

НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій
Кафедра комп'ютерних наук

Методи та системи штучного інтелекту

*Тема №2. Моделі представлення знань в
системах штучного інтелекту*

Київ - 2026

Зміст

- 1. Логічні моделі знань.**
- 2. Продукційні моделі (продукційні правила);**
- 3. Семантичні моделі (семантичні мережі);**
- 4. Фреймові моделі (фреймові структури).**



В сучасних системах штучного інтелекту застосовуються такі моделі представлення знань:

- логічні моделі;
- продукційні моделі (продукційні правила);
- семантичні моделі (семантичні мережі);
- фреймові моделі (фреймові структури);
- штучні нейронні мережі (ШНМ).

1. Логічні моделі знань

Логічні моделі (ЛМ) використовуються, насамперед, в експертних системах управління, діагностування та проектування складних об'єктів.

Логічні моделі управління призначені для вироблення управлінських рішень, регламентуючих процес функціонування деякого реального об'єкта, званого об'єктом управління (ОУ).

При побудові логічних моделей управління виходять з таких положень.

Об'єкт управління розглядається як складна система, яка являє собою цілісну множину елементів, пов'язаних між собою взаємними відносинами.

У кожен момент часу ОУ знаходиться в одному з можливих станів.

Стан об'єкта управління у будь-який момент часу описується набором суттєвих змінних і параметрів, які мають назву характеристик стану системи:

$$z = (z_1, z_2, \dots, z_n).$$

Характеристики стану ОУ можуть піддаватися кількісному вимірюванню або оцінюватися якісними категоріями («високий», «низький» тощо).

Для кожної характеристики стану ОУ заданий допустимий діапазон її значень.

Для кожної характеристики стану ОУ заданий допустимий діапазон її значень.

Якщо всі характеристики стану ОУ у певний момент часу належать допустимим діапазонам, то говорять, що ОУ знаходиться у нормальному стані.

Якщо хоча би одна характеристика стану ОУ вийшла за межі свого допустимого діапазону, то вважається, що ОУ знаходиться в аномальному стані.

Перехід ОУ в аномальний стан викликає необхідність розв'язання задачі управління, тобто вироблення управлінського рішення, здатного привести ОУ у нормальний стан.

В системах управління логічні моделі у більшості випадків будуються за такими схемами:

<ситуація> → <дія> або <дія> → <результат>.

Якщо «дія» вимагає використання будь-яких ресурсів, то в ці схеми включається додатковий компонент і вони набувають такий вигляд:

<ситуація> & <ресурс> → <дія>;

<ресурс> & <дія> → <результат>.

У найбільше складних задачах управління до складу ЛМ можуть входити різноманітні за своєю структурою вирази, наприклад:

<умова реалізації дії> & <дія> → <результат>;

<умова наявності ресурсу> → <ресурс>;

<шукана дія> & <ресурс> → <необхідний результат>.

У наведених схемах символ «→» відображає логічну операцію імплікації, тобто логічного слідування.

Зміст імплікації відбивається висловлюванням «якщо ..., – то ...».

Термін «ситуація» має декілька наукових визначень:

Ситуація – це виникнення безлічі подій, збіг життєвих обставин і положень, що відкриваються сприйняттям та діяльністю людини.

Ситуація – це відхилення від ідеальної картини.

Ситуація – набір значень ознак, що описують стан об'єкта управління в певний момент часу.

У найпростіших випадках поняття «ситуація» ототожнюється із станом ОУ: ситуація вважається нормальною, якщо ОУ знаходиться у нормальному стані, і аномальною – в іншому випадку.

У більш складних задачах, коли практична реалізація управлінських рішень передбачає використання певних ресурсів, під ситуацією розуміють сукупність стану ОУ та стану цих ресурсів (наявність, справність тощо).

Дія – виконання управляючої операції, здатної привести ОУ у нормальний стан.

Ресурс – сукупність матеріальних, технологічних та інших засобів, необхідних для виконання тієї чи іншої управляючої операції.

Результат – зміна значення характеристики поточного стану ОУ, яка вийшла за межі допустимого діапазону.

Логічна модель представлення знань являє собою систему математичних виразів, записаних на мові предикатів першого порядку.

Предикат – логічна функція, яка в залежності від значень своїх аргументів (термів) приймає одно з двох значень: 1 («істина») або 0 («хибність»).

Предикати, які мають єдиний аргумент, називають одномісними.

Предикати, які мають декілька аргументів (наприклад,), називають -місними або -арними.

Одномісні предикати відбивають властивості деякого єдиного об'єкта, а багатомісні – відношення між об'єктами.

Наприклад, якщо одномісний предикат $P(x)$ має зміст « x – ціле число», то для значення аргументу $x=2$ він прийме значення «істина»: $P(x)=1$, а при $x=2.7$ – значення «хибність»: $P(x)=0$.

Двомісний предикат $Q(x,y)$, що має зміст « x більше y », при умові $x=5$ і $y=3$ прийме значення $Q(x,y)=1$ «істина»: , а при $x=4$ і $y=7$ – значення «хибність»: $Q(x,y)=0$.

В логічних моделях предикатними аргументами (термами) можуть бути константи, незалежні змінні або математичні функції.

Логічні моделі <ситуація> → <дія>

У найпростіших випадках логічна модель управління будується за схемою <ситуація> → <дія>.

Наприклад:

<йде дощ> → <взяти парасольку>;

<в кімнаті темно> → <ввімкнути лампу>;

<червоне світло світлофора> → <гальмувати> тощо.

Найпростіша логічна модель управління, побудована за схемою <ситуація> → <дія>, являє собою множину предикатних виразів такої структури:

$$(\forall i \in I^c)(\forall j \in J_i)[P(c_i) \rightarrow X(u_j)], \quad (2.1)$$

де I^c – множина аномальних ситуацій;

J_i – множина номерів управляючих операцій, здатних привести ОУ у нормальний стан в i -й аномальній ситуації;

c_i – вектор ознак i -ї аномальної ситуації;

u_j – ідентифікатор j -ї управляючої операції;

P – предикат, який відбиває факт виникнення i -ї аномальної ситуації;

X – предикат, що має смисл «реалізувати j -у управляючу операцію»;

\forall – квантор спільності.

Буквальний сенс формули (2.1) можна висловити таким чином: якщо виникла i -а аномальна ситуація, то необхідно виконати j -у управляючу операцію, яка належить множині тих управляючих операцій, що здатні привести ОУ у нормальний стан.

Це правило розповсюджується на всі можливі аномальні ситуації ($i \in I^c$) і всі управляючі операції з відповідної множини ($j \in J_i$).

Логічні моделі формуються у загальному вигляді окремо для кожного типу задач, що виникають у системі управління. Тому розв'язанню тієї чи іншої задачі передуює процедура налаштування ЛМ на конкретну ситуацію, що склалася.

Налаштування моделі (2.1) на ситуацію полягає у визначенні значень предикатів $P(c_i)$, $i \in I^c$.

Предикат $P(c_i)$ приймає значення «істина», якщо вектор ознак поточного стану ОУ, які вводяться користувачем системи або поступають на її вхід від зовнішніх датчиків чи інших технічних пристроїв, співпадають з вектором ознак i -ї аномальної ситуації c_i . В інших випадках предикат $P(c_i)$ приймає значення «хибність».

Припустимо, що значення «істина» прийняв предикат $P(c_{i^*})$.

Це означає, що сталася i^* -а аномальна ситуація, $i^* \in I^c$.

Для цієї аномальної ситуації модель (2.1) передбачає множину J_{i^*} номерів управляючих операцій, здатних привести ОУ у нормальний стан.

Якщо ця множина містить єдиний елемент $j^* \in J_{i^*}$, це означає, що необхідно виконати управляючу операцію u_{j^*} .

Формально це твердження виражається таким чином:

$$[P(c_{i^*}) = 1] \rightarrow (\forall j \in J_{i^*}) [X(u_j) = 1].$$

Якщо множина J_{i^*} містить єдиний елемент (тобто $|J_{i^*}| = 1$), то виконується єдина можлива управляюча операція u_{j^*} , $j^* \in J_{i^*}$.

Якщо множина J_{i^*} складається з декількох елементів (тобто $|J_{i^*}| > 1$), це означає, що для приведення ОУ у нормальний стан існують декілька управляючих операцій.

У цьому випадку для практичної реалізації вибирається та управляюча операція u_{j^*} , $j^* \in J_{i^*}$, яка є переважною за заданим критерієм.

Логічні моделі <ситуація> & <ресурс> → <дія>

Якщо для виконання управляючих операцій необхідно застосувати ті чи інші **ресурси**, логічна модель управління будується за схемою <ситуація> & <ресурс> → <дія>:

$$(\forall i \in I^c)(\forall j \in J_i)[P(c_i) \& \bigwedge_{t \in T_j} R(r_{jt}, s_{jt}) \rightarrow X(u_j)], \quad (2.2)$$

де R – предикат, що відбиває факт наявності або відсутності ресурсів, необхідних для реалізації j -ї управляючої операції у i -й ситуації;

T_j – множина типів ресурсів, необхідних для реалізації j -ї управляючої операції;

r_{jt} – ідентифікатор ресурсу t -го типу, необхідного для реалізації j -ї управляючої операції;

s_{jt} – показник стану ресурсу r_{jt} ;

\bigwedge – загальний знак кон'юнкції.

Налаштування моделі (2.2) на ситуацію полягає у:

- 1) визначенні значень предикатів $P(c_i)$, $i \in I^C$;
- 2) визначенні значень предикатів $R(r_{jt}, s_{jt})$, $j \in J_i$, $t \in T_j$.

Предикат $P(c_i)$ приймає значення «істина», якщо вектор ознак поточного стану ОУ, які вводяться користувачем системи або поступають на її вхід від зовнішніх датчиків чи інших технічних пристроїв, співпадають з вектором ознак i -ї аномальної ситуації c_i . В інших випадках предикат $P(c_i)$ приймає значення «хибність».

Припустимо, що значення «істина» прийняв предикат $P(c_{i^*})$.

Це означає, що сталася i^* -а аномальна ситуація, $i^* \in I^C$.

Предикат $R(r_{jt}, s_{jt})$ приймає значення «істина», якщо:

- ресурс t -го типу в наявності;
- його стан за кількісними та якісними показниками дозволяє його застосування для виконання j -ї управляючої операції.

Наприклад, для побудови шухляди необхідно мати 10 дошок довжиною 1 метр. Якщо вони є, то предикат $R(\text{дошка}, 10 \text{ штук}, 1 \text{ метр})$ приймає значення «істина». У всіх інших випадках (в наявності лише 7 дошок; в наявності 10 дошок, але довжиною лише 0,5 метра тощо) даний предикат приймає значення «хибність».

Для i^* -ї аномальної ситуації модель (2.2) передбачає множину J_{i^*} номерів управляючих операцій, здатних привести ОУ у нормальний стан.

Для виведення управлінського рішення на множині J_{i^*} необхідно виділити підмножину $J_{i^*}^R$ управляючих операцій, які забезпечені необхідними ресурсами:

$$J_{i^*}^R = \{j \in J_{i^*} : \bigwedge_{t \in T_j} R(r_{jt}, s_{jt}) = 1\}.$$

Значення предикатів $X(u_j)$, $j \in J_{i^*}^R$, відбиваючих необхідність реалізації j -ї управляючої операції, визначаються згідно формули:

$$[P(c_{i^*}) \& \bigwedge_{t \in T_{i^*}} R(r_{jt}, s_{jt}) = 1] \rightarrow (\forall j \in J_{i^*}^R)[X(u_j) = 1].$$

Якщо підмножина $J_{i^*}^R$ містить єдиний елемент $j^* \in J_{i^*}^R$ (тобто $|J_{i^*}^R| = 1$), це означає, що необхідно виконати управляючу операцію u_{j^*} .

Якщо $|J_{i^*}^R| > 1$, то для практичної реалізації вибирається та управляюча операція u_{j^*} , $j^* \in J_{i^*}^R$, яка є переважною за заданим критерієм.

Схеми $\langle \text{ситуація} \rangle \rightarrow \langle \text{дія} \rangle$ та $\langle \text{ситуація} \rangle \& \langle \text{ресурс} \rangle \rightarrow \langle \text{дія} \rangle$ використовуються тоді, коли будь-яка управляюча операція, передбачена логічною моделлю, безумовно приводить ОУ до нормального стану.

У цих випадках кількісне вимірювання чи якісне оцінювання результатів реалізації управляючих операцій втрачає сенс.

Однак, у більш складних задачах вироблення управлінських рішень така необхідність існує. Це обумовлює використання більш складних моделей, побудованих за схемою «ресурс» & «дія» \rightarrow «результат».

Логічні моделі «ресурс» & «дія» → «результат»

Логічні моделі, побудовані за схемою «ресурс» & «дія» → «результат», не містять ознак аномальних ситуацій, які вимагають розв'язання задачі вироблення управлінського рішення. Але «результат» кожної управляючої операції формулюється таким чином, щоб її виконання призводило ОУ до нормального стану.

Нехай $z = (z_i | i = 1, m)$ – вектор характеристик стану ОУ.

Логічні моделі, побудовані за схемою «ресурс» & «дія» → «результат», мають таку структуру:

$$(\forall i = \overline{1, m})(\forall j \in J_i) [\bigwedge_{t \in T_j} R(r_{jt}, s_{jt}) \& X(u_j, r_{jt}) \rightarrow D(z_i, a_{ij})], \quad (2.3)$$

де J_i – множина номерів управляючих операцій, виконання яких призводить до зміни значення i -ї характеристики стану ОУ;

T_j – множина типів ресурсів, необхідних для реалізації j -ї управляючої операції;

u_j – ідентифікатор j -ї управляючої операції;

r_{jt} – ідентифікатор ресурсу t -го типу, необхідного для реалізації j -ї управляючої операції;

s_{jt} – показник стану ресурсу r_{jt} ;

a_{ij} – показник зміни характеристики z_i стану ОУ під впливом j -ї управляючої операції;

$R(r_{jt}, s_{jt})$ – предикат, який відбиває факт наявності або відсутності ресурсу t -го типу, необхідного для реалізації управляючої операції u_j ;

$X(u_j, r_{jt})$ – предикат, що відбиває факт виконання управляючої операції u_j з використанням ресурсу r_{jt} ;

$D(z_i, a_{ij})$ – предикат, який відбиває факт зміни значення i -ї характеристики стану ОУ в результаті реалізації управляючої операції u_j .

Залежно від специфіки ОУ компоненти моделі (2.3) можуть нести різне смислове навантаження. Наприклад, показник s_{jt} стану ресурсу r_{jt} може констатувати деякий факт («увімкнутий», «вимкнений») або відбивати кількісну характеристику ресурсу, необхідного для виконання j -ї управляючої операції.

У випадку оцінювання характеристик стану ОУ ($z_i; i = \overline{1, m}$) **якісними категоріями** показник a_{ij} вказує напрям зміни значення характеристики z_i під впливом j -ї управляючої операції («збільшиться», «зменшиться»).

При **кількісному вимірюванні** характеристик стану ОУ ($z_i; i = \overline{1, m}$) показник a_{ij} є числом, яке відбиває зміну значення характеристики z_i в результаті виконання j -ї управляючої операції.

Варіанти:

1) Характеристики стану ОУ ($z_i; i = \overline{1, m}$) оцінюються **якісними категоріями**.

Показник a_{ij} вказує напрям зміни значення характеристики z_i під впливом j -ї управляючої операції («збільшиться», «зменшиться»).

2) Характеристики стану ОУ ($z_i; i = \overline{1, m}$) піддаються **кількісному вимірюванню**.

Показник a_{ij} є числом, яке відбиває зміну значення характеристики z_i в результаті виконання j -ї управляючої операції.

Виведення рішень при оцінюванні характеристик стану ОУ якісними категоріями

Виведенню рішень на основі логічної моделі управління передують її налаштування на ситуацію. При оцінюванні характеристик стану ОУ якісними категоріями налаштування моделі (2.3) на ситуацію полягає у:

При оцінюванні характеристик стану ОУ якісними категоріями налаштування моделі (2.3) на ситуацію полягає у:

1) фіксації номеру i^* характеристики z_{i^*} стану ОУ, значення якої вийшло за межі допустимого діапазону;

2) визначенні показника b_{i^*} напряму необхідної її зміни («збільшити», «зменшити»);

3) визначенні значень предикатів $R(r_{jt}, s_{jt})$, $j \in J_{i^*}$, $t \in T_j$.

Предикат $R(r_{jt}, s_{jt})$ приймає значення «істина», якщо ресурс r_{jt} наявності і його стан s_{jt} дозволяє використати його для практичної реалізації управляючої операції u_j . В іншому випадку предикат $R(r_{jt}, s_{jt})$ приймає значення «хибність».

Алгоритм відшукування управляючої операції, здатної привести характеристику z_i^* в допустимий діапазон, передбачає наступні дії:

1. Формування множини номерів управляючих операцій, здатних привести до необхідної зміни характеристики z_i^* :

$$J_i^D = \{j \in J_i^* : a_{i^*j} = b_i^*\}.$$

2. Визначення підмножини номерів управляючих операцій, що входять до складу множини J_i^D , для реалізації яких є необхідні ресурси:

$$J_i^R = \{j \in J_i^D : \bigwedge_{t \in T_j} R(r_{jt}, s_{jt}) = 1\}.$$

Якщо виявляється, що $J_i^R = \emptyset$, це означає, що модель (2.3) не передбачає можливості зміни значення характеристики z_i^* у ситуації, що склалася.

3. Вибір (при $|J_i^R| > 1$) найбільш прийнятній (за заданим критерієм) управляючої операції u_{j^*} , $j^* \in J_i^R$, яка підлягає реалізації.

Виведення рішень при кількісному вимірюванні характеристик стану ОУ

При кількісному вимірюванні характеристик стану ОУ **налаштування** моделі (2.3) на ситуацію полягає у:

- 1) фіксації номеру i^* характеристики z_{i^*} стану ОУ, значення якої вийшло за межі допустимого діапазону;
- 2) визначенні значень предикатів $R(r_{jt}, s_{jt})$, $j \in J_{i^*}$, $t \in T_j$.

Предикат $R(r_{jt}, s_{jt})$ приймає значенні «істина», якщо ресурс r_{jt} в наявності і його стан s_{jt} дозволяє використати його для практичної реалізації управляючої операції u_j . В іншому випадку предикат $R(r_{jt}, s_{jt})$ приймає значенні «хибність».

Алгоритм відшукування управляючої операції, здатної привести характеристику z_{i^*} у допустимий діапазон, передбачає наступні дії:

1. Формування множини номерів управляючих операцій, здатних привести значення цієї характеристики в допустимий діапазон:

$$J_{i^*}^D = \{j \in J_{i^*} : z_{i^*}^{(1)} \leq z_{i^*} + a_{i^*,j} \leq z_{i^*}^{(2)}\}.$$

Якщо $J_i^D = \emptyset$, обчислення припиняються. Інакше виконується наступний пункт цього алгоритму.

Якщо $J_i^D = \emptyset$, це означає, що модель (2.3) не передбачає жодної управляючої операції, яка була б здатна без залучення інших операцій привести характеристику z_i у допустимий діапазон.

У цьому випадку доцільно розглянути можливість одночасної реалізації декількох управляючих операцій u_j , $j \in J_i^D$, які призводять до зміни значення даної характеристики.

2. Формування підмножини номерів управляючих операцій, що входять до складу множини J_i^D , для реалізації яких є необхідні ресурси:

$$J_i^R = \{j \in J_i^D : \bigwedge_{r \in I_j} R(r_{jt}, s_{jt}) = 1\}.$$

Якщо $J_i^R = \emptyset$, обчислення припиняються. Інакше виконується наступний пункт цього алгоритму.

Якщо $J_i^R = \emptyset$, це означає, що ні для однієї управляючої операції, здатної привести характеристику z_i у допустимий діапазон, немає необхідних ресурсів. У такій ситуації задача не має рішень.

3. Вибір управляючої операції u_{j^*} , $j^* \in J_i^R$, що підлягає реалізації.

Якщо підмножина J_i^R складається з єдиного елемента j^* , то саме він і визначає управляючу операцію u_{j^*} , що підлягає реалізації.

При $|J_i^R| > 1$ для реалізації вибирається найбільш прийнятна (за заданим критерієм) управляюча операція u_{j^*} , $j^* \in J_i^R$.

Логічні моделі (2.1)-(2.3) дозволяють за допомогою простих алгоритмів знаходити управлінські рішення у ситуаціях, коли за межами допустимого діапазону знаходиться **єдина** характеристика стану ОУ і для її нормалізації достатньо **однієї** управляючої операції.

На практиці часто зустрічаються більш складні задачі, коли:

а) за межами допустимого діапазону знаходиться **єдина** характеристика стану ОУ, але для її нормалізації **однієї** управляючої операції недостатньо;

б) за межами допустимих діапазонів знаходяться **декілька** характеристик стану ОУ;

с) існує **побічний ефект**, який полягає у тому, що одна і та же управляюча операція може одночасно змінювати значення декількох характеристик стану ОУ.

У цих випадках для виведення управлінських рішень застосовуються більш складні алгоритми, засновані на методах комбінаторної оптимізації.

Логічні моделі діагностування

Задачі діагностування відрізняються від задач управління тим, що в задачах управління ми шукаємо **дію**, здатну нормалізувати аномальну ситуацію, а в задачах діагностування – **причину** виникнення аномальної ситуації.

В задачах діагностування **ситуація** ототожнюється зі станом об'єкта діагностування (ОД).

Стан ОД визначається вектором значень контрольованих параметрів – характеристик стану ОД:

$$z = (z_j \mid j = \overline{1, n}).$$

Надалі під ОД будемо розуміти складну технічну систему, в якій можуть виникати як поодинокі, так і множинні пошкодження (відмови) з ефектом накладання їх наслідків.

Ефект накладання наслідків полягає в тому, що одночасне виникнення кількох пошкоджень, кожне з яких впливає на ту чи іншу характеристику стану ОД, може призвести до зміни її значення на величину, що є арифметичною сумою приростів, викликаних кожним із цих пошкоджень окремо.

Логічні моделі діагностування будуються за таким схемами:

<окреме пошкодження> → <одна характеристика>;

<окреме пошкодження> → <множина характеристик>;

<комбінація пошкоджень> → <множина характеристик>.

Логічні моделі діагностування, побудовані за схемою **<окреме пошкодження> → <одна характеристика>**, застосовуються для визначення поодиноких пошкоджень у випадках, коли за межами допустимого діапазону знаходиться одна характеристика стану ОД.

Логічні моделі, побудовані за схемою **<окреме пошкодження> → <множина характеристик>**, використовуються тоді, коли межами допустимого діапазону одночасно опиняються декілька характеристик стану ОД.

Логічні моделі, побудовані за схемою **<комбінація пошкоджень> → <множина характеристик>**, застосовується для визначення комбінацій пошкоджень.

Логічні моделі діагностування у найпростіших випадках будуються за схемою <окреме пошкодження> \rightarrow <одна характеристика> і відображаються формулою:

$$(\forall i = \overline{1, m}) [X(p_i) \rightarrow D(z_{j(i)}, h_{i, j(i)})], \quad (2.4)$$

де m – кількість видів пошкоджень, які можуть виникати в ОД;

$X(p_i)$ – предикат, що констатує факт виникнення i -го пошкодження;

$D(z_j, h_{ij})$ – предикат, який відбиває зміну значення характеристики z_j стану ОД на величину h_{ij} внаслідок виникнення i -го пошкодження;

$j(i)$ – номер характеристики стану ОД, яка змінює значення внаслідок виникнення i -го пошкодження.

Алгоритм визначення окремого пошкодження при зміні значення однієї характеристики стану ОД передбачає такі дії:

1) Фіксація номеру j^* характеристики стану ОД, яка вийшла за межі допустимого діапазону.

2) Визначення пошкодження p_{i^*} за ознакою:

$$\delta_{j^*} - \varepsilon_{j^*} \leq h_{i^*, j^*} \leq \delta_{j^*} + \varepsilon_{j^*},$$

де δ_j – відхилення фактичного значення характеристики z_j від еталонного;

ε_j – допустима похибка вимірювання значення відхилення δ_j .

Ознака виникнення комбінації пошкоджень формулюється таким чином: не існує такого i -го ($1 \leq i \leq m$) пошкодження, для якого множина J_i номерів характеристик стану ОД, що змінюють значення внаслідок його виникнення, співпадає з множиною J^* номерів характеристик стану ОД, які вийшли за межі допустимого діапазону:

$$(\bigcap \exists i : 1 \leq i \leq m) (J^* = J_i),$$

де J^* – множина номерів характеристик стану ОД, які вийшли за межі допустимого діапазону.

Визначення комбінацій пошкоджень здійснюється за такою процедурою:

1) Фіксація множини J^* номерів характеристик стану ОД, які вийшли за межі допустимого діапазону.

2) Зіставлення предикатам $X(p_i)$ бівалентних змінних:

$$x_i \in \{0, 1\}, i \in I^X,$$

де I^X – множина ідентифікаторів пошкоджень, які можуть призвести до зміни значень характеристик $z_j, j \in J^*$:

$$I^X = \{i : (1 \leq i \leq m) \& (J_i \cap J^* \neq \emptyset)\}.$$

3) Формування алгебраїчної моделі:

$$\delta_j - \varepsilon_j \leq \sum_{i \in I^X} h_{ij} x_i \leq \delta_j + \varepsilon_j; j \in J^*.$$

4) Розв'язання задачі методом спрямованого перебору варіантів.

4) Розв'язання задачі методом спрямованого перебору варіантів.

Інтерпретація отриманого результату: в ОД виникли пошкодження p_i , $i \in I^*$, де $I^* = \{i \in I^X : x_i = 1\}$.

Приклад 2.2. Визначити множину можливих пошкоджень, які можуть призвести до зміни значень характеристик стану ОД z_1 , z_3 та z_5 при таких вхідних даних:

– множина номерів можливих пошкоджень:

$$I = \{1, 2, 3\};$$

– множини номерів характеристик стану ОД, які змінюють свої значення внаслідок виникнення пошкоджень $i \in \{1, 2, 3\}$:

$$J_1 = \{1, 2, 3\}; \quad J_2 = \{2, 4, 6\}; \quad J_3 = \{3, 5, 7\};$$

– множина номерів характеристик стану ОД, які вийшли за межі допустимого діапазону:

$$J^* = \{1, 3, 5\}.$$

Визначення елементів множини I^X номерів пошкоджень, які можуть призвести до зміни значень характеристик z_1 , z_3 та z_5 :

1) для $i = 1$:

$$J_1 \cap J^* = \{1, 2, 3\} \cap \{1, 3, 5\} = \{1, 3\} \neq \emptyset;$$

2) для $i = 2$:

$$J_2 \cap J^* = \{2, 4, 6\} \cap \{1, 3, 5\} = \emptyset;$$

3) для $i = 3$:

$$J_3 \cap J^* = \{3, 5, 7\} \cap \{1, 3, 5\} = \{3, 5\} \neq \emptyset.$$

Отже, множина I^X номерів пошкоджень, які можуть призвести до зміни значень характеристик z_1 , z_3 та z_5 , має такий склад:

$$I^X = \{1, 3\}.$$

2. Продукційні моделі знань

Продукційні моделі (ПМ) використовуються для розв'язання задач управління, діагностування та проектування складних об'єктів.

Продукційні моделі доцільно застосовувати у тих випадках, коли для досягнення бажаного результату необхідно здійснити не одиночну дію, а послідовність певних дій.

У таких ситуаціях остаточне рішення основної задачі являє собою ланцюг рішень окремих задач.

Приклад прийняття рішення в ситуації з «пробитим» колесом автомобіля.

Одне із можливих рішень цієї задачі формулюється таким чином: «Якщо тиск у шині автомобіля нижче допустимого та є запасне колесо з нормальним тиском у шині, справний домкрат і балонний ключ, то необхідно самостійно замінити колесо».

А якщо запасне колесо є, але тиск в ньому нижче допустимого? Для врахування такої ситуації треба додати до логічної моделі ще один математичний вираз з таким змістом: «Якщо запасне колесо є, але тиск у його шині нижче допустимого та є насос, то необхідно запасне колесо накачати».

Для нормалізації такої ситуації необхідно послідовно виконати дві дії: спочатку накачати запасне колесо, а потім замінити їм «пробите» колесо автомобіля. Тому і рішення задачі повинне складатися з двох відповідних компонентів.

Структура продукційних моделей

Продукційна модель – це система предикатних виразів, які називають продукційними правилами або продукціями:

$$P_j = \bigwedge_{i \in I_j} A_i \rightarrow A_j; \quad j = \overline{1, n}, \quad (2.6)$$

де P_j – позначення j -ї продукції;

n – кількість продукцій у продукційній моделі;

A_i та A_j – предикати, що мають назву посилки та заключення (умовиводу) відповідно;

I_j – множина номерів предикатів, що входять до продукції P_j як посилки.

Предикат може бути посилкою в одній продукції і заключенням в іншій. Тому розрізняють:

- **вхідні** предикати, які входять у продукційну модель виключно як посилки;
- **проміжні** предикати, які входять до деяких продукцій як посилки, а до інших – як заключення;
- **кінцеві (термінальні)** предикати, що входять у продукційну модель виключно як заключення.

Вхідні предикати описують поточний стан ОУ у ситуації, що склалася.

Проміжні предикати відбивають умови застосовності управлінських рішень, а також попередні дії, що забезпечують можливість практичної реалізації цих рішень.

Термінальні предикати визначають можливі управлінські рішення.

Нехай I^A – множина номерів предикатів, які входять у ліві частини всіх виразів моделі (2.6):

$$I^A = \bigcup_{j=1}^n I_j.$$

Тоді множини номерів вхідних (I^B), проміжних ($I^П$) та термінальних (J^T) предикатів визначатимуться за формулами:

$$I^B = I^A \setminus J;$$

$$I^П = I^A \cap J;$$

$$J^T = J \setminus I^П,$$

де $J = \{1, \dots, n\}$.

В інтелектуальних системах управління вхідні, проміжні та термінальні предикати мають такий зміст:

- вхідні предикати описують поточний стан ОУ у ситуації, що склалася;
- проміжні предикати відбивають умови застосовності управлінських рішень, а також попередні дії, що забезпечують можливість практичної реалізації цих рішень;
- термінальні предикати визначають можливі управлінські рішення.

Значення вхідних предикатів визначаються користувачем системи при налаштуванні моделі на ситуацію.

Приклад 2.4. Визначити множини номерів вхідних, проміжних та термінальних предикатів для продукційної моделі:

$$A_2 \wedge A_5 \wedge A_7 \rightarrow A_1;$$

$$A_4 \wedge A_6 \wedge A_8 \rightarrow A_2;$$

$$A_5 \wedge A_7 \wedge A_9 \rightarrow A_3;$$

$$A_6 \wedge A_8 \wedge A_9 \rightarrow A_4.$$

$$I^A = \{2, 4, 5, 6, 7, 8, 9\}; J = \{1, 2, 3, 4\}.$$

$$I^B = \{2, 4, 5, 6, 7, 8, 9\} \setminus \{1, 2, 3, 4\} = \{5, 6, 7, 8, 9\};$$

$$I^{\Pi} = \{2, 4, 5, 6, 7, 8, 9\} \cap \{1, 2, 3, 4\} = \{2, 4\};$$

$$J^T = \{1, 2, 3, 4\} \setminus \{2, 4, 5, 6, 7, 8, 9\} = \{1, 3\}.$$

Налаштування продукційної моделі на ситуацію

Налаштування продукційної моделі (2.6) на ситуацію полягає у визначенні та фіксації значень вхідних предикатів, що відбивають поточний стан ОУ.

Налаштування продукційної моделі (2.6) на ситуацію полягає у визначенні та фіксації значень вхідних предикатів, що відбивають поточний стан ОУ.

Нехай I^C – множина номерів вхідних предикатів, значення яких характеризують ситуацію, що склалася; $I^C \subseteq I^B$.

Якщо зміст предиката A_i ($i \in I^C$) відповідає ситуації, то йому привласнюється значення 1 (тобто «істина»), якщо ні – значення 0 (тобто «хибність»).

Множину номерів вхідних предикатів, які в процесі настройки на ситуацію отримали значення «істина», будемо позначати символом I_1^C , а множину вхідних предикатів, яким привласнене значення «хибність», – символом I_0^C :

$$I_1^C = \{i \in I^C : A_i = 1\};$$

$$I_0^C = \{i \in I^C : A_i = 0\}.$$

Алгоритм виведення рішень на основі продукційної моделі

Задача логічного виведення рішення на основі продукційної моделі (2.6) полягає у відшуванні такого термінального предиката A_j ($j^* \in J^T$), який приймає значення «істина» для заданого набору значень вхідних предикатів A_i ; $i \in I^C$.

Згідно правил, які встановлені для продукційних моделей, заключний предикат продукції приймає значення «істина», якщо одночасно є істинними всі її предикати-посилки, та значення «хибність», якщо хоча би один з її предикатів-посилок є хибним:

$$A_j = \begin{cases} 1, & \text{якщо } (\forall i \in I_j) (A_i = 1) \\ 0, & \text{якщо } (\exists i \in I_j) (A_i = 0) \end{cases}; \quad j \in J. \quad (2.7)$$

Продукція, у якої заключний предикат приймає значення «істина», зветься **актуалізованою**.

Позначення

J_1 – множина номерів предикатів-заключень, які отримали значення «істина»:

$$J_1 = \{j \in J : A_j = 1\};$$

J_0 – множина номерів предикатів-заключень, які отримали значення «хибність»:

$$J_0 = \{j \in J : A_j = 0\};$$

J^H – множина номерів предикатів-заключень, які після підстановки значень вхідних предикатів $A_i \in \{0, 1\}$, $i \in I^C$ залишилися невизначеними:

$$J^H = J \setminus (J_1 \cup J_0);$$

J_1^T – множина номерів термінальних предикатів, які отримали значення «істина»:

$$J_1^T = J^T \cap J_1;$$

J_0^T – множина номерів термінальних предикатів, які отримали значення «хибність»:

$$J_0^T = J^T \cap J_0;$$

J^{TH} – множина номерів термінальних предикатів, які після підстановки значень вхідних предикатів $A_i \in \{0, 1\}$, $i \in I^C$ залишилися невизначеними:

$$J^{TH} = J^T \cap J^H;$$

J_1^Π – множина номерів проміжних предикатів, які отримали значення «істина»:

$$J_1^\Pi = I^\Pi \cap J_1;$$

J_0^Π – множина номерів проміжних предикатів, які отримали значення «хибність»:

$$J_0^\Pi = I^\Pi \cap J_0;$$

$J^{\Pi H}$ – множина номерів проміжних предикатів, які після підстановки значень вхідних предикатів $A_i \in \{0, 1\}$, $i \in I^C$ залишилися невизначеними:

$$J^{\Pi H} = I^\Pi \cap J^H.$$

Алгоритм виведення рішення на основі продукційної моделі (2.6) передбачає послідовне виконання таких операцій:

1) Підстановка значень вхідних та проміжних предикатів в продукційну модель (2.6).

У першому циклі реалізації алгоритму в продукційну модель (2.6) підставляються значення вхідних предикатів $A_i \in \{0, 1\}$, $i \in I^C$, які відбивають ситуацію, що склалася.

При реалізації всіх наступних циклів в модель (2.6), налаштовану на ситуацію, підставляються значення проміжних предикатів $A_j \in \{0, 1\}$, $j \in (J_1^{\Pi} \cup J_0^{\Pi})$, визначених у попередньому циклі.

2) Визначення значень предикатів-заключень згідно формули (2.7).

3) Перевірка факту встановлення шуканого рішення.

Якщо

$$(J_1^T \neq \emptyset) \& (|J_1^T| = 1),$$

це означає, що шукане управлінське рішення знайдено.

Воно визначається єдиним термінальним предикатом $A_{j^*} = 1$, $j^* \in J_1^T$.

Якщо

$$(J_1^T \neq \emptyset) \& (|J_1^T| > 1),$$

це означає, що знайдено декілька можливих управлінських рішень.

Для практичної реалізації вибирається найбільш прийнятне (за заданим критерієм) з них.

В обох випадках обчислювальний процес завершується.

Інакше виконується наступний пункт алгоритму.

4) Перевірка умови існування шуканого рішення.

Якщо

$$(\forall j \in J^T) (A_j = 0) ,$$

це означає, що задача не має рішень.

Найчастіше це пояснюється тим, що розробники продукційної моделі не передбачили ситуацію, яка склалася в ОУ.

На цьому обчислення припиняються.

Інакше виконується наступний пункт алгоритму.

5) Перевірка наявності проміжних предикатів, які отримали значення в результаті підстановки в модель (2.6) значень вхідних предикатів $A_i \in \{0, 1\}$, $i \in I^C$.

Якщо

$$J_1^{\Pi} \cup J_0^{\Pi} \neq \emptyset,$$

то виконується наступний пункт алгоритму, в іншому випадку – пункт 7.

б) Підстановка значень проміжних предикатів A_j , $j \in (J_1^{\Pi} \cup J_0^{\Pi})$ в ліві частини виразів продукційної моделі (2.6).

Після цього реалізується повторний цикл алгоритму, починаючи з пункту 2.

7) Формування множин номерів вхідних предикатів, які потребують від користувача додаткового завдання значень.

Для кожного термінального предиката A_j , $j \in J^{TH}$, значення якого ще не визначено, множина I_j^{BD} номерів вхідних предикатів, значення яких необхідно додатково задати, визначається формулою:

$$I_j^{BD} = I_j^B \setminus I^C, \quad j \in J^{TH},$$

де I_j^B – множина номерів вхідних предикатів, що входять у вираз моделі (2.6), який відповідає термінальному предикату $j \in J^{TH}$:

$$I_j^B = I_j \cap I^B.$$

Перелік вхідних предикатів A_i , $i \in I_j^{BD}$, $j \in J^{TH}$, значення яких необхідно додатково визначити, надається користувачу системи.

З метою скорочення тривалості подальшого процесу розв'язання задачі з моделі (2.6) видаляються продукції, в яких предикати-заключення вже отримали конкретні значення.

На цьому обчислення призупиняються.

Після отримання від користувача значень вказаних вхідних предикатів наведена послідовність дій реалізується повторно, починаючи з пункту 1.

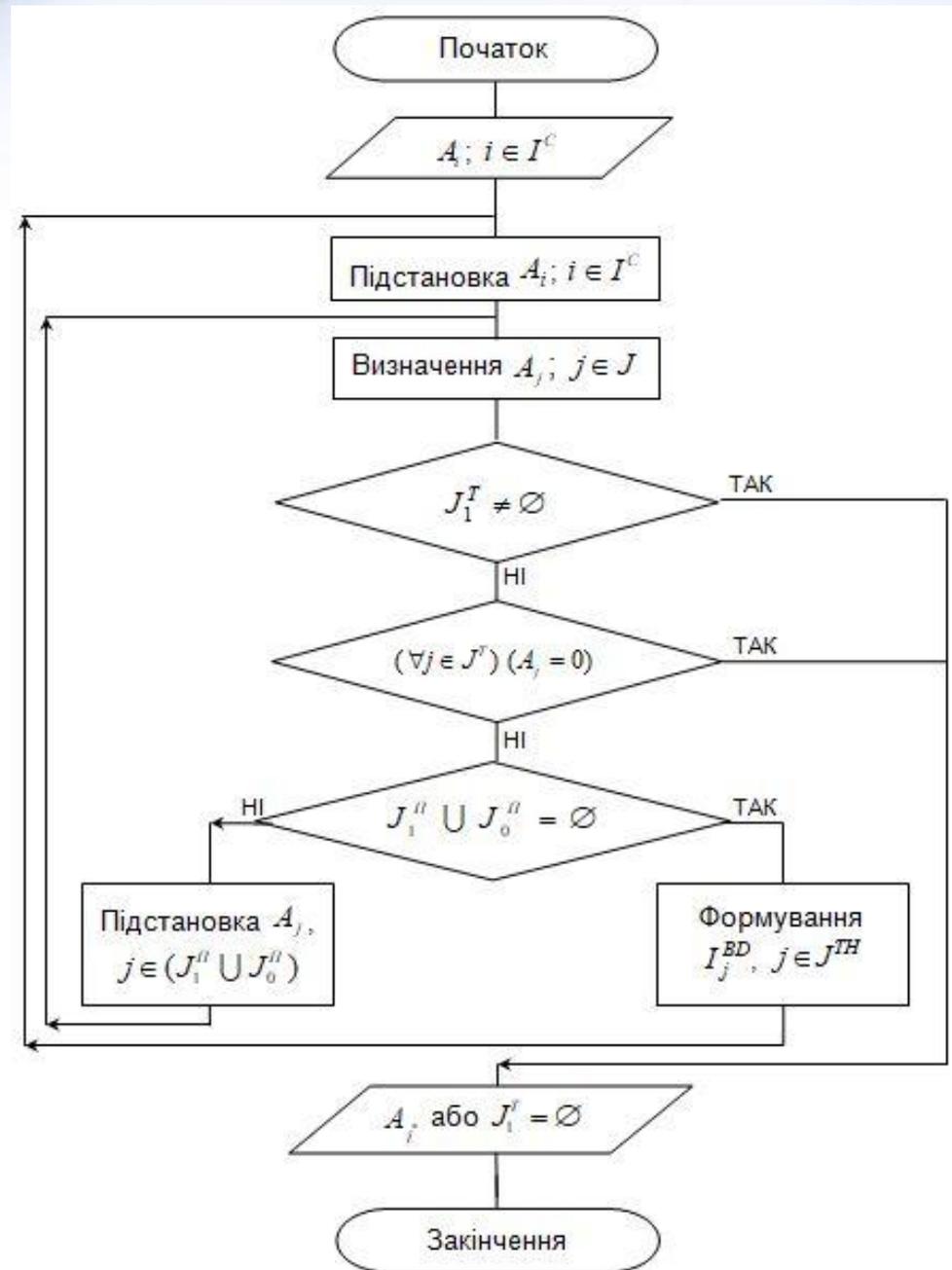


Схема алгоритму виведення рішень на основі продукційної моделі

3. Семантичні моделі знань

Семантичні моделі представлення знань (семантичні мережі) використовуються, головним чином, в інтелектуальних системах обробки текстів природної мови (ПМ-системах), системах автоматичного перекладу текстів, діалогових інформаційно-пошукових системах.

Структура семантичних мереж

Семантична мережа має вигляд розміченого орієнтованого графа, вершини якого відповідають **поняттям** предметної області, а дуги – **відношенням** між ними.

Відношення задають базову структуру організації знань, дослідження якої дозволяє виводити логічним шляхом інші знання.

До категорій **понять** предметної області відносять:

- об'єкти, суб'єкти (агенти), предмети;
- події, дії, операції;
- властивості, місця та ін.

Об'єкт – філософська категорія, яка виражає щось, на що спрямована практична, теоретична, пізнавальна або дослідницька діяльність суб'єкта (спостерігача, дослідника).

Суб'єкт – фізична або юридична особа, консолідована група осіб (спільнота), діяльність якої полягає у дослідженні існуючих об'єктів, створенні нових об'єктів, впливі на об'єкт тощо.

В системах управління об'єктом є керований процес, а суб'єктом – орган управління цим процесом (управляюча підсистема).

Предмет – частина об'єкта, певний його аспект, досліджуваний в якомусь конкретному випадку; те, на що спрямована думка або дія об'єкта.

Типи відношень:

- теоретико-множинні (частина – ціле, множина – підмножина, клас – підклас, клас – елемент класу);
- логічні (диз'юнкція, кон'юнкція, інверсія, імплікація);
- функціональні, визначувані дієсловами («виробляє», «впливає» тощо);
- кількісні (більше, менше, рівно);
- просторові (далеко від ..., близько від ..., за, під, над);
- часові (раніше, пізніше, протягом ...);
- атрибутивні зв'язки (мати властивість, мати значення ...).

Для семантичних мереж характерна ієрархія понять або система класифікації, на основі якої поняття, що відносяться до предметної області, класифікуються на деяку кількість категорій або класів на основі їх загальних властивостей.

В ієрархічній структурі понять використовуються **предикати відношень** типів IS-A, PART-OF та AKO (скорочення від A KIND-OF).

Відношення типу **IS-A** («це», «є», «являє собою») означає **наслідування**, при якому екземпляр поняття нижнього рівня містить всі атрибути, які має екземпляр поняття верхнього рівня (прототип).

Наприклад, речення «АН-124 є літаком» можна представити у такій формі:

IS-A (АН-124, Літак).

Вираз **PART-OF** («частина», «елемент», «входить до складу») відбиває відношення **«ціле – частина»**.

Наприклад, фраза «Академічна група IT-461Б входить до складу Факультету кібербезпеки, комп'ютерної та програмної інженерії» можна представити таким чином:

PART-OF (Академічна група IT-461Б, Факультет ККІП).

Предикат типу **AKO** («типу», «виду», «на кшталт», «має властивість») відображає відношення **«об'єкт – властивість»**.

Наприклад, висловлювання «Літак АН-124 оснащений турбореактивними двигунами» можна представити так:

AKO (Літак АН-124, Турбореактивний).

Загальною формою подібних висловлювань є предикатний вираз:

$$\forall x \forall y [P(x) \rightarrow Q(y)]. \quad (2.7)$$

У першому прикладі:

x – предикатний аргумент, який має значення **об'єкту** «АН-124»;

y – предикатний аргумент, який має значення **класу об'єктів** «Літак»;

$P(x)$ – предикат, який відбиває умову « x – АН-124»;

$Q(y)$ – предикат, який відбиває умову « y – Літак».

У другому прикладі:

x – предикатний аргумент, який має значення **елементу** «Академічна група ІТ-461Б»;

y – предикатний аргумент, який має значення **множини** «Факультет ККП»;

$P(x)$ – предикат, який відбиває умову « x – Академічна група ІТ-461Б»;

$Q(y)$ – предикат, який відбиває умову « y – Факультет ККП».

У третьому прикладі:

x – предикатний аргумент, який має значення **об'єкту** «Літак АН-124»;

y – предикатний аргумент, який має значення **властивості об'єкту** (тип двигуна) «Турбореактивний»;

$P(x)$ – предикат, який відбиває умову « x – Літак АН-124»;

$Q(y)$ – предикат, який відбиває умову « y – Тип двигуна: турбореактивний».

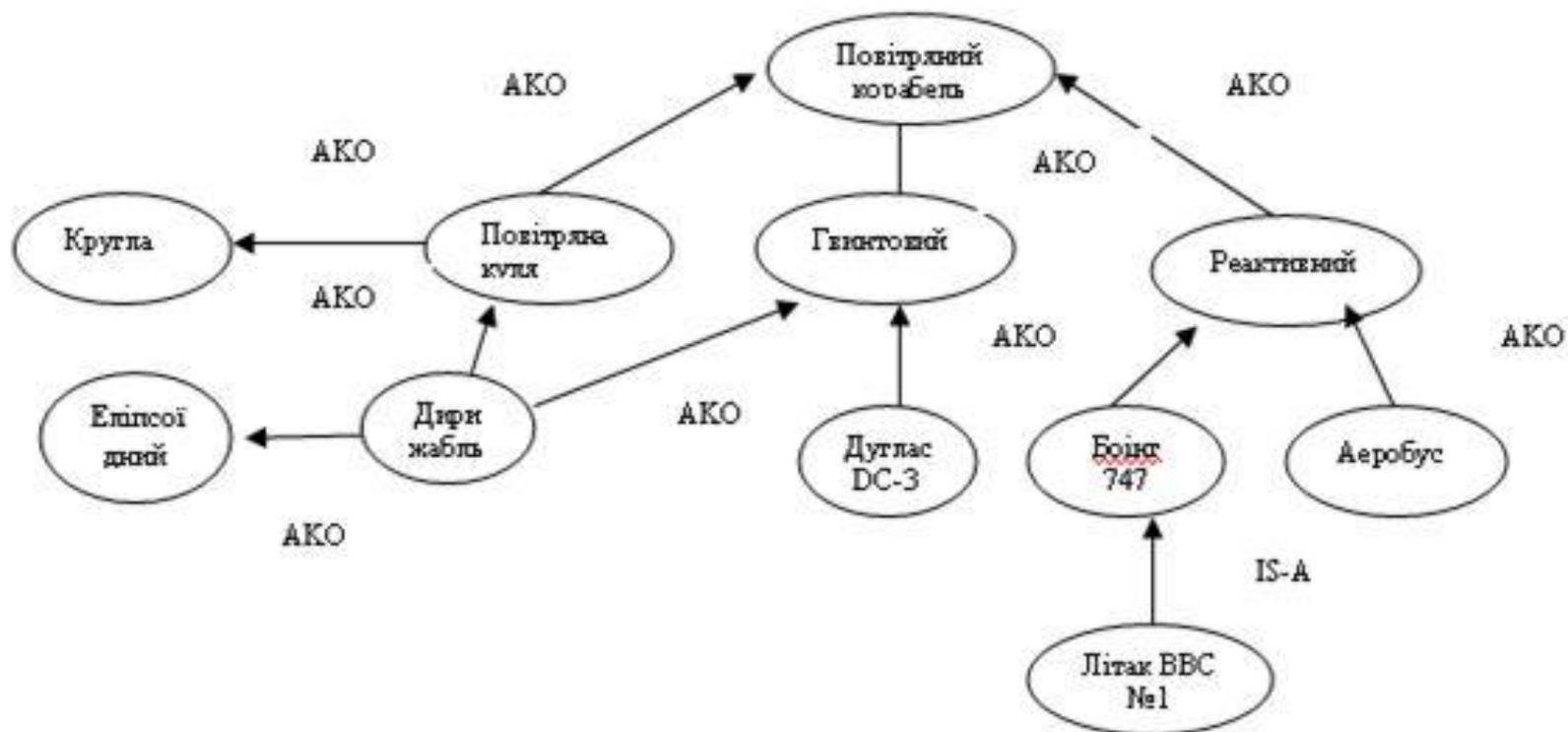


Рис. 2.1 – Приклад представлення семантичною мережею знань про літальні апарати

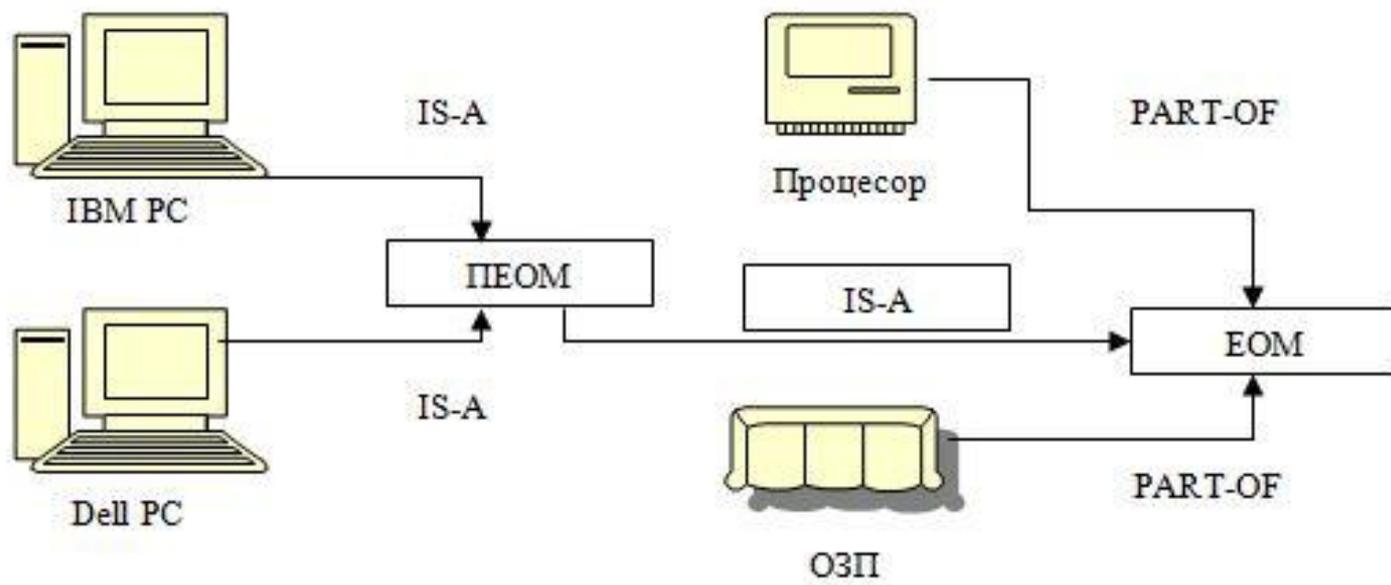


Рис. 2.2 – Приклад представлення семантичною мережею знань про обчислювальні пристрої

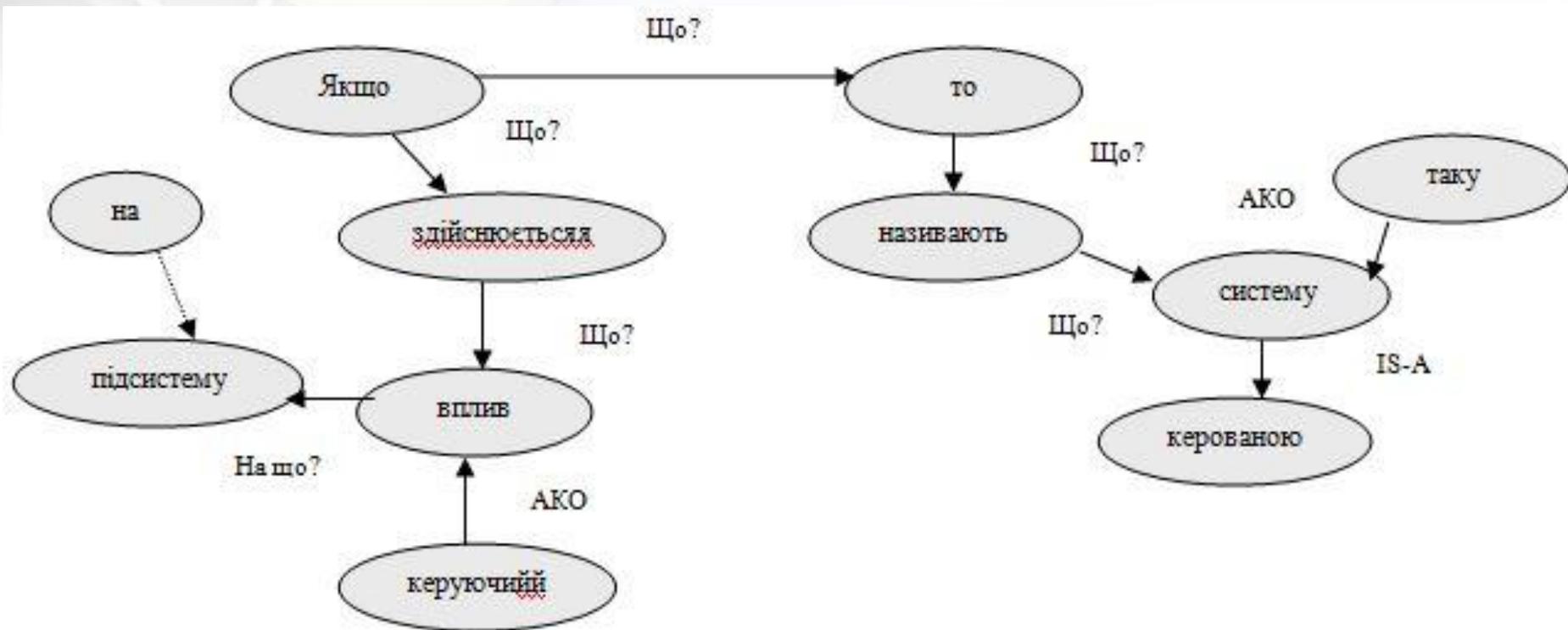


Рис. 2.3 – Приклад семантичної мережі, що відбиває речення «Якщо на підсистему здійснюється керуючий вплив, то таку систему називають керованою»

Процедуру побудови семантичної моделі представлення знань розглянемо на такому прикладі.

Нехай предметна область описується висловлюванням: «Якщо верстат закінчив обробку, робот вантажить касету з деталями на робокар, який перевозить їх на склад».

На першому етапі побудови семантичної мережі виділяються основні об'єкти і поняття, про які йде мова в даному описі, а також події і дії, які встановлюють взаємозв'язки між цими об'єктами і поняттями.

З аналізу змісту початкової фрази випливає, що в ній взаємодіють такі основні поняття: «Верстат», «Деталь», «Касета», «Робот», «Робокар» і «Склад».

До числа основних подій, які у вхідній фразі описуються дієсловом, слід віднести: «Закінчив», «Вантажить», «Перевозить».

На основі встановлених понять і подій визначаються елементарні дії:

F1 – закінчив;

F2 – вантажить;

F3 – перевозить.

Зазначені дії у вхідній фразі описані у загальній формі. Тому їх необхідно конкретизувати, зв'язавши з виділеними поняттями.

В результаті вказані дії набувають такі формулювання:

F1 – верстат закінчив обробку деталі;

F2 – робот вантажить касету на робокар;

F3 – робокар перевозить касету на склад;

F4 – касета містить деталі.

На другому етапі встановлюються зв'язки між виділеними діями.

Спочатку розглянемо зв'язок лише між двома подіями: «F1 – закінчив» та «F2 – вантажить».

Графічне цей зв'язок можна представити у вигляді фрагмента семантичної мережі, наведеного на рис. 2.4.

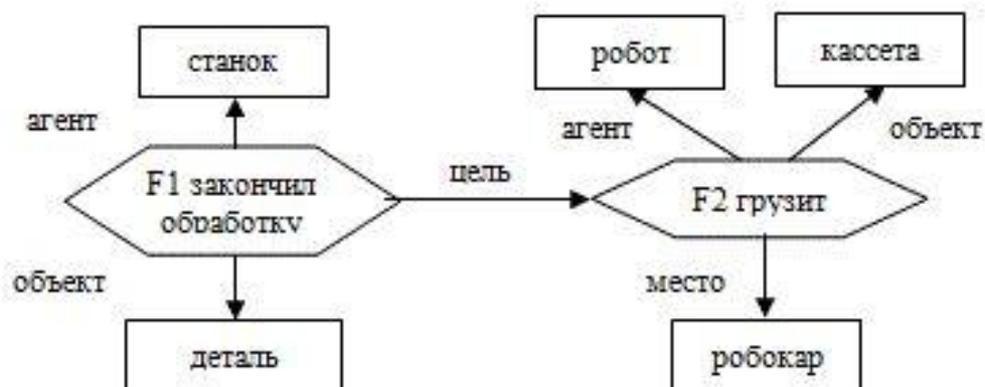


Рис. 2.4 – Фрагмент мережі, що описує зв'язок між подіями F1 та F2.

Після виконання описаної процедури для всіх передбачених дій і поєднань всіх можливих подій можна побудувати узагальнену модель семантичної мережі, яка описує знання про предметну область, що розглядається.

Для наведеного прикладу така мережа буде мати вигляд, представлений на рис. 2.5.

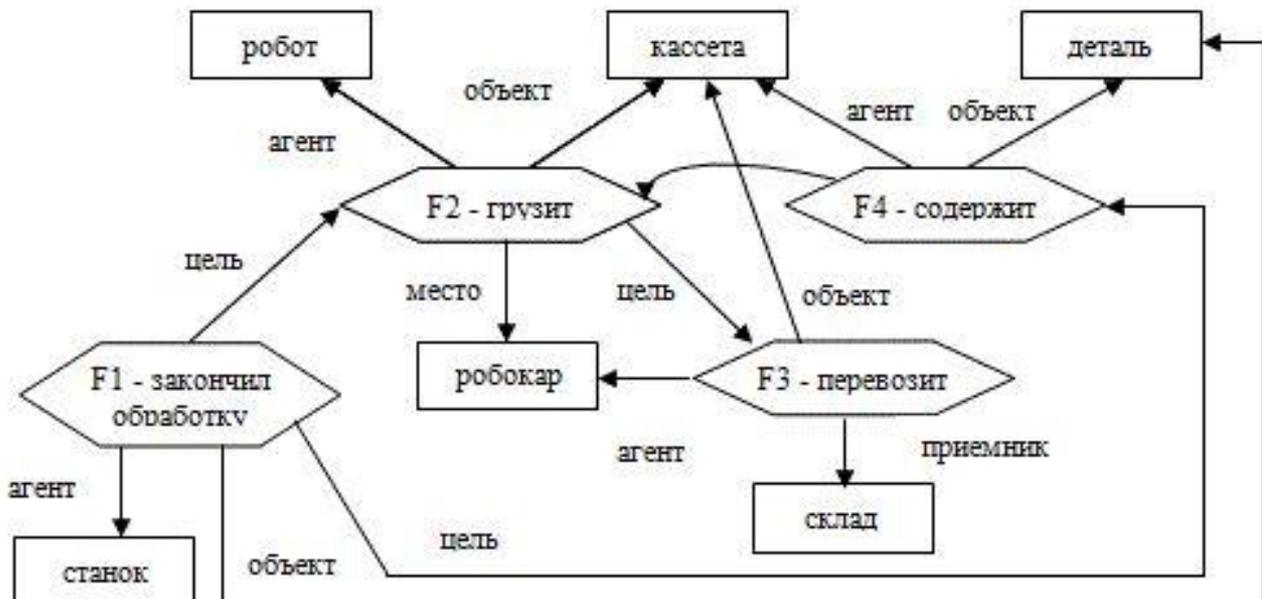


Рис. 2.5 – Узагальнена семантична мережа

Виведення рішень на семантичних мережах

На даний час визначені три основні підходи до побудови алгоритмів логічного виведення на семантичних мережах.

Перший з них оснований на так званому методі зіставлення (порівнювання). Він передбачає побудову підмережі, що відповідає запиту (питанню) користувача до інтелектуальної системи, з подальшим встановленням фрагменту базової семантичної мережі, який має аналогічну структуру.

Спочатку шуканим змінним, приписаним до вершин підмережі запиту, привласнюються гіпотетичні значення. Після встановлення зазначеного фрагменту їм привласнюються конкретні значення, які містяться у відповідних вершинах базової мережі. Далі на основі конкретизованої таким чином підмережі запиту автоматично формується відповідь на запит користувача системи.

Розглянемо приклад семантичної мережі, яка відбиває ієрархію підпорядкованості співробітників деякої організації (рис. 2.6).

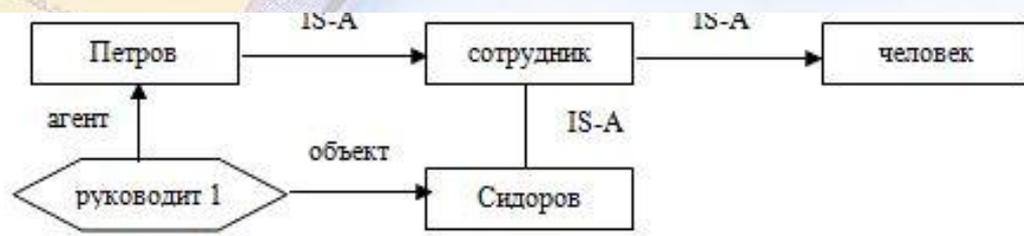


Рис. 2.6 – Семантична мережа «Підпорядкованість співробітників організації»

Наведені зв'язки показують підпорядкованість першого співробітника. Інші співробітники зв'язуються через вершини мережі «керує 2», «керує 3» і т. д.

Запит «Хто керує Сидоровим?» представляється у вигляді підмережі, наведеної на рис. 2.7.

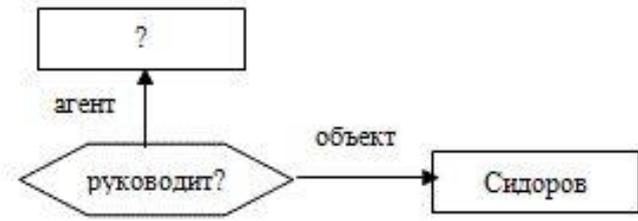


Рис. 2.7 – Семантична підмережа запиту «Хто керує Сидоровим?»

Зіставлення загальної мережі з підмережею запиту починається з пошуку вершини «керує», яка має гілку «об'єкт», спрямовану до вершини «Сидоров». Потім здійснюється перехід по гілці «агент», що і призводить до відповіді «Петров».



Другий підхід, що отримав назву «перехресного пошуку», полягає у встановленні відношень між двома концептуальними об'єктами. Формально він зводиться до побудови шляхів між вершинами графа, які відбивають концептуальні об'єкти, і виявленні вузла мережі, де перетинаються ці шляхи. На основі значень змінних, які приписані знайденому вузлу, формується відповідь користувачу системи.

Третій підхід передбачає опис семантичної мережі у вигляді множини диз'юнктивів (2.7). Це дає змогу реалізувати дедуктивні алгоритми логічного виведення, засновані на принципі резолюції Дж. Робінсона, або застосувати комбінаторні алгоритми спрямованого пошуку варіантів.

4. Фреймові моделі (структури) знань

Фреймові моделі представлення знань використовуються, головним чином, в інтелектуальних системах обробки інформації, які взаємодіють з користувачами в режимі «питання – відповідь».

Інтелект – основа цілепокладання, планування ресурсів та побудови стратегії досягнення мети.

Структура фрейму

Фрейм (англ. *Frame* – рамка, каркас, кадр) – структура даних, яка використовується для опису поняття або ситуації.

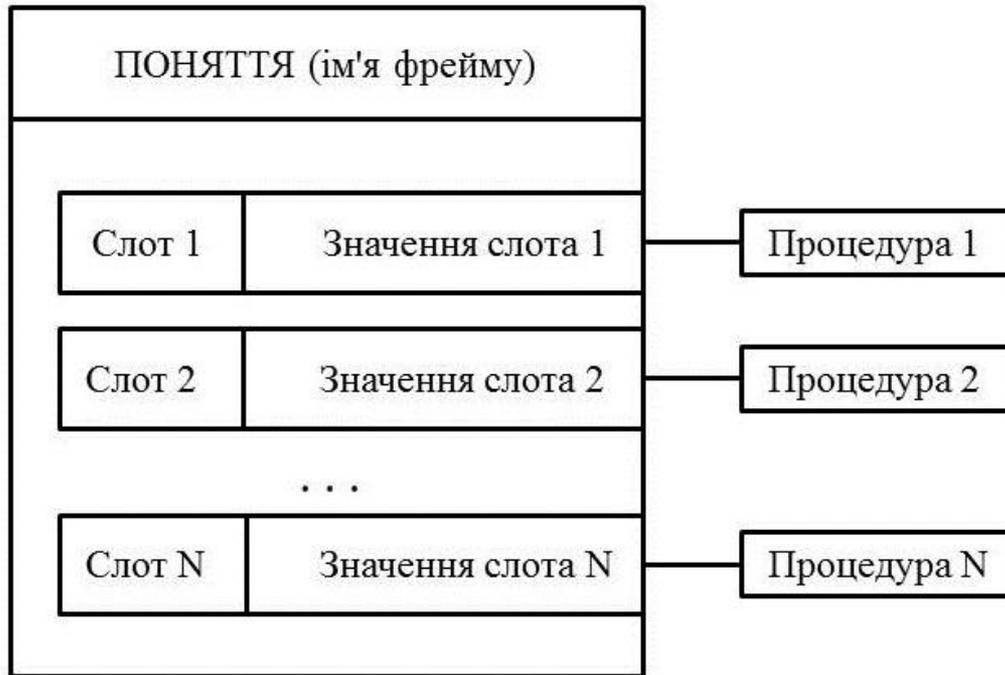
Фрейм складається із власного ім'я та слотів.

Слотом фрейму називається елемент даних для фіксації знань про об'єкт, якому відведений цей фрейм.

Слот – це атрибут, який має власне ім'я та значення.

Значенням слоту можуть бути:

- числа або математичні співвідношення;
- тексти на природній мові або програми;
- правила виводу або посилання на інші слоти цього фрейму або інших фреймів;
- набір слотів нижчого рівня;
- екземпляр атрибуту або інший фрейм.



Розрізняють **фрейми-прототипи** (протофрейми) та **фрейми-екземпляри**.

Фрейми-прототипи описують відношення між об'єктами (або дії) на абстрактному рівні, а фрейми-екземпляри – на конкретному.

Фрейм-прототип:

ПЕРЕМІЩАТИ (хто?, що?, звідки?, куди?, коли?, ...).

Відповідний фрейм-екземпляр:

ПЕРЕМІЩАТИ (лаборант Петренко, комп'ютер, лабораторія 6-102, приймальна комісія НАУ, вчора о 14-30 год., ...).

Інший приклад фрейму-прототипу:

ПЕРСОНАЛЬНИЙ КОМП'ЮТЕР (фірма-виробник?, процесор?, пам'ять?, ...).

Відповідний фрейм-екземпляр:

ПЕРСОНАЛЬНИЙ КОМП'ЮТЕР (ASUS, Intel Celeron, 512Мб, ...).

З кожним слотом можуть бути пов'язані одна або декілька **процедур**, які виконуються, коли змінюються значення слотів (уводиться нова інформація, видаляється інформація, запрошується інформація із порожнього слоту).

Найчастіше із слотами зв'язуються такі процедури:

«якщо – додано»;

«якщо – видалено»;

«якщо – треба».

Процедура «якщо – додано» виконується, коли у слот уводиться нова інформація.

Процедура «якщо – видалено» виконується при видаленні інформації із слоту.

Процедура «якщо – треба» виконується у тих випадках, коли запрошується інформація із слоту, а він виявляється порожнім.

Логічне виведення на фреймовій моделі

Задача логічного виведення на фреймовій моделі полягає у **конкретизації фрейму**, який зіставлений запиту користувача до інтелектуальної системи.

При конкретизації фрейму йому та його слотам привласнюються конкретні імена і відбувається заповнення слотів. Таким чином із протофреймів виводяться фрейми-екземпляри, тобто фрейми, які містять конкретну інформацію.

Перехід від початкового протофрейма до фрейму-екземпляру є, як правило, багатокроковим внаслідок поступового уточнення значень слотів.

Логічне виведення здійснюється шляхом обміну повідомленнями між фреймами різного рівня ієрархії.

Зв'язки між фреймами задаються значеннями спеціального слоту з ім'ям «Зв'язок».

При формуванні відповіді на запит користувача спочатку отримує управління кореневий фрейм.

Кореневий фрейм – фрейм-прототип, який містить всі елементи запиту користувача.

Далі динамічно формується ланцюжок фреймів наступного рівня ієрархії, необхідний для реалізації запиту.

Основною операцією при роботі з фреймами є **пошук за зразком**.

Зразок – це фрейм, в якому заповнені не всі слоти.

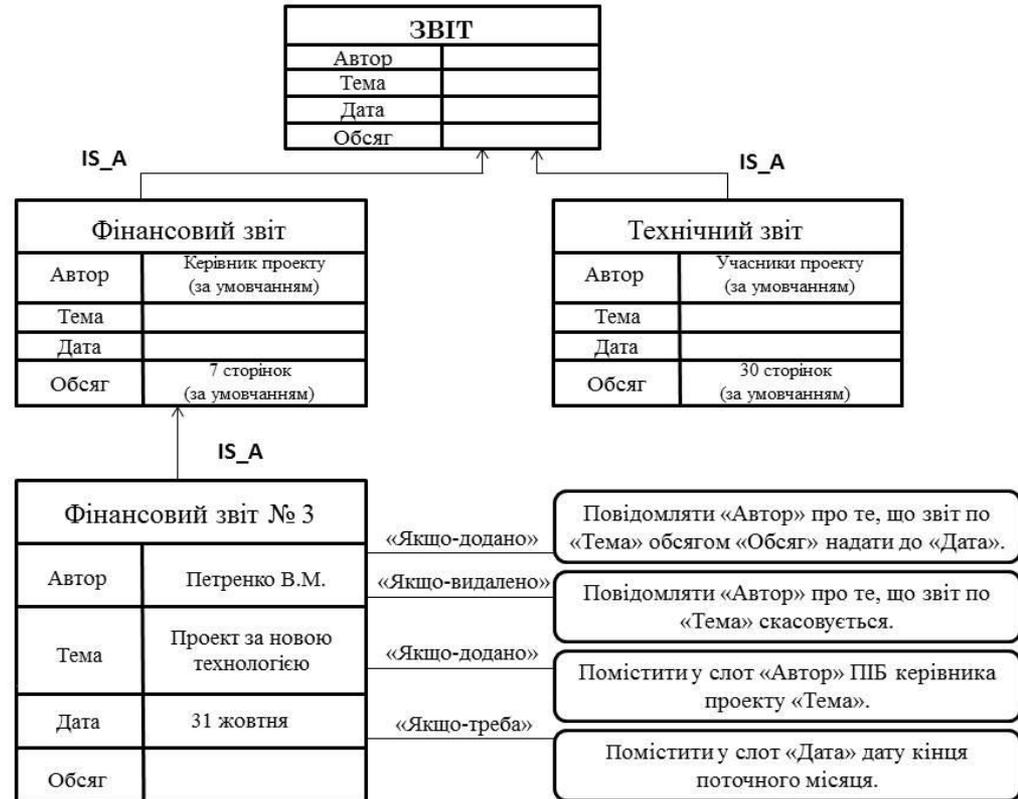
При цьому використовуються спеціальні процедури наповнення слотів конкретними значеннями, а також введення в систему нових фреймів-прототипів і нових зв'язків між ними.

Алгоритм «пошуку за зразком» передбачає такі дії:

- фіксується кореневий фрейм;
- формується ланцюжки фреймів наступних рівнів ієрархії, що ведуть від кореневого фрейму до фрейму, який відповідає запиту.

Процедура завершується заповненням конкретними значеннями всіх слотів фрейму, що відповідає запиту.

Наприклад, користувач звертається до інтелектуальної системи з запитом: «Мені потрібний фінансовий звіт про виконання проекту за новою технологією».



Фрагмент фреймової моделі, що використовується для формування відповіді на запит «Мені потрібний фінансовий звіт про виконання проекту за новою технологією»

Після введення у систему вказаного запиту виконуються такі дії:

1) Формується порожній фрейм «Фінансовий звіт №3», який зіставляється запиту до інтелектуальної системи.

2) У слот «Тема» цього фрейму вноситься текст «Проект за новою технологією».

3) Ініціюється процедура «якщо – додано», яка пов'язана із слотом «Тема».

Вона здійснює пошук керівника проекту та вносить його прізвище у слот «Автор» даного фрейму.

Якщо керівник проекту не буде знайдений, то слот «Автор» наслідуватиме значення класу, а саме текст «Керівник проекту».

4) Виконується процедура «якщо – додано», пов'язана із слотом «Автор», оскільки у цей слот було вписано нове значення, що відповідає запиту.

Ця процедура передбачає складання повідомлення, яке має бути відправленим керівнику проекту. Але для цього необхідно встановити дату виконання.

5) Виконується процедура «якщо – додано», яка переглядає слот «Дата» і, знайшовши його порожнім, активує процедуру «якщо – треба», пов'язану з цим слотом.

Процедура «якщо – треба» аналізує поточну дату та встановлює найближчу до неї дату кінця поточного м'яся, після чого вписує її у в слот «Дата».

б) Виконується процедура «якщо – додано», пов'язана із слотом «Автор», яка встановлює, що для формування відповіді на запит необхідно визначити ще одне значення, а саме – обсяг звіту.

Слот «Обсяг» не пов'язаний з процедурами і тому не може отримати конкретне значення безпосередньо. Проте на вищому від фрейму, що розглядається, рівні ієрархії знаходиться вузол загальної концепції фінансового звіту під назвою «Фінансовий звіт», що містить значення цього обсягу – 7 сторінок.

Тому процедура IS_A, яка реалізує концепцію спадкоємства властивостей класу, використовує це значення обсягу фінансового звіту і складає наступне повідомлення: «Пан Петренко В.М., підготуйте фінансовий звіт за проектом нової технології до 31 жовтня обсягом 7 сторінок».

Якщо в якийсь момент прізвище керівника проекту буде видалений із слоту «Автор», система автоматично відправить йому повідомлення, що його звіт не потрібен.

У розглянутому прикладі заповнювачами служили конкретні екземпляри атрибутів (заповнювані за умовчанням).

Крім них заповнювачами слотів можуть бути імена інших фреймів системи, на які робиться посилання, а також фасети (наприклад, «агрегат», «діапазон», «за умовчанням» тощо).