перераховуються в ймовірності $p(E|E')$ [18, 19].

Значення апостеріорної ймовірності $p(H|E')$, обчислене з урахуванням ймовірності підтвердження свідчення E, визначає ефективне відношення правдоподібності $D_{E'}$

$$D_{E'} = \frac{p(H|E')}{1-p(H|E')} \cdot \frac{1-p(H)}{p(H)}. \quad (2.28)$$

Це відношення дозволяє виконувати корекцію апостеріорних шансів на користь гіпотези H [12]:

$$O(H|E') = D_{E'} \cdot O(H). \quad (2.29)$$

Правила логічного виведення допускають об'єднання свідчень E_1, E_2, ..., E_n за допомогою логічних зв'язок І, АБО, НІ. У такій ситуації обробка кожного правила виконується з урахуванням принципів нечіткої логіки. Для свідчень, пов'язаних логічною операцією І, вибирається мінімальна з ймовірностей. Для логічної операції АБО береться максимальна з ймовірностей. Операція логічного заперечення НІ призводить до обчислення зворотної ймовірності.

При використанні баєсового підходу для обробки неточних знань виникають дві основні проблеми. По-перше, повинні бути відомі всі апріорні умовні ймовірності свідчень, а також апріорні ймовірності гіпотез (або відповідні відношення правдоподібності і шанси). По-друге, з умов теореми Баєса слідує, що всі гіпотези, що розглядаються системою, повинні бути несумісні, а ймовірності $p(E|H_i)$ і $p(E)$ – незалежні. В окремих галузях (наприклад, в медицині) остання вимога не виконується.

Крім цього, при введенні нової гіпотези, необхідно заново перераховувати таблиці ймовірностей, що також обмежує можливість застосовування баєсового підходу. Тим не менше, він досить широко застосовується в експертних системах, завдяки якісному теоретичному фундаменту. Важливо, що після багаторазового застосування теореми Баєса вплив всіх вихідних припущень на результат стає мінімальним. Тому, хоча апріорні ймовірності можуть визначатися наближено, їх співвідношення обчислюється досить надійно. Свій подальший розвиток розглянутий підхід отримав в баєсових мережах довіри (БМД) [9].

2.5 Логічне виведення на основі коефіцієнтів впевненості

2.5.1 Загальна характеристика коефіцієнтів впевненості

В ідеальному світі можна обчислити ймовірність $P(d_i|E)$, де d_i – i-та діагностична категорія, а E являє собою всі необхідні додаткові свідчення чи фундаментальні знання, використовуючи тільки ймовірності $P(d_i|s_j)$, де

s_j є j-м клінічним спостереженням (симптомом). Раніше йшлося про те, що правило Баєса дозволяє виконати такі обчислення тільки в тому випадку, якщо, по-перше, доступні всі значення $P(s_j|d_i)$, і, по-друге, правдоподібне припущення взаємної незалежності симптомів.

У системі MYCIN застосований альтернативний підхід на основі правил впливу, що у такий спосіб зв'язують наявні дані (свідчення) з гіпотезою рішення:

ЯКЩО пацієнт має показання і симптоми s_i^...^s_k і
мають місце визначені фонові умови t_1^...^t_m ,
ТО можна з упевненістю сказати, що пацієнт
страждає захворюванням d_i.

Коефіцієнт впевненості τ приймає значення в діапазоні [-1,+1]. Якщо τ=+1, то це означає, що при дотриманні всіх згаданих умов укладач правила абсолютно впевнений у *правильності* висновку d_i. Якщо ж τ=-1, то виходить, що при дотриманні всіх згаданих умов існує абсолютна впевненість у *помилковості* цього висновку. Відмінні від +1 додатні значення коефіцієнта вказують на ступінь впевненості в *правильності* висновку d_i, а від'ємні значення – на ступінь впевненості в його *помилковості*.

Основна ідея полягає в тому, щоб за допомогою породжуючих правил такого виду, спробувати замінити обчислення $P(d|s_1 \Lambda ... \Lambda s_k)$ наближеною оцінкою та у такий спосіб зімітувати процес ухвалення рішення експертом-людиною. Результати застосування правил такого виду зв'язуються з коефіцієнтом впевненості остаточного висновку за допомогою такої формули:

$$CF(d_i \mid s_1 \wedge ... \wedge s_k \wedge t_1 \wedge ... \wedge t_m) = \tau \times \min(CF(s_1),...,CF(s_k),CF(t_1),...,CF(t_m)), \quad (2.30)$$

де $CF(a)$ – коефіцієнт впевненості у вірогідності значення параметра a, а додаткові умови $t_1 \Lambda ... \Lambda t_m$ являють собою фонові знання, що обмежують застосування конкретного правила. Найчастіше виявляється, що ці умови можуть бути інтерпретовані значеннями "істина" чи "хиба", тобто відповідні коефіцієнти приймають значення +1 або -1. Таким чином, відмінні від одиниці значення коефіцієнтів характеризують тільки симптоми s_1, ..., s_k. Роль фонових знань полягає в тому, щоб дозволити чи заборонити застосування правила в цьому конкретному випадку. Нехай, наприклад, існує діагностичне правило, що зв'язує появу болю у черевній порожнині з можливою вагітністю. Застосування цього правила блокується фоновим знанням, що воно справедливо тільки стосовно пацієнтів-жінок.

Бучанан і Шортліфф стверджують, що застосування правила Баєса в будь-якому випадку не дозволяє одержати точні значення, оскільки умовні ймовірності, які використовуються, суб'єктивні [20]. Як ми вже бачили, це основний аргумент проти застосування ймовірнісного підходу. Однак така аргументація припускає *об'єктивістську* інтерпретацію поняття ймовірності. Тобто передбачається, що "хибні" значення все-таки існують, але ми не можемо їх одержати. В такому разі правило Баєса не можна використовувати. Цей аргумент має явно схоластичний відтінок, оскільки будь-яка експертиза, проведена інженером зі знань, зовсім очевидно зводиться до подання тих знань про предметну область, якими володіє людина-експерт (ці знання, звичайно ж, є суб'єктивними), а не до відтворення абсолютно адекватної моделі світу. З погляду теорії, доцільніше використовувати математично коректний формалізм до неточних даних, чим формалізм, що математично некоректний до тих же неточних даних.

Перл звернув увагу на важливу практичну перевагу підходу, заснованого на правилах [21]. Обчислення коефіцієнтів упевненості висновку має явно виражений модульний характер, оскільки не потрібно брати до уваги ніякої іншої інформації, окрім тієї, що існує в даному правилі. При цьому не має ніякого значення як саме отримані коефіцієнти впевненості, що характеризують вихідні дані.

При побудові експертних систем ця особливість використовується часто. Припускається, що для всіх правил, що мають справу з визначеним параметром, передумови кожного правила логічно незалежні. Аналізуючи систему MYCIN, Шортліфф порадив згрупувати всі залежні ознаки в єдине правило, а не розподіляти їх по множині правил [22].

Нехай існує залежність між ознаками E_1 та E_2. Шортліфф рекомендує згрупувати їх у єдине правило:

якщо E_1 і E_2, то приходимо до висновку H з упевненістю τ,

а не розподіляти на два правила:

якщо E_1, то приходимо до висновку H з упевненістю τ;
якщо E_2, то приходимо до висновку H з упевненістю τ.

В основі цієї рекомендації лежить один з наслідків теорії ймовірностей, з якого випливає, що $P(H|E_1, E_2)$ не може бути простою функцією від $P(H|E_1)$ і $P(H|E_2)$.

Вирази для умовної ймовірності не можуть у цьому сенсі розглядатися як модульні. Вираз $P(B|A)=\tau$ не дозволяє стверджувати, що $P(B)=\tau$ за наявності A, якщо тільки A не є єдиною відомою ознакою. Якщо крім A ми маємо у розпорядженні знання E, то потрібно спочатку обчислити $P(B \mid A, E)$, а вже потім можна буде що-небудь сказати і про значення $P(B)$. Така чутливість до контексту може стати основою дуже могутнього

механізму логічного висновку, але, як уже не раз підкреслювалося, за це прийдеться платити істотним підвищенням складності обчислень.

2.5.2 Приклад застосування коефіцієнтів впевненості в експертних системах

Метод коефіцієнтів впевненості був вперше застосований в експертній системі MYCIN. На відміну від баєсового підходу, який використовує теорію ймовірностей для підтвердження гіпотез, метод в ЕС MYCIN базується на евристичних міркуваннях, які були запозичені з практичного досвіду роботи експертів. Коли експерт оцінює ступінь достовірності деякого висновку, він використовує такі поняття як "точно", "досить ймовірно", "можливо", "нічого не можна сказати" і т. д. Розробники системи MYCIN вирішили відобразити ці розмиті поняття на шкалі коефіцієнтів впевненості, що змінюються в діапазоні від -1 до +1. Для цього були введені дві оцінки – *MB* і *MD*. Оцінка *MB* відображає ступінь істинності деякого факту (свідчення) і приймає значення від 0 до +1. Оцінка *MD* відповідає ступеню хибності деякого факту і приймає значення в діапазоні від -1 до 0. Коефіцієнт впевненості факту, що позначається *CF*, визначається як різниця оцінок *MB* та *MD*:

$$CF = MB - MD. \tag{2.31}$$

Тут $0<MB<1$ (при $MD=0$) і $-1<MD<0$ (при $MB=0$). Якщо коефіцієнт впевненості приймає значення, що дорівнює +1, то факт вважається істинним. Якщо $CF=-1$, то факт хибний. Шкала зміни значень коефіцієнтів упевненості наведена на рис. 2.2 [18].

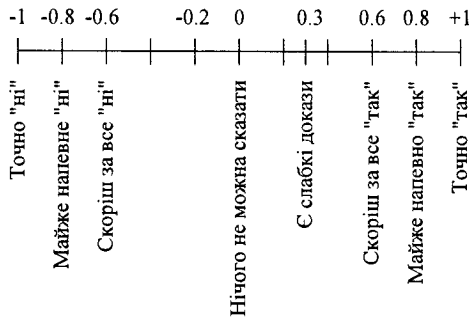

Рисунок 2.2 – Шкала коефіцієнтів впевненості

У ході логічного виведення над фактами, що становлять передумови правил, виконуються логічні операції. В результаті цього утворюються складні висловлювання, коефіцієнти впевненості яких обчислюються за такими правилами:

1) при логічному зв'язку *І* між фактами P_1 і P_2:

$$CF(P_1 \wedge P_2) = MIN(CF(P_1), CF(P_2)); \tag{2.32}$$

2) при логічному зв'язку АБО між фактами P_1 і P_2:

$$CF(P_1 \vee P_2) = MAX(CF(P_1), CF(P_2)). \tag{2.33}$$

В системі MYCIN коефіцієнти впевненості приписуються не тільки фактам, а й правилам. Таким способом забезпечується облік ненадійності правил, які часто формуються на основі евристичних міркувань. Позначимо коефіцієнт впевненості правила через CF_R. Коефіцієнт CF_R відповідає ступеню істинності висновку правила при істинних передумо-вах. Якщо передумови характеризуються коефіцієнтом впевненості $CF_{перед} \neq 1$, то коефіцієнт впевненості висновку $CF_{висн}$ обчислюють за вира-зом:

$$CF_{\text{висн}} = CF_{\text{перед}} \cdot CF_R. \tag{2.34}$$

Розглянемо приклад правила $(P_1 \wedge P_2) \vee P_3 \Rightarrow C_1(0{,}9)$, де P_1, P_2, P_3 – факти, що утворюють передумови правила; C_1 – висновок правила; 0,9 – коефіцієнт впевненості правила. Нехай факти характеризуються такими коефіцієнтами впевненості: $CF(P_1)=0{,}8$; $CF(P_2)=0{,}6$; $CF(P_3)=0{,}7$. Визначимо коефіцієнт впевненості висновку $CF(C_1)$:

$$CF(P_1 \wedge P_2) = \min(0{,}8; 0{,}6) = 0{,}6,$$
$$CF\big((P_1 \wedge P_2) \vee P_3\big) = \max(0{,}6; 0{,}7) = 0{,}7,$$
$$CF(C_1) = 0{,}7 \cdot 0{,}9 = 0{,}62.$$

В процесі логічного виведення один і той же висновок може підтверджуватися різними правилами, кожне з яких приписує висновку свій коефіцієнт впевненості. Очевидно, що коефіцієнт впевненості висновку, що підтверджується кількома правилами повинен збільшитися. Чим більше буде підтверджень на користь деякого висновку, тим ближче до одиниці повинен бути його коефіцієнт впевненості. У загальному випадку комбінація свідчень на підтримку деякого висновку виконується за такими виразами, що використовуються в системі EMYCIN [12, 19]:

$$CF = CF_1 + CF_2 - CF_1 \cdot CF_2, \text{ якщо } CF_1 > 0, CF_2 > 0;$$
$$CF = CF_1 + CF_2 + CF_1 \cdot CF_2, \text{ якщо } CF_1 < 0, CF_2 < 0;$$
$$CF = \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)}, \text{ якщо } CF_1 \cdot CF_2 \leq 0, CF_1 \neq \pm 1, CF_2 \neq \pm 1.$$

Якщо $CF_1 = \pm 1$ і $CF_2 = \mp 1$, то $CF = 1$.

Коефіцієнти впевненості висновків, які формуються трьома або більше правилами, можна вивести послідовно, застосовуючи наведені вище вирази.

На закінчення відзначимо, що метод коефіцієнтів впевненості завдяки своїй простоті знаходить широке застосування в багатьох системах, що підтримують виведення на ненадійних знаннях. Недоліки методу пов'язані з відсутністю теоретичного фундаменту, а також складністю підбору значень коефіцієнтів впевненості.

2.5.3 Коефіцієнти впевненості та умовні ймовірності

Учений Адамс показав, що якщо використовується проста ймовірнісна модель на основі правила Баєса, то в системі MYCIN коефіцієнти впевненості гіпотез не відповідають ймовірностям гіпотез при заданих ознаках [23]. На перший погляд, якщо коефіцієнти впевненості використовуються тільки для упорядкування альтернативних гіпотез, це не дуже страшно. Але Адамс також показав, що можлива ситуація, коли при використанні коефіцієнтів впевненості дві гіпотези будуть ранжовані в зворотному порядку відносно відповідних ймовірностей. Розглянемо це питання докладніше.

Позначимо через *P(h) суб'єктивне*, тобто сформоване на основі висновку експерта значення ймовірності того, що гіпотеза *h* справедлива, тобто значення *P(h)* відображає ступінь впевненості експерта в справедливості гіпотези *h*. Ускладнимо стан справ і додамо нову ознаку *e* на користь цієї гіпотези, такий що $P(h|e)>P(h)$. Ступінь довіри експерта до справедливості гіпотези збільшиться, і це збільшення виразиться відношенням

$$MB(h,e) = \frac{P(h \mid e) - P(h)}{1 - P(h)}, \tag{2.35}$$

де *MB* означає відносну *міру довіри*.

Якщо ж ознака *e* свідчить проти гіпотези *h*, тобто $P(h|e)<P(h)$, то збільшиться *міра недовіри* експерта до справедливості цієї гіпотези. Міру недовіри *MD* можна виразити таким відношенням:

$$MD(h,e) = \frac{P(h) - P(h \mid e)}{P(h)}. \tag{2.36}$$

Адамс звернув увагу на те, що рівні довіри до однієї і тієї же гіпотези з врахуванням різних додаткових ознак не можуть бути визначені незалежно. Якщо деяка ознака є абсолютним діагностичним індикатором конкретного захворювання, тобто якщо всі пацієнти із симптомом s_i страждають захворюванням d_j, то ніякі інші ознаки вже не можуть змінити

діагноз, тобто рівень довіри до висунутої гіпотези. Іншими словами, якщо існує пара ознак s_1, s_2 і

$$P(d_i \mid s_1) = P(d_i \mid s_1 \wedge s_2) = 1, \tag{2.37}$$

то

$$P(d_i \mid s2) = P(d_i). \tag{2.38}$$

Адамс також критично поставився до об'єднання (кон'юнкції) гіпотез. Модель, покладена в основу MYCIN, припускає, що рівень довіри до поєднання гіпотез $d_1 \wedge d_2$ повинний відповідати *найменшому* з рівнів довіри окремих гіпотез, а рівень недовіри – *найбільшому* з рівнів недовіри окремих гіпотез. Припустимо, що гіпотези d_1 і d_2 не тільки не незалежні, але і взаємно виключають один одного. Тоді $P(d_1 \wedge d_2|e) = 0$ за наявності будь-якої ознаки e і незалежно від ступеня довіри чи недовіри до d_1 або d_2.

Бучанан і Шортліфф визначили коефіцієнт впевненості як деякий артефакт, що дозволяє чисельно оцінити комбінацію рівнів довіри чи недовіри до гіпотез [20]. Він являє собою різницю між мірою довіри і недовіри:

$$CF(h \mid e_a \wedge e_f) = MB(h, e_f) - MD(h, e_a), \tag{2.39}$$

де e_f – ознака, що свідчить на користь гіпотези h, а e_a – ознака, що свідчить проти гіпотези h. Однак отримане в такий спосіб значення аж ніяк не еквівалентне умовній ймовірності існування гіпотези h за умови $e_a \wedge e_f$, яке випливає з правила Баєса:

$$P(h \mid e_a \wedge e_f) = \frac{P(e_a \wedge e_f \mid h) P(h)}{P(e_a \wedge e_f)}. \tag{2.40}$$

Таким чином, хоча ступінь довіри зв'язаний з визначеним правилом і може бути співвіднесений із суб'єктивною оцінкою ймовірності, коефіцієнт упевненості є комбінованою оцінкою. Його основне призначення полягає в такому:

– керувати ходом виконання програми при формуванні суджень;

– керувати процесом пошуку мети в просторі станів: якщо коефіцієнт впевненості гіпотези виявляється в діапазоні [+0.2, -0.2], то пошук блокується;

– ранжувати набір гіпотез після обробки всіх ознак.

Адамс, однак, показав, що ранжування гіпотез на основі коефіцієнтів впевненості може дати результат, протилежний тому, який буде отриманий при використанні ймовірнісних методів. Він продемонстрував це на такому прикладі.

Припустимо, що d_1 і d_2 – це дві гіпотези, а e – ознака, що свідчить як на користь однієї гіпотези, так і на користь іншої. Нехай між апріорними ймовірностями існує відношення $P(d_1) \geq P(d_2)$ і $P(d_1|e) > P(d_2|e)$. Іншими словами, суб'єктивна ймовірність справедливості гіпотези d_1 більше ніж гіпотези d_2, причому це співвідношення зберігається і після того, як до уваги береться додаткова ознака. Адамс показав, що за цих умов можливе зворотне співвідношення $CF(d_1,e) < CF(d_2,e)$ між коефіцієнтами впевненості гіпотез.

Припустимо, що ймовірності мають такі значення: $P(d_1)$=0.8, $P(d_2)$=0.2, $P(d_1|e)$=0.9, $P(d_2|e)$=0.8. Тоді підвищення довіри до d_1 буде дорівнювати (0.9-0.8)/0.2=0.5, а підвищення довіри до d_2 – (0.8-0.2)/0.8=0.75. Звідси випливає, що $CF(d_1,e) < CF(d_2,e)$, незважаючи на те, що і $P(d_1|e) > P(d_2|e)$.

Адамс назвав це явище "небажаною властивістю" коефіцієнтів довіри. Уникнути такої ситуації можна, якщо всі апріорні ймовірності будуть рівні. Нескладно продемонструвати, що ефект у приведеному вище прикладі з'явився наслідком того, що ознака e більше свідчить на користь гіпотези d_2, чим на користь d_1, тому що апріорна ймовірність останньої більш висока. Однак прирівнювання апріорних ймовірностей явно не погоджується зі стилем мислення тих, хто ставить діагноз, оскільки існує досить велика відмінність у частоті поєднань різних хвороб з однаковими симптомами. Отже, експерти будуть привласнювати їм зовсім різні значення суб'єктивних ймовірностей.

Послідовне застосування правил у системі MYCIN також пов'язано з існуванням деяких теоретичних проблем. Використовувана при цьому функція комбінування, заснована на припущенні, що якщо ознака e впливає на деяку проміжну гіпотезу h з ймовірністю $P(h|e)$, а гіпотеза h входить в остаточний діагноз d з ймовірністю $P(d|h)$, то

$$P(d \mid e) = P(d \mid h)P(h \mid e). \qquad (2.41)$$

Таким чином, складається враження, що транзитивне відношення в послідовності правил висновку суджень справедливо на першому кроці, але не справедливо в загальному випадку. Для того щоб існував зв'язок між правилами, популяції, пов'язані з цими категоріями, повинні бути вкладені приблизно так, як показано на рис. 2.3.

Адамс прийшов до висновку, що успіх практичного застосування системи MYCIN та інших подібних систем пояснюється тим, що в них використовуються досить короткі послідовності комбінування правил, а розглянуті гіпотези досить прості.

Інше критичне зауваження відносно MYCIN було висловлено Горвіцем і Гекерманом. Воно стосується використання коефіцієнтів впевненості як міри *зміни* довіри, у той час, коли в дійсності вони встановлюються експертами як ступінь *абсолютної* довіри [24].

Пов'язуючи коефіцієнти довіри з правилами, експерт відповідає на запитання: "Наскільки ви упевнені в правдоподібності того чи іншого висновку?" При застосуванні в MYCIN функцій комбінування додаткових ознак ці коефіцієнти стають мірою *відновлення* ступеня довіри, що призводить до несумісності цих значень з теоремою Баєса.

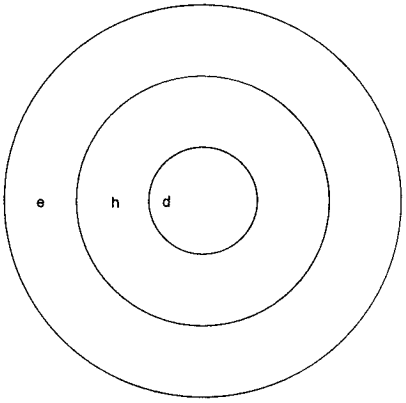

Рисунок 2.3 – Популяції, що дозволяють використовувати $P(d|e) = P(d|h)P(h|e)$

2.6 Альтернативні підходи до побудови моделей логічного виведення на основі нечіткої логіки

Крім використання коефіцієнтів впевненості, у літературі описані й інші підходи, альтернативні ймовірнісному. Зокрема, багато уваги приділяється нечіткій логіці (fuzzy logic) і теорії функцій довіри (belief functions). Про функції довіри йтиметься в розділі 3, а в цьому розділі розглянемо основні аспекти нечіткої логіки. Буде показано, чому підхід, заснований на ідеях нечіткої логіки, останнім часом усе ширше використовується при створенні експертних систем.

2.6.1 Нечіткі множини

Знання, яке експерт використовує при оцінюванні ознак чи симптомів звичайно базується скоріше на відносинах між класами даних і класами гіпотез, чим на відносинах між окремими даними і конкретними гіпотезами. Більшість методик вирішення проблем у тій чи іншій формі містять в собі класифікацію даних (сигналів, симптомів і т. п.), що розглядаються як конкретні представники деяких більш загальних категорій.

Рідко коли ці більш загальні категорії можуть бути чітко окреслені. Конкретний об'єкт може мати частину характерних ознак визначеної категорії, а частину не мати. Приналежність конкретного об'єкта до визначеного класу може бути розмита. Запропонована Заде [25] *теорія нечітких множин (fuzzy set theory)* являє собою формалізм, призначений для формування суджень про такі категорії і приналежних до них об'єктів. Ця теорія лежить в основі нечіткої логіки *(fuzzy logic)* [26] і *теорії можливостей (possibility theory)* [27].

Класична теорія множин базується на двозначній логіці. Вирази у вигляді $a \in A$, де a являє собою індивідуальний об'єкт, а A – множину подібних об'єктів, можуть приймати тільки значення "істина" або "неправда". Після появи поняття "нечітка множина" класичні множини іноді стали називати *твердими*. Твердість класичної теорії множин стала джерелом ряду проблем при спробі застосувати її до нечітко визначених категорій.

Розглянемо категорію, визначену словом "швидкий" (fast). Якщо застосувати це визначення до автомобілів, то який автомобіль можна вважати швидким? У класичній теорії ми можемо визначити множину A "швидких автомобілів" або перерахуванням (склавши список усіх членів множини), або ввівши до розгляду деяку характеристичну функцію f таку, що для будь-якого об'єкта $XF(X)$="істина" тоді і тільки тоді, коли $x \in A$.

Наприклад, ця функція може відбирати тільки ті автомобілі, що мають швидкість більш 150 миль за годину:

істина, якщо CARX і TOPSPEEDX> *150,* неправда в протилежному випадку.істина,якщоCAR(X)iTOP_SPEED(X)>□впротилежномувипадку (2.42)

Множина, визначена такою характеристичною функцією, подається формулою:

$$\{X \in CAR \mid TOP-SPEED\ (X) > 150\}. \qquad (2.43)$$

З цієї формули слідує, що елементами нової множини є ті елементи множини *CAR*, які мають максимальну швидкість понад 150 миль за годину.

А що можна сказати про множину (категорію) "швидких" автомобілів? У випадку, коли границі множини розмиті і належність елементів до множини може бути якимось чином ранжована, можна говорити про те, що окремий об'єкт (автомобіль) більш-менш типовий для цієї множини (категорії). Можна за допомогою деякої функції f охарактеризувати

ступінь належності об'єктів до такої множини. Функція $f(Y)$ визначена на інтервалі [0,1]. Якщо для об'єкта X функція $f(X)=1$, то об'єкт є членом множини, якщо $f(X)=0$, то об'єкт не є членом множини. Усі проміжні значення означають *ступінь членства* об'єкта X у цій множині. У прикладі з автомобілями знадобиться функція, що оперує з максимальною швидкістю кожного претендента на членство. Можна визначити її таким чином, що $f_{FAST}(80)=0$, $f_{FAST}(80)=1$, а проміжні значення являють собою деяку монотонну гістограму, що має значення в інтервалі між нулем і одиницею.

Тоді множина "швидких автомобілів" може бути охарактеризована функцією

$$f_{FAST-CAR}(X) = f_{FAST}(TOP - SPEED(X)), \tag{2.44}$$

яка визначена на множині всіх автомобілів. Таким чином, членами множини стають пари (об'єкт, ступінь), наприклад:

$$FAST - CAR = \{((Porche - 944, 0.9), BMW - 316, 0.5), (Chevy - Nova, 0.1)\}.$$

2.6.2 Багатозначна нечітка логіка

Роль, яку в класичній теорії множин відіграє двозначна булева логіка, у теорії нечітких множин відіграє багатозначна нечітка логіка, у якій припущення про приналежність об'єкта множині, наприклад, *FAST-CAR(Porche-944)* можуть приймати дійсні значення в інтервалі від 0 до 1. Виникає питання як, використовуючи концепцію невизначеності, обчислити значення істинності складного виразу, такого як *FAST-CAR(Chevy-Nova)*.

За аналогією з теорією ймовірності, якщо F являє собою нечіткий предикат, то операція заперечення реалізується за формулою $F(X)=1-F(X)$.

Але аналоги операцій кон'юнкції і диз'юнкції в нечіткій логіці не мають ніякого зв'язку з теорією ймовірностей. Розглянемо такий вираз:

"Porche 944 є швидким (fast), представницьким (pretentious) автомобілем".

У класичній логіці припущення *FAST-CAR(Porche-944)* $\wedge$ *PRETENTIOUS-CAR(Porche-944)* є істинним в тому і тільки в тому випадку, якщо істині обидва члени кон'юнкції. В нечіткій логіці існує угода: якщо F і G є нечіткими предикатами, то

$$f_{(F \wedge G)}(X) = min(f_F(X), f_G(X)). \tag{2.45}$$

Таким чином, якщо *FAST-CAR(Porche-944)*=0.9 і *PRETENTIOUS-CAR(Porche-944)*=0.7, то *FAST-CAR(Porche-944) Λ PRETENTIOUS-CAR(Porche-944)* = 0.7.

А тепер розглянемо вираз: *FAST-CAR(Porche-944) Λ FAST-CAR (Porche-944)*. Ймовірність істинності цього твердження дорівнює 0, оскільки *P(FAST-CAR(Porche-944) | FAST-CAR(Porche-944))*=0. Але в нечіткій логіці значення цього виразу буде дорівнювати 0.1. Який зміст має це значення? Його можна вважати показником належності автомобіля до нечіткої множини автомобілів середньої швидкості, що в чомусь близькі до швидких, а в чомусь – до повільних.

Зміст виразу *FAST-CAR(Porche-944)*=0.9 полягає в тому, що ми тільки на 90% впевнені в тому, що цей автомобіль є швидким саме через невизначеність самого поняття "швидкий автомобіль". Цілком доцільно припустити, що існує деяка впевненість у тому, що *Porche-944* не належить до швидких автомобілів. Наприклад, він повільніший за автомобіль, який приймає участь у гонках "Формула-1".

Аналог операції диз'юнкції в нечіткій логіці визначається в такий спосіб:

$$f_{(FVG)}(X) = \max(f_F(X), f_G(X)). \tag{2.46}$$

Тут також очевидна повна протилежність з теорією ймовірностей, у якій

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B). \tag{2.47}$$

Розглянемо такі припущення і значення істинності їхньої приналежності в нечіткій множині *FAST-CAR*:

FAST-CAR(Porche-944)∨*FAST-CAR(Porche-944)*=0.9,
FAST-CAR(BMW-316)∨*FAST-CAR(BMW-316)*=0.5,
FAST-CAR(Chevy-Nova)∨*FAST-CAR(Chevy-Nova)*=0.2.

Значення ймовірності істинності кожного з цих припущень, як це визначено в теорії ймовірностей, дорівнює 1. У нечіткій логіці більш високі значення для автомобілів *Porche-944* і *Chevy-Nova* пояснюються тим фактом, що ступінь належності кожного з цих об'єктів до нечіткої множини *FAST-CAR* вищий. Нечіткість концепції "швидкий чи не швидкий" більш сприятлива для них, чим для більш повільного *BMW-316*.

Оператори мають властивості комутативності, асоціативності і взаємної дистрибутивності. Застосуємо до них принцип *композитивності*, як до операторів у стандартній логіці. Тобто значення складених виразів

обчислюються тільки за значеннями виразів-компонентів. У цьому оператори нечіткої логіки складають повну протилежність законам теорії ймовірностей, згідно з яким при обчисленні ймовірностей кон'юнкції і диз'юнкції величин потрібно брати до уваги умовні ймовірності.

2.6.3 Базові аспекти теорії можливостей

Нечітка логіка має справу із ситуаціями, коли сформульоване питання і знання, які ми розташовуємо, містять нечітко окреслені поняття. Однак нечіткість формулювання понять є не єдиним джерелом невизначеності. Іноді ми просто не впевнені в самих фактах. Якщо стверджується: "Можливо, що Джон зараз у Парижі", то говорити про нечіткість понять Джон і Париж не доводиться. Невизначеність закладена в самому факті, чи дійсно Джон знаходиться в Парижі.

Теорія можливостей є одним з напрямків у нечіткій логіці, у якому розглядаються точно сформульовані питання, що базуються на неточних знаннях. У цьому підрозділі розглянемо тільки основні ідеї цієї теорії. Найкраще це зробити на прикладі.

Припустимо, що в шухляді знаходиться 10 куль, але відомо, що тільки деякі з них червоні. Яка ймовірність того, що на удачу із шухляди буде витягнута червона куля?

Просто обчислити шукане значення, ґрунтуючись на знаннях, що тільки кілька куль *червоні* (red), не можна. Проте для кожного значення X з $P(RED)$ у діапазоні [0,1] можна в такий спосіб обчислити можливість, що $P(RED) = X$.

По-перше, визначимо "декілька" (several) як нечітку множину, наприклад, так: $f_{SEVERAL}$={(3, 0.2), (4, 0.6), (5, 1.0), (6, 1.0), (7, 0.6), (8, 0.3)}.

У цьому визначенні вираз (3, 0.2) $\in f_{SEVERAL}$ означає, що 3 з 10 навряд чи можна визнати як "декілька", а вирази (5, 1.0) $\in f_{SEVERAL}$ та (6, 1.0) $\in$ $f_{SEVERAL}$ означають, що значення 5 і 6 з 10 ідеально узгоджуються з поняттям "декілька". Зверніть увагу на те, що в визначення нечіткої множини не входять значення 1 і 10, оскільки інтуїтивно зрозуміло, що "декілька" означає "більше одного" і "не всі". Нечітка множина, визначена на множині чисел, називається *нечіткими числами (fuzzy numbers)*. За таким же принципом, що і множина $f_{SEVERAL}$, можна визначити нечіткі множини f_{FEW} для поняття "мало" та f_{MOST} для поняття "майже".

Тепер розподіл можливостей для $P(RED)$ можна подати у вигляді такої формули:

$$f_{p(\mathrm{Red})} = SEVERAL\,|\,10, \tag{2.48}$$

після підстановки значень в яку отримуємо:

$$\{(0.3, 0.2), (0.4, 0.6), (0.5, 1.0), (0.6, 1.0), (0.7, 0.6), (0.8, 0.3)\}.$$

Вираз (0.3, 0.2) $\in f_{p(Red)}$ означає, що шанс на те, що $P(RED)$=0.3, дорівнює 20%. Можна розглядати $f_{p(Red)}$ як *нечітку ймовірність (fuzzy probability)*.

Припускаючи, що майже будь-яке поняття може бути областю визначення такої функції, природно ввести і поняття "нечітке значення правдоподібності". Ми часто оцінюємо деяке твердження як "дуже правдоподібне" чи "частково правдоподібне". Таким чином, можна уявити собі нечітку множину f_{TRUE}:[0, 1]→[0, 1], де і область визначення, і область значень функції f_{TRUE} є можливими значеннями правдоподібності в нечіткій логіці. Отже, можна одержати $TRUE(FASR\text{-}CAR(Porsche\text{-}944))$=1 навіть при $FASR\text{-}CAR(Porsche\text{-}944)$=0.9, оскільки (0.9, 1.0) $\in f_{TRUE}$. Це означає, що будь-яке припущення щодо значення 0.9 розглядається як "досить правдоподібне". Таким чином, можна з упевненістю сказати, що *Porsche-944* є швидким автомобілем, незважаючи на те, що на ринку є і більш швидкісні.

2.7 Особливості проблеми невизначеності

Одна з головних переваг формалізму нечіткої логіки щодо застосування в експертних системах полягає в можливості комбінування його логічних операторів. Раніше ми вже відзначали, що для правила в ЕС MYCIN якщо пацієнт має показання і симптоми $s_1 \Lambda ... \Lambda s_k$ і мають місце визначені фонові умови $t_1 \Lambda ... \Lambda t_m$, то можна з упевненістю τ стверджувати, що пацієнт страждає захворюванням d_i.

Оцінювання набору симптомів $s_1 \Lambda ... \Lambda s_k$, відповідно до аксіом теорії ймовірностей, охоплює обчислення добутків вигляду:

$$P(s_1 \mid s_2 \wedge ... \wedge s_k) P(s_2 \mid s_3 \wedge ... \wedge s_k) ... P(s_k). \qquad (2.49)$$

Така операція в гіршому випадку потребує обчислення k-1 оцінки ймовірностей поверх тих, які необхідно для s_i. Було також показано, що в ЕС MYCIN кон'юнкція інтерпретується як оператор нечіткої логіки При цьому обчислюється $min(s_1 \Lambda ... \Lambda s_k)$. Це може іноді привести до результатів, цілком протилежних тим, що випливають з теорії ймовірностей. При порівнянні результатів, отриманих за допомогою різних методів обробки невизначеності в практичних системах, були знайдені й інші приклади помилкових висновків. Це порівняння показало, що методи, засновані на нечіткій логіці, менш надійні, чим ті, котрі використовують баєсовий підхід [28].

З іншого боку, можна відзначити, що людині також не властиво будувати судження на основі баєсового підходу. Дослідження Канемана і Тверського показали, що люди схильні не брати до уваги колишній досвід і віддавати перевагу більш свіжій інформації [29]. Деякі дослідники

думають, що людям властиво переоцінювати свою компетентність [30], причому більшість мають слабке уявлення про теорію оцінок [31].

Частково привабливість нечіткої логіки для проектувальників експертних систем складається в її близькості до природної мови. Таким термінам, як "швидкий", "небагато", "правдоподібно", найчастіше дається інтерпретація на основі повсякденного досвіду й інтуїції. Це спрощує процес інженерії знань, оскільки подібні судження людини-експерта можна безпосередньо перетворити у вираження нечіткої логіки.

Тут були викладені тільки основні ідеї, щоб читач міг отримати перше уявлення про концепцію невизначеності знань і даних, та пов'язаних з цим проблем. Але навіть з цього короткого викладу ясно, що ще має бути дуже багато зроблено для того, щоб мати повне поняття про адекватне уявлення невизначеності в технічних системах.

2.8 Контрольні питання

1. На які категорії поділяється невизначеність інформації?
2. До чого призводить невизначеність знань про предметну область?
3. Що таке умовна ймовірність?
4. Наведіть правило Баєса в найпростішому вигляді та поясніть його.
5. У чому полягає складність обчислення оцінок умовної ймовірності?
6. Яким чином складність обчислення оцінок умовної ймовірності знижується?
7. Назвіть проблеми використання теорії ймовірностей в експертних системах.
8. Наведіть альтернативний підхід, застосований системі MYCIN на основі правил впливу.
9. Опишіть значення, які приймає коефіцієнт впевненості τ і як їх потрібно трактувати.
10. У чому полягає роль фонових правил?
11. Опишіть модель, покладену в основу MYCIN, яка визначає рівень довіри і недовіри до сполучення гіпотез $d_1 \Lambda d_2$.
12. Які бувають види невизначеностей? Охарактеризуйте кожну із них.
13. В чому полягає основна ідея баєсового методу? Проілюструйте його ідею на прикладі.
14. В чому полягає основна ідея логічного виведення на основі коефіцієнтів впевненості?
15. Яким значенням впевненості відповідають крайні позначки на шкалі коефіцієнтів впевненості? Чому дорівнює позначка 0?
16. Наведіть коефіцієнт впевненості, який запропонували Бучанан і Шортліфф. Поясніть значення аргументів.
17. Яке основне призначення коефіцієнта впевненості?
18. Що таке "небажана властивість" коефіцієнта ймовірності і як її можна уникнути?

19. Що таке теорія нечітких множин, для чого вона застосовується?

20. В чому полягає теорія можливостей?

21. Назвіть одну з головних переваг застосування нечіткої логіки в експертних системах.

22. В основу якої теорії покладено теорію нечітких множин?

23. На чому базується класична теорія множин?

24. Охарактеризуйте дослідження Канемана і Тверського.

25. Бучачан і Шортліфф стверджують, що застосування правила Баєса в будь-якому випадку не дозволяє одержати точні значення. Чому?

2.9 Вправи

1. Яка ймовірність того, що з повної колоди буде витягнута одна зі старших карт (король, дама чи валет)?

2. Яка ймовірність того, що в кожному із двох послідовних кидків грального кубика випаде число більше трьох?

3. Припустимо, що ймовірність відмови одного з двигунів тримоторного літака дорівнює 0.01. Яка ймовірність того, що відмовлять усі три двигуни, якщо вважати, що роботоздатність одного двигуна не залежить від стану двох інших?

4. Припустимо, що поняття "декілька" визначене як нечітка множина: $f_{трохи}$ = {(3, 0.8), (4, 0.7), (5, 0.6), (6, 0.5), (7, 0.4), (8, 0.3)}. В ящику знаходиться 15 кульок і відомо, що декілька з них – сині. Яка ймовірність того, що з ящика буде витягнуто саме синю кульку?

5. Припустимо, що поняття "незвичайна оцінка з десяти" визначене як нечітка множина: $f_{незвичайна}$ = {(0, 1.0), (1, 0.9), (2, 0.7), (3, 0.5), (4, 0.3), (5, 0.1), (6, 0.1), (7, 0.3), (8, 0.5), (9, 0.9), (10, 0.9)}, а поняття "висока оцінка з десяти" визначене як нечітка множина: $F_{висока}$ = {(0, 0), (1, 0), (2, 0), (3, 0.1), (4, 0.2), (5, 0.3), (6, 0.4), (7, 0.6), (8, 0.7), (9, 0.8), (10, 1.0)}.

Побудуйте складену функцію "незвичайно висока оцінка з десяти".

3 ОСНОВИ ТЕОРІЇ ДЕМПСТЕРА-ШЕФЕРА

У теорії Демпстера-Шефера (Dempster-Shafer) передбачається, що гіпотези – компоненти простору гіпотез Θ – є взаємно виключаючими, а набір гіпотез – вичерпним. У термінології авторів простір гіпотез Θ називається *областю аналізу (frame of discernment)*. Також передбачається, що ми маємо у своєму розпорядженні засіб одержання свідчень не тільки на користь окремих гіпотез h_1, ..., h_n приналежних Θ, але і на користь підмножин гіпотез A_1, ..., A_k, які можуть перекриватися.

Можна розглядати ці свідчення, як елементи множини Ψ і побудувати відображення

$$\Gamma : \Psi \to 2^{\Theta}, \tag{3.1}$$

що буде зв'язувати кожен елемент в Ψ із підмножиною простору Θ. Така підмножина називається *фокальним елементом*. Зазначимо, що припущення про вичерпну повноту набору гіпотез означає, що жоден з елементів $\psi \in \Psi$ не відображається на порожню множину. Іншими словами, для будь-якого свідчення існує хоча б одна гіпотеза, вірогідність якої підтверджує це свідчення.

Теорія Демпстера-Шефера пропонує засоби обчислення функції довіри на таких множинах гіпотез і правила об'єднання функцій довіри, сформульованих на підставі різних свідчень.

3.1 Функції довіри

У теорії Демпстера-Шефера m – це функція *присвоєння базових ймовірностей (bpa – basic probability assignment)*, що визначена на множині 2^{Θ} значень з інтервалу [0,1], така, що $m(\varnothing)=0$ і $\sum(m(A_i))=1$; підсумовування виконується за всіма $A_i \in 2^{\Theta}$.

Сумарна довіра *Bel* для будь-якого фокального елемента A може бути знайдена підсумовуванням значень m по всіх підмножинах в A. Таким чином, *Bel* є функцією, визначеної на множині 2^{Θ} значень з інтервалу [0,1], такий, що

$$Bel(A) = \sum_{B \subset A} m(B). \tag{3.2}$$

$Bel(\Theta)$ завжди дорівнює 1, незалежно від значення $m(\Theta)$. Це випливає з визначення функції присвоєння базових ймовірностей. Співвідношення $Bel(\Theta)=1$ означає таке: можна з повною впевненістю стверджувати, що в просторі Θ обов'язково присутня коректна гіпотеза, оскільки за визначе-

нням набір гіпотез є вичерпним. Значення $m(\Theta)$ відображає вагу свідчення, ще не врахованого в підмножинах, що входять у простір Θ. Значення *Bel* і *m* будуть рівні для множин, які містять єдиний елемент.

Оцінка ймовірності фокального елемента *A* буде обмежена знизу оцінкою довіри до *A*, а зверху – *оцінкою привабливості A*, що дорівнює $1-Bel(A^c)$, де A^c – доповнення до *A*.

Оцінка привабливості A, Pls(A), являє собою ступінь сумісності свідчення з гіпотезами в *A* і може бути обчислена за такою формулою:

$$Pls(A)=\sum_{A\cap B\neq\emptyset} m(B). \tag{3.3}$$

Оскільки визначена в такий спосіб оцінка привабливості *A* є не що інше, як міра нашої недовіри до $\neg A$, то можна записати, що:

$$Pls(A)=1-Bel(\neg A). \tag{3.4}$$

Значення оцінки привабливості *A* можна розглядати як межу, до якої можна поліпшити гіпотези з *A* за наявності свідчень на користь гіпотез-конкурентів. Зручно розглядати інформацію, що міститься в оцінці *Bel* для цієї підмножини, у вигляді *довірчого інтервалу* у вигляді *[Bel(A), Pls(A)]*. Ширина інтервалу може слугувати оцінкою непевності в справедливості гіпотез з *A* при наявному наборі свідчень.

Правила Демпстера дозволяють обчислити нове значення функції довіри за двома її значеннями, що базується на різних спостереженнях. Позначимо Bel_1 і Bel_2, два значення функції довіри, яким відповідають два значення функції присвоєння базових ймовірностей, m_1 і m_2. Правило дозволяє обчислити нове значення $m_1 \oplus m_2$, а потім і нове значення функції довіри $Bel_1 \oplus Bel_2$, ґрунтуючись на визначеннях, приведених вище.

Для гіпотези *A* значення $m_1 \oplus m_2(A)$ є сумою всіх добутків у вигляді $m_1(X)m_2(Y)$, де *X* і *Y* поширюються на всі підмножини в Θ, перетином яких є *A*. Якщо в таблиці перетинів буде виявлений порожній елемент, виконується нормалізація. У процедурі нормалізації значення *k* визначається як сума всіх ненульових значень, присвоєних у множині Θ, потім $m_1 \oplus m_2(\emptyset)$ присвоюється значення нуль, а значення $m_1 \oplus m_2$ для всіх інших множин гіпотез поділяється на $(1-k)$.

Таким чином,

$$m_1 \oplus m_2(A)=\frac{\sum_{X\cap Y=A} m_1(X)m_2(Y)}{1-\sum_{X\cap Y=\emptyset} m_1(X)m_2(Y)}. \tag{3.5}$$

Потрібно враховувати, що значення m_1 і m_2 сформовані за незалежними джерелами свідчень у межах того ж простору гіпотез. Зверніть увагу і на

той факт, що внаслідок комутативності операції множення правило Демпстера дає той самий результат при будь-якому порядку об'єднання свідчень.

3.2 Приклад застосування теорії Демпстера-Шефера в експертних системах

Гордон (Gordon) і Шортліфф (Shortliffe) запропонували використовувати теорію Демпстера-Шефера як альтернативу операціям з коефіцієнтами впевненості, застосовуваним в ЕС MYCIN. Вони звернули увагу на те, що при визначенні організмів ЕС MYCIN часто звужує множину розглянутих гіпотез до визначеної підмножини, включаючи в неї, наприклад, тільки грамовід'ємні мікроорганізми (це приклад того, що Кленсі (Clancey) назвав застосуванням структурних знань). Правила, що породжують таке звуження простору гіпотез, нічого не говорять про відносну правдоподібність відібраних гіпотез.

При використанні баєсового підходу можна було б припустити, що відібрані гіпотези про шукані мікроорганізми мають рівні апріорні ймовірності, і, отже, рівномірно розподілити між цими гіпотезами ваги свідчень. Але це може призвести до того, що система не буде здатна відрізнити випадки, коли є однакові свідчення на користь кожної гіпотези, від випадків, коли такі свідчення відсутні зовсім. *Функція присвоєння базових ймовірностей* у теорії Демпстера-Шефера не робить розходження між апріорними й апостеріорними ймовірностями, а тому й не приводить до такого розподілу ймовірностей.

Функції довіри в теорії Демпстера-Шефера дозволяють також уникнути й іншого наслідку застосування баєсового підходу, що суперечить нашій інтуїції. При баєсовому підході суб'єктивна інтерпретація ймовірностей означає, що, довіряючи деякою мірою гіпотезі H, ми тим самим змінюємо ступінь довіри до інших гіпотез, тобто

$$P(H) = 1 - P(\neg H). \tag{3.6}$$

Однак одним із слабких місць моделі підтвердження, яка використовує коефіцієнти впевненості у ЕС MYCIN, є те, що свідчення, яке частково підтверджує визначену гіпотезу, не може розглядатися одночасно і як свідчення, що спростовує цю гіпотезу. У теорії Демпстера-Шефера зміна ступеня довіри до підмножини гіпотез A не змушує до зміни ступеня довіри до інших гіпотез, оскільки

$$Bel(A) + Bel(A^c) \le 1. \tag{3.7}$$

Залишок після підсумовування ступенів довіри до A та A^c – це ступінь ігнорування гіпотези A.

Гордон і Шортліфф показали також, яким чином можна застосувати теорію Демпстера-Шефера в MYCIN для висновку суджень про гіпотези на підставі свідчень, що надійшли. Тріада (об'єкт – атрибут – значення), включена в праву частину правил, являє собою в кожному з них єдину гіпотезу (тобто множина гіпотез, що складається з єдиного елемента), "відповідальну" за це значення визначеного атрибута у визначеному об'єкті. Отже, будь-яка множина таких тріад, що мають ті ж самі об'єкти і ті ж самі атрибути, наприклад (ORGANISM-1 IDENTITY *<значення>*), створює простір гіпотез у тому сенсі, як це трактується в теорії Демпстера-Шефера. Якщо параметр має єдине значення, то умова взаємної виключності гіпотез не порушується. Набір значень у правилах також має бути вичерпним.

Таким чином, правила в системі повинні бути побудовані як свого роду опис функцій довіри в теорії Демпстера-Шефера. Якщо посилання в правилі підтверджує висновок про гіпотезу H зі ступенем d і якщо d має значення, що перевищує визначений поріг активізації правила, то значення коефіцієнта впевненості, пов'язаного з цією гіпотезою H, можна розглядати як функцію присвоєння базових ймовірностей, яка присвоює значення d множині $\{H\}$, що складається з однієї гіпотези. А значення $1\text{-}d$ – простору Θ. Якщо ж посилання спростовує гіпотезу зі ступенем упевненості d, то ми привласнюємо значення d множині $\{H\}^c$, значення $1\text{-}d$ – простору Θ, а значення коефіцієнта впевненості, пов'язаного з цією гіпотезою H, буде дорівнювати $-d$.

Виділено три варіанти комбінування свідчень у результаті виконання правил при використанні моделі Демпстера-Шефера.

1. Обидва правила або підтверджують, або спростовують один і той самий висновок $\{H\}$, причому правила характеризуються базовими ймовірностями m_1 і m_2. У цьому випадку деяка вага свідчення буде розподілена між $\{H\}$ і Θ. Обновлені значення довіри для цих двох множин будуть мати вигляд $m_1 \oplus m_2(\{H\})$ та $m_1 \oplus m_2(\Theta)$. При цьому немає необхідності застосовувати k-нормалізацію, оскільки $\{H\} \cap \Theta \neq \varnothing$. Виявляється, що в цьому випадку теорія Демпстера-Шефера дає той самий результат, що і метод обробки коефіцієнтів упевненості.

2. Одне правило підтверджує гіпотезу $\{H\}$ зі ступенем m_1, а інше правило її спростовує зі ступенем m_2, тобто підтверджує $\{H\}^c$. У цьому випадку необхідна нормалізація, оскільки $\{H\} \cap \{H\}^c = \varnothing$. Інакше значення ймовірностей будуть комбінуватися, як і раніше: $m_1 \oplus m_2(\{H\})$, $m_1 \oplus m_2(\{H\}^c)$ і $m_1 \oplus m_2(\Theta)$. У цьому випадку результати відрізняються від отриманих при використанні коефіцієнтів упевненості. Якщо застосувати правило Демпстера, то виявляється, що таке суперечливе свідчення призводить до зни-

ження підтримки та гіпотези $\{H\}$ і її опонентів $\{H\}^c$, а росте довіра до Θ. (У результаті появи суперечливого свідчення для кожної з множин гіпотез збільшується оцінка *привабливості Pls*, оскільки підтримка опонента знижується. Цей результат не узгодиться з інтуїтивним уявленням про привабливість. Але потрібно зазначити, що в теорії Демпстера-Шефера цей термін має дещо відмінний від звичайного зміст). Застосування тих функцій комбінування коефіцієнтів упевненості, які використовуються з MYCIN, позначиться тільки на тій гіпотезі, що характеризується великим значенням коефіцієнта впевненості.

3. Правила виносять висновки, що стосуються двох конкуруючих гіпотез $\{H_1\}$ і $\{H_2\}$, тобто двох множин, кожна з яких містить тільки по одному елементу. Якщо $\{H_1\} \cap \{H_2\} = \emptyset$, то буде потрібна нормалізація і потрібно буде обчислити значення оцінок $m_1 \oplus m_2(\{H_1\})$, $m_1 \oplus m_2(\{H_2\})$ і $m_1 \oplus m_2(\Theta)$. Правило Демпстера і у цьому випадку виявляється більш загальним, чим функції комбінування коефіцієнтів впевненості в MYCIN. Це проявляється в тому, що якщо між $\{H_1\}$ і $\{H_2\}$ існує відношення підмножини, то довіра до підмножини буде розцінюватися як довіра до супермножини, але не навпаки. Таким чином, при використанні моделі Демпстера-Шефера поява нового свідчення впливає більше, ніж при використанні колишньої моделі, заснованої на коефіцієнтах упевненості.

Гордон і Шортліфф запропонували наближені методи обчислень, що дозволяють знизити обсяг обчислювальних операцій порівняно з оригінальною теорією Демпстера-Шефера. Вони також звернули увагу на те, що поділ простору пошуку, схожий до виконаного у системі INTERNIST, допоможе виділити досить малу множину конкуруючих гіпотез, що утворять поточну область аналізу. Однак в таких системах, як INTERNIST, при формуванні множини конкуруючих гіпотез неможливо виконати пряме відображення вигляду $\Gamma : \Psi \to 2^{\Theta}$ між окремими свідченнями і множинами гіпотез, вважаючи, що симптоми можуть бути причетні до розділення множин гіпотез на рівні ієрархії.

За останні десять років популярність теорії Демпстера-Шефера неухильно зростає. Вона знаходить застосування в різних сферах діяльності, наприклад при вирішенні задач діагностування [32] і машинного зору [33]. Хоча ця теорія і не дозволяє вирішити проблему умовної залежності, вона надає інженеру зі знань визначену гнучкість у тому, що можна встановлювати ступінь довіри до підмножин у просторі гіпотез, що складаються з більш, ніж одного елемента. Таке встановлення може слугувати засобом кодування залежностей між групами свідчень. Ієрархічна організація областей розпізнавання сприяє спрощенню такої технології обробки.

3.3 Методика Перла

Альтернативою теорії Демпстера-Шефера є методика Перла [34], в якій свідчення враховуються на основі баєсового підходу до групування і поширення впливу свідчень на достовірність гіпотез. Як і в методиці, запропонованій Гордоном (Gordon) і Шортліффом (Shortliffe), передбачається, що в просторі гіпотез виділено деяку підмножину гіпотез, що являють собою інтерес у визначеному семантичному контексті, причому ця підмножина має ієрархічну структуру.

Також передбачається, що ще до отримання свідчень з кожною окремою гіпотезою пов'язано визначене значення ступеня довіри до її правдоподібності. Перл не уточнює, яким саме способом формуються ці вихідні значення, але скоріш за все це повинен зробити експерт в предметній області при формулюванні гіпотез.

Від експерта також потрібно виділити множину гіпотез S, на які безпосередньо поширюється певна множна свідчень E. Якщо свідоцтва з E безпосередньо впливають на гіпотези з S, то повинен існувати якийсь причинний механізм, що зв'язує кожен член множини S зі свідченнями, причому він є унікальним для кожного з них. Однак самі по собі свідчення в множині E не несуть ніякої інформації, яка дозволила б нам віддати перевагу одному з членів S перед іншими.

Це відображення множин один на одного дозволяє ввести поняття умовної незалежності між свідченнями і окремими гіпотезами h_i:

$$P(E \mid S, h_i) = P(E \mid S, h_i), \text{ для всіх } h_i \in S. \tag{3.8}$$

За допомогою відношення ймовірностей можна кількісно оцінити ступінь, з якою свідчення підтверджують або спростовують множину гіпотез S:

$$\lambda_S = \frac{P(E \mid S)}{P(E \mid \neg S)}. \tag{3.9}$$

Вплив свідчень E на множину S обчислюється таким чином. Кожна окрема гіпотеза h_i, що належить множині S, отримує вагу $W_i=\lambda_S$, тоді як кожна гіпотеза з доповнюючої множини S отримує вагу $W_i=1$. Все це виконується на фазі розподілу ваг.

Потім, коли настає фаза оновлення, обчислюється нове значення функції довіри $BEL'(h_i)$ за її колишнім значенням $BEL(h_i)$:

$$BEL'(h_i) = P(h_i \mid E) = \alpha_S W_i BEL(h_i), \tag{3.10}$$

де α_S – коефіцієнт нормалізації, заданий співвідношенням

$$\alpha_S = (\sum_i W_i BEL'(h_i))^{-1}. \qquad (3.11)$$

Таким чином, ступінь довіри, призначена множині гіпотез, розподіляється між членами цієї множини як функція їх апріорних ймовірностей. Водночас ступінь довіри, призначений групі гіпотез, є сумою відповідних показників елементів цієї групи. Оновлення значень показників довіри може виконуватися рекурсивно, тобто апостеріорні оцінки, отримані на підставі одних свідчень, можуть використовуватися як апріорні оцінки для наступного циклу оновлення при отриманні нових свідчень.

Вся схема обчислень заснована на припущенні про умовну незалежність і дотриманні симетричності множиною S^C, доповнюючим S. Зі співвідношення $P(E \mid S^C, h_i) = P(E \mid S^C, h_i)$, для всіх $h_i \in S^C$ випливає, що $P(E \mid h_i) = P(E \mid S)$, якщо $h_i \in S$, інакше $P(E \mid S^C)$.

З цього співвідношення і правила Баєса випливає, що $P(h_i \mid S) = \alpha_S \lambda_S P(h_i)$, якщо $h_i \in S$, інакше $\alpha_S P(h_i)$.

Але, хоча Перл використовує формалізм Баєса, часткове свідчення на користь якоїсь гіпотези не може бути витлумачено як часткова підтримка заперечення цієї гіпотези. Свідчення на користь підмножини гіпотез S не може бути витлумачено як свідчення на користь доповнення до цієї підмножини S^C.

Розподіл свідчень на користь підмножини між окремими гіпотезами відновлює точковий розподіл ймовірностей на просторі гіпотез, але це відбувається за рахунок точності оцінок для окремих гіпотез. Перл стверджує, що немає необхідності розподіляти загальний показник, взятий для всієї підмножини S, на його елементи доти, доки не будуть отримані додаткові свідчення (або всі можливі). Нормалізацію також можна відкласти доти, доки отримані свідчення не підштовхнуть систему до виділення певних гіпотез (можливо, різних). Наприклад, якщо отримані свідчення E_1, ..., E_n відповідно на користь гіпотез S_1, ..., S_n, то ваги будуть комбінуватися мультиплікативно

$$W_i(E_1,...,E_n) = W_{1,i} W_{2,i} ... W_{n,i}, \qquad (3.12)$$

де $W_{k,i} = \lambda_{Sk}$, якщо $h_i \in S$, інакше 1.

Перл запропонував також і альтернативний механізм оновлення, який дозволяє уникнути нормалізації та включає поширення перегляду параметрів гіпотез як вверх, так і вниз по ієрархічній структурі за допомогою передачі повідомлень. З точки зору практичної реалізації цей механізм здається більш привабливим, ніж правило Демпстера. Перл стверджує, що метод розповсюдження, заснований на передачі повідомлень, досить прозорий, оскільки шляхи впливу мають семантичне обґрунтування. Відмова від глобальної нормалізації дозволяє краще зрозуміти

результати на проміжних етапах розповсюдження. Залишається тільки один числовий параметр – відношення ймовірностей, – сенс якого достатньо зрозумілий.

3.4 Контрольні питання

1. Що передбачається щодо гіпотез та їх наборів у теорії Демпстера-Шефера?
2. Що таке область аналізу в теорії Демпстера-Шефера?
3. Що таке фокальний елемент?
4. Чи існує в теорії Демпстера-Шефера хоча б одне свідчення, яке не підтверджує жодна з гіпотез?
5. Що таке функція m в теорії Демпстера-Шефера?
6. На якому інтервалі і на якій множині визначена функція m?
7. Чому дорівнює сумарна довіра Bel для будь-якого фокального елемента A?
8. На якому інтервалі та на якій множині визначена функція Bel?
9. Чому дорівнює $Bel(\Theta)$ і чи залежить це від $m(\Theta)$?
10. Що означає співвідношення $Bel(\Theta)=1$?
11. Що відображає значення $m(\Theta)$?
12. Що таке оцінка привабливості?
13. Що дозволяють обчислити правила Демпстера?
14. Назвіть переваги теорії Демпстера-Шефера порівняно з Баєсовим підходом.
15. Що запропонували Гордон і Шортліфф для зменшення обсягу обчислень та виділення досить малої множини конкуруючих гіпотез порівняно з оригінальною теорією Демпстера-Шефера?
16. Які можливості надає інженеру зі знань теорія Демпстера-Шефера?
17. В чому перевага ієрархічної організації областей розпізнавання?
18. Що є альтернативою теорії Демпстера-Шефера?
19. Що вимагається від експерта при формулюванні гіпотез?
20. Як обчислюється вплив свідчень E на множину S?

4 БАЄСОВІ МЕРЕЖІ ДОВІРИ

У роботі [35] описаний формалізм, якому автор присвоїв назву мережі Баєса (далі, баєсова мережа довіри – БМД). Цей механізм можна розглядати як узагальнення описаних у цьому розділі ієрархічних мереж довіри. У БМД дуги між вузлами також являють собою причинні залежності, але допускається ситуація, коли деякі вузли мають множину батьківських елементів, причому структура мережі може містити петлі. Відновлення оцінок довіри виконується за допомогою передачі повідомлень, як і у випадку строгої ієрархічної організації, хоча дія цього механізму очевидна тільки для полідерев, тобто мереж, у яких між будь-якими двома вузлами існує єдиний шлях.

Викликає інтерес порівняння формалізму Перла і теорії Демпстера-Шефера.

У формалізмі Перла потрібно привласнювати апріорні оцінки довіри окремим подіям, а в теорії Демпстера-Шефера оцінка поширюється на всю область аналізу.

У формалізмі Перла визначення функції $Bel(h_i)$ через $P(h_i)$ і $Bel'(h_i)$ через $P(h_i|E)$ дозволяє більш коректно обґрунтувати ці функції на основі виведень теорії ймовірностей, чого не можна сказати про правила комбінування Демпстера [36].

Учений Йєн [37] звернув увагу на те, що у формалізмі Перла губиться поняття *довірчого інтервалу*, всередині якого можуть змінюватися ймовірнісні оцінки. Довірчі інтервали дуже зручно використовувати в експертних системах, оскільки вони дозволяють аналізувати "якість" гіпотез, можливості їхнього удосконалювання й асоційованого ступеня невизначеності.

У своїй книзі [35] Перл зовсім справедливо відзначає, що теорія Демпстера-Шефера заснована на неповній ймовірнісній моделі, а тому може дати тільки часткові відповіді. Замість того, щоб безпосередньо оцінити, наскільки близька гіпотеза до того, щоб її можна було вважати істинною. Теорія Демпстера-Шефера обґрунтовує, наскільки отримане свідчення повинно підштовхнути нас до переконання, що ця гіпотеза істинна. У цьому відношенні теорія Демпстера-Шефера значно більше нагадує об'єктивістські методи перевірки значимості з використанням довірчих інтервалів, чим суб'єктивістські методи на основі баєсового підходу [38].

Але, незважаючи на відмічені розбіжності, в обох підходах є багато спільного. Асоціювання свідчення із підмножинами гіпотез у рамках формалізму Перла не суперечить відображенню однієї множини на іншу в теорії Демпстера-Шефера. Обидва варіанти можна розглядати як використання метафори "масового розподілу" в тому значенні, що основна увага приділяється розподілу отриманих свідчень у контексті структурованого

простору альтернатив, причому обидва методи дозволяють обчислювати значення функції довіри на основі простих ймовірнісних оцінок.

4.1 Основні терміни математичного апарату баєсових мереж довіри

БМД – це компактне подання спільного розподілу ймовірностей для множини змінних X. БМД складається з таких елементів [39]:

– спрямованого ациклічного графа G, кожний вузол якого відповідає випадковій змінній $X_i \in X$;

– множини умовних ймовірнісних розподілів, по одному для кожного вузла графа G.

Для БМД виконується марківська умова: кожний вузол (змінна) не залежить від усіх інших вузлів (змінних), за винятком своїх батьківських елементів $Par(X_i)$, цю залежність параметризує множина умовних розподілів ймовірностей. Для МБ визначене мережеве правило: якщо МБ має n вузлів, які являють собою n випадкових змінних $X_1, \ldots, X_n$, то:

$$P(X_i, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Par(X_i)). \quad (4.1)$$

Гібридна БМД $B = (X, D, P)$ визначається через спрямований ациклічний граф $G = (X, E)$ і його функції:

$$P = \{P(x_i \mid Par(x_i))\}, \quad (4.2)$$

де $Par(x_i)$ – набір батьківських вузлів x_i;

X – це множина змінних, які діляться на дискретні Δi неперервні Γ, тобто $X = \Gamma \cup \Delta$.

Структура графа G обмежена тим, що неперервні змінні не можуть мати вузли-нащадки дискретних змінних. Умовний розподіл неперервних змінних задається лінійною гаусівською моделлю:

$$P(x_i \mid I = i, Z = z) = N(\alpha(i) + \beta(i) \times z, y(i)) x \in \Gamma, \quad (4.3)$$

де Z та I – множини неперервних і дискретних батьківських елементів x_i, відповідно, а $N(\mu, \sigma)$ – багатовимірний нормальний розподіл. Мережа являє собою спільний розподіл усіх його змінних, заданих добутком усіх таблиць умовних ймовірностей.

Умовний ймовірнісний розподіл X_i називають ***лінійним умовним ймовірнісним розподілом***, який після виключення усіх нульових β_{ij} має вигляд [39]:

$$P(X_i \mid X_1, \ldots, X_{i-1}) = N(X_i; \beta_{i,0} + \sum_{j=1}^{i=1} \beta_{i,j} X_j, \sigma_i^2). \quad (4.4)$$

Лінійний умовний ймовірнісний розподіл для кореневих вузлів є просто одновимірним гаусіаном. БМД, в якій усі умовні ймовірнісні розподіли є лінійними, називається ***лінійною гаусівською*** (ЛГ). Кожний багатовимірний гаусіан може бути поданий лінійним гаусіаном і навпаки – кожна БМД з лінійним умовним ймовірнісним розподілом являє собою нормальний спільний розподіл [39].

Для БМД існують точні і наближені алгоритми формування висновку. Точні методи надають точний результат, а за допомогою наближених намагаються отримати найбільше наближення. Точні методи застосовують для мереж, що належать до класу об'єднаних між собою мереж, також відомих як полідерева. Мережа належить до цього класу, якщо основний ненаправлений граф має не більше одного шляху між двома вузлами. Коли мережа має багато з'єднань між вузлами, для перетворення її у полідерево можна використати метод кластеризації, а потім сформувати точний висновок.

Наближені методи формування логічного висновку для звичайних БМД

Для практичних задач часто виникає проблема побудови і застосування мереж великого розміру, для яких формування точного висновку є неможливим. У такому випадку доцільно застосовувати наближені методи формування висновку. Більшість наближених методів для МБ містять: спрощення моделі, випадковий відбір проб, розповсюдження довіри за наявності петель.

Методи спрощення моделі виконують її спрощення до тих пір, поки не стає можливим застосування точного алгоритму. Деякі поширені методи спрощення містять видалення слабкої залежності або дуг, дискетизацію неперервних вузлів, лінеаризацію нелінійних відношень, абстракцію простору станів тощо.

Стохастичний відбір проб, також відомий, як метод Монте-Карло, є найбільш відомим методом формування наближеного висновку і широко використовується на практиці. Зазвичай, він використовується для усіх типів БМД. Однак, алгоритми відбору проб можуть займати досить багато часу для наближення до надійних результатів. Особливі складнощі виникають у випадку малоймовірних свідчень, коли алгоритми не дають результатів навіть при великих обсягах зразків. По суті, алгоритми стохастичного відбору проб спочатку генерують випадкові зразки відповідно до деяких попередньо вибраних розподілів, а потім наближують апостеріорні розподіли послідовності вузлів частотою появи зразків. Точність висновку залежить від обсягу зразків і вибраних розподілів, які використовувалися для відбору, але може бути незалежною від структури мереж і умовних ймовірнісних розподілів змінних. Стохастичний відбір проб поділяють на два види: вибірку за значущістю і методи Монте-Карло для марківських мереж (МКММ).

Перший алгоритм відбору проб запропоновано в 1988 р., його називають ***логічним відбором проб*** [39]. Цей алгоритм використовує простий метод генерування проб наперед (що відповідає нормальному впорядкуванню у спрямованому графі) відповідно до апріорного розподілу мережі і відкидає зразки, які не узгоджені зі свідченням. Якщо раніше створений екземпляр вузла відрізняється від значення, яке спостерігається, то вхідний зразок відкидається. Алгоритм дуже неефективний для малоймовірних свідчень і його неможливо використати для свідчень у вигляді неперервних змінних, оскільки ймовірність генерування однакових неперервних змінних дорівнює нулю. Покращеним варіантом є ***метод зваженої правдоподібності***, запропонований для подолання проблем, які виникають при використанні логічного відбору проб. Цей метод не вибирає вузли, які вже спостерігалися, а замість цього використовує значення вузла, що спостерігався, і оцінює зразок за вірогідністю умовного свідчення для зразка. Цей метод працює значно краще логічного відбору проб і може застосовуватися до дуже великих і складних мереж. Потрібно зауважити, що швидкість збіжності цього методу досить мала для малоймовірних свідчень, оскільки метод також використовує апріорний розподіл для генерування зразків. Якщо свідчення малоймовірне, метод зваженої правдоподібності навряд чи має представницькі зразки невідомих апостеріорних розподілів заданого свідчення, оскільки вага випадкового зразка дуже мала.

Причина невдалого застосування методів логічного відбору проб і зваженої правдоподібності для малоймовірних свідчень полягає в тому, що вони використовують саме апріорний алгоритм генерування проб. Відомо, що виконання методів відбору проб залежить не лише від розміру зразка, але, що найважливіше, від типу вибіркового розподілу. Замість використання апріорної функції використовують функцію "значущості" для відбору проб. Необхідно відзначити, чим ближчою є функція значущості до реального невідомого розподілу, тим ефективнішим і точнішим буде алгоритм відбору проб.

Метод відбору проб. Ідея методу відбору проб полягає у випадковому присвоєнні значень випадковим змінним (ці присвоєння називають зразками) і оцінюванні властивостей спільного розподілу з використанням цих зразків.

Формально, вводиться припущення, що існує деякий спільний розподіл $P(X)$ і необхідно оцінити математичне сподівання деякої функції f цього розподілу, тобто необхідно знайти оцінку $E_p\,[f(X)]$. Необхідно зауважити, що таке формулювання є дуже загальним. Зокрема, задаючи деяке свідчення $e=e_1, \ldots, e_n$, можна скоротити задачу обчислення ймовірності $P(e)$ для цієї форми. Для цього вводять індикаторну функцію $1_e(X)$ як $1_e(X) = \prod_i 1_{e_1}(X)$ (при цьому, $1_e(X)$ дорівнює 1, якщо X узгоджується з e_i і 0 – у протилежному випадку). Тепер можна записати, що:

$$P(e) = \sum_x 1_e(x) \cdot P(x) = E_p[1_e(X)]. \quad (4.5)$$

Для оцінювання сподівання $E[\ f(X)]$ необхідно визначити випадкову змінну $Y = f(X)$ і $E(Y) = E[f(X)]$. Далі роблять припущення, що $\sigma_Y^2 < \infty$, де $\sigma_Y^2 \stackrel{\text{def}}{=} Var(Y)$. А також припускають, що існує послідовність з N незалежних рівномірно розподілених зразків $x[1]$, $x[2]$, ..., $x[N]$ з розподілу $P(X)$ і $y[i] = f(x[i])$.

Використавши ці припущення, отримують:

$$\frac{1}{N}\sum_{i=1}^{N} y[i] = \frac{1}{N}\sum_{i=1}^{N} f(x[i]). \quad (4.6)$$

За центральною граничною теоремою для великих N сума має нормальний розподіл із середнім $E[f(X)]$ і дисперсією $\frac{\sigma_Y^2}{N}$. Тому сума збігається до $E[f(X)]$ і швидкість збіжності пропорційна $1/\sqrt{N}$. Іншими словами, можна дати відповідь на загальне ймовірнісне запитання для вибору рівномірно розподілених зразків із розподілу:

$$E[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x[i]). \quad (4.7)$$

4.2 Робота з баєсовими мережами довіри в середовищі Hugin Lite

Розглянемо приклад побудови найпростішої експертної системи (ЕС) на основі БМД, що дозволяє оцінювати стан плодових дерев.

Досвідчений садівник, який професійно володіє знаннями про фізіологію рослин, одного разу виявив, що його улюблене яблуневе дерево втратило листя. Садівник хоче з'ясувати, чому це трапилося. Будучи професіоналом у своїй галузі (експертом), він прагне змоделювати ситуацію шляхом побудови ЕС, яка знадобилася б і іншим менш досвідченим садівникам. Він знає, що ця ситуація може бути змодельована БМД, що містить три вершини подій: "Хворіє", "Засохло" та "Облетіло". При цьому кожна подія може приймати всього лише один із двох можливих станів (рис. 4.1):

- "Хворіє" – стан "хворіє" або "ні";
- "Засохло" – стан "засохло" або "ні";
- "Облетіло" – стан "облетіло" або "ні".

ЯКЩО "Дерево засихає при нестачі вологи"
АБО "Дерево хворіє"
ТО "Листя опадає"

БМД, наведена на рис. 4.1, моделює той факт, що існує причинно-наслідкова залежність від вершини "Хворіє" до вершини "Облетіло" і від вершини "Засохло" до вершини "Облетіло". Це відображено стрілками на БМД.

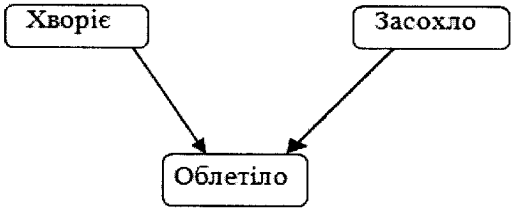

Рисунок 4.1 – Приклад БМД

Коли існує причинно-наслідкова залежність від вершини *A* до іншої вершини *B*, то очікується, що коли *A* перебуває в деякому певному стані – це впливає на стан *B*. Іноді зовсім не очевидно, який напрямок повинна мати стрілка в БМД. У розглянутому прикладі є залежність від стану "Хворіє" до стану "Облетіло", тому що коли дерево хворіє – це може викликати опадання його листя. Опадання листя є наслідком хвороби, а не хвороба – наслідком опадання листя.

На рис. 4.1 зображено графічне подання БМД. Однак, це тільки якісне подання. Перед тим, як повноцінно назвати це баєсовою мережею довіри, необхідно визначити кількісне подання, тобто множину таблиць апріорних і умовних ймовірностей (табл. 4.1).

При реалізації розглянутої БМД у середовищі HUGIN потрібно мати на увазі, що ця система не у всіх режимах підтримує роботу з шрифтами кирилиці. Тому, надалі, будемо користуватися позначеннями вершин англійською мовою.

Таблиця 4.1 – Апріорні й умовні ймовірності

P (Хворіє)	
Хворіє = "Хворіє"	Хворіє = "ні"
0.1	0.9

P (Засохло)	
Засохло = "Засохло"	Засохло = "ні"
0.1	0.9

P (Облетіло/ Хворіє, Засохло)				
	Засохло = "Засохло"		Засохло = "ні"	
Хворіє = "Хворіє"		Хворіє = "ні"	Хворіє = "Хворіє"	Хворіє = "ні"
Облетіло = "Облетіло"	0.95	0.85	0.9	0.02
Облетіло = "ні"	0.05	0.15	0.1	0.98

4.2.1 Основні прийоми роботи з системою HUGIN при побудові баєсових мереж довіри

Після запуску середовища Hugin Lite [40] відкриється вікно системи HUGIN, що містить панель меню, панель інструментів і панель графічного відображення мережі (рис. 4.2).

Після запуску система автоматично встановлюється в режим редагування елементів ЕС (edit mode – select tool), що дозволяє приступити до побудови нової БМД і визначення стану її вершин.

Іншим важливим режимом роботи системи є режим виконання (run mode – команда меню run), що дозволяє використовувати БМД для одержання необхідних результатів.

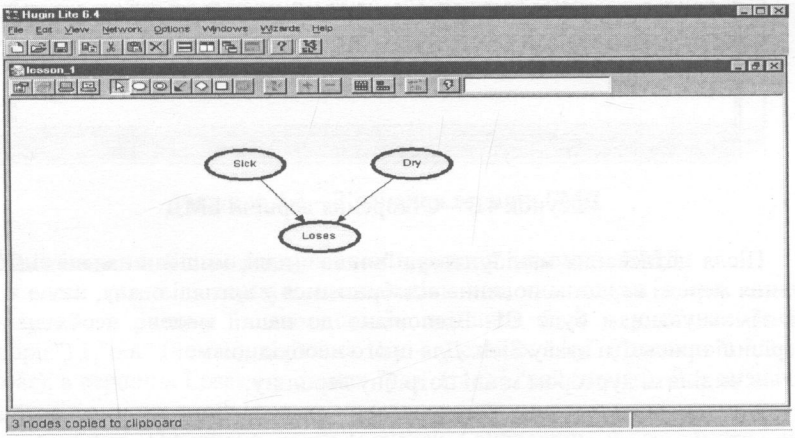

Рисунок 4.2 – Вікно програми в системі Hugin Lite

Додавання нових вершин у проектовану БМД

Побудова БМД починається з визначення окремих вершин, що входять у проектовану БМД. При цьому для створення вершин з дискретними станами, визначення їхніх властивостей і встановлення причинно-наслідкових зв'язків між вершинами використовуються наведені на рис. 4.2 інструменти інтерфейсу системи, які активуються шляхом натискання лівої кнопки миші (ЛКМ) на їхніх піктограмах.

Для побудови розглянутого прикладу БМД, перше, що необхідно зробити – створити вершину Sick ("Хворіє"). Для цього необхідно:

– установити режим додавання вершин з дискретними станами, вибравши відповідну піктограму в панелі інструментів (рис. 4.3);

– натиснути ЛКМ у будь-якому місці панелі відображення мережі, де передбачається розміщення вершини, яка додається.

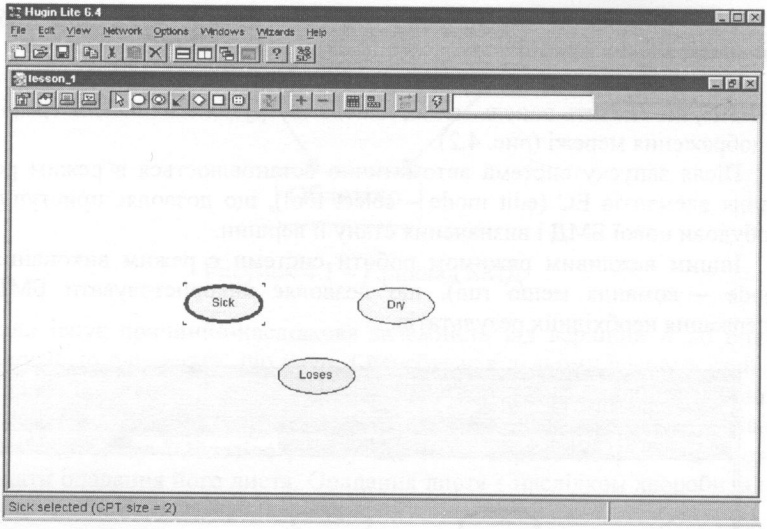

Рисунок 4.3 – Створення вершин БМД

Після натискання маніпулятора "миша" (далі, миші) в панелі відображення мережі вершина повинна відобразитися у вигляді овалу, назва якого за замовчуванням буде C1. Відповідно до нашої моделі, необхідно цій вершині присвоїти назву Sick. Для цього необхідно:

– виділити курсором миші потрібну вершину;

– установити режим визначення властивостей вершини [Node Properties], натиснувши на відповідній піктограмі панелі інструментів;

– змінити вміст полів Name і Label на Sick та натиснути "OK".

Встановлення причинно-наслідкових зв'язків між вершинами проектованої БМД

Проектована БМД має вигляд, наведений на рис. 4.4. Наступний етап проектування БМД полягає у встановленні причинно-наслідкових зв'язків між подіями. Ці зв'язки в моделі БМД відображаються у вигляді стрілок, що з'єднують між собою вершини БМД. Для додавання стрілок від вершини Sick до вершини Loses і від Dry до Loses необхідно:

– натиснути піктограму додавання зв'язків (рис. 4.3);

– протягнути мишею стрілку від Sick до Loses, натиснувши ліву кнопку маніпулятора "мишка" (далі, ЛКМ) й тримаючи натиснутою клавішу SHIFT;

– протягнути мишею стрілку від Dry до Loses, натиснувши ЛКМ.

Тепер маємо повне якісне подання, подібне до зображеного на рис. 4.4. Наступний крок – це встановлення станів і таблиць умовних ймовірностей для кожної вершини.

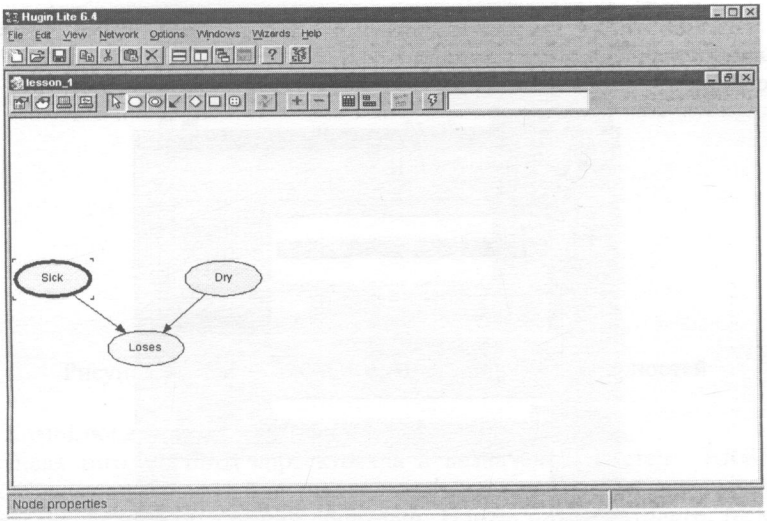

Рисунок 4.4 – З'єднання вершин БМД

Визначення всіх можливих станів кожної вершини БМД

Раніше кожна з вершин визначалась так, що кожна з них могла перебувати в одному із двох станів: вершина Sick – у станах "sick" ("хворіє") і "not" (немає), вершина Dry – у станах "dry" ("засохло") і "not" (немає), а вершина Loses – у станах "yes" (так) і "no" (немає). Для визначення станів вершин необхідно:

– зробити активною вершину Sick, вибравши її в списку, що розкривається на панелі інструментів, або двічі натиснувши на ній ЛКМ;

– у вікні, що з'явилося, для вершини Sick вибрати закладку States (рис. 4.5);

– перевести курсор на поле, що містить текст State 1 і ввести в нього текст "sick", задаючи один зі станів вершини, натиснувши кнопку Raname. Потім State 2 замінити на "not", задаючи другий стан вершини;

– натиснути кнопку OK.

Встановлення значень таблиць умовних ймовірностей кожної вершини БМД

Процес заповнення таблиці умовних ймовірностей розглядається на прикладі вершини Sick:

– вибираємо вершину Sick;

– вибираємо команду Network>>Run;

– у вікні редагування вершин, що з'явилося, натискаємо у прямокутнику із хрестиком, розкривши стан вершини;

– натискаємо ЛКМ на піктограмі Enter Likelihood evidence (рис. 4.5);

– у вікні Insert Likelihood, що з'явилося, установлюємо значення ймовірності 0,1 для стану вершини "Sick" і 0,9 для стану вершини "not" (рис. 4.6).

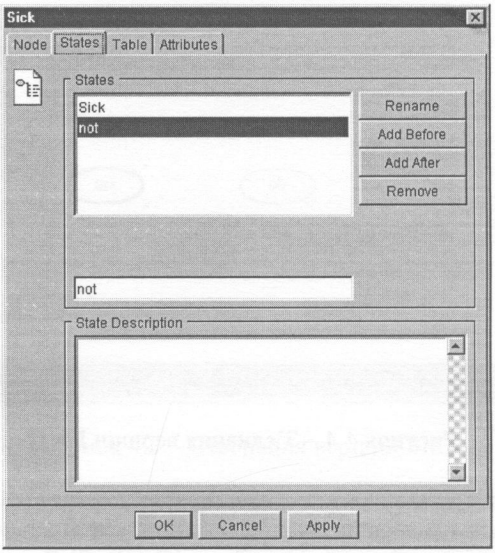

Рисунок 4.5 – Встановлення дискретних станів вершини Sick

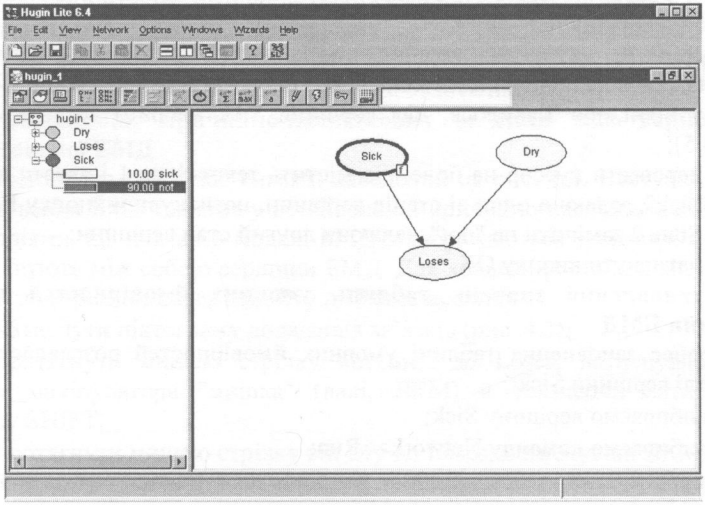

Рисунок 4.6 – Створення таблиці умовних ймовірностей для вершини Sick

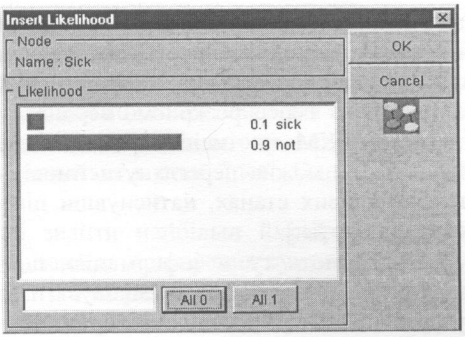

Рисунок 4.7 – Вікно встановлення умовних ймовірностей

Компілювання спроектованої БМД

Після того, як БМД спроектована й визначена в системі HUGIN, її необхідно скомпілювати й подивитися, як вона працює. Для цього необхідно перевести систему в режим обчислень. Із цією метою натисніть ЛКМ на піктограмі режиму обчислень [switch to run mode] у панелі інструментів. Компіляція створеної БМД пройде швидко й система перейде в режим обчислень ["run" mode].

У цьому режимі вікно мережі поділене на дві вертикальні секції (рис. 4.8). Ліва секція являє собою панель списку вершин [node list pane], а права – панель відображення мережі [network pane].

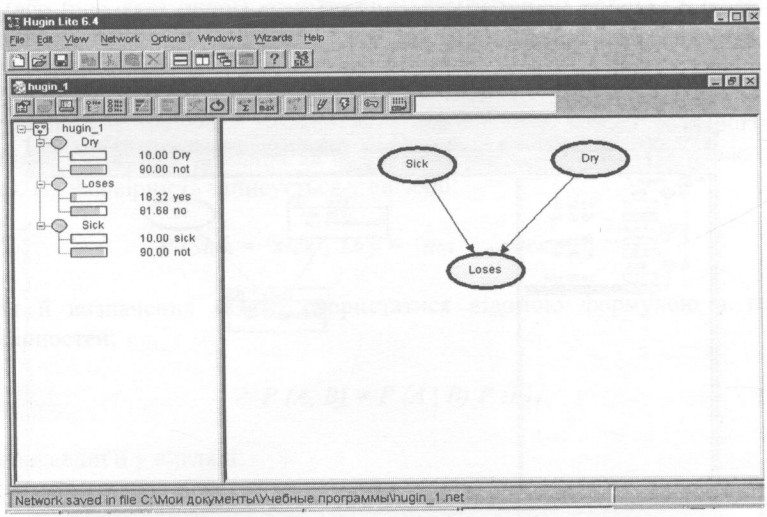

Рисунок 4.8 – Вигляд вікна в режимі обчислень

У лівій частині вікна відображаються вершини БМД, всі можливі стани кожної з вершин, а також ймовірності перебування вершини в кожному зі станів. Подивитися ці ймовірності можна, двічі натиснувши ЛКМ на імені вершини в списку вершин. Тепер розкриємо вершини Loses і Sick. Для цього двічі натискаємо ЛКМ на імені вершини Loses, а потім двічі натискаємо на Sick. Також можна переглянути ймовірності перебування всіх вершин у всіх можливих станах, натиснувши піктограму розкриття списку вершин [expand node list].

Вікна контролю показують ту ж інформацію, що й панель списку вершин, але з'являється можливість розташовувати ці вікна поруч із відповідними вузлами БМД у панелі відображення мережі. Можна відкрити таке вікно для будь-якої вершини, але найкращий варіант – це відкривати лише ті вікна, які становлять особливий інтерес. У протилежному випадку, ці вікна будуть займати занадто багато місця на екрані дисплея.

Відкриємо такі вікна для вершин Sick та Loses і повторимо розрахунок. Спочатку необхідно ініціалізувати БМД, тобто привести її до вихідного стану. Для цього натискаємо піктограму початкової ініціалізації у вигляді кільця зі стрілкою [initialize network].

Після того, як мережа ініціалізована, можна відкрити вікна контролю для вершин Sick і Loses. Із цією метою:

– вибираємо Sick і Loses (тримаємо натиснутою клавішу SHIFT для вибору більше ніж однієї вершини);

– вибираємо команду "Show Monitor Windows" (показувати вікна контролю) з пункту меню "View" (Вигляд).

Вигляд екрана показаний на рис. 4.9.

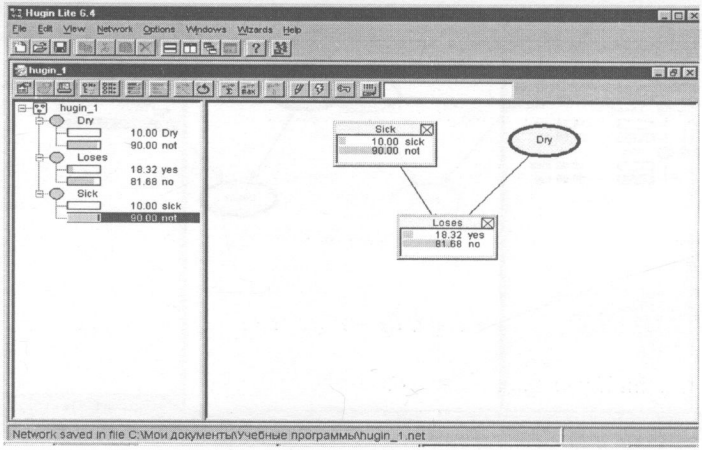

Рисунок 4.9 – Вигляд вікна в режимі контролю

Найбільш ймовірні комбінації станів

На основі аналізу факту про опадання листя, можна з ймовірністю 0.47 зробити припущення, що дерево засихає, і з ймовірністю 0.49, що дерево хворіє. При цьому, як для події Sick, так і для події Dry, більш ймовірними є стани "not". Це надає впевненості в тому, що найкраща комбінація станів, коли Sick і Dry знаходяться в стані "not". Однак, це неправильний висновок.

Для того, щоб знайти найбільш ймовірну комбінацію станів всіх вершин, необхідно, замість поширення сум, використовувати розповсюдження максимумів. Піктограма поширення максимумів [Max-propagate normal] знаходиться на панелі інструментів праворуч від піктограми поширення сум. При натисканні ЛКМ на цій піктограмі, можна отримати нове поширення ймовірностей у вікнах відображення мережі та вершин. При цьому кожна з станів вершин, що має значення 100.00, буде належати до найбільш вірогідної комбінації станів. Для розглянутого прикладу буде отримана одна унікальна комбінація, при якій найбільш вірогідним є те, що подія Sick (Хворіє) знаходиться в стані "sick", а подія Dry (Засохло) – в стані "not".

З отриманого результату випливає, що незважаючи на те, що стан Sick = "sick" менш ймовірний, ніж стан Sick = "not", саме цей стан Sick = "sick" входить в найбільш ймовірну комбінацію станів вершин БМД при надходженні факту про опадання листя, в той час як стан Sick = "not" в неї не входить. Після отримання висновку про найбільш вірогідну комбінацію станів вершин БМД, можна дізнатися ймовірності настання цієї (або будь-якої іншої) комбінації станів за умови надходження нових свідчень.

Розрахунок ймовірності комбінації станів

Розглянемо способи розрахунку ймовірності найбільш вірогідної комбінації станів, отриманих за наявності факту, що яблуня втрачає своє листя. Ця ймовірність записується у вигляді:

$$P\ (Sick = "sick",\ Dry = "not" \mid Loses = "yes"),$$

і для її визначення можна скористатися відомою формулою з теорії ймовірностей:

$$P\ (A,\ B) = P\ (A \mid B)\ P\ (B), \tag{4.8}$$

переписавши її у вигляді:

$$P\ (A \mid B) = P\ (A,\ B) / P\ (B), \tag{4.9}$$

де $P\ (A \mid B)$ – умовна ймовірність появи події A за появи події B;

P (A, B) – спільна ймовірність появи події *A* і *B*.

Кожного разу при виконанні розрахунку розповсюдження сум для БМД, в нижньому лівому кутку вікна HUGIN буде відображатися ймовірність (значення *P (All)*) одночасного настання кількох подій, яку можна записати таким чином:

$$P(A1, A2, ..., An).$$

Якщо замінити подію *A* на Sick = "yes", Dry = "not" і *B* на Loses = "yes", то вищенаведена формула для визначення умовної ймовірності може бути переписана у вигляді:

$$P(Sick = 'yes', Dry = 'not' | Loses = 'yes') = P(Sick = 'sick', Dry = 'not', Loses = 'yes') / P(Loses = 'yes') \quad (4.10)$$

Якщо вибрати стан "yes" для вершини Loses і зробити розрахунок поширення сум, то можна побачити ймовірність перебування вершини Loses в стані "yes". Її значення буде 0.1832, тобто *P* (Loses = "yes") = 0.1832. Після визначення значення ймовірності *P* (Loses = "yes") можна визначити значення ймовірності *P* (Sick = "sick", Dry = "not", Loses = "yes"). Це робиться таким чином:

– вводимо Sick = "sick", Dry = "not" і Loses = "yes" в БМД;

– виконуємо розрахунок поширення сум (натисніть іконку sum-propagate normal);

– визначаємо значення ймовірності *P* (Sick = "sick", Dry = "not", Loses = "yes") як значення *P (All)* в нижньому лівому кутку;

Це значення повинно дорівнювати 0.081. Тепер можна розрахувати необхідну ймовірність:

P (Sick = "yes", Dry = "not" | Loses = "yes") = 0.081 / 0.1832 = 0.442.

Таким чином, на основі факту про опадання листя, в ЕС зроблено висновок про те, що найбільш вірогідною є ситуація, пов'язана з тим, що дерево захворіло, а не засохло. При цьому ймовірність такої ситуації дорівнює 0.442.

4.2.2 Проектування діаграм впливу

Призначення та основні компоненти діаграм впливу

Основною метою є знайомство з основами проектування ЕС з використанням діаграм впливу (ДВ), методами їх реалізації в системі Hugin Lite і основними прийомами роботи з ними.

Діаграми впливу – це інструментарій розробки ЕС і систем підтримки прийняття рішень. Вони являють собою БМД, розширені поняттями користі (utility) та рішення (decisions). У них, крім вершин шансів, як в БМД, використовуються ще два типи вершин: вершини рішень, які

позначаються в ДВ прямокутниками, і вершини користі, що позначаються ромбами.

Вершини рішень, а точніше вказівки, що містяться в них, визначають старшинство у часі:

– стрілка від вершини шансів (випадкової змінної) (*V*) до вершини рішень (*D*) вказує, що значення випадкової змінної відомо на момент прийняття рішення;

– стрілка від вершини рішень до якої-небудь іншої вершини шансів *V* або вершини користі (*U*) вказує час, впорядковуючи рішення (рис. 4.10).

При цьому мережа повинна залишатися ациклічною і повинен існувати безпосередній шлях, що містить всі вершини рішень в мережі. В процесі прийняття рішення необхідно не просто знайти рішення, а знайти найкраще в якомусь сенсі рішення. З цією метою в ДВ вершини користі зв'язуються зі станом мережі, і кожна з них містить функцію корисності, яка пов'язує кожну конфігурацію стану її батьківських елементів з корисністю. Вершини корисності не мають спадкоємців, тобто стрілка може бути спрямована тільки до них.

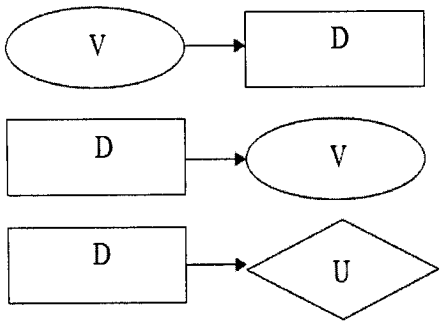

Рисунок 4.10 – БМД у вигляді діаграми впливу

Приклад побудови простої діаграми впливу

Після того як садівник оцінив стан своїх дерев, перед ним постає завдання прийняття рішення про доцільність вкладення матеріальних коштів на їх лікування. Для цього до вихідної БМД додаємо ще три вершини. Нові вершини: "Хворіє 1", "Всохло 1" і "Опало 1" точно такі ж, як і їхні аналоги в попередній моделі, але відображають майбутній час, наприклад, час збору врожаю. Нові вершини мають ті ж стани, що і старі (рис. 4.11).

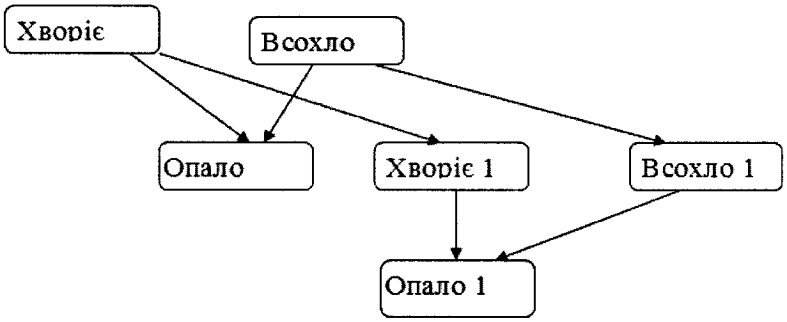

Рисунок 4.11 – Початкова БМД

У новій моделі відображаються залежності від "Хворіє" до "Хворіє 1" і від "Всохло" до "Всохло 1". Це пов'язано з тим, що якщо дерево хворіє зараз, то, ймовірно, воно буде хворіти і в майбутньому. Те ж можна сказати і про всихання дерева. Звичайно, сила залежності залежить від того, як далеко в майбутнє ми хочемо заглянути. Можна було б встановити залежність і від "Опало" до "Опало 1", але в цій моделі, для її спрощення, це робити не доцільно. Разом з тим, у садівника є можливість змінити ситуацію в майбутньому, якщо він вчасно вирішить проблему про причини опадання листя зі своїх плодових дерев, і, тим самим, зберегти врожай. Прикладами заходів, що попереджають хвороби дерев, є: оприскування дерев, очікування дощу та інше.

Дії, пов'язані з лікуванням дерева, можуть бути додані в модель у вигляді вершини рішень "Лікування", яка може мати два стани ("Лікування" = "так", "Лікування" = "ні"). При цьому, ця вершина повинна бути змодельована зі стрілкою до вершини "Хворіє 1", що викликано тим, що лікування вплине на майбутнє здоров'я дерева.

З додаванням вершини "Лікування" від моделі БМД (рис. 4.11) переходимо до діаграми впливу, в якій необхідно також визначити функцію корисності, яка дозволяє обчислити користь від прийняття рішення. Це здійснюється додаванням в діаграму впливу вершин корисності, кожна з яких визначає вклад в загальну вигідність. Діаграма впливу для розглянутого прикладу приведена на рис. 4.12.

Вершина "Витрати" містить інформацію про витрати на лікування, а вершина "Врожай" – інформацію про доходи від збору врожаю. При цьому, звичайно, кількість урожаю визначається здоров'ям дерев. Тому вершина "Врожай" залежить від стану вершини "Хворіє 1", вказуючи, що продукція залежить від здоров'я саду в момент збору врожаю.

Модель, подана на рис. 4.12, дає завершене якісне подання ДВ. Для отримання кількісного уявлення необхідно побудувати таблиці умовних

ймовірностей для кожної з вершин шансів і задати таблиці прибутковості для кожної з вершин корисності. Вершини прийняття рішення не мають таблиць умовних ймовірностей.

Рисунок 4.12 – Кінцеве подання ДВ

При цьому таблиці умовних ймовірностей для *P* ("Хворіє"), *P* ("Всохло") і *P* ("Опало" | "Хворіє", "Всохло") будуть мати той же вигляд, що і в попередньому прикладі. Таблиця умовних ймовірностей для *P* ("Опало 1" | "Хворіє 1","Всохло 1") буде аналогічна *P* ("Опало"|"Хворіє", "Всохло"). Таблиці умовних ймовірностей для решти станів повинні бути задані на основі знань експерта.

Мета побудови ДВ – прийняття рішення, пов'язаного з вершиною "Лікування" для того, щоб отримати найбільшу очікувану вигідність. Навіть у такому простому прикладі, ручний розрахунок досить складний і тому виникає необхідність у використанні інструменту для розробки ЕС, наприклад, такого, яким є ЕС Hugin Lite. Відповіддю для прийняття рішення про витрати на лікування буде обчислення загальної функції корисності за умови, що ймовірність *P* ("Опало" = "опало") = 1, а реалізована модель має вигляд, аналогічний наведеному на рис. 4.13.

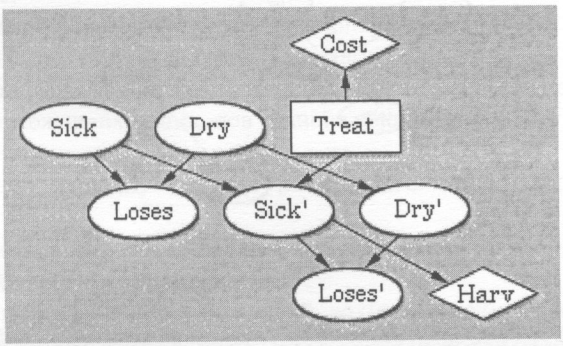

Рисунок 4.13 – Діаграма впливу для цього прикладу

Завантаження БМД для редагування. Копіювання вершин при розробці моделі БМД

У діаграмі впливу (рис. 4.13) є три вузли, аналогічні тим, які є в моделі. Можна скопіювати цю групу таким чином:

– розтягуємо прямокутник навколо всіх трьох вузлів (рухаємо мишу, утримуючи ЛКМ);

– вибираємо опцію "Copy" (копіювати) з пункту меню "Edit";

– потім вибираємо опцію "Paste" (вставити) у тому ж пункті "Edit";

– переміщуємо групу в потрібне місце.

Система Hugin згенерує нові імена і мітки для нових вершин. Можна зберегти імена або змінити їх. Однак, не можна використовувати імена Sick', Dry' або Loses', оскільки вони містять керуючий символ " ' ", який заборонено використовувати в іменах.

Мітки можуть використовувати довільний набір символів і їх можна змінити таким чином:

– виділяємо вершину курсором миші;

– відкриваємо вікно властивостей вершини (вузла). Для цього натискаємо в панелі інструментів кнопку "Node Properties" або вибираємо цей же режим з меню, що викликається натисканням правої кнопки миші;

– змінюємо значення поля "Label", наприклад, Sick_1;

– натискаємо "OK".

Після виконання цих дій для всіх трьох нових вершин, БМД буде виглядати так, як показано на рис. 4.14.

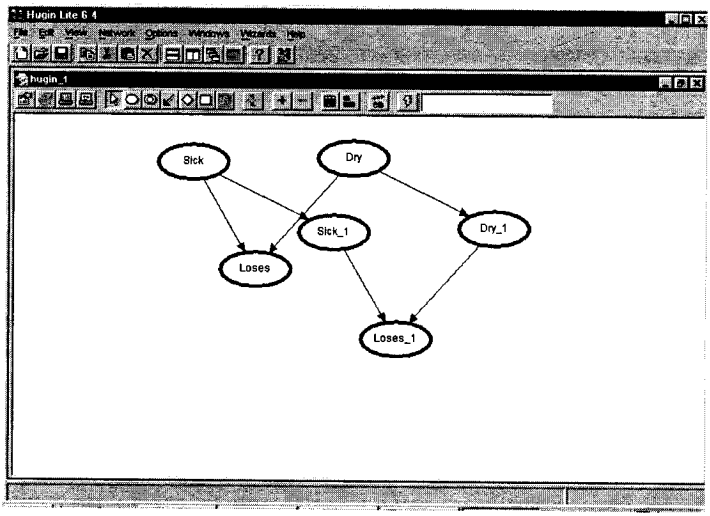

Рисунок 4.14 – Вигляд вікна HUGIN для діаграми впливу

На наступному кроці необхідно додати причинно-наслідкові зв'язки (стрілки) від Sick до Sick_1 і від Dry до Dry_1.

Це можна зробити таким чином (на прикладі вершин Sick і Sick_1):

– натискаємо ЛКМ в панелі інструментів кнопку з піктограмою стрілки ("Link Tool");

– продовжуємо стрілку від Sick до Sick_1, утримуючи ЛКМ у натиснутому стані.

Якщо при цьому утримувати клавішу Shift, то це дасть можливість створювати кілька зв'язків, не натискаючи кожного разу на кнопку "Link Tool" панелі інструментів.

4.2.3 Перехід від баєсової мережі довіри до діаграми впливу шляхом додавання вершини корисності

Натепер побудована мережа є БМД. Зробимо першу зміну, яка наблизить БМД до діаграми впливу. Ця зміна полягає в додаванні вершини корисності. Ця вершина буде характеризувати дохід (Корисність) від отриманого врожаю (вершина Harv на рис. 4.13). Для побудови цієї вершини необхідно виконати такі дії:

– натиснути кнопку "Utility Tool", розташовану в панелі інструментів праворуч від піктограми "Link tool";

– натиснути ЛКМ в тому місці робочої області, де передбачається розміщення вершини (краще праворуч від вершини Loses_1);

– змінити ім'я і мітку нової вершини на Harv.

Оскільки корисність безпосередньо визначається якістю врожаю, який у свою чергу залежить від стану здоров'я дерев до моменту збору врожаю, то необхідно встановити причинно-наслідковий зв'язок (додати стрілку) від Sick_1 до Harv.

Для цього:

– натискаємо в панелі інструментів кнопку "Link Tool";

– продовжуємо стрілку від Sick_1 до Harv.

На наступному етапі потрібно визначити значення корисності вершини Harv для всіх можливих станів, пов'язаної з нею вершини Sick_1.

З цією метою:

– виділяємо вершину Harv (з розкривного списку або подвійним натисканням ЛКМ);

– у таблицю визначення вершини Harv (команди View >> Open Table) вводимо значення з таблиці на рис. 4.15.

Ці значення визначають можливий дохід (корисність) від зібраного врожаю залежно від стану саду та можуть бути задані на основі експертних оцінок, наприклад, в грошовому еквіваленті. Вигляд вікна на цьому етапі показаний на рис. 4.16.

U (Harv)

Sick='sick'	Sick='not'
3000	20000

U (Harv)

Sick='sick'	Sick='not'
3000	20000

Рисунок 4.15 – Виведення значення з таблиці

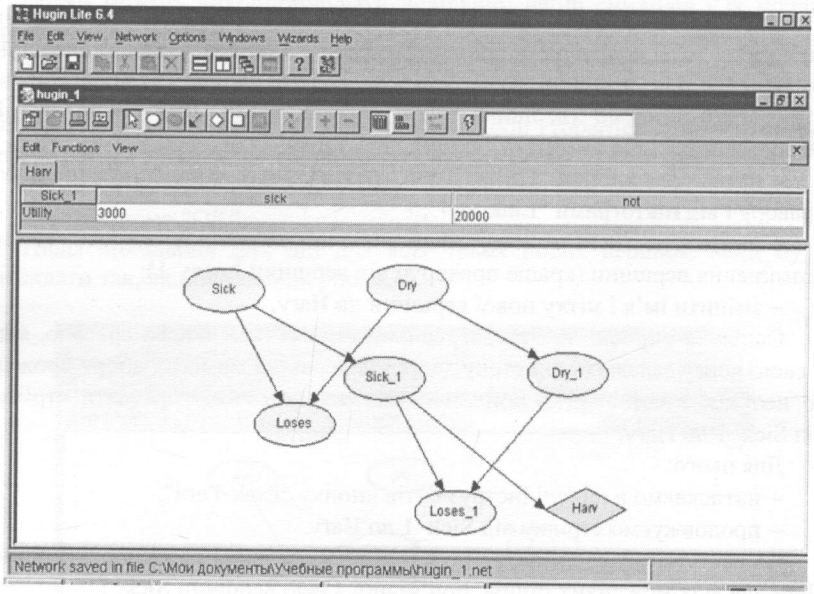

Рисунок 4.16 – Вигляд вікна ЕС на проміжному етапі створення ДВ

Додавання до ДВ вершини рішень і ще однієї вершини корисності

Процес прийняття рішення з використанням проектованої ДВ полягає у відповіді на запит про доцільність додаткових витрат на лікування дерев з метою підвищення їх врожайності, а, отже, і доходу від зібраного врожаю. З цією метою, відповідно до рис. 4.13, додамо в проектовану ДВ вершину рішення Treat (від англ. "Обробка").

Робиться це так само, як і у випадку з вершинами шансів або вершинами корисності:

– натискаємо в панелі інструментів кнопку "Discrete decision tool" с піктограмою прямокутника (праворуч від "Utility Tool");

– натискаємо мишею в робочій області (краще правіше вершини Dry);

– змінюємо ім'я і мітку створеної вершини на Treat.

Можливі стани вершини рішень задаються точно так само, як і стани для вершин шансів, а саме:

– виділяється вершина Treat (із списку або подвійним клацанням миші);

– натискається закладка States;

– вибираючи дії Action 1 і Action 2, змінюється назва можливих станів на "treat" і "not". Після кожної зміни натискається кнопка Rename.

Рішення про поточну обробку дерев проти їх захворювання зробить вплив на їх захворюваність в момент збирання врожаю, тобто вершина Treat має вплив на вершину Sick_1. Для обліку в діаграмі впливу цього причинно-наслідкового зв'язку додайте стрілку від Treat до Sick_1.

Разом з тим, рішення про оброблення дерев потребує певних матеріальних витрат. Це моделюється вершиною корисності Cost. Значення величини корисності вершини Cost для всіх можливих станів вершини Treat, виражені в тому ж самому еквіваленті, що і для вершини Harv, подані на рис. 4.17. Додаємо вершину корисності Cost до проектованої діаграми впливу. Для цього необхідно:

– додати нову вершину корисності (бажано праворуч від вершини Treat);

– змінити ім'я і мітку вершини на Cost;

– додати стрілку від Treat до Cost.

U (Cost)	
Treat='treat'	Treat='not'
- 8000	0

Рисунок 4.17 – Значення величини корисності вершини Cost

Стани для вершини корисності Cost заповнюються як показано на рис. 4.17. Для цього використовується команда View >> Open Tables. Вигляд вікна ЕС на цьому етапі побудови діаграми впливу показаний на рис. 4.18.

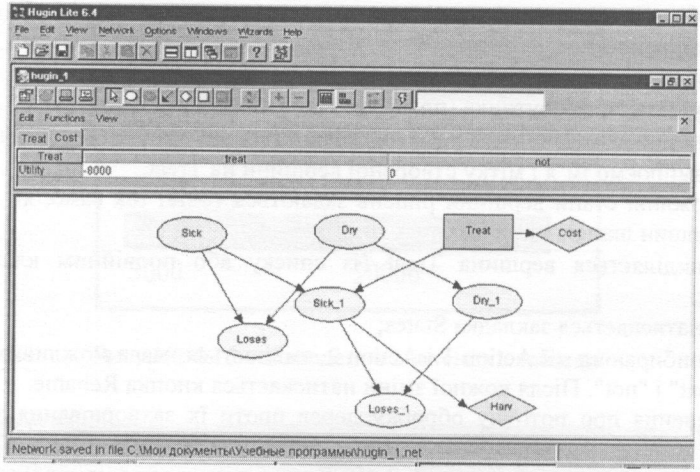

Рисунок 4.18 – Вигляд вікна ЕС на заключному етапі побудови діаграми впливу

Заповнення таблиць умовних ймовірностей

Після того, як при проектуванні ДВ були скопійовані вершини Sick_1 і Dry_1, вони успадкували від вершин Sick і Dry свої таблиці умовних ймовірностей. Однак, оскільки Sick_1 і Dry_1 є залежними не від тих же самих вершин, що і Sick та Dry, то їх таблиці умовних ймовірностей не відповідають дійсності і потребують коректування. Нові таблиці умовних ймовірностей для цих вершин, отримані на основі обробки знань експертів, можуть мати вигляд, аналогічний наведеному на рис. 4.19 і рис. 4.20.

P (Sick_1 | Sick, Treat)				
	Treat='treat'		Treat='not'	
	Sick='sick'	Sick='not'	Sick='sick'	Sick='not'
Sick='sick'	0.20	0.01	0.99	0.02
Sick='not'	0.80	0.99	0.01	0.98

Рисунок 4.19 – Нова таблиця умовних ймовірностей для вершини Sick

P(Dry_1 | Dry)		
	Dry='dry'	Dry='not'
Dry'='dry'	0.60	0.05
Dry'='not'	0.40	0.95

Рисунок 4.20 – Нова таблиця умовних ймовірностей для вершини Dry

Для остаточного визначення проектованої діаграми впливу необхідно:

– заповнити таблицю умовних ймовірностей для вершини Sick_1 даними з таблиці, що наведена на рис. 4.19;

– заповнити таблицю умовних ймовірностей для вершини Dry_1 даними з таблиці, що наведена на рис. 4.20.

Після того, як до вихідної БМД були додані три вершини шансів, вершина рішення, дві вершини корисності, а також заповнені всі необхідні таблиці умовних ймовірностей, побудову діаграми впливу можна вважати завершеною.

Після цього необхідно зберегти результати з побудови ДВ у вигляді нового файлу на диску. Якщо все було зроблено правильно, ДВ повинна мати вигляд, як показано на рис. 4.21.

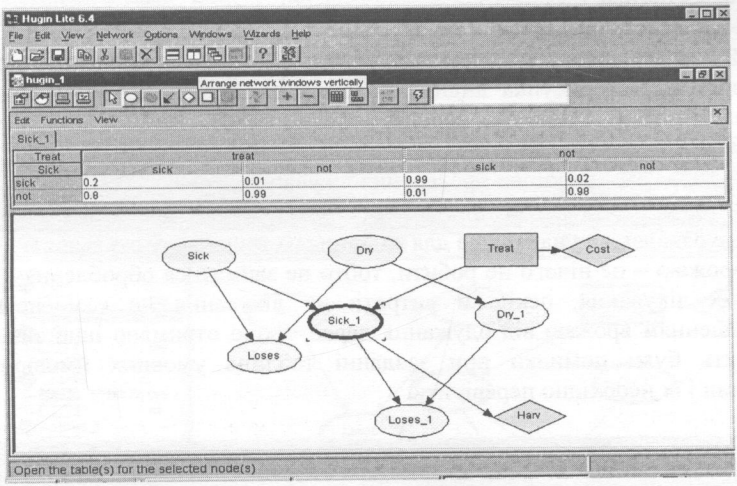

Рисунок 4.21 – Таблиця умовних ймовірностей для вершини Sick_1

4.2.4 Процес роботи і прийняття рішення з використанням експертної системи на основі діаграми впливу

Після завершення процесу моделювання знань, експертна система на основі ДВ готова до роботи. Для роботи з нею, перш за все, необхідно її відкомпілювати. Для цього необхідно з режиму редагування перейти в режим виконання. З цією метою натискаємо кнопку "Recompile".

Якщо процес компіляції діаграми не вдався, отже допущено помилку при її побудові. У цьому випадку спочатку перевіряємо, чи правильно встановлено зв'язки. Потім перевіряємо значення таблиць для кожної вершини. Після успішної компіляції можна приступити до роботи з ЕС і процесу прийняття рішення.

На початковому етапі єдине, що знає садівник – це те, що його дерева втрачають листя. Іншими словами, єдиним свідченням, яке можна повідомити ЕС, є факт опадання листя, тобто той факт, що подія Loses знаходиться в стані "yes". Яке рішення краще прийняти садівникові в цій ситуації?

Для вирішення цього завдання:

– відображаємо в списку вершин у лівій частині екрана допустимі стани вершин Loses і Treat, натиснувши ЛКМ по їх графічному позначенню в списку вершин;

– встановлюємо як абсолютно достовірний для вершини Loses стан "yes" (двічі натиснувши ЛКМ на цьому стані);

– проводимо поширення сум ймовірностей, натиснувши кнопку "Sum-propagate normal";

– дивимось на значення "treat" і "not" вершини Treat.

Якщо все виконано правильно, то будуть отримані значення, показані на рис. 4.22. З рисунка видно, що сумарна корисність події Treat, що визначається на основі всіх вершин корисності ДВ, приймає (для прикладу в грошовому еквіваленті) значення:

– 10234,39 – за умови проведення лікування дерев;

– 11514,01 – в іншому випадку.

Це означає, що найкраще для садівника з точки зору отримання доходу від врожаю – це нічого не робити, тобто не займатися обробленням дерев для їх лікування, оскільки витрати на лікування не компенсуються збільшенням врожаю від одужання дерев. Якщо отримано інші значення, значить були помилки при завданні таблиць умовних ймовірностей вершин і їх необхідно перевірити.

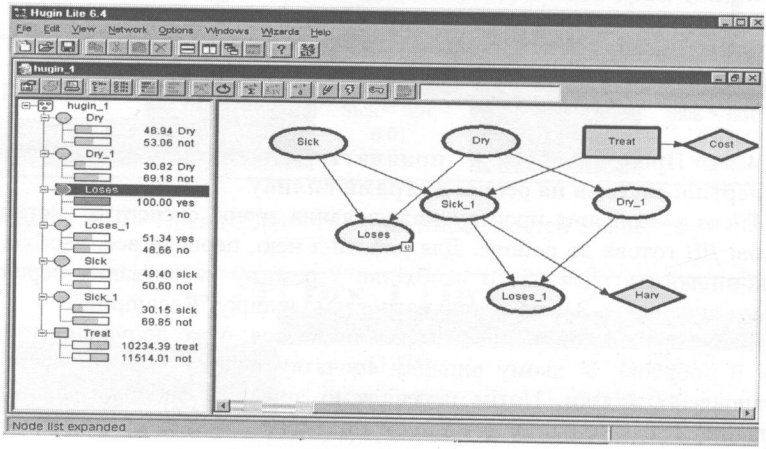

Рисунок 4.22 – Моделювання створеної діаграми впливу

При виявленні нових фактів, думка садівника щодо того, що робити, може змінитися. Розглянемо той випадок, коли садівник має інформацію про достатню кількість опадів, що випали за останній час, за якої дерева ніяк не можуть страждати від посухи. Введемо в ЕС це нове додаткове свідчення, яке являє собою факт, що для вершини Dry абсолютно достовірний стан "not". Для цього:

– розкриємо в списку вершин ДВ список станів вершини Dry;

– встановимо абсолютно достовірним стан "not" для Dry (двічі натиснувши ЛКМ на стані "not");

– проведемо нове поширення ймовірностей для діаграми впливу (натискаємо кнопку "Sum-propagate normal" на панелі інструментів);

– дивимось на значення сумарної корисності вершини Treat при її різних станах, а саме "treat" і "not".

Повинно вийти (рис. 4.23):

– 9138.33 – при проведенні будь-яких заходів;

– 5918.33 – за їх відсутності.

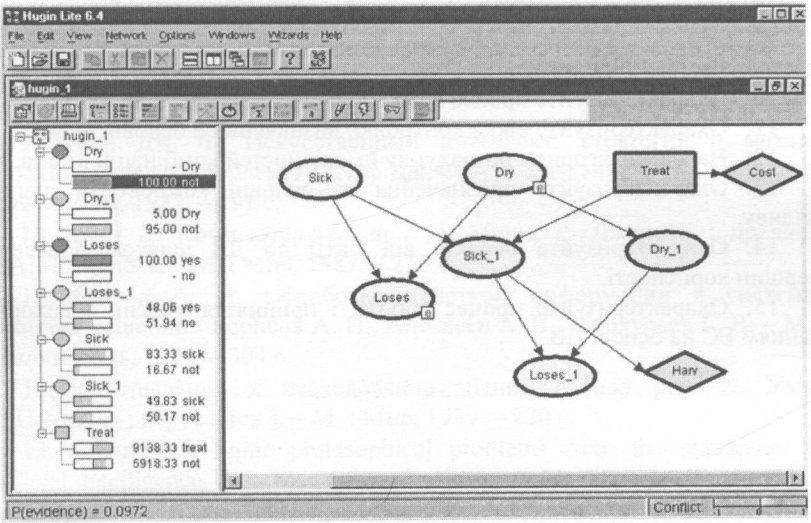

Рисунок 4.23 – Моделювання створеної діаграми впливу

У цьому випадку очевидно, що найкращим виходом із ситуації для садівника буде оброблення дерев. Причина відмінностей цих двох ситуацій полягає в тому, що в першому варіанті враховується ступінь незнання про те, чи страждає дерево від посухи. Отже, витрати, пов'язані з лікуванням, не будуть відшкодовані через можливе засихання дерев.

Аналогічним чином можуть бути досліджені будь-які інші факти і комбінації фактів із різним ступенем довіри до кожного з них.

4.3 Контрольні питання

1. Охарактеризуйте баєсові мережі довіри (БМД): визначення, основи побудови.
2. Наведіть приклад БМД для своєї предметної області.
3. Охарактеризуйте наближені методи формування висновку для звичайних мереж Баєса.
4. Охарактеризуйте причинно-наслідкові залежності між вершинами БМД.
5. Охарактеризуйте апріорні й умовні ймовірності в БМД.
6. Наведіть алгоритм додавання нових вершин у проектовану БМД.
7. Наведіть алгоритм встановлення причинно-наслідкових зв'язків між вершинами проектованої БМД.
8. Наведіть алгоритм визначення всіх можливих станів кожної з вершин БМД.
9. Наведіть алгоритм заповнення значень таблиць умовних ймовірностей кожної з вершин БМД.
10. Охарактеризуйте процес компілювання спроектованої БМД.
11. Охарактеризуйте вікна контролю: визначення й основні функції.
12. Наведіть алгоритм розрахунку ймовірностей комбінацій станів.
13. Охарактеризуйте призначення та основні компоненти діаграм впливу.
14. Охарактеризуйте перехід від БМД до ДВ шляхом додавання вершин корисності.
15. Охарактеризуйте процес роботи і прийняття рішень з використанням ЕС на основі ДВ.

СПИСОК ЛІТЕРАТУРИ

1. Бондарев В. Н. Искусственный интеллект: учеб. пособие для вузов / В. Н. Бондарев, Ф. Г. Аде. – Севастополь : СевНТУ, 2002. – 615 с.

2. Shepard R. N. Mental rotation of three-dimensional objects / R. N. Shepard, J. Metzler. – American Association for the Advancement of Science, 1971. – 703 p.

3. Интеллект : [в 3-х кн.]. – Кн. 2. Модели и методы : справочник / [под. ред. Д. А. Поспелова]. – М. : Радио и связь, 1990. – 304 с.

4. Кузин Л. Т. Основы кибернетики : [в 2-х т.]. Т. 2. Основы кибернетических моделей : учеб. пособие для вузов / Кузин Л. Т. – М. : Энергия, 1979. – 584 с.

5. Хант. Э. Искусственный интеллект / Хант Э. ; [пер. с англ., под ред. В. Л. Стефанюка]. – М. : Мир, 1971. – 558 с.

6. Нильсон Н. Принципы искусственного интеллекта / Нильсон Н. – М. : Радио и связь, 1985. – 376 с.

7. Гладун В. П. Процессы формирования новых знаний / Гладун В. П. – София : СД "Педагог 6", 1994. – 192 с.

8. Гладун В. П. Эвристический поиск в сложных средах / Гладун В. П. – К. : Наук. думка, 1977. – 166 с.

9. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. / Люгер Дж. Ф. – [4-е издание]. – М. : Финансы и статистика, 1987. – 191 с.

10. Мичи Д. Компьютер-творец / Д. Мичи, Р. Джонсон ; предисл. Д. А. Поспелова. – М. : Мир, 1987. – 255 с.

11. Борисов А. Н. Обработка нечеткой информации в системах принятия решений / Борисов А. Н., Алексеев А. В., Меркурьев Г. В. – М. : Радио и связь, 1989. – 304 с.

12. Представление и исследование знаний / [под ред. Х. Уэно, М. Исидзука ; пер. с япон.]. – М. : Мир, 1989. – 220 с.

13. McCarthy J. Some philosophical problems from the standpoint of artificial intelligence. / J. McCarthy, P. Hayes // Machine Intelligence 4. – Edinburgh : Edinburgh University Press, 1969. – 463 – 502 pp.

14. Charniak E. Introduction to Artificial Intelligence / E. Charniak, D. McDermott. – MA : Addison-Wesley, 1985. – 701 p.

15. Pearl J. Reverend Bayes on inference engines: a distributed hierarchical approach. / J. Pearl // National Conference on Artificial Intelligence. – 1982. – 133 – 136 pp.

16. Cheeseman P. In defense of Probability. / P. Cheeseman // 8th International Joint Conference on Artificial Intelligence. – 1985. – 1002 – 1009 pp.

17. Cooper G. F. The computational complexity of probabilistic inference using Bavesian belief networks / G. F. Cooper // Artificial Intelligence. – 1990. – 393 – 405 pp.

18. Бакаев Л. А. Экспертные системы и логическое программирование / Л. А. Бакаев, А. А. Гриценко. – К. : Наук, думка, 1992. – 220 с.

19. Элти Дж. Экспертные системы: концепции и примеры / Дж. Элти, М. Кумбс ; пер. с англ. – М. : Финансы и статистика, 1987. – 191 с.

20. Buchanan B. G. Rule-Based Expert Systems / B. G. Buchanan, E. H. Shortliffe – MA : Addison-Wesley, 1984. – 769 p.

21. Pearl J. Probabilistic Reasoning for Intelligent Systems / J. Pearl. – Los Altos, CA : Morgan Kaufmann, 1988. – 552 p.

22. Adams J. B. A probability model of medical reasoning and the MYCIN model. / J. B. Adams // Mathematical Biosciences. – 1976. – 177 – 186 pp.

23. Horvitz E. The inconsistent use of measures of certainty in artificial intelligence research / E. Horvitz, D. Heckerman // Uncertainty in Artificial Intelligence. – 1986. – 137 – 151 pp.

24. Horvitz E. A framework for comparing formalisms for plausible reasoning / E. Horvitz, D. Heckerman, C. Langlotz // Proc. National Conference on Artificial Intelligence. – 1986. – 210 – 214 pp.

25. Zadeh L. A. Fuzzy sets / L. A. Zadeh // Information and Control. – 1965. – 338 – 353 pp.

26. Zadeh L. A. Fuzzy logic and approximate reasoning / L. A. Zadeh // Synthese. – 1975. – 407 – 428 pp.

27. Zadeh L. A. Fuzzy sets as a basis for a theory of possibility / L. A. Zadeh // Fuzzy Sets and Systems. – 1978. – 3 – 28 pp.

28. Wise B. P. A framework for comparing uncertain inference systems to probability / B. P. Wise, M. Henrion // Uncertainty in Artificial Intelligence. – Amsterdam : North-Holland, 1986. – 69 – 83 pp.

29. Kahneman D. Subjective probability: a judgement of representativeness / D. Kahneman, A. Tversky // Cognitive Psychology. – 1972. – 430 – 454 pp.

30. Kahneman D. Judgement under Uncertainty: Heuristics and Biases / Kahneman D., Slovic P., Tversky A. – Cambridge : Cambridge University Press, 1982. – 544 p.

31. Tversky A. Judgment under uncertainty: Heuristics and biases / A. Tversky, D. Kahneman // Science. – Los Altos, CA : Morgan Kaufrnarm, 1974. – 544 p.

32. Biswas G. MIDST: an expert system shell for mixed initiative reasoning / G. Biswas, T. S. Anand // International Symposium on Methodologies for Intelligent Systems. – 1987. – 1 – 8 pp.

33. Provan G. M. The application of Dempster-Shafer theory to a logic-based visual recognition system / G. M. Provan // Uncertainty in Artificial Intelligence. – Amsterdam : North Holland, 1990. – 389 – 405 pp.

34. Pearl J. On evidential reasoning in a hierarchy of hypotheses / J. Pearl // Artificial Intelligence. – 1986. – 9 – 15 pp.

35. Pearl J. Probabilistic Reasoning for Intelligent Systems / J. Pearl. – Los Altos, CA : Morgan Kaufmann, 1988. – 552 p.

36. Shafer G. A Mathematical Theory of Evidence / Shafer G. – Princeton, NJ: Princeton University Press, 1976. – 314 p.

37. Yen J. A reasoning model based on an extended Dempster-Shafer theory / J. Yen // National Conference on Artificial Intelligence. –1986. – 125 – 131 pp.

38. Neapolitan R. E. Probabilistic Reasoning in Expert Systems: Theory and Algorithms / R. E. Neapolitan. – New York : Wiley, 1990. – 448 p.

39. Джексон П. Введение в экспертные системы / Джексон П. – [3-е изд.]. – М. : Вильямс, 2001. – 624 с.

40. Hugin Expert [Електронний ресурс]. – Тип доступу: http://www.hugin.com.

АЛФАВІТНИЙ ПОКАЖЧИК

Навчальне видання

Яровий Андрій Анатолійович
Арсенюк Ігор Ростиславович
Месюра Володимир Іванович

ЕКСПЕРТНІ СИСТЕМИ
Частина 2

Навчальний посібник

Редактор Є. Плетньова

Оригінал-макет підготовлено А. Яровим

Підписано до друку 30.11.2017 р.
Формат 29,7×42 ¼. Папір офсетний.
Гарнітура Times New Roman.
Ум. друк. арк. 6,04.
Наклад 50 (1-й запуск 1-20) пр. Зам. № 2017-414.

Видавець та виготовлювач
Вінницький національний технічний університет,
інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
Тел. (0432) 59-85-32, 59-87-38.
press.vntu.edu.ua; *e-mail:* kivc.vntu@gmail.com
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.