

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
КАФЕДРА ЕЛЕКТРОННОЇ ТЕХНІКИ

Методичні вказівки
до лабораторних робіт
з дисципліни
«ІНЖЕНЕРІЯ ЗНАНЬ І РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ»
ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 122 КОМП'ЮТЕРНІ НАУКИ
ДЕННОЇ ТА ЗАОЧНОЇ ФОРМИ НАВЧАННЯ

Луцьк
2023

УДК
М

Методичні вказівки до лабораторних робіт з дисципліни «Інженерія знань і розробка експертних систем» для студентів спеціальності 122 Комп'ютерні науки усіх форм навчання. / укладач: В.В. Шамаєв. – Луцьк: ДонНТУ, 2023. – 66 с.

Методичні вказівки містять основні положення щодо вимог, виконання, структури та оформлення звітів з практичних робіт і лабораторних робіт з дисципліни «Інженерія знань і розробка експертних систем» магістрами ДонНТУ, що навчаються за спеціальністю 122 Комп'ютерні науки усіх форм навчання. Наведено зразки виконання окремих алгоритмів, видів текстового матеріалу, схем, таблиць, формул та ілюстрацій.

Призначено для магістрів спеціальності 122 Комп'ютерні науки

Укладач: В.В. Шамаєв, доц., к.т.н., доц. каф. ЕТ.

Рецензент: Н.О. Маслова, доц., к.т.н., доц. каф. ПМІ

Відповідальний за випуск: О.В. Вовна, проф., д.т.н., зав. каф. ЕТ..

Затверджено навчально-методичним відділом ДонНТУ

Протокол НМВ № _ від __ . ____ .202_ р.

Розглянуто на засіданні кафедри електронної техніки

Протокол № 9 від 22.03.2023 р.

Зміст:

Вступ.	4
1. Основні положення дисципліни «Інженерія знань і розробка експертних систем»	5
2. Лабораторна робота № 1. Проектування бази знань на основі логічної моделі .	12
3. Лабораторна робота № 2. Проектування бази знань на основі продукційної моделі.	20
4. Лабораторна робота № 3. Проектування бази знань на основі семантичної мережі.	28
5. Лабораторна робота № 4. Проектування бази знань на основі фреймової моделі.	31
6. Лабораторна робота № 5. Застосування методів нечіткої логіки в інтелектуальних системах.	34
7. Лабораторна робота № 6. Розробка проекту експертної системи на мові Visual Prolog	48
Список рекомендованої літератури.	53
Додаток А. Розгорнутий словник термінів і їх визначення	55
Додаток Б. Узагальнений алгоритм структурування знань.	59
Додаток В. Приклад структури БД, як основи бази знань (БЗ).	60
Додаток Г. Структура і архітектура експертної системи.	61
Додаток Д. Методи пошуку і отримання знань.	64

Вступ

Основна мета виконання лабораторного практикуму – узагальнення, систематизація та поглиблення теоретичного матеріалу, і отримання практичних навичок вирішення базових прикладних задач в області інженерії знань і розробки експертних систем.

Виконання завдань з лабораторних робіт зумовлює поглиблене вивчення наступних тем: моделі представлення знань, класифікація знань, відмінність знань від даних, класифікація моделей знань, проектування бази знань, логіко-алгебраїчне представлення знань, продукційні моделі і системи, семантичні мережі, класифікація семантичних мереж, фрейми і фреймові моделі, Frame Representation Language (FRL), продукційно-фреймові моделі, логічні моделі, предикати, набути навички застосування методів дедукції, індукції і абдукції,

Основна мета виконання наведених завдань з лабораторних робіт полягає у наданні студентам уяви про основні принципи побудови експертних систем підтримки рішень з елементами штучного інтелекту.

Кожна тема (лабораторна робота) має мінімально необхідний попередній опис теоретичних положень, необхідних для виконання роботи, основних положень з їх практичного впровадження, формулювання завдань, порядок і приклади їх виконання, склад звіту з виконання лабораторної роботи, перелік контрольних питань, перелік додаткових навчальних джерел. Лабораторні роботи, опис яких наведено, є початковими в тематичному циклі з інженерії знань і розробки експертних систем та систем штучного інтелекту, у тому разі продукційних і фреймових моделей, семантичних моделей, експертних систем з нечіткою логікою, мультиагентних, а також експертних систем когнітивного моделювання і допомагають студентам засвоїти важливі відповідні основні тематичні поняття.

1. Основні положення дисципліни «Інженерія знань і розробка експертних систем»

Під терміном «інженерія знань» розуміють сукупність методів, моделей і технічних прийомів, спрямованих на формування систем, призначених для пошуку рішень проблем на основі наявних знань. Фактично під цим терміном розуміють методологію, теорію і технологію, що охоплюють методи аналізу, видобутку, обробки і представлення знань. Сутність штучного інтелекту полягає в науковому аналізі та автоматизації інтелектуальних функцій, властивих людині. Ця задача вирішується методом застосування знань експертів, для створення інтелектуальної системи, здатної не тільки запам'ятовувати, але й аналізувати і використовувати в подальшому знання, які зберігаються у відповідних базах знань (БЗ) [1,6,22].

Виникнення терміну «інженерія знань» відноситься до 60-70-х років ХХ сторіччя, коли у Стенфордському університеті (США) групою науковців під керівництвом Е. Фейгенбаума була розроблена система DENDRAL (пізніше – MYCIN). Обидві системи отримали назву експертних зважаючи на їх здібності накопичувати в комп'ютерній пам'яті і використовувати для вирішення проблем знання експертів. Дана область отримала термін «інженерії знань». В основі інженерії знань два підходи: перетворення знань та побудова моделей.

- I. Перетворення знань. Процес зміни експертизи і перехід від знань до їх програмної реалізації. На ньому будувалася розробка *Knowledge Based Systems*. Формат представлення знань – правила. Недоліками є неможливість подання неявних знань і різних видів знань в адекватній формі, складність відображення великої кількості правил.
- II. Побудова моделей. Створення штучного інтелекту вважається різновидом моделювання; побудова комп'ютерної моделі, покликаної вирішувати завдання в конкретній області нарівні з експертами. Модель не здатна імітувати на когнітивному рівні діяльність експерта, але дозволяє отримати аналогічний результат.

Моделі і методи інженерії знань спрямовані на розробку інтелектуальних інформаційних комп'ютерних систем, основною метою яких є отримання знань та їх подальша організація.

Одним із способів реалізації штучного інтелекту є нейронна мережа.

Машинне навчання – область розробки штучного інтелекту, спрямована на вивчення методів побудови самонавчальних алгоритмів. Необхідність у цьому виникає при відсутності чіткого вирішення конкретного завдання. У такій ситуації доцільно розробити механізм, здатний створити метод пошуку рішення, а не шукати його. «Глибоке навчання» - алгоритми машинного навчання, для роботи яких потрібна велика кількість обчислювальних ресурсів. Поняття в більшості випадків асоціюється з нейронними мережами.

Розрізняють два види штучного інтелекту: вузько направлений, або слабкий, і загальний, або сильний. Дія слабого спрямована на пошук рішення вузького списку завдань. Найбільш яскравими представниками вузько спрямованого виду відносять голосові помічники Google Assistant, Siri і Аліса. Здатності сильного штучного інтелекту дозволяють йому виконувати широке коло практичних задач.

В теорії, під «*машинним навчанням*» розуміють методи в сфері штучного інтелекту, що застосовуються для створення машини, здатної навчатися на власному досвіді. Під процесом навчання передбачають обробку машиною величезних масивів даних і пошук у них закономірностей [5,8,17,22].

Поняття ***Machine learning*** і ***Data science***, незважаючи на свою схожість, все ж різняться і справляються кожен зі своїми завданнями. Обидва інструменти є компонентами систем штучного інтелекту [5].

В нашому навчальному курсі під терміном «*машинне навчання*» ми розуміємо *алгоритми, на підставі яких комп'ютер здатний робити висновки, не дотримуючись жорстко заданих правил*. Машина шукає закономірності в складних завданнях з великою кількістю параметрів, знаходячи більш точні відповіді, на відміну від мозку людини. Результатом методу є точне прогнозування.

Data science. *Наука про способи аналізу даних і вилучення з них цінних знань та інформації (data mining)*, застосовується у сукупності з машинним навчанням з технологіями взаємодії з великими обсягами даних. *Data science* дозволяє проаналізувати дані і відшукати правильний підхід для подальшої сортування, обробки, вибірки і пошуку інформації.

Нейронні мережі – це один з видів машинного навчання, вони здатні моделювати роботу людського мозку за допомогою штучних нейронів. Їх дія спрямована на рішення поставленої задачі і самонавчання на підставі отриманого досвіду з мінімізацією помилок.

Метою машинного навчання вважається часткова або повна автоматизація пошуку рішень різних аналітичних завдань. З цієї причини машинне навчання повинне давати на підставі отриманих даних максимально точні прогнози. Результат навчання системи – прогноз і запам'ятовування результату з можливістю подальшого відтворення і вибору одного з варіантів.

Розглянемо існуючі види класифікації машинного навчання [5,17,22].

За ознакою наявності експерта (фахівця з машинного навчання):

- *З експертом.* Застосовується, коли використання знань передбачає навчання комп'ютера розпізнавання сигналів і об'єктів.
- *Без експерта.* Базується на алгоритмах, які виявляють подібності і відмінності об'єктів, аномалій з подальшим їх розпізнаванням.
- *З підкріпленням.* Застосовуються, якщо комп'ютер повинен коректно виконати завдання у зовнішньому середовищі з безліччю можливих рішень.

За видом використовуваних алгоритмів:

- *Класичне навчання.* Алгоритми навчання, розроблені для статистичного аналізу, тобто для роботи з даними.
- *Глибоке навчання та нейронні мережі.* Сучасний підхід до машинного навчання. Нейромережі застосовуються в задачах розпізнавання відео і зображень, машинного перекладу, в процесах прийняття і аналізу рішень.

Зазначимо, що в інженерії знань також існують моделі, які поєднують кілька різних підходів. При грамотній комбінації різних видів і алгоритмів машинного навчання можлива автоматизація рутинних процесів, але творча частина – складання і виконання стратегій – залишається за людиною.

В контексті створення штучного інтелекту виділяють дві головні проблеми:

- 1) *Правомірність визнання за людиною свідомості та волі, і, відповідно, для визнання штучного інтелекту розумним потрібні ті ж самі підходи;*
- 2) *Порівняння штучного інтелекту з людським розумом і його здібностями, не враховує індивідуальних особливостей усіх систем і тягне за собою їх дискримінацію через відкидання значення їх діяльності.*

Проблеми створення штучного інтелекту криються в тому числі у формуванні образів і асоціативної пам'яті [5,17].

Подібні схеми у людини формуються асоціативно, на відміну від роботи комп'ютера; в протилежність людському розуму, комп'ютер відшукує конкретні папки і файли, а не вибирає схеми асоціативних зв'язків.

Інша проблема – переклад тексту (людської мови) програмами-перекладачами здійснюється автоматично, а підсумковий результат представлено набором слів. Для коректного перекладу від комп'ютерної системи вимагається розуміння змісту фрази, що сьогодні ще важко реалізується штучним інтелектом.

Відсутність проявів волі у штучного інтелекту також вважається проблемою на шляху до його створення. Іншими словами, у комп'ютера відсутні особисті бажання, на відміну від наявності можливостей для проведення складних логіко - асоціативних операцій.

Сучасні системи штучного інтелекту поки не мають стимулів до подальшого існування і вдосконалення, вплинути на це можна за допомогою створення логіко - асоціативного зворотного зв'язку між комп'ютером і людиною і поліпшення алгоритмів самонавчання комп'ютера.

Вищенаведене, обумовлює той факт, що сьогодні використання систем штучного інтелекту в наукових, промислових і інших цілях поки що не дуже ефективне. Розробники інтелектуальних комп'ютерних систем (зокрема, автономних роботів) на основі систем штучного інтелекту стикаються з проблемою неспроможності штучного інтелекту повноцінно самонавчатися, автоматично розпізнавати, формалізувати і коректно аналізувати отриману інформацію в режимі реального часу.

Початок досліджень знань відноситься до першої половини 60-х років минулого століття, і йому присвячено роботи Е. Фейгенбаума, Г. Саймона, Д. Слейгла, А. Ньюелла. З середини 70-х років XX ст. у галузі штучного інтелекту почалися дослідження, пов'язані зі спробами автоматизувати методи вирішення людиною завдань у сфері професійної і науково-технічної діяльності: ухвалення рішень, планування, проектування і конструювання, доведення теорем, ігри й інші завдання, які прийнято називати інтелектуальними. Систематичні дослідження за уявленням знань в інтелектуальних системах проводяться як у нашій країні, так і за кордоном. В нашій країні значного просування добилися дослідницькі колективи, керовані Д. А. Поспеловим, Е. В. Поповим, А. С. Наріньяні та ін. ^[1,8,22].

Розглянемо основні проблемні питання і термінологію, які застосовуються в інженерії знань. (*Розгорнутий словник термінів і їх визначення наведено у додатку А*).

Основні труднощі пов'язані з тим, що системи, засновані на знаннях, як правило, розробляються для погано формалізованих предметних галузях (ПРГ), в яких знання неточні, неповні, суперечливі й мінливі. Тож необхідні розроблення спеціальних засобів представлення таких знань, а також пошук ефективних методів роботи з ними.

Робота із знаннями лежить в основі сучасного періоду розвитку штучного інтелекту. Будь-яка предметна галузь діяльності може бути описана у вигляді сукупності відомостей про структуру в цій галузі, основних її характеристиках, процесах, що відбуваються в ній, а також про способи виконання завдань, що виникають. Усі ці відомості утворюють знання про предметну галузь. Для вирішення задач у даній предметній галузі необхідно зібрати знання про неї і створити концептуальну модель цієї області.

Джерелами знань можуть бути документи, мережі, хмарні технології, статті, книги, фотографії тощо. Із цих джерел слід «витягнути» знання, що містяться в них. Цей процес виявляється достатньо важким, бо треба наперед оцінити важливість і потрібність тих або інших знань для роботи інтелектуальної системи. Фахівці, які займаються питаннями, пов'язаними із знаннями, називаються інженерами зі знань або інженерами знань ^[1,8,22].

Робота зі знаннями складається з наступних етапів:

1. *здобування знань із різних джерел*: (формалізація якісних знань; інтеграція знань);
2. *придбання знань від професіоналів*: (організація роботи з експертами; оцінка і формалізація знань; узгодження знань);
3. *представлення знань*: (моделі знань - семантичні мережі; фрейми; онтології; логічні системи; - продукційні системи представлення знань; бази знань);
4. *маніпулювання знаннями* (поповнення знань; класифікація знань; узагальнення знань; висновки на знаннях: методи резолюції; квазіаксіоматичні системи; міркування за допомогою знань);
5. *пояснення на знаннях*.

Одержані від експертів знання оцінюють з погляду їх відповідності отриманим раніше знанням і формалізують для введення в пам'ять системи.

Знання, одержані від різних експертів, погоджують між собою, щоб уникнути випадків їх несумісності і суперечливості. Введення в базу знань (БЗ) знань не проблема, на відміну від виявлення і введення суб'єктивних експертних знань. Для витягу і формалізації експертних знань розроблено багато стратегій і моделей подання знань. (В додатку Б наведено узагальнений алгоритм структурування знань).

В інтелектуальних системах для зберігання й використання знань створюються спеціальні системи представлення знань, що включають сукупність процедур, необхідних для запису знань, добування їх із пам'яті та підтримки зберігання знань у робочому стані [1,8,22].

Системи представлення знань формуються у вигляді баз знань (БЗ). Розрізняють три рівні представлення знань: (1) *користувача - проблемно-орієнтований*; (2) *«tool»-орієнтований* (знання представлені мовою програмування системи); (3) *системний рівень* (біти) - іманентний системі (внутрішньо - комп'ютерне уявлення) [8,17,22]

Приймаючи до уваги те, що усі системи штучного інтелекту є системами, що базуються на знаннях, та основними властивостями таких систем є: роздільне зберігання знань, представлених у символічній формі, та компонентів обробки цих знань; мінливість системи робити висновки і приймати рішення на підставі збереженої інформації, неявно представлені в системі; можливість виводу пояснень, тобто, на вимогу користувача вивести зрозумілу для нього логіку міркувань (кроки резолюції); здатність до навчання, тобто, здатність виводити нові знання на підставі одержаної раніше інформації (про успіх або невдачу минулого сеансу роботи з програмою).

Технологія побудови експертних систем - це процес взаємодії розробників експертної системи з експертами в певній предметній області.

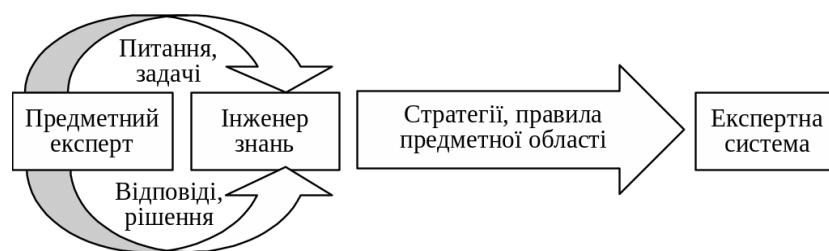


Рис.1. Загальна схема розробки експертної системи [17]

Основа експертної системи - база знань (БЗ), її особливістю є *інституційна пам'ять* (експерти можуть з часом мінятися, а їхній досвід – лишається). Важливою характеристикою експертних систем є наявність *прогностичних можливостей*, використання умовної логіки «If... Then... Else...», тобто, системи можуть видавати відповіді на поведінку в конкретній ситуації і показувати, як зміняться ці відповіді у нових ситуаціях [8,17, 22].

В експертних системах використовуються три методи подання знань:

- Правила (подання знань, побудоване на правилах і таке, що базується на використанні виразу Якщо (умова IF) - То (дія DO). Зіставлення частин Якщо правил з фактами може створити *завершену послідовність висновків*.).
- Семантичні мережі (використовують мережу вузлів, пов'язаних між собою відношеннями і організованих ієрархічно. Вузол надає концепцію, яка може бути описана за допомогою атрибутів та значень, пов'язаних з цим вузлом).
- Фрейми (подання знань, що базуються на використанні фреймів).

Архітектура інформаційної системи з інтелектуальним інтерфейсом на основі використання баз знань наведена на рисунку 2.

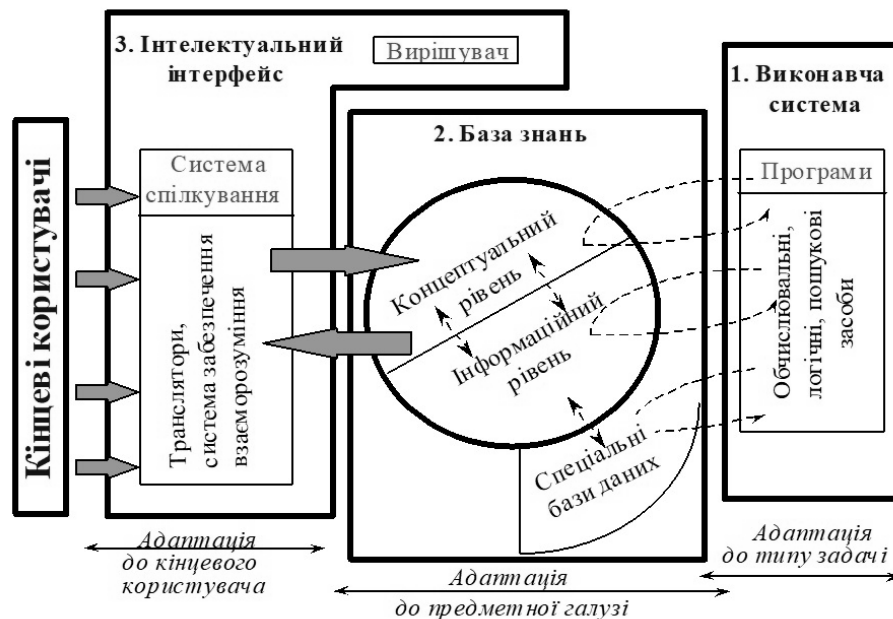


Рис. 2. Архітектура інтелектуальної інформаційної системи.

Основна різниця між експертними системами та традиційними програмами полягає в тому, що *експертні системи маніпулюють знаннями, а звичайні програми - даними*.

Функціонал, структура і архітектура різновидів ЕС наведені в додатку Г.

2. Лабораторна робота № 1.

Тема: Проектування бази знань на основі логічної моделі.

Мета: Дослідження логічної моделі представлення знань, формування навичок формалізації знань і створення баз знань (БЗ) для вирішення прикладних завдань.

Теоретичні відомості:

База знань - це сукупність відомостей (про реальні об'єкти, процеси, події або явища), що відносяться до певної теми або задачі, організована так, щоб забезпечити зручне представлення цієї сукупності як в цілому так і будь-якої її частини. Це означає, що *система управління базою знань* (саме знань, а не даних) повинна забезпечити уявлення і обробку моделі, зіставною по своїй складності з моделлю що використовується свідомістю людини.

Важливий параметр бази знань - *якість знань*, що накопичені в ній. Бази знань містять *релевантну інформацію*, мають сучасні системи пошуку інформації і детально пророблену *структуру і формат знань*.

Релевантність (англ. *relevance*) - міра відповідності отриманого результату бажаному (тобто, відповідності результатів пошуку завданню поставленому в пошуковому запиті). Релевантність визначає, наскільки повно той або інший документ відповідає критеріям, вказаним в запиті користувача.

Головна мета створення простих баз знань - допомогти не фахівцям знайти існуючий опис способу вирішення якої-небудь проблеми в певній предметній області.

База знань - важливий компонент інтелектуальної системи. Найбільш поширений клас інтелектуальних систем - *експертні системи*.

Експертні системи призначені для знаходження способу вирішення специфічних проблем, базуючись на записах баз знань і на користувацькому описі ситуації.

В нашому навчальному курсі ми досліджуємо особливості створення саме експертних систем, тому вміння розробки баз знань – є першим і головним етапом створення ЕС.

База знань – це основний компонент експертної системи. Кожна база знань містить факти і правила.

Факти – це фрази без умов, вони містять твердження, які завжди абсолютно вірні. Правила містять твердження, істинність яких залежить від деяких умов, включених в тіло правила. Факти містять короткострокову інформацію (тобто вони можуть мінятися, наприклад, в ході час консультації).

Правила – містять довготривалу інформацію про те, як породжувати нові факти або гіпотези з того, що зараз відомо.

Основна відмінність застосування баз знань полягає в тому, що БЗ володіє «творчими» можливостями. Факти в базах даних (БД) зазвичай пасивні: вони там є, або їх немає. База знань, на відміну від БД активно намагається поповнити інформацію, якої бракує.

Логічна модель представлення знань.

Основна ідея логічного підходу полягає в тому, щоб розглядати всю систему знань необхідну для вирішення прикладних завдань, як сукупність фактів (тверджень), які представляються як формули в деякій логіці. Знання відображаються сукупністю таких формул, а отримання нових знань зводиться до реалізації процедур логічного висновку.

В основі логічних моделей представлення знань лежить поняття формальної теорії, кортеж S:

$$S = \langle B, F, A, R \rangle,$$

де: B - зліченна множина базових символів (алфавіт);

F - множина, елементи якої називають формулами;

A – виділена підмножина апріорі справедливих формул (аксіом);

R - зліченна множина відношень між формулами, яку називають правилами висновку.

Факти представляються як формули в деякій логіці (першого або вищого порядку, багатозначної, нечіткої або ін.) Система знань відображається сукупністю таких формул і, представлена в комп'ютерній системі утворює БЗ.

Формули неподільні і при модифікації БЗ можуть лише додаватися або віддалятися.

Логічні методи забезпечують розвинений апарат виведення нових фактів з тих, які явно представлені в базі знань (БЗ).

Основним примітивом маніпуляції знаннями є операція висновку.

На першому етапі необхідно виконати формалізацію завдання.

Формалізація завдання. Уявимо собі, що фахівця - біолога по радіозв'язку просять визначити вид деякої рослини. Оскільки він не бачить конкретний екземпляр, а питає у не фахівця у області ботаніки, то задає додаткові, уточнюючі питання, щоб одержати відомості, необхідні для визначення рослини. Спробуємо представити алгоритм такої радіо консультації у графічному вигляді (рис. 1.1) [6].

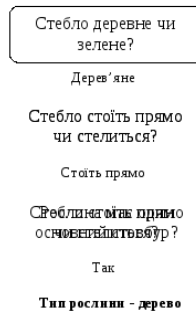


Рис. 1.1. «Телефонна консультація» [6]

Результатом такої консультації буде логічний висновок: «Ґрунтуючись на Вашій відповіді, можна припустити, що тип рослини – дерево».

Проте, наведена діаграма ілюструє тільки один з можливих варіантів питань і відповідей (тобто експертизи). Аналогічним чином можна представити і все інші варіанти відповідей, і хід консультації. Можливість проведення такої консультації (визначення послідовності питань і їх змісту) обумовлена знаннями фахівця (експерта) в конкретній області.

Представлення знань у вигляді «дерева рішень» Базуючись на знаннях експерта діаграму всіх можливих результатів даної консультації можна представити у наступному вигляді (рис. 1.2) [6].

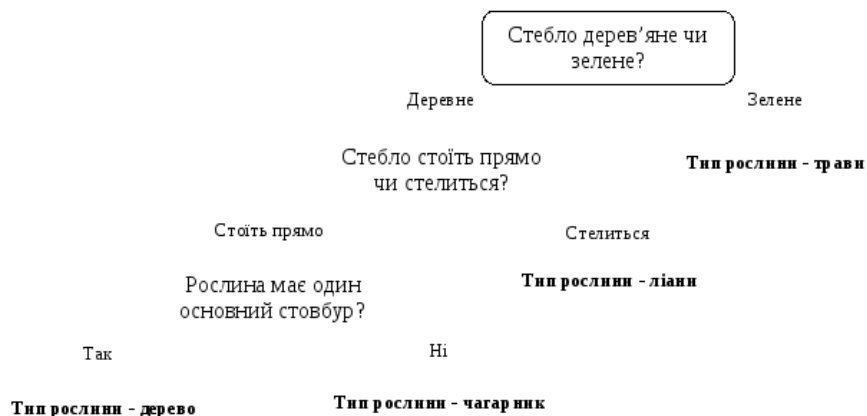


Рис. 1.2. Дерево рішення задачі [6]

Таке графічне представлення моделі даних називається «деревом рішення», воно об'єднує всі гілки розгалуженого алгоритму пошуку виду невідомої рослини. Але, у випадку, якщо консультація експерта повинна бути глибшою і визначати, наприклад, клас рослини, то для «типу рослини – дерево» необхідно побудувати своє «дерево рішень», яке після отримання знань від експерта також можна представити у графічному вигляді. Нова частина буде «піддеревом» початкового «дерева рішень» [6].

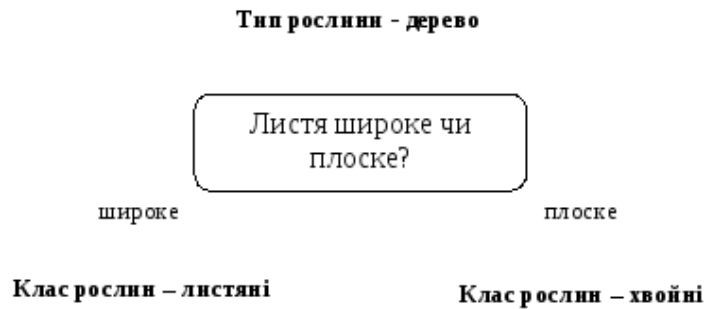


Рис. 1.4. «Піддерево-1» рішення задачі [6]

Існує дві причини, з яких алгоритм «дерева рішень» доцільно розподілити на окремі фрагменти:

- «дерево рішень» швидко стає довгим і важко оглядовим;
- ділення «дерева рішень» на секції спрощує запам'ятовування мети, яка переслідується в процесі придбання знань.

Коли «піддерево» створене, завершальна його частина може бути скопійована в корінь новостворюваної гілки «дерева рішень», і для неї на основі знань, одержаних від експерта, створюється своє «піддерево рішень» [6].

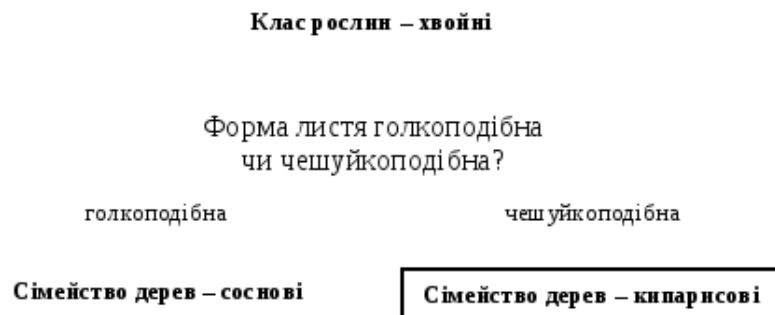


Рис. 1.5. «Піддерево-2» рішення задачі [6]

Діаграми, приведені на рисунках 1.3 – 1.5 – це модель незавершеної ботанічної бази знань (БЗ), яка вирішує тільки частину загального завдання.

На основі вище наведеного, можна зробити висновок, що при розробці моделі БЗ будь-якої предметної області на основі «дерева рішень» необхідно:

- загальне завдання розподілити на ряд підзадач;
- для кожної з підзадач розробити своє «дерево рішень» (для спрощення створення і відладки БЗ).

В рамках логічної моделі знання представляються в системі логіки предикатів першого порядку^[7].

Наведемо приклади є логічними моделями подання фактів (в даному випадку – «Самощенко відвідує лекції» і «Самощенко – студент») за допомогою предикатів і носять назву атомарної формули.

ВІДВІДУВАННЯ (Самощенко, лекції)

СТУДЕНТ (Самощенко)

Наступні приклади є правильно побудованими логічними формулами, що включають квантори існування $\exists$ і спільності $\forall$. ($\exists x$) [ДЕЛЬФІН (x) $\vee$ РОЗУМНИЙ (x)]

($\forall x$) [ВОРОНА (x) - "КОЛІР (x , Чорний)]

Ці формули можуть бути інтерпретовані так: «якийсь дельфін наділений розумовими здібностями» і «все ворони мають чорне забарвлення».

Логічний висновок здійснюється за допомогою силогізму (якщо з А слід В, а з В слід С, то з А слід С).

Перевагами логічної моделі представлення знань є єдиність теоретичного обґрунтування і можливість реалізації системи формально точних визначень і висновків^[7].

Однак при вирішенні достатньо складних задач, спроба представити неформалізовані знання експерта, серед яких переважають евристики, в системі строгої логіки наштовхується на серйозні перешкоди.

Це пов'язано з тим, що на відміну від строгої логіки, так звана, «людська логіка» володіє нечіткої структурою. Тому велика частина досягнень в області систем з базами знань до справжнього моменту пов'язана із застосуванням нелогічних моделей.

Розглянемо кілька методів створення баз знань (БЗ) на основі логічної моделі представлення знань, на мові Visual Prolog з використання обмеженої кількості типів логічних структур^[8].

Рекурсія - це спосіб опису функцій або процесів через самих себе. Рекурсія використовується як основа для запису рекурсивних конструкцій. *Рекурсивність* - це властивість мови програмування, яка полягає в тому, що конструкції мови допускають явне включення конструкції одного типу. Це дозволяє написати програму, використовуючи невелику кількість типів логічних структур, що досягається за рахунок використання рекурсивних процедур, тобто процедур, назва яких з'являється хоча б в одному виклику з його тіла. Ітеративні процедури є окремим випадком рекурсивних процедур.

Введемо ще одне відношення – *ancestor (предок)*. Визначимо його в термінах батьківського відношення. Відносини предків можна представити за допомогою двох правил:

1. Правило визначення прямих предків

X – предок Z, якщо X – батько Z

2. Правило визначення далеких предків, тобто тих, між якими існує ланцюжок батьківських відносин за типом батьків.

Для всіх X і Z, якщо є такий Y, що:

- a) X – батько Y (прямий предок);
- б) Y – предок Z (віддалений предок).

У мові Visual Prolog є *функція оптимізації хвостової рекурсії*. Копії викликаних предикатів не створюються, а обчислення виконуються в області пам'яті батьківського присудка, що дозволяє підвищити ефективність розрахунків [8].

Розглянемо приклад недетермінованого рекурсивного предикату. Наступна програма на мові Visual Prolog рекурсивно визначає відносини «предок» як транзитивне замикання батьківських відносин.



class facts - relatives

parent: (string Батько, string Дитина).

clauses

parent("Микола", "Марія").

parent("Ганна", "Марія").

parent("Марія", "Павло").

parent("Марія", "Петро").

parent("Петро", "Степан").

class predicates

ancestor: (string Предок, string Нащадок) nondeterm (o,o) (i,o).

clauses

ancestor(X, Y):-

parent(X, Y).

ancestor(X, Y):-

parent(X, Z),

ancestor(Z, Y).

run():-

ancestor(X, Y),

write(X, " - ", Y), nl,

fail;

_ = readLine().

Завдання:

Скомпілювати і записати мовою логіки предикатів базу знань (БЗ) логічного типу, яка містить:

- 1) не менше 15-20 фактів,
- 2) мінімум 4-8 правил.

Вказівки до роботи:

1. Визначте об'єкти предметної області, які будуть описуватися базою знань. Виділіть їх властивості і зв'язки між ними, які будуть описані.
2. Складіть список предикатів, що описують властивості предметів і відносини.
3. Складіть список фактів про предметну область у вигляді формул (виразів) на мові логіки предикатів.
4. Визначити правила / умови виводу для отримання нових знань про об'єкти предметної області (див. приклад «предок (X, Y)»)
5. Запишіть ці правила мовою логіки предикатів, використовуючи символи: (кон'юнкція, диз'юнкція, заперечення, імплікація, еквівалентність) $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$

P.S. Остання цифра залікової книжки – кількість фактів (якщо менше 5, тоді 15, в інших випадках - 20). Передостання цифра залікової книжки – кількість правил (якщо менше 5, тоді 4, в інших випадках -8).

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Короткий виклад основних теоретичних понять.
3. Постановка завдання з характеристикою порядку виконання робіт.
4. Результати роботи програми для різних початкових ситуацій (скріншоти екрану з результатами роботи програми). Пояснення за результатами роботи програми.
5. Пояснення отриманих результатів.
6. Текст (лістинг) програми.
7. Відповіді на контрольні питання.

Контрольні питання:

1. Охарактеризувати особливості логічної моделі представлення знань.
2. Основні типи моделей представлення знань.
3. Поняття предикатів. Поняття висловлювання.
4. Особливості формування логічної моделі БЗ на мові PROLOG (змінні, структура, алфавіт, оператори тощо)
5. Розробіть логічну модель представлення знань для заданої предметної області згідно свого варіанту (5-7 фактів, 2-3 правила).
6. Правила використання присудку.

3. Лабораторна робота № 2.

Тема: Проектування бази знань на основі продукційної моделі.

Мета: Дослідження продукційної моделі представлення знань, формування навичок формалізації знань і створення баз знань (БЗ) для вирішення прикладних завдань.

Теоретичні відомості:

Продукційна модель представлення знань базується на використанні продукцій. У продукції немає єдиної, окремої теорії як і немає чіткого математичного апарату, тому процедури і етапи створення баз знань на основі продукцій можуть значно відрізнятися. Продукційні моделі були запропоновані Е. Постом у 70-і роки XX віку [2,15,16,23].

Асоціативні зв'язки між поняттями предметної області та відповідними реакціями виражають у вигляді правил-продукцій виду «IF-TO», що адекватно відображують процеси логічного виводу людини.

Бази знань продукційного типу є сукупністю незалежних правил-продукцій та понять предметної області.

У загальному вигляді продукцію зображають у вигляді:

$$R = \langle I, Q, P, A \Rightarrow B, N \rangle,$$

де: I - ім'я продукції;

Q - сфера застосування продукції (метазнання системи);

P - умова застосування ядра продукції;

$A \Rightarrow B$ - ядро продукції,

A - умова продукції;

B - наслідок продукції;

N - послідовність дій після застосування продукції.

Ядро продукції найчастіше має вигляд «якщо A то B ». У базах знань інтелектуальних систем умова продукції A виступає як деяке речення-зразок, логічний вираз, за яким здійснюється пошук у базі знань. Наслідок продукції B виступає як дія, процедура, яка виконується при успішному завершенні пошуку. При цьому B може бути і вказівкою на одержання нової інформації.

Системи обробки знань, які використовують продукційні правила, отримали назву *продукційних систем*. До складу продукційної системи входять база правил, база даних та інтерпретатор правил (рис.2.1) [15,16].

База правил - це область пам'яті, яка містить базу знань - сукупність знань, які представлені в формі правил вигляду «якщо-то»; база даних - це область пам'яті, яка містить фактичні дані (факти), що описують дані, які вводяться, і стан системи.

Бази даних у різних системах мають різну форму, однак усі вони можуть бути описані як група даних, які містять ім'я даних, атрибути і значення атрибутів. Інтерпретатор є механізмом виведення, тобто тим компонентом системи, який формулює висновок, використовуючи базу правил і базу даних.

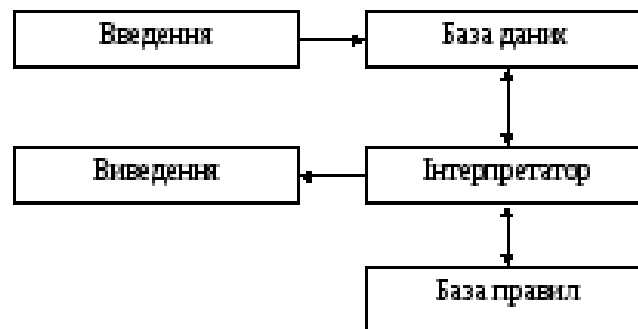


Рис. 2.1. Схема інтелектуальної системи на основі продукційної моделі

Математичний опис продукційної моделі - це **кортеж «і, С, R, F»**

де: і – унікальний ідентифікатор продукції у базі знань;

С – ядро продукції, зазвичай має вигляд імплікації – $f_1 \wedge f_2 \wedge f_3 \Rightarrow F$;

R – правило використання продукції, в математичній моделі описується як кон'юнкція – $f_1 \wedge f_2 \wedge f_3$. Якщо кон'юнкція приймає значення true, продукційну модель можна використовувати для прийняття рішень;

F – набір функцій, які повинні бути використані, або набір даних для логічного виводу, або набір даних для фактологічної бази.

Розглянемо приклад інтелектуальної системи на основі БЗ продукційної моделі, яка повинна керувати налаштуваннями навантаження в електричній мережі, керуючи джерелами енергозабезпечення (сонячної та вітрової). Результатом роботи системи повинна бути оптимальне завантаження мережі, діапазон якого визначено завдяки експертній оцінці.

Спочатку визначимо факти, якими буде оперувати наша система:

f_1 – чи працюють тільки сонячні панелі;

f_2 – чи працюють лише вітрові генератори;

f_3 – чи 100% мережа навантажена від сонячних панелей;

f_4 – чи 100% мережа навантажена від вітрових генераторів;
 f_5 – чи енергія від сонця;
 f_6 – чи енергія від вітру;
 f_7 – чи потік енергії з обох джерел;
 f_8 – крок переходу на те чи інше джерело енергії.

На етапі формування фактологічної бази даних системи, доцільно, по-перше, визначити об'єкти, з якими буде працювати система; по друге, визначити їх властивості; а вже з них обирати ті, які необхідні для створення бази знань (БЗ). Не варто відкидати властивості об'єктів, які було визначено, але не обрано для фактологічної бази. Продукційні системи можна масштабувати на етапі проектування.

Почнемо формувати базу правил нашої продукційної системи:

$\langle 0, \langle f_1 \wedge f_5 \rangle, \langle \neg f_4, \neg f_7 \rangle, \langle \text{«Увімкнути контролер сонячних панелей}(f_8)\text{»} \rangle \rangle$

$\langle 1, \langle f_2 \wedge f_6 \rangle, \langle \neg f_3, \neg f_7 \rangle, \langle \text{«Увімкнути контролер вітрогенератора}(f_8)\text{»} \rangle \rangle$

$\langle 2, \langle f_1 \wedge f_2 \wedge f_5 \rangle, \langle \neg f_3, \neg f_7 \rangle, \langle \text{«Вимкнути контролер вітрогенератора()»} \rangle \rangle$

$\langle 3, \langle f_1 \wedge f_2 \wedge f_6 \rangle, \langle \neg f_4, \neg f_7 \rangle, \langle \text{«Вимкнути контролер сонячних панелей()»} \rangle \rangle$

На даному етапі чотирьох продукцій повинно вистачити.

Промовимо продукцію: «Продукція нуль, ЯКЩО (f_1) Увімкнути контролер вітрогенератора І (f_5) Енергія від вітру ТОДІ (блок F) Викликати функцію з параметром f_8 – Увімкнути контролер сонячних панелей На(f_8) ЛИШЕ ТОДІ І ТІЛЬКИ ТОДІ, ЯКЩО (блок R) (f_3) Вітрогенератор НЕ працює на повну потужність і (f_7) Вихідної потужності енергії недостатньо».

На наступному кроці розробляємо (або обираємо з існуючих) алгоритм за яким обробляються продукції.

Загальну схему структури бази знань (БЗ) на основі моделі продукційного представлення знань зображено на рисунку 2.2, а алгоритм вирішувача наведено на рисунку 2.3.

Структура проекту бази знань (БЗ) складається з наступних компонентів:

– «*Система логічного вводу*» – компонент, задача якого переробити вхідні дані у зрозумілий для бази знань формат. Однією з можливих функцій є розподіл даних ^[15,16,23].

Наприклад, якщо деякі дані є параметрами фактологічної бази даних, а деякі параметрами, які конфігурують роботу вирішувача, – система логічного вводу застосує їх відповідно до їх ролі у базі знань (БЗ) [10].

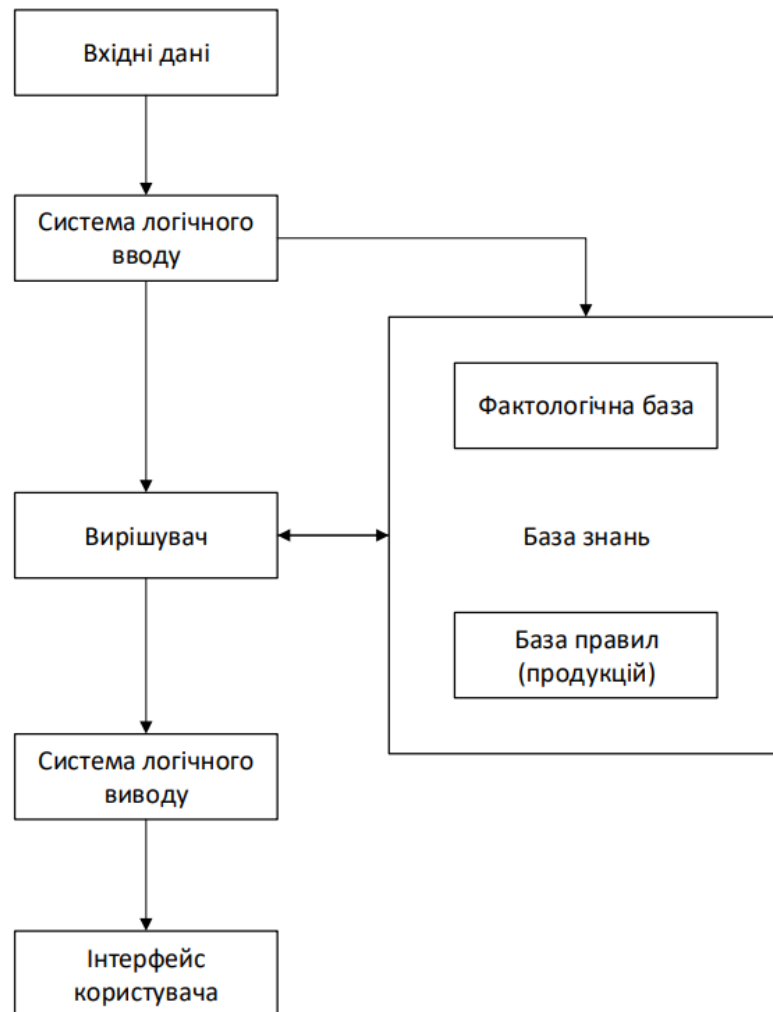


Рис.2.2. – Загальна схема структури бази знань (БЗ) на основі моделі продукційного представлення знань

– **«База знань»** – базу знань (БЗ) можливо розділити на дві частини: фактологічна база та база правил. Така структура характерна для більшості інтелектуальних систем; в залежності від виду представлення знань її вигляд буде відрізнятися. Зокрема, для інтелектуальних інформаційних систем на основі продукційної моделі представлення знань характерне розділення бази на фактологічну базу («робочу пам'ять») і на базу правил (базу продукцій).

– **«Фактологічна база»** – або робоча пам'ять відображає контент, який знаходиться у полі зору інтелектуальної системи, факти, які вона використовує в своїх розрахунках. У цій же пам'яті відбуваються зміни значень фактів.

– «**База правил (продукцій)**» – зазвичай, база даних, в якій певним чином, в залежності від типу та особливостей інтелектуальної інформаційної системи, зберігаються продукції.

– «**Система логічного виводу**» – система, або підсистема, в задачі якої входить підготовка результатів обробки запитів за допомогою бази знань в інтелектуальній інформаційній системі, для надання користувачу у певному форматі відображення даних.

– «**Вирішувач**» – головний елемент інтелектуальної інформаційної системи. Задача вирішувача обробити дані з фактологічної бази в БЗ і застосувати їх відповідно правилам у базі продукцій, тобто, фактично задача вирішувача – конвертувати дані БЗ в знання, які можна надати користувачам.

Алгоритм роботи вирішувача з реалізованим у ньому «наївним» алгоритмом обробки продукцій зображено на рисунку 2.3 [10, 12].

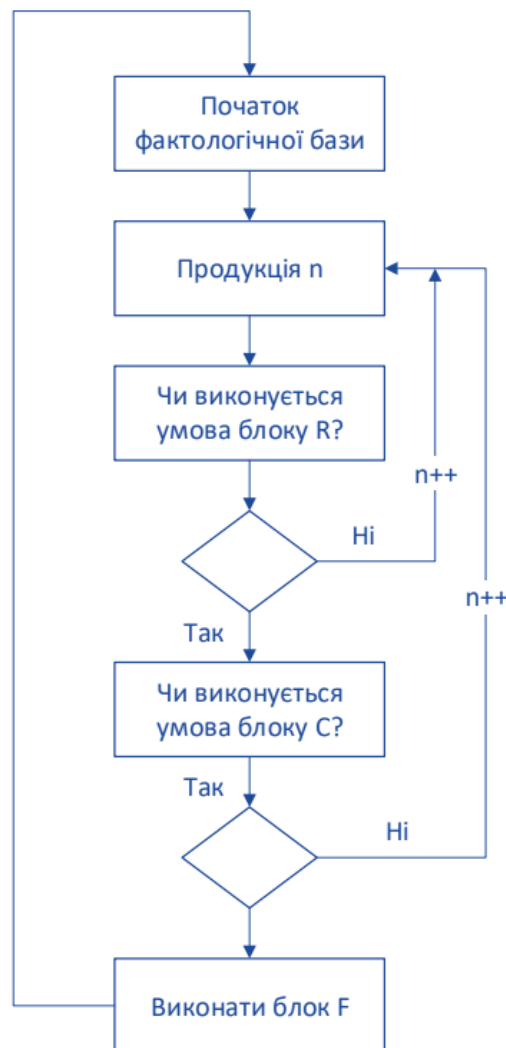


Рис.2.3. – Загальна схема алгоритму вирішувача в БЗ.

Якщо складові компоненти продукційної системи (продукції і факти) БЗ мають незначну кількість продукцій і фактів – розробка алгоритмів оптимізації не потрібна, тому переходимо до розробки «наївного» алгоритму обробки продукцій.

Суть «наївного» алгоритму полягає у послідовному переборі продукцій і виконанні тих, які можна використати згідно результатів обчислення блоку R. Після того як продукція виконана, алгоритм переходить до початку списку продукцій ^[10].

Розглянемо алгоритм як логічну послідовність кроків. Нагадаємо, що для роботи наведеної інтелектуальної системи необхідні зовнішні модулі, за допомогою яких буде визначено момент, коли вихідна енергія від обох джерел буде оптимізована, і змінять відповідний факт у робочій пам'яті системи, також вони генерують повідомлення, яке із джерел енергії зараз в активному стані (сонячні панелі або вітрогенератор). Взаємодія з системою відбувається завдяки модифікації її робочої пам'яті. Встановимо пам'ять в певний стан:

$f_1 = \text{true};$

$f_2 = \text{true};$

$f_3 = \text{false};$

$f_4 = \text{false};$

$f_5 = \text{false};$

$f_6 = \text{true};$

$f_7 = \text{false};$

$f_8 = 1.$

Алгоритм формує список продукцій і переходить до першої в списку, підставляючи їх значення з робочої пам'яті (фактологічної бази). На першому етапі це застосовується до блоку R. Отже:

1. Продукція «0».
2. Блок $R = \langle \neg f_3, \neg f_7 \rangle = \langle \text{NOT false AND NOT false} \rangle = \text{true}$.
3. Переходимо до ядра продукції (оскільки $R = \text{true}$) і намагаємося його активувати $C = \langle f_1 \wedge f_5 \rangle = \langle \text{true AND false} \rangle = \text{false}$. Спроба невдала.
4. Переходимо до наступної продукції.
5. Продукція «1».
6. Блок $R = \langle \neg f_4, \neg f_7 \rangle = \langle \text{NOT false AND NOT false} \rangle = \text{true}$

7. $C = \langle f_2 \wedge f_6 \rangle = \langle \text{true AND true} \rangle = \text{true}$
8. Оскільки $C = \text{true}$ – переходимо до блоку F, викликаємо функцію «Увімкнути°контролер°сонячних°панелей°На(f_8)» з параметром фактологічної бази f_8 . Нехай, в цей момент сонячна генерація складає 100%, тоді в цей момент буде змінено факт $f_4 = \text{true}$.
9. Оскільки ядро продукції було активовано – алгоритм переходить на початок списку продукцій.
10. Продукція «0».
11. $R = R = \langle \neg f_3, \neg f_7 \rangle = \langle \text{NOT false AND NOT false} \rangle = \text{true}$.
12. $C = C = \langle f_1 \wedge f_5 \rangle = \langle \text{true AND false} \rangle = \text{false}$.
13. Продукція «1»
14. $R = \langle \neg f_4, \neg f_7 \rangle = \langle \text{NOT true AND NOT false} \rangle = \text{false}$.
15. Продукція «2»
16. $R = \langle f_3, \neg f_7 \rangle = \langle \text{false AND NOT false} \rangle = \text{false}$.
17. Продукція «3»
18. $R = \langle f_4, \neg f_7 \rangle = \langle \text{true AND NOT false} \rangle = \text{true}$.
19. $C = \langle f_1 \wedge f_2 \wedge f_6 \rangle = \langle \text{true AND true AND true} \rangle = \text{true}$.
20. F = Вимкнути°контролер°сонячних°панелей () – В цей момент функція вимикає живлення від сонячних панелей - $f_2 = \text{false}$, а оскільки єдиним працюючим джерелом залишається вітрогенератор, то $f_5 = \text{true}$, $f_6 = \text{false}$. Ядро продукції активовано – алгоритм виконує перехід в початок списку продукцій.
21. Продукція «0»
22. $R = \langle \neg f_3, \neg f_7 \rangle = \langle \text{NOT false AND NOT false} \rangle = \text{true}$.
23. $C = \langle f_1 \wedge f_6 \rangle = \langle \text{true AND true} \rangle$
24. F = Увімкнути°контролер°сонячних°панелей°На(f_8)
25. Перехід до продукції «0»

Як можна побачити, тепер система перейде в цикл, до того моменту поки $f_4 = \text{true}$, але з цього можна буде зробити висновок що сонячного світла взагалі немає, або система буде запускати сонячні панелі на f_8 доти, доки $f_7 = \text{true}$, у цьому випадку у всіх продукцій блок R перестане виконуватись, і система буде пробігати по списку нічого не змінюючи оскільки досягнуто цільового стану.

Завдання: Розробити проект БЗ на основі продукційної моделі.

1. Складіть проект БЗ на основі продукційної моделі. Вкажіть у звіті наступну інформацію: а) предметна область; б) кратка характеристика предметної області; в) завдання, що вирішуються за допомогою БЗ.
2. Складіть і запишіть 10-15 продукційних правил (виду, IF ...TO), що входять в проект БЗ, яка допоможе вирішити обрану вами задачу(вибір навчального закладу, гаджета, тощо).

Примітка. Для написання правил використовуйте синтаксис: If ... To, де (AND {}) | (АБО {}); (може мати вигляд AND OR) = ; (наприклад, «напруга != 5V» або «напруга > 5V») = {}; = (=); (наприклад, «message = «помилка!»») Наприклад: IF intention = «навчатися» And road = «з вибоїнами», To recommended transport = «позашляховик». Якщо температура > 50°, To cooler_state_1 = «включити».

*P.S. Остання цифра залікової книжки – кількість продукцій (якщо менше 5, тоді 5).
Передостання цифра залікової книжки – кількість фактів (якщо менше 5, тоді 10).*

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Постановка завдання з характеристикою порядку виконання робіт.
3. Результати роботи програми (скріншоти з результатами роботи програми). Пояснення за результатами роботи програми.
4. Пояснення отриманих результатів.
5. Алгоритм, структура БЗ, текст (лістинг) програми.
6. Відповіді на контрольні питання.

Контрольні питання:

1. Структура продукційної моделі, ядро продукції.
2. Особливості фактологічної бази?
3. Алгоритм роботи вирішувача в БЗ на основі продукційної моделі.
4. Опишіть особливості представлення продукційної моделі представлення знань (імена змінних, структура, алфавіт, оператори).
5. Дайте формальний (математичний) опис продукційної моделі представлення знань.
6. Механізми виводу в продукційних системах (наведіть приклад).

4. Лабораторна робота № 3.

Тема: Проектування бази знань (БЗ) на основі семантичної мережі.

Мета: Дослідження моделі знань на основі семантичної мережі, формування навичок формалізації знань і створення БЗ для вирішення прикладних завдань.

Теоретичні відомості:

Семантичні мережі – це одна з поширених форм представлення знань. Були запропоновані у 60 рр. XX ст. як відображення уявлень про понятійний апарат людини. Їх поширення обумовлено можливостями семантичних мереж відображати довільної структури понять предметної області та наявністю розроблених математичних методів вирішення задач на графах [4,10, 12].

В загальному випадку семантичну мережу може бути зображено у вигляді графу: $G=(X, R)$,

де: $X=\{x_i\}$, — множина вершин графу,

$R=\{<x_i, x_j>\}$, — множина дуг.

Вершини графу відповідають поняттям предметної області, а дуги - відношенням між ними (рис.3.1).

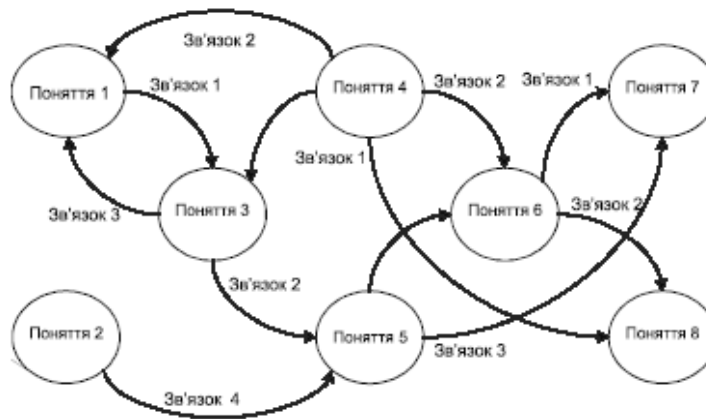


Рис. 3.1. Структура семантичної мережі

Як додаткові характеристики графу, залежно від реалізації, можуть бути введені типи вершин, дуг, імена і вагові коефіцієнти відношень тощо.

Характерна особливість семантичних мереж – обов'язкова присутність трьох типів відносин (рис.3.2):

- клас, до якого належить певне поняття;
- властивості, що виділяють це поняття серед інших певного класу;
- приклад певного поняття.



Рис.3.2. Базові відношення семантичних мереж [12].

Для побудови семантичної мережі, яка описує структуру події, потрібно виділити саму дію (яка зазвичай, описується дієсловом і є «центром» семантичної мережі). Потім виділяють інші поняття даної події:

- *агент* (діюча особа);
- *умова* (логічна залежність між двома подіями);
- *інструмент* (предмет, явище або пристрій, що виконує дію);
- *місце* (де проходить подія);
- *об'єкт* (предмет, на який скерована дія);
- *мета* (для чого виконується дія);
- *адресат* (особа, яка використовує результати дії);
- *якість* (якісне або кількісне обмеження обсягу дії)

На рис.3.3 наведено приклад семантичної мережі події, яку можна описати фразою «Для отримання заліку студент Петренко написав на оцінку «відмінно» гелевою ручкою контрольну роботу в аудиторії 15».

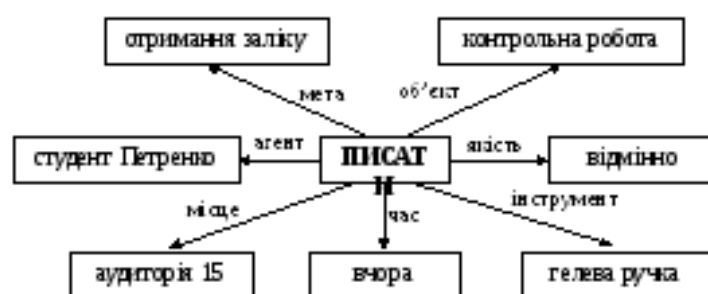


Рис.3.3. Приклад семантичної мережі, що описує подію [10].

Перевагами семантичних мереж є універсальність і можливість відображення теоретично необмеженої кількості понять і відношень предметної області. Семантичні мережі придатні для масштабування, в них ефективно реалізується пояснення рішень, є можливість експорту - імпорту знань в будь-які інші типи моделей представлення знань [4,10, 12].

До недоліків семантичних мереж можна віднести високу обчислювальну складність реалізації процедур пошуку і логічного виводу для складних ієрархічних багаторівневих семантичних мереж.

Завдання: Розробити проект БЗ на основі семантичної мережі.

1. Складіть проект бази знань (БЗ) у вигляді семантичної мережі. Вкажіть у звіті наступну інформацію: а) предметна область; б) кратка характеристика предметної області; в) завдання, що вирішуються за допомогою БЗ.
2. Розробіть схему (у вигляді графа) вмісту (контенту) бази знань (БЗ) на основі моделі семантичної мережі.

Примітка: База даних може складатися з одного або декількох графів із загальною кількістю вершин (сутностей, понять) не менше 20).

P.S. Остання цифра залікової книжки – кількість графів (якщо менше 5, тоді 3, в інших випадках - 5).

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Постановка завдання з характеристикою порядку виконання робіт.
3. Результати роботи програми (скріншоти з результатами роботи програми). Пояснення за результатами роботи програми.
4. Пояснення отриманих результатів.
5. Схема, графи, структура БЗ, текст (лістинг) програми.
6. Відповіді на контрольні питання.

Контрольні питання:

1. Структура моделі БЗ на основі семантичної мережі.
2. Типи відносин у семантичній мережі.
3. Алгоритм побудови семантичної мережі.
4. Опишіть особливості застосування семантичних мереж представлення знань.
5. Переваги і недоліки БЗ заснованих на семантичних мережах представлення знань.
6. Як відображається предметна галузь у семантичних мережах.

5. Лабораторна робота № 4.

Тема: Проектування бази знань на основі фреймової моделі.

Мета: Дослідження фреймової моделі представлення знань, формування навичок формалізації знань і створення баз знань (БЗ) для вирішення прикладних завдань.

Теоретичні відомості:

Фреймові моделі знань запропоновані у 70-і роки ХХ ст. як окремий випадок семантичних мереж для відображення стереотипних ситуацій предметної області. Фреймові моделі у порівнянні із семантичними мережами дають більш формалізований і в той же час досить гнучкий «механізм» представлення знань. У рамках фреймових моделей вдалося значною мірою об'єднати декларативні знання про об'єкти і процедурні знання про методи витягу і перетворення інформації для досягнення заданих цілей [4,10,15,16].

Термін «фрейм» (від англ.: *frame* - рамка) асоціюється з уявленням людиною про предмети, об'єкти, стереотипні ситуації, яке завжди обрамлено (обмежено) певними характеристиками та властивостями об'єкта чи ситуації.

Кожному об'єкту (ситуації) ставиться у відповідність формальна конструкція – «фрейм», а кожній характеристиці, властивості об'єкта - одне поле (слот) фрейма.

Формально фрейм можна подати такою структурою:

$$f [< v_1, g_1 >, < v_2, g_2 > \dots, < v_n, g_n >],$$

де: f - ім'я фрейму;

пара $[< v_i, g_i >]$ – i -й слот,

v_i - ім'я слота,

g_i – значення слота.

Фрейми умовно розподіляють на дві групи – «**фрейми – образи**» і «**рольові фрейми**» (або «**фрейми-сценарії**»). В рольових фреймах в якості імен слотів виступають запитання, відповіді на які є значеннями слотів. Наприклад,

фрейм-образ: «периферійні пристрої[<роутер, 5 шт.>, <комутатори, 10 шт.>, ..., <...>],

фрейм-сценарій: «перевести[<що? роутер>, <звідки? Київ>, <куди? Покровськ>... <...>]».

Слот може містити як декларативну, так і процедурну інформацію. Значенням слота може бути число, математичний вираз, текст, програма, тощо. Слотом також може бути набір слотів більш низького рівня. Значеннями слотів можуть бути також імена інших фреймів, що забезпечує зв'язок між фреймами. Слот може містити не тільки конкретне значення, але й ім'я процедури, що дозволяє обчислити його за заданим алгоритмом, а також одну або декілька продукцій (евристик), за допомогою яких це значення можна знайти. У слот може входити не одне, а декілька значень. Іноді слот містить компонент, що називається фасетом і задає діапазон або перелік можливих значень. Фасет указує також граничні значення заповнювача слота.

Специфічною функцією у фреймах є демон. Демон задає процедуру, яка автоматично запускається, коли в його слот підставляється деяке значення або коли проводиться звірення значень.

Отже, з кожним слотом може бути зв'язана різноманітна інформація:

- процедура;
- способи отримання інформації про слот;
- значення, що приймаються за замовчуванням;
- правила логічного виведення.

Якщо, слоти фрейма не містять конкретних значень, то фрейм називають прототипом, еталоном, формуляром, фреймом-інтенсіоналом або просто фреймом. Фрейми з конкретними значеннями слотів називають фреймами-екземплярами, фреймами-прикладками, фреймами-примірниками. Наведемо приклад фрейму-екземпляра «Студент».

Ім'я слота	Значення слота
Прізвище	Самощенко
Ім'я	Микола
По батькові	Васильович
Вищий навчальний заклад	Донецький національний технічний університет
Факультет	КІТА
Спеціальність	123 «Комп'ютерна інженерія»
Курс	четвертий
Група	KI-19

Сукупність фреймів можна зобразити у вигляді графу. Вершини графа відповідають фреймам-екземплярам, а дуги - зв'язкам між ними.

Унаслідок сіткової організації та типізації вершин і зв'язків фреймові моделі претендують на компроміс між універсальністю і ефективністю логічного виводу. Проте, відсутність формальної семантики ускладнює логічну інтерпретацію і порівняння властивостей різних фреймових моделей. Крім того, внаслідок типізації вершин висуваються обмеження щодо поповнення бази новими типами знань.

Завдання: Розробити проект БЗ на основі фреймової моделі.

1. Складіть проект бази знань (БЗ) у вигляді фреймової моделі. Вкажіть у звіті наступну інформацію: а) предметна область; б) кратка характеристика предметної області; в) завдання, що вирішуються за допомогою БЗ.
2. Схематично (у вигляді графічного об'єкта) зобразити вміст (контент) бази знань згідно з фреймовою моделлю. База знань (БЗ) повинна складатися з кількох протофреймів (незаповнених - оболонок) із загальною кількістю слотів не менше 10 і кількох екзофреймів (заповнених - «екземплярів») із загальною кількістю слотів не менше 20.

P.S. Остання цифра залікової книжки – кількість слотів протофреймів (якщо менше 5, тоді 10, в інших випадках - 15).

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Постановка завдання з характеристикою порядку виконання робіт.
3. Результати роботи програми (скріншоти з результатами роботи програми). Пояснення за результатами роботи програми.
4. Пояснення отриманих результатів.
5. Схема, графи, структура БЗ, текст (лістинг) програми.
6. Відповіді на контрольні питання.

Контрольні питання:

1. Структура моделі БЗ на основі фреймової моделі.
2. Поняття слот і поле фрейму.
3. Охарактеризуйте поняття слота-образа і слота-сценарія.
4. Поняття демона у фреймовій моделі.
5. Охарактеризуйте поняття протофреймів і екзофреймів.

6. Лабораторна робота № 5.

Тема: Застосування методів нечіткої логіки в інтелектуальних інформаційних системах.

Мета: Ознайомитися з загальними принципами застосування методів нечіткої логіки в проектуванні інтелектуальних систем, розробити експертну систему нечіткого виведення за методом Мамдані з використанням основних функцій пакету Fuzzy Logic Toolbox програмного середовища MatLab

Теоретичні відомості:

Логіка в класичному сенсі - це формалізоване уявлення механізму мислення. Однак в реальності існує не тільки *формальна логіка (булева)*, а стільки логік, скільки побажаємо, тому що все визначається вибором відповідної системи аксіом, а всі твердження, що побудовані на їх основі, повинні бути строгими, без протиріч, пов'язані за правилами, встановленими для цієї системи аксіом [6,8].

Нечітка логіка - узагальнення класичної (формальної, булевої) логіки для ситуації, коли істинність розглядається як лінгвістична змінна, що приймає не кількісні, а якісні значення типу: «дуже істинно», «більш-менш істинно», «дуже сильно» «не дуже сильно», тощо [6,8].

Зазначені лінгвістичні значення представляються нечіткими множинами. **Основна відмінність нечіткої логіки від класичної полягає в тому, що замість значень «True» і «False» в нечіткій логіці використовується ступінь істинності, що приймає значення з нескінченної множини від 0 (Хибно) до 1 (Істина) включно.**

Логічні операції в нечіткій логіці, неможливо представити таблицями істинності. Вони задаються функціями і лише у крайніх випадках - коли значення змінних виключно 1 або 0 - можуть бути представлені таблицями істинності класичної логіки.

Математична теорія нечітких множин (fuzzy sets) і нечітка логіка (fuzzy logic) - результат узагальнення класичної теорії множин і класичної логіки. Основи нечіткої логіки було введено професором Лотфі Заде (Берклі, Каліфорнія, США) в 1965 р., який започаткував теорію нечітких множин [8].

Зазвичай, термін **«нечітка логіка»** використовується в двох різних значеннях. У вузькому сенсі, *нечітка логіка* - це логічне числення, що є розширенням багатозначної логіки. В широкому сенсі, *нечітка логіка* рівнозначна теорії нечітких множин.

Архітектура Fuzzy Logic складається з чотирьох основних частин: фазифікація; база правил; блок логічного висновку; дефазифікація (рис. 6.1) [8].

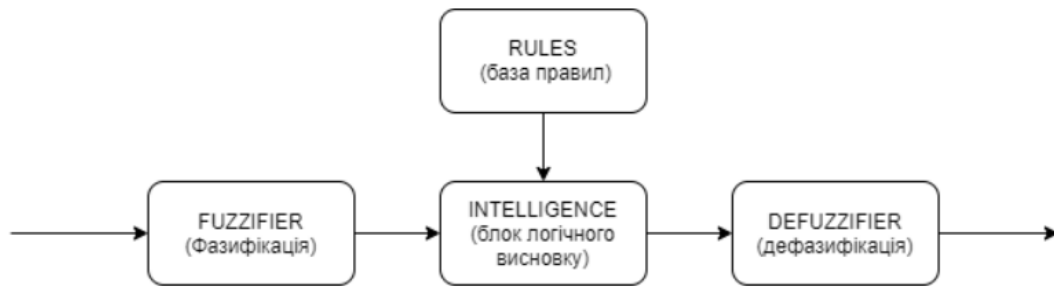


Рисунок 6.1 – Загальна структура нечіткої логіки

Головним базовим поняття *Fuzzy Logic* є *FIS-структура* – *система нечіткого виведення (Fuzzy Inference System)*. Вона містить а собі необхідні процеси для реалізацію функціонального відображення «входи-виходи» на основі нечіткого логічного виведення.

При використанні системи нечіткого виведення на етапі блоку нечіткого виведення з'являється два поняття – *функція належності (Membership Functions)*, яка вказує ступінь належності елемента нечіткій підмножині і *лінгвістична змінна*, яка є множиною нечітких змінних та дає словесний опис деякому нечіткому числу, що буде отримане в результаті розрахунків [25].

При формуванні лінгвістичних змінних множина їх значінь називається *терм_множина*, яка включає в себе всі можливі значення лінгвістичної змінної. Термом називають будь-який елемент даної множини [6,8].

При вирішенні практичних задач методами нечіткої логіки, зазвичай мають на увазі *системи нечіткого виведення*, які використовуються для управління технічними пристроями і процесами.

Розробка і застосування систем нечіткого виведення включає ряд етапів, реалізація яких виконується за допомогою розглянутих вище основних положень нечіткої логіки.

Інформацією, яка поступає на вхід системи нечіткого виведення, є виміряні деяким чином вхідні змінні. Ці змінні відповідають реальним змінним процесу управління. Інформація, яка формується на виході системи нечіткого виведення - вихідні змінні, тобто керуючі змінні процесу управління.

Системи нечіткого виведення призначені для перетворення значень вхідних змінних процесу управління у вихідні змінні на основі використання нечітких правил продукцій [8,25].

Для цього системи нечіткого виведення повинні містити базу правил нечітких продукцій і реалізовувати нечітке виведення висновків на основі посилок або умов, представлених у формі нечітких лінгвістичних висловів.

Отриманні висновки в системах нечіткого виведення базуються на розподілі процесу виведення на послідовність етапів, реалізація яких виконується на основі основних положень нечіткої логіки (рис. 6.2 і 6.3) ^[8].

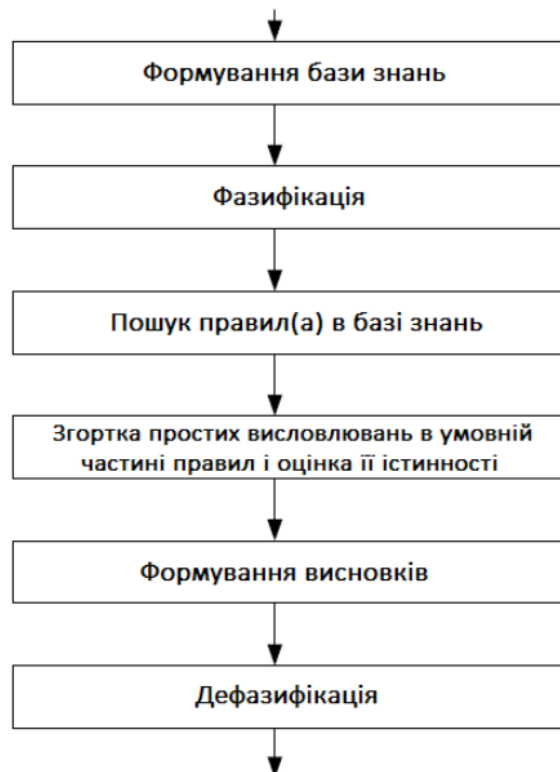


Рис. 6.2. Етапи процесу нечіткого виведення

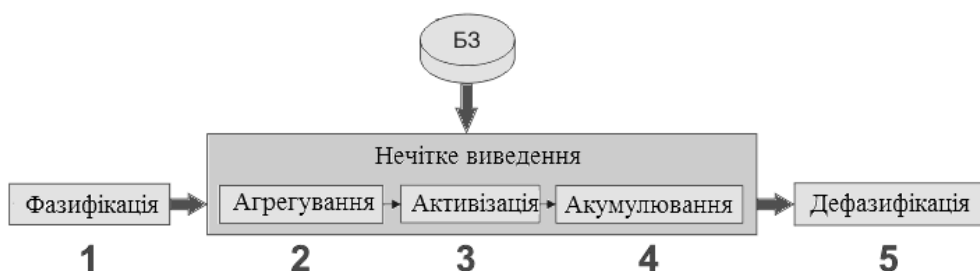


Рис. 6.3. Послідовність етапів процесу нечіткого виведення ^[25]

Основним компонентом, який визначає якість керування (прийняття рішення) за допомогою методів нечіткої логіки, як в будь якої інтелектуальної системі, буде *база знань (БЗ) системи нечіткого умовного виведення*, що являє собою множину узгоджених правил нечіткого умовного виведення ^[6,25].

При побудові вказаної бази знань повинні бути виконані наступні умови:

1. Правила, що утворюють БЗ системи, не повинні бути суперечливими.
2. Система правил повинна бути повною і ненадмірною

Наступним кроком буде формування бази правил систем нечіткого виведення. **База правил систем нечіткого виведення** призначена для формального представлення емпіричних знань або знань експертів.

В системах нечіткого виведення використовують правила нечітких продукцій, в яких умови і висновки сформульовані в термінах нечітких лінгвістичних висловів.

База правил нечітких продукцій є скінченною множиною правил нечітких продукцій, що узгоджені відносно до використовуваних в них лінгвістичних змінних у вигляді структурованого тексту [6,8]:

Правило_1: ЯКЩО "Умова_1" ТО "Висновок_1" (F_1)

Правило_2: ЯКЩО "Умова_2" ТО "Висновок_2" (F_2)

.....

Правило_n: ЯКЩО "Умова_n" ТО "Висновок_n" (F_n)

або в еквівалентній формі:

RULE_1: IF Condition_1 THEN Conclusion_1 (F_1)

RULE_2: IF Condition_2 THEN Conclusion_2 (F_2)

. (8.5)

RULE_n: IF Condition_n THEN Conclusion_n (F_n)

де $F_i (i \in \{1, 2, \dots, n\})$ - коефіцієнти визначеності або вагові коефіцієнти відповідних правил. Коефіцієнти можуть приймати значення з інтервалу $[0, 1]$. У випадку коли вагові коефіцієнти відсутні, вважають, що їх значення дорівнюють 1.

Узгодженість правил відносно використовуваних лінгвістичних змінних означає, що в якості умов і висновків правил можуть використовуватися тільки нечіткі лінгвістичні вислови, при цьому в кожному з нечітких висловів повинні бути визначені функції приналежності значень терм-множин для кожної з лінгвістичних змінних.

В системах нечіткого виведення лінгвістичні змінні, які використовуються в нечітких висловах під_умов правил нечітких продукцій, часто називають *вхідними лінгвістичними змінними*, а змінні, які використовуються в нечітких висловах під_висновків правил нечітких продукцій, часто називають *вихідними лінгвістичними змінними* [8].

При формуванні бази правил нечітких продукцій необхідно визначити: множину правил нечітких продукцій: $P=\{R_1, R_2, \dots, R_n\}$, множину вхідних лінгвістичних змінних: $V=\{\beta_1, \beta_2, \dots, \beta_m\}$ і множину вихідних лінгвістичних змінних: $W=\{w_1, w_2, \dots, w_s\}$. База правил нечітких продукцій вважається заданою, якщо задані множини P, V, W .

У системі нечіткого виведення (рис. 6.4) за етап фазифікації відповідає спеціальний блок - *фазифікатор* (блок фазифікації). Цей блок перетворює чіткі («crisp») величини, отримані за межами нечіткої системи, в нечіткі величини, що описуються лінгвістичними змінними в базі знань. Блок рішень використовує нечіткі продукційні правила, закладені в базі знань, для перетворення нечітких вхідних даних в необхідні виходи, які носять також нечіткий характер. Блок *дефазифікації* перетворює нечіткі дані з виходу блоку рішень в чітку величину для подальшого використання [25].

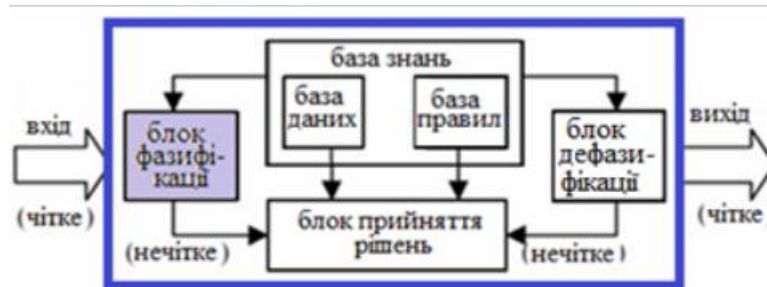


Рис. 6.4. Система нечіткого виведення

В інженерних задачах, застосовується метод нечіткого виводу Мамдані (запропонований в 1975 р. Ібрагімом Мамдані). Метод Мамдані є нечіткою системою виведення (НСВ). НСВ являє собою систему, яка використовує теорію нечітких множин для відображення входів (функцій в разі нечіткої класифікації) та виходів (класів в разі нечіткої класифікації) [21].

Метод нечіткого виводу Мамдані складається з наступних етапів, які можна реалізувати у середовищі Matlab:

- формування бази правил;
- фазифікація;
- агрегування під_умов;
- активізація під_висновків;
- акумулювання висновків;
- дефазифікація.

Кожен етап виконується послідовно, кожен наступний етап отримує в якості вхідних - значення, які були вихідними (отримані в результаті роботи) попереднього. Для вхідних значень метод Мамдані використовує певну базу правил. Загальна схема роботи нечіткої логіки зображена на рисунку 6.5 ^[21].

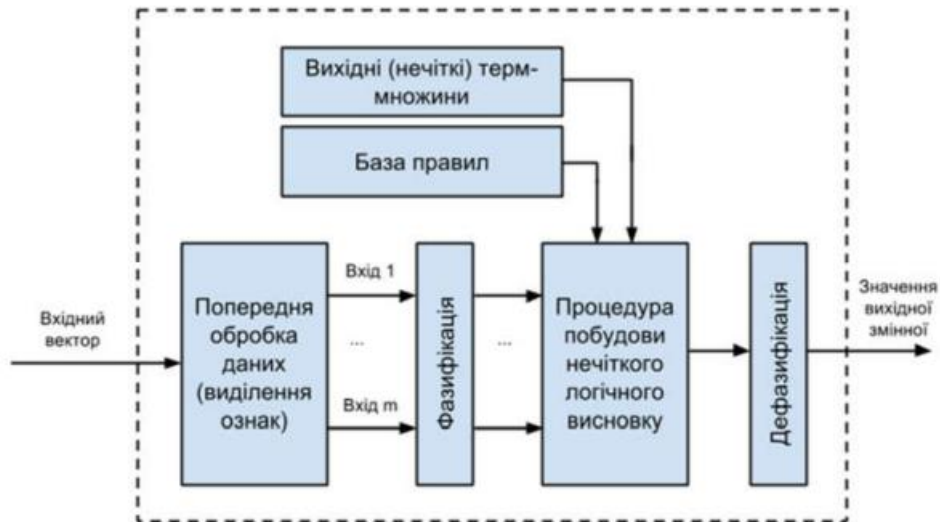


Рисунок 6.5. Загальна схема роботи нечіткої логіки

Як ми казали, **функція належності** є апаратом для символного (математичного) опису нечіткої множини, тобто переходу від «істини» до «хибності». Жодних обмежень на вибір конкретної функції належності немає.

На практиці, зазвичай, використовують ті з них, які допускають аналітичне подання у вигляді простих математичних функцій. Необхідність типізації окремих функцій належності обумовлена реалізацією відповідних функцій в інструментальних засобах, наприклад, у MATLAB.

Для реалізації процесу нечіткого моделювання в середовищі MATLAB призначений спеціальний пакет розширення *Fuzzy Logic Toolbox*. За його допомогою можна розробляти та розв’язувати нечіткі моделі ^[21].

Застосуємо метод нечіткого моделювання для процесів збагачення корисних копалин (зокрема енергетичного вугілля), які відбуваються на збагачувальних фабриках.

Для створення нечіткої моделі ознайомимося з основними технологічними особливостями об’єкта керування.

– дроблення і подрібнення - технологічна операція та процес руйнування і зменшення розмірів грудок мінеральної сировини (корисної копалини) під дією зовнішніх механічних, теплових, електричних сил, направлених на

подолання внутрішніх сил зчеплення, що зв'язують між собою частинки твердого тіла. *Дроблення і подрібнення* за фізикою процесу не мають між собою принципових відмінностей. Умовно прийнято вважати, що при дробленні одержують продукти крупніші 5 мм, а при подрібненні - дрібніші 5 мм. Розмір максимальних зерен, до якого необхідно роздробити або подрібнити корисну копалину при її підготовці до збагачення, залежить від розміру включень основних компонентів, що входять до складу корисної копалини, і від технічних можливостей обладнання.

- грохочення і класифікація – це методи розподілу корисної копалини на продукти різних фракцій крупності (класи крупності). Грохочення здійснюється розсіванням корисної копалини на решетах і ситах з каліброваними отворами на дрібний (підрешітний) продукт і крупний (надрешітний). Грохочення застосовується для розділення корисних копалин за розміром на ситах (поверхнях, з розмірами отворів від часток міліметра до декількох сотень міліметрів). Сита можуть бути барабанні, вібраційні, хитні, обертові, сортувальні, тощо. Класифікація матеріалу за крупністю здійснюється у водному або повітряному середовищі і базується на використанні розбіжності у швидкостях осадження частинок різної крупності. Великі частинки осаджуються швидше і концентруються в нижній частині класифікатора, дрібні частинки осаджуються повільніше і виносяться з апарата водним або повітряним потоком. Класифікація застосовується для розділення дрібних і тонких продуктів по зерну розміром не більше 1 мм.

- флотація - це найефективніший метод збагачення вугільних шламів. Схеми флотації, що приймаються, визначаються маркою збагачуваного вугілля, його зольністю, вмістом тонких класів і проміжних фракцій, складом неорганічної частини і особливо наявністю глинистих порід, що піддаються розмоканню, вимогами щодо якості кінцевих продуктів і техніко-економічними показниками. При низькій зольності вихідних шламів (не більше 18-20 %) і легкій збагачуваності застосовується «пряма схема». При флотації шламів більш важкої збагачуваності і більшій їх зольності застосовують схему з перечищенням концентрату основної флотації. Перечищення концентрату здійснюється в окремих флотаційних машинах.

- зневоднення флотаційного концентрату здійснюється фільтруванням у дискових вакуум-фільтрах і сушінням. Відходи флотації сумісно з відходами гвинтових сепараторів згущуються у циліндроконічних згущувачах і зневоднюються у фільтр-пресах. Застосування фільтр-пресів дозволяє здійснити на вуглезбагачувальних фабриках замкнений цикл по воді.

Розглянемо постановку задачі. Необхідно розробити систему управління процесом утворення товарних (комерційних) фракцій, яка ґрунтується на нечіткому моделюванні. На перебіг процесів зневоднення та утворення фракцій продукту потрібного розміру безпосередньо впливає значення тиску в фільтр-пресі (змінна_1), кількість і розмір фільтруючих плит приймаємо для даного виду фільтр-пресу вважаємо const, а також параметр концентрації дрібної фракції на виході фільтр-преса (змінні_2), вплив якої буде відчутно через певний час.

В технологічному процесі фахівці співставляють ці дві змінні для прийняття управлінських рішень.

Параметр тиску у фільтр-пресі визначимо як керуючу змінну.

Факторами моделі визначимо 2 нечіткі лінгвістичні змінні: «концентрація дрібної фракції на виході» і «вологість продукту», а як вихідну змінну - нечітку лінгвістичну змінну «тиск у фільтр-пресі» (далі «тиск»).

За Терм - множини першої лінгвістичної змінної «концентрація дрібної фракції на виході» візьмемо множину $T1 = \{\text{«Занадто мала»}, \text{«Мала»}, \text{«Відповідна»}, \text{«Велика»}, \text{«Завелика»}\}$, а за терм - множину другої лінгвістичної змінної «вологість продукт» використаємо множину $T2 = \{\text{«низька»}, \text{«нормальна»}, \text{«значна»}\}$. За терм - множину вихідної лінгвістичної змінної «тиск» візьмемо множину $T3 = \{\text{«Дуже малий»}, \text{«Малий»}, \text{«Зменшений»}, \text{«Нормальний»}, \text{«Збільшений»}, \text{«Великий»}, \text{«Дуже великий»}\}$. При цьому терм першої вхідної змінної (концентрація дрібної фракції) оцінюватимемо за 40 – відсотковою шкалою, де цифрі «0» відповідає найменша концентрація дрібної фракції на виході, а цифрі «40» – найбільша концентрація дрібної фракції на виході. Терм другої вхідної змінної («вологість») оцінюватимемо за 10 - бальною шкалою, при якій цифрі «0» відповідає найменша вологість продукту, а оцінки «10» - найбільша. Що стосується терма вихідної змінної «тиск», то її шкала буде проградуйована в Бар від 7 до 50 Бар в порядку зростання.

Розглянемо послідовність виконання етапів (процесів) з розробки вказаної системи і моделі нечіткого висновку в інтерактивному режимі:

1. Ініціюємо редактор системи нечіткого висновку FIS, для чого у вікні команд набрати ім'я відповідної функції *fuzzy*. Після виконання цієї команди на екрані з'явиться графічний інтерфейс редактора FIS з ім'ям системи нечіткого висновку *Untitled* і типом системи нечіткого висновку (Мамдані), запропонованим за умовчанням.

2. Оскільки ми маємо систему нечіткого висновку з двома входами, то необхідно додати в систему FIS, що розробляється, ще одну вхідну змінну. Для цього слід виконати команду меню Edit > Add Variable... > Input. У результаті виконання цієї команди на діаграмі системи нечіткого висновку з'явиться новий жовтий прямокутник з ім'ям другої вхідної змінної: *input2*. (рис. 6.6)

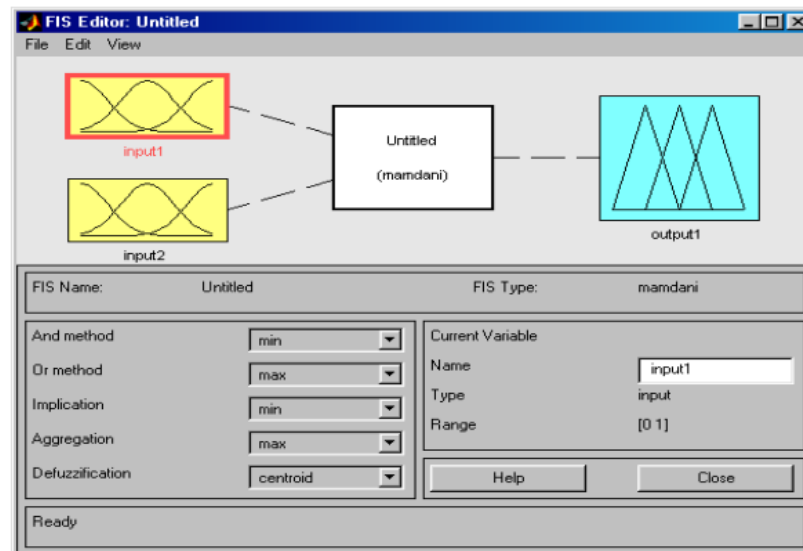


Рис. 6.6. Вікно редактора FIS після додавання другої вхідної змінної.

3. Змінимо імена вхідних і вихідних змінних, які MATLAB встановив за умовчанням. Для цього виділимо прямокутник з ім'ям відповідної змінної, кликнувши на його зображенні на діаграмі (сторони виділеного прямокутника мають червоний колір). Після цього введемо нове ім'я змінної в полі *Name* у правій частині редактора FIS. Результат надання нових імен змінним системи нечіткого висновку наведено на рис. 6.7.

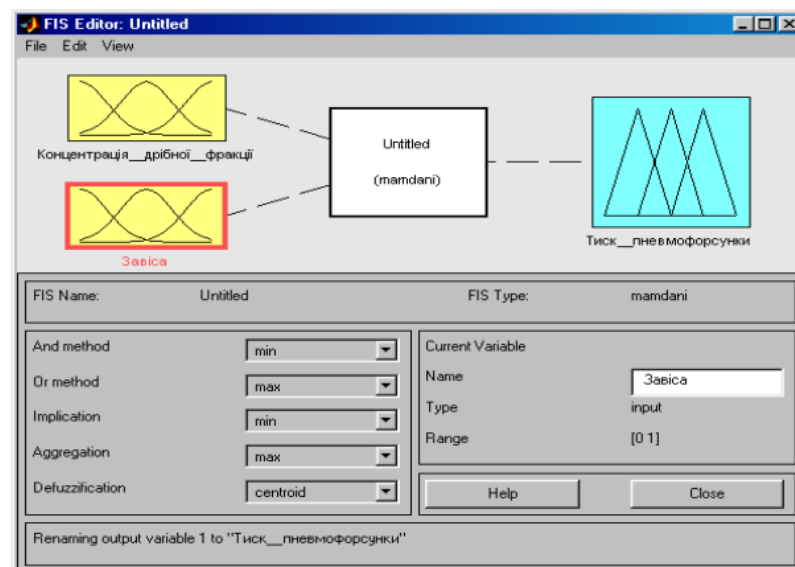


Рис.6.7. Вікно редактора FIS після надання нових імен змінним.

Змінимо ім'я системи нечіткого висновку (*Untitled*), запропоноване за умовчанням. Для цього збережемо створену *структуру FIS* в зовнішньому файлі з ім'ям *mytip.fis*, виконавши команду меню *File>Export>To Disk....* При цьому буде викликано стандартне діалогове вікно збереження файлу, в якому користувачу пропонують ввести ім'я відповідного файлу (розширення файлу приписується автоматично). Залишимо без зміни запропоновані системою MATLAB за умовчанням: метод нечіткого *логічного І* (*And method*) - значення «min», метод нечіткого *логічного АБО* (*Or method*) - значення «max», метод імплікації (*Implication*) – значення «min», метод агрегації (*Aggregation*) – значення «max» і метод дефаззифікації (*Defuzzification*) – значення «centroid». Очевидно, що ці значення можуть бути змінені користувачем.

4. Тепер необхідно визначити терми і їх функції належності для вхідних і вихідних змінних нашої системи нечіткого висновку. Для цього скористаємось редактором функцій належності, який можна викликати або подвійним кліком лівої клавіші миші на значку прямокутника з ім'ям відповідної змінної; або командою меню *Edit>Membership Functions...* (заздалегідь виділіть прямокутник з ім'ям відповідної змінної).

Після виклику редактора функцій належності кожної із змінних за умовчанням пропонується 3 терми з трикутними функції належності (рис.6.8).

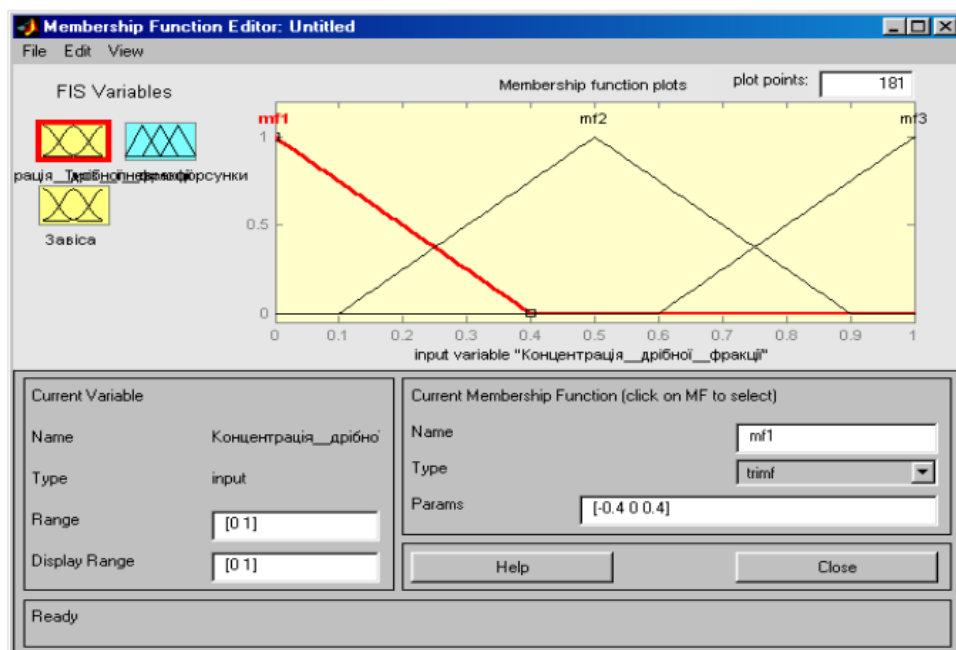


Рис.6.8. Вікно редактора функцій належності з функціями належності для термів змінної «концентрація дрібної фракції», запропонованих MATLAB

По перше, змінимо діапазон значень вхідних змінних, для чого в полях введення **Range** і **Display Range** змінимо верхнє значення з 1 на 40 (відсотків).

Аналогічно виконаємо зміни відповідних діапазонів для вхідної змінної «вологість» [0 10] і для вихідної змінної «тиск» [7.0 - 50.0]. Згоду на зміни підтвердимо натисканням клавіші Enter. Оскільки для вибраної задачі необхідно 5 термів, а система MATLAB пропонує за умовчанням тільки 3, то командою меню Edit>Remove All MFs видаляємо запропоновані терми, а командою меню Edit>Add MFs... створюємо 5 нових термів. (рис. 6.9).

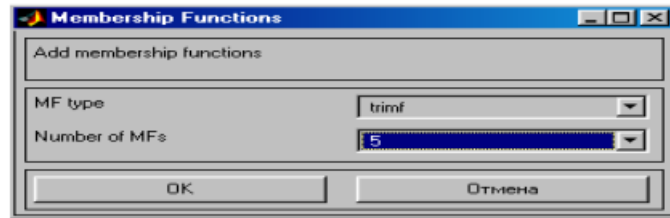


Рис.6.9. Вікно створення необхідної кількості термів.

Далі змінимо назви термів першої вхідної змінної «концентрація дрібної фракції», створених нами (*mf1*, *mf2*, *mf3*, *mf4*, *mf5*) на «Занадто мала», «Мала», «Відповідна», «Велика», «Завелика» відповідно. Після чого змінимо тип крайніх функцій належності, запропонованих за умовчанням, на трапецієвидні функції (*trapmf*), вибравши відповідний пункт в полі Type.

Встановимо параметри функцій: для терма «занадто мала» - параметри [0 0 2 6], для терма «мала» – [2 6 10], для терма «відповідна» – [6 10 14], для терма «велика» – [10 14 18], для терма «завелика» – [14 18 20 20]. Вигляд редактора функцій належності після внесених змін зображений на рис. 6.10.

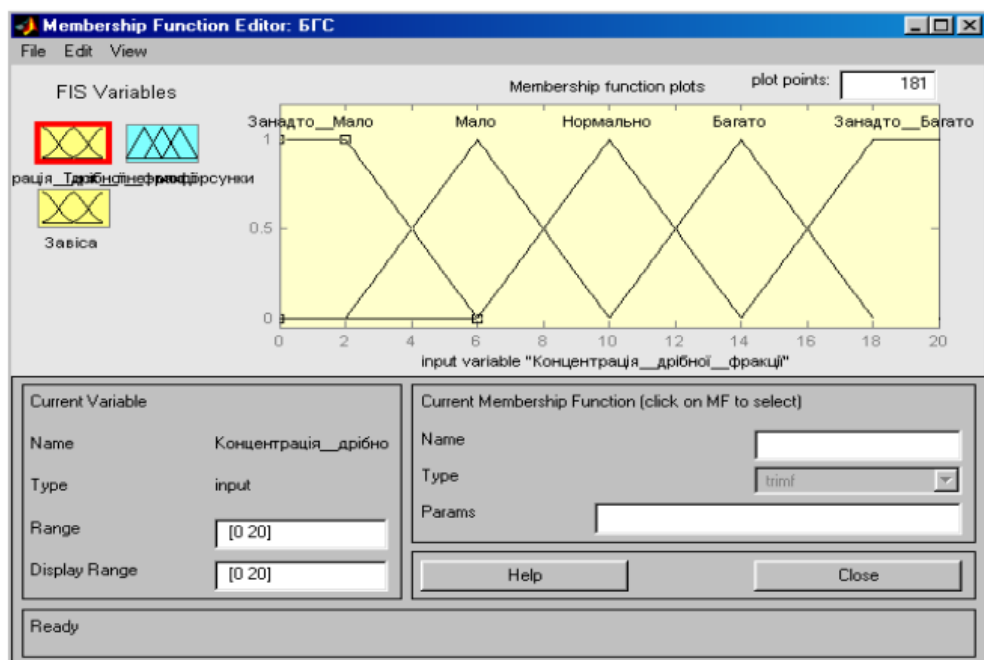


Рис.6.10. Вікно редактора функцій належності після зміни назви термів і типу функцій належності для 1-ої вхідної змінної «концентрація фракції»

Аналогічно змінимо назви термів 2-ої вхідної змінної «вологість». Далі змінимо, запропонований за умовчанням, тип функцій належності на функції типу Гауса (*gaussmf*), вибравши відповідний пункт в *полі Type*. Параметри функцій належності залишимо без змін. За терм_множину другої лінгвістичної змінної «Вологість» використано «низька», «нормальна», «значна». Вигляд редактора функцій належності після внесених змін наведено на рис. 6.11.

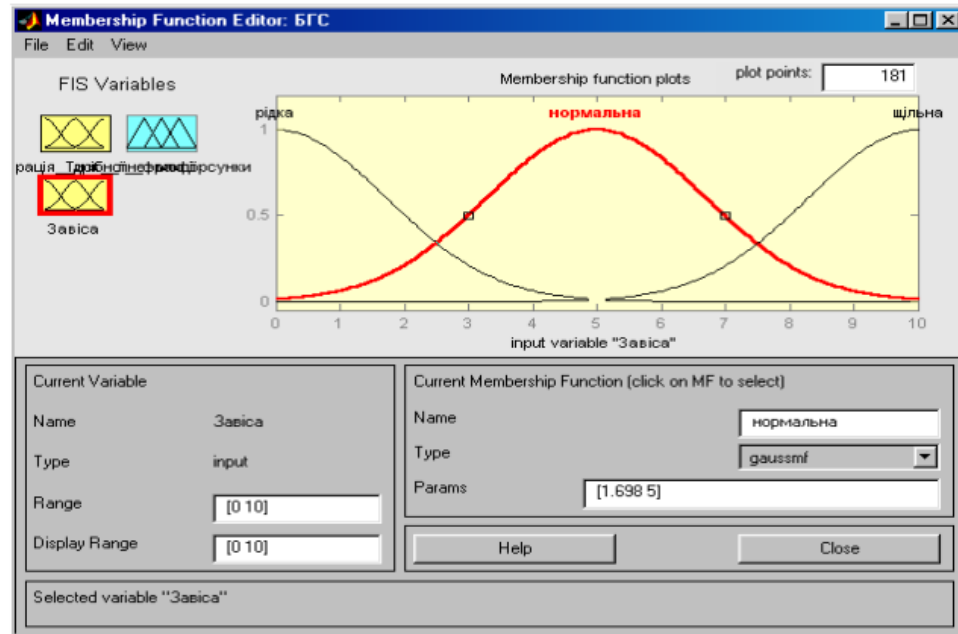


Рис.6.11. Вікно редактора функцій належності після зміни назви термів і типу їх функцій належності для 2-ої вхідної змінної «вологість»

Нарешті, змінимо назви термів і параметри функцій належності для вихідної змінної «Тиск». Нам необхідно 7 термів, а MATLAB пропонує за умовчанням тільки 3. Командою меню *Edit>Remove All MFs* видаляємо запропоновані за замовчуванням терми. Командою меню *Edit>Add MFs* створюємо 7 нових термів. Далі змінюємо назви термів вихідної змінної «Тиск», створених нами (*mf1*, *mf2*, *mf3*, *mf4*, *mf5*, *mf6*, *mf7*) на «дуже малий», «малий», «зменшений», «нормальний», «збільшений», «великий», «дуже великий» відповідно. Далі змінимо тип крайніх функцій належності, запропонованих за умовчанням, на трапецієвидні функції (*trapmf*), вибравши відповідний пункт в *полі Type*. Параметри функцій встановлюємо наступним чином: для терма «дуже малий» - параметри [7.0 9.0 10.0], для терма «малий» – [10.5 12.5 14.5], для терма «Зменшений» – [15.5 17.5 19.5], для терма «Нормальний» – [20.5 25.5 30.5], 47 для терма «збільшений» – [32.5 33.5 35.5], для терма «великий» – [40.5 42.5 44.5], для терма «Дуже великий» – [46.0 48.0 50.0]. Вид вікна редактора функцій належності після внесених змін наведено на рис. 6.12.

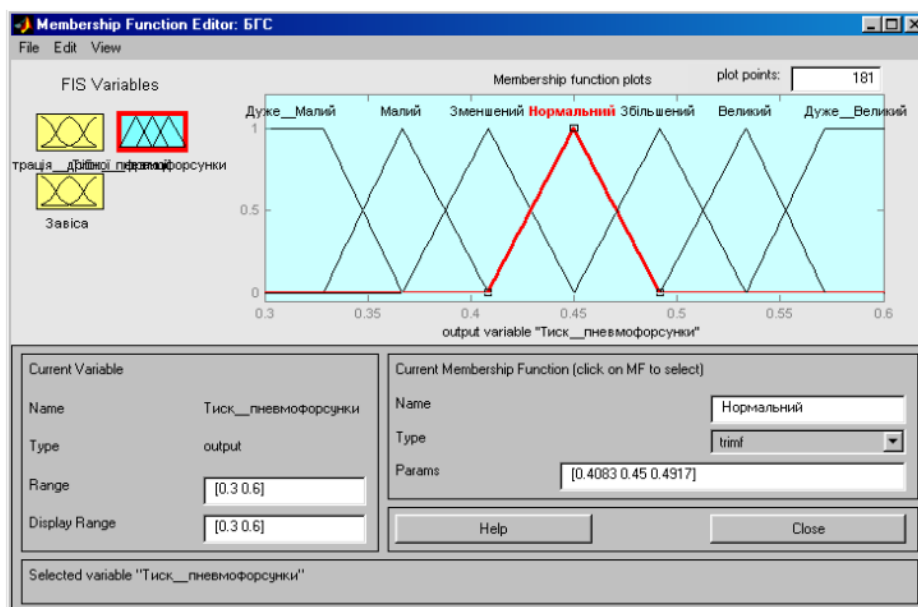


Рис.6.12. Вікно редактора функцій належності після зміни назви термів і типу функцій належності для 2-ої вихідної змінної «Тиск»

Завдання:

1. Створити модель процесу гранулоутворення мінеральних добрив у барабанному грануляторі - сушарці за методикою, описаною в роботі.
2. Розробити математичну модель на базі нечіткого моделювання для власного технологічного об'єкта керування, узгодженого з викладачем.
3. Дослідити типи функцій належності, які надає пакет, і способи їх математичного подання

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Структурно – параметрична схема об'єкта керування.
3. Опис лінгвістичних змінних, які використані у моделі процесу.
4. Скріншоти вікна редактора MATLAB із зображеннями та характеристиками функцій належності.
5. Результати роботи програми для різних початкових ситуацій (скріншоти екрану з результатами роботи програми).
6. Пояснення отриманих результатів.
7. Текст (лістинг) програми.
8. Відповіді на контрольні питання.

Контрольні запитання

1. Поняття нечіткої множини і лінгвістичної змінної.
2. Функція приналежності нечіткої множини.
3. Операції над нечіткими множинами.
4. Чіткі відповідності і взаємозв'язки. Нечіткі збіги і відносини.
5. Терми і функції належності лінгвістичної змінної.
6. Нечіткий висновок. Етапи та особливості нечіткого висновку.
7. Особливості нечіткої логіки в Matlab (Fuzzy toolbox).
8. Як додати вхідну змінну в редакторі пакету Fuzzy Logic?
9. Як змінити тип функції належності?
10. Обчислення значень функції приналежності.

7. Лабораторна робота № 6.

Тема: Розробка проекту експертної системи на мові Visual Prolog

Мета – розробка, впровадження та налагодження навчальної експертної системи, створеної на мові Visual Prolog (*PROgramming LOGic*).

Теоретичні відомості:

Експертні системи (ЕС) – це системи штучного інтелекту (інтелектуальні системи), призначені для розв’язання погано формалізованих і слабо структурованих задач в певних проблемних областях на основі закладених в них знань фахівців-експертів. В даний час ЕС впроваджуються в різні види людської діяльності, де застосування точних математичних методів і моделей ускладнено або навіть неможливо. До них відносяться: медицина, навчання, підтримка прийняття рішень і управління в складних ситуаціях, тощо [4,7,11,12].

Основними компонентами експертної системи є робоча пам'ять, яку також називають базою даних (БД), база знань (БЗ), блоки пошуку рішення, пояснення, вилучення та накопичення знань, навчання та організації взаємодії з користувачем. Робоча пам'ять, БЗ і блок рішень складають ядро ЕС [4,11].

База даних (робоча пам'ять) призначена для зберігання вихідних і проміжних даних розв'язуваної в даний момент задачі. Цей термін збігається за назвою, але не за значенням, з терміном, що використовується в інформаційно-пошукових системах (ІПС) і системах управління базами даних (СКБД) для позначення всіх даних (в першу чергу довгострокових), що зберігаються в системі [11].

База знань в ЕС призначена для зберігання довгострокових даних, що описують розглянуту предметну область (а не поточні дані) і правила, що описують відповідні перетворення даних в цій області [12].

Для побудови ЕС використовуються різні інструментальні засоби: універсальні мови програмування, мови штучного інтелекту, інструментальні системи та середовища та системи оболонки.

Системи оболонки - це найпростіший засіб формалізації (автоформалізації) експертних знань, які практично не вимагають участі посередників в особі інженера знань або програміста при їх використанні. Інженер знань лише допомагає експерту вибрати найбільш підходящу для його проблемної області оболонку.

Розглянемо алгоритм функціонування експертної системи, яку ми намагаємось створити. Наприклад, при роботі з предметною областю «Аудіо техніка» визначається присудок `technique_is(рядок)`, який дозволяє виявити аудіо пристрої, які найбільше задовольняють заданим користувачем характеристикам. При цьому база знань (БЗ) експертної системи задається шляхом перерахування властивостей об'єктів, які у них є або відсутні:

```
technique_is("CD-плеєр") :-  
    it_is("аудио"),  
    positive("роботаєт с", "CD-дисками"),  
    negative("имеет", "динамик").
```

Додатні і негативні предикати необхідні для того, щоб запросити у користувача відповідь про наявність деякої властивості об'єкті і зберегти цю відповідь в робочій пам'яті:

```
positive(X,Y) :-  
    xpositive(X,Y),!.  
positive(X,Y) :-  
    not(xnegative(X,Y)),  
    ask(X,Y,Answer),  
    remember(X,Y,Answer),  
    Answer = yes.  
negative(X,Y) :-  
    xnegative(X,Y),!.  
negative(X,Y) :-  
    not(xpositive(X,Y)),  
    ask(X,Y,Answer),  
    remember(X,Y,Answer),  
    Answer = no.  
ask(X,Y,yes) :-  
    write(X," ",Y,"?"),  
    readln(Reply),  
    frontchar(Reply,'y',_),!.  
ask(_,_,no).  
remember(X,Y,yes) :-  
    assertz(xpositive(X,Y)).  
remember(X,Y,no) :-  
    assertz(xnegative(X,Y)).
```

Вбудовані предикати *asserta* і *assertz*, дозволяють додавати нові твердження в базу даних. Обидва предикати поведуться абсолютно однаково, за винятком того, що *asserta* додає твердження на початок бази даних, тоді як *assertz* (або *assert*) додає твердження в кінець бази даних.

Завдання.

1. Розробити проект простої експертної системи в середовищі Visual Prolog відповідно до наведеної нижче таблиці варіантів завдання або запропонувати свою предметну область (узгодивши її з викладачем).
2. Виконати тестування і відладку експертної системи у Visual Prolog для різних наборів відповідей користувача на питання, задані експертною системою.

Варіанти завдань.

№ варіанту	Опис
1	Розробити проект експертної системи зі <u>зворотною схемою</u> міркувань для визначення причини непрацездатності комп'ютера.
	Ряд можливих причин непрацездатності – 4.
	Кількість ознак непрацездатності – 15.
	Максимальна глибина дерева визначення причини непрацездатності комп'ютера за відповідними параметрами – 3.
2	Розробити проект експертної системи з <u>прямою схемою</u> міркувань для визначення причини несправності комп'ютера.
	Ряд можливих причин непрацездатності – 12.
	Кількість ознак непрацездатності – 6.
	Максимальна глибина дерева визначення причини непрацездатності за відповідними параметрами – 4.
3	Розробити проект експертної системи зі <u>зворотною схемою</u> міркувань для визначення причини, чому авто не заводиться.
	Кількість можливих причин – 4.
	Кількість ознак, що дозволяють визначити причину – 11.
	Максимальна глибина дерева визначення причини за відповідними параметрами – 3.

№ варіанту	Опис
4	Розробити проект експертної системи з <u>прямою</u> схемою міркувань для визначення причини, чому автівка не заводиться.
	Кількість можливих причин – 6.
	Кількість ознак, що дозволяють визначити причину – 7.
	Максимальна глибина дерева визначення причини за відповідними характеристиками – 5.
5	Розробити проект експертної системи зі <u>зворотною</u> схемою міркувань для розробки рекомендацій студенту з планування свого часу для успішного складання іспитів.
	Кількість можливих рекомендацій – 4.
	Кількість факторів, що впливають на розробку рекомендації – 11.
	Максимальна глибина дерева визначення причини за відповідними факторами – 4.
6	Розробити проект експертної системи з <u>прямою</u> схемою міркувань для надання рекомендацій студенту щодо планування свого часу для успішного складання іспитів.
	Кількість можливих рекомендацій – 10.
	Кількість факторів, що впливають на розробку рекомендації – 4.
	Максимальна глибина дерева визначення причини за відповідними факторами – 5.

Зміст звіту по лабораторній роботі:

1. Формулювання мети завдання.
2. Короткий виклад основних теоретичних понять.
3. Постановка завдання з характеристикою порядку виконання робіт.
4. Результати роботи програми для різних початкових ситуацій (скріншоти екрану з результатами роботи програми). Пояснення за результатами роботи програми.
5. Пояснення отриманих результатів.
6. Текст (лістинг) програми.
7. Відповіді на контрольні питання.

Контрольні питання:

1. Етапи розробки експертних систем.
2. Інструментальні засоби побудови експертних систем.
3. Охарактеризуйте особливості програми, створеної на мові Prolog.
4. Опишіть алгоритм вирішення задачі логічним висновком з раніше відомих положень, який запропоновано у мові Visual Prolog.
5. Особливості логіки предикатів.
6. Наведіть структуру програми на мові Visual Prolog.
7. Охарактеризуйте розділ core.
8. Охарактеризуйте розділ constants(констант).
9. Охарактеризуйте розділ domains (доменів).
10. Охарактеризуйте розділ class facts (фактів).
11. Охарактеризуйте розділ predicates (предикатів).
12. Охарактеризуйте розділ clauses (пропозицій).
13. Охарактеризуйте розділ goal (цілей).

Список рекомендованої літератури:

1. Навчання машин та штучний інтелект: конспект лекцій / уклад. Верескун М.В.. – Маріуполь: ДВНЗ «ПДТУ», 2019. – 84 с.
2. Інтелектуальні інформаційні системи: навч. посіб. / С. Шаров, Д. Лубко, В. Осадчий. – Мелітополь: Вид. МДПУ ім. Б. Хмельницького, 2015. – 144 с.
3. Стратегія інноваційного розвитку економіки: бізнес, наука, освіта. Праці VII Міжнародної науково-практичної конференції (29 вересня – 2 жовтня 2015 р.). – Харків.: НТУ «ХП», 2015.– 292 с
4. Баклан І.В. Експертні системи. Курс лекцій // К.: НАУ, 2012. – 132 с.
5. Федорчук Є. Н. Програмування систем штучного інтелекту. Експертні системи. / Видавництво Львівської політехніки, 2012 р. – 168 с.
6. Fuzzy Logic – Algorithms, Techniques and Implementations / Edited by Elmer P. Dadios. – Rijeka, Croatia: InTech, 2012. – 282 pp.
7. Advansed Artificial Intelligence / Zhongzhi Shi. – World Scientific, 2011. – 613 pp.
8. Ross T. J. Fuzzy Logic with Engineering Applications / Timothy J. Ross. – John Wiley & Sons, 2010. – 585 pp.
9. Pool D. L. Artificial Intelligence / David L. Poole, Alan K. Mackworth. – Cambridge University Press, 2010. – 662 pp.
10. Jones M. T. Artificial Intelligence: A Systems Approach / M. Tim Jones. – Infinity Science Press LLC, 2008. – 498 pp.
11. Munakata T. Fundamentals of the New Artificial Intelligence / Toshinori Munakata. – Springer, 2008. – 255 pp.
12. Siler W. Fuzzy Expert Systems and Fuzzy Reasoning / William Siler, James J. Buckley. – New Jersey: John Wiley & Sons, 2005. – 405 pp.

Інформаційні ресурси

1. Mühlenbrock, M., Tewissen, F., & Hoppe, H. U. A Framework System for Intelligent Support in Open Distributed Learning Environments. [Електронний ресурс] // International Journal of Artificial Intelligence in Education, 9, pp. 256-274. Режим доступу: <https://link.springer.com/article/10.1007/s40593-015-0058-8> (дата звернення 15.05.2023).
2. OWL. [Електронний ресурс] // OWL Web Ontology Language Guide W3C. Режим доступу: <http://www.w3.org/TR/owl-guide/> (дата звернення 15.05.2023).
3. Ritter, S. A Model-tracing tutor on the World-Wide Web. Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web" [Електронний ресурс] // AI-ED'97, 8th World Conference on Artificial Intelligence in Education, pp. 11-17. 18 August 1997. Kobe, Japan, ISIR. Режим доступу: http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Ritter/Ritter.html. (дата звернення 15.05.2023).

4. A. Serengul Smith-Atakan, Ann Blandford. ML Tutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System. [Електронний ресурс] // HAL Open science. Режим доступу: <https://telearn.archives-ouvertes.fr/hal-00197318/document>. (дата звернення 15.05.2023).
5. Інженерія знань. Штучний інтелект. [Електронний ресурс] // Інженерія знань. Машинне навчання. Режим доступу: presa.com.ua (дата звернення 15.05.2023).
6. Придбання і формалізація знань. [Електронний ресурс] // Експертні системи. StudFiles. Режим доступу: <https://studfile.net/preview/5745314/page:5/> (дата звернення 15.05.2023).
7. Предикатна логіка. [Електронний ресурс] // Філософія. Режим доступу: <https://tureligious.com.ua/predykatna-lohika/> (дата звернення 15.05.2023).
8. Дейнега Л., Камінська Ж., Левада І., Сердюк С. Практичне програмування мовою Visual Prolog (навчальний посібник) [Електронний ресурс] // Вид-во Запорізького національного технічного університету. Режим доступу: http://eir.zntu.edu.ua/bitstream/123456789/1186/1/Practical_programming_in_Visual_Prolog.pdf (дата звернення 15.05.2023).
9. Методичні вказівки до лабораторних робіт з курсу «Експертні системи» [Електронний ресурс] // Вид-во Кіровоградського національного технічного університету. Режим доступу: <http://core.ac.uk/download/158807205.pdf> (дата звернення 15.05.2023).
10. Моделі подання та методи обробки чітких знань. [Електронний ресурс] // Українськи підручники та статті. Режим доступу: <https://posibniki.com.ua/post-modeli-podannya-ta-metodi-obrobki-chitkih-znan-psshi> (дата звернення 15.05.2023).
11. Тема 10. Експертні та навчальні системи. [Електронний ресурс] // Експертні та навчальні системи. StudFiles. Режим доступу: <https://studfile.net/preview/9790466/> (дата звернення 15.05.2023).
12. Інженерія знань. Прикладні системи штучного інтелекту. [Електронний ресурс] // Українськи підручники та статті. Режим доступу: <https://posibniki.com.ua/post-inzheneriya-znan> (дата звернення 15.05.2023).
13. Довгий С. Комп'ютерні онтології і їх використання у навчальному процесі. [Електронний ресурс] // Монографія, НАПН України. Режим доступу: <https://core.ac.uk/download/pdf/32309159.pdf> (дата звернення 15.05.2023).
14. Стрижак О. Програмно-інформаційні засоби формування систем знань навчального призначення. [Електронний ресурс] // Посібник, НАПН України. Режим доступу: [Strijak_2014_posibnyk.indd](#) (дата звернення 15.05.2023).

15. Конспект лекцій з дисципліни «Експертні системи». [Електронний ресурс] // Вид-во Львівського національного університету ім. Франко, 2020. Режим доступу: https://financial.lnu.edu.ua/wp-content/uploads/2019/09/ES_konspekt-lektsiy.pdf (дата звернення 15.05.2023).
16. Видря Ю. Формування знань про предметну область для ідентифікації об'єктів. [Електронний ресурс] // Магістерська дис. ЕЛАКРІ. Навчально-науковий інститут атомної і теплової енергетики. Київ, 2019. Режим доступу: <https://ela.kpi.ua/handle/123456789/27627> (дата звернення 15.05.2023).
17. Класифікація знань. Прикладні системи штучного інтелекту. [Електронний ресурс] // Українськи підручники та статті. Режим доступу: <https://posibniki.com.ua/post-klasifikaciya-znan-psshi> (дата звернення 15.05.2023).
18. Тема 10.5 Основні терміни та визначення в ІЗ і ЕС. [Електронний ресурс] // Вид-во Запорізького національного технічного університету. Режим доступу: <https://studfile.net/preview/9790466/page:2/> (дата звернення 15.05.2023).
19. Міхалевська Г. Основні концепції побудови експертних систем. [Електронний ресурс] // Informational technologies. Режим доступу: [Basic concepts of building expert systems.doc](https://khnu.km.ua/doc/Basic_concepts_of_building_expert_systems.doc) (khnu.km.ua) (дата звернення 15.05.2023).
20. Шулима О., Шендрик В., Давідсон П. Модель підтримки прийняття рішень при плануванні енергозабезпечення будівель з використанням альтернативних джерел енергії. [Електронний ресурс] // Труды міжнародної науково-практичної конференції «Математичне моделювання процесів в економіці і управлінні проектами і програмами (ММПП-2016)». Режим доступу: [2016.pdf](https://kname.edu.ua/2016.pdf) (kname.edu.ua) (дата звернення 15.05.2023).
21. Jorge Santiago Nolasco Valenzuela. Python Aplicaciones prácticas. [Електронний ресурс] // © De la edición: Ra-Ma 2018/ Режим доступу: https://webooks.co/images/team/academicos/ingenieria/informatica/programacion/lenguajepython/5.Python_Aplicaciones_practicas_Jorge_Sant.pdf (дата звернення 15.05.2023).
22. Федорук П., Масловський С. Модель представлення знань в адаптивній системі дистанційного навчання та контролю знань «EduPRO» [Електронний ресурс] // Штучний інтелект, № 3 [56], 2011. С. 463 – 472. Режим доступу: <http://dspace.nbuv.gov.ua/handle/123456789/60073> (дата звернення 15.05.2023).
23. Тема 5. Нечітка логіка. Основні визначення. [Електронний ресурс] // (Стислий конспект). Теорія алгоритмів та математична логіка. Режим доступу: https://elearning.sumdu.edu.ua/free_content/lectured:075b2e8a0bfe48bcef0ab3106c6d51679abc41f9/latest/117286/index.html (дата звернення 15.05.2023).

Основні терміни та визначення. [8,9,11,21]

- **Абстракція (abstraction)** - подання тільки тієї інформації, що потрібна для досягнення заданої мети, є необхідним засобом керування складними процесами. Інформація (у широкому розумінні) - це будь-які відомості про певний об'єкт, процес або явище.
- **Алгоритми розв'язання** – логічна послідовність етапів, необхідних для виконання визначених задач.
- **Апостеріорні знання** (від лат. a posteriori - з наступного) - знання, отримані за допомогою органів почуттів - є протилежними стосовно апріорних знань.
- **Апріорні знання** (від лат. a priori - з попереднього) передують знанням, отриманим за допомогою органів почуттів, і не залежать від них.
- **Асоціація** - це зв'язок між двома незалежними поняттями, за якого необхідно врахувати відповідність між екземплярами сутностей, належних екстенсіоналам понять одного або різних типів.
- **Алгоритм** - формальна процедура, яка гарантує отримання оптимального або коректного рішення.
- **База даних** - набір фактів, тверджень і висновків, що використовуються у разі зіставлення з правилами в системі, яка ґрунтується на правилах.
- **База знань** - частина системи, базованої на знаннях або експертної системи, що містить предметні знання.
- **Базові елементи знання** - інформація про властивості об'єктів реального світу.
- **Відношення** - виражають як елементарні властивості базових елементів, так і відношення між концепціями.
- **Диспетчер** - частина механізму виведення, яка вирішує, коли і в якому порядку застосовувати різні частини предметних знань.
- **Декларативні знання** (дескриптивне уявлення) - знання про те, чи є певне твердження істинним або помилковим - включають факти або аксіоми і правила, що стосуються цих фактів.
- **Евристики** - правило, що спрощує чи обмежує пошук рішення у предметній галузі, яка є складною або недостатньо вивченою.

- **Експерт** - людина, яка має досвід винятково ефективно розв'язувати задачі, що належать до конкретної предметної царини.
- **Експертні знання** - спеціалізований різновид знань і навичок, яким володіють експерти.
- **Експертна система** - комп'ютерна програма, яка використовує експертні знання для забезпечення високоефективного рішення у вузькій предметній царині.
- **Емпіричні знання**, отримані шляхом спостереження за навколишнім середовищем, є ще одним видом суб'єктивних знань.
- **Епістемологія** - наука про знання. У рамках цієї науки розглядаються характер, структура і походження знань.
- **Засіб побудови експертної системи** - мова програмування та пакет програм, що використовуються під час створення експертної системи.
- **Засоби підтримки** - програми та апаратура, пов'язані з засобами побудови експертних систем.
- **Зміст поняття** - сукупність усіх істотних ознак (властивостей) певного поняття, які дозволяють однозначно ідентифікувати це поняття. Іншими словами, обсяг поняття включає всі предмети і об'єкти ПРГ, які можуть бути описані цим поняттям.
- **Знання** - це формалізована система суджень із принциповою і єдиною організацією, заснована на об'єктивній закономірності, що простежується у визначеній ПРГ (принципи, зв'язки, закони), установленій у результаті розумової діяльності людини, спрямованої на узагальнення досвіду, отриманого нею у результаті практичної діяльності, яка дозволяє ставити і вирішувати завдання в цій сфері.
- **Інженер знань** - людина, яка проектує і створює експертну систему.
- **Інженерія знань** - розділ теорії штучного інтелекту, що вивчає процеси і методи одержання, подання і формалізації знань для розроблення систем, заснованих на знаннях, зокрема, експертних систем.
- **Інструмент** - скорочене позначення засобу побудови експертної системи.
- **Інтерпретатор** - частина механізму виведення, що вирішує, яким чином застосовувати предметні знання.

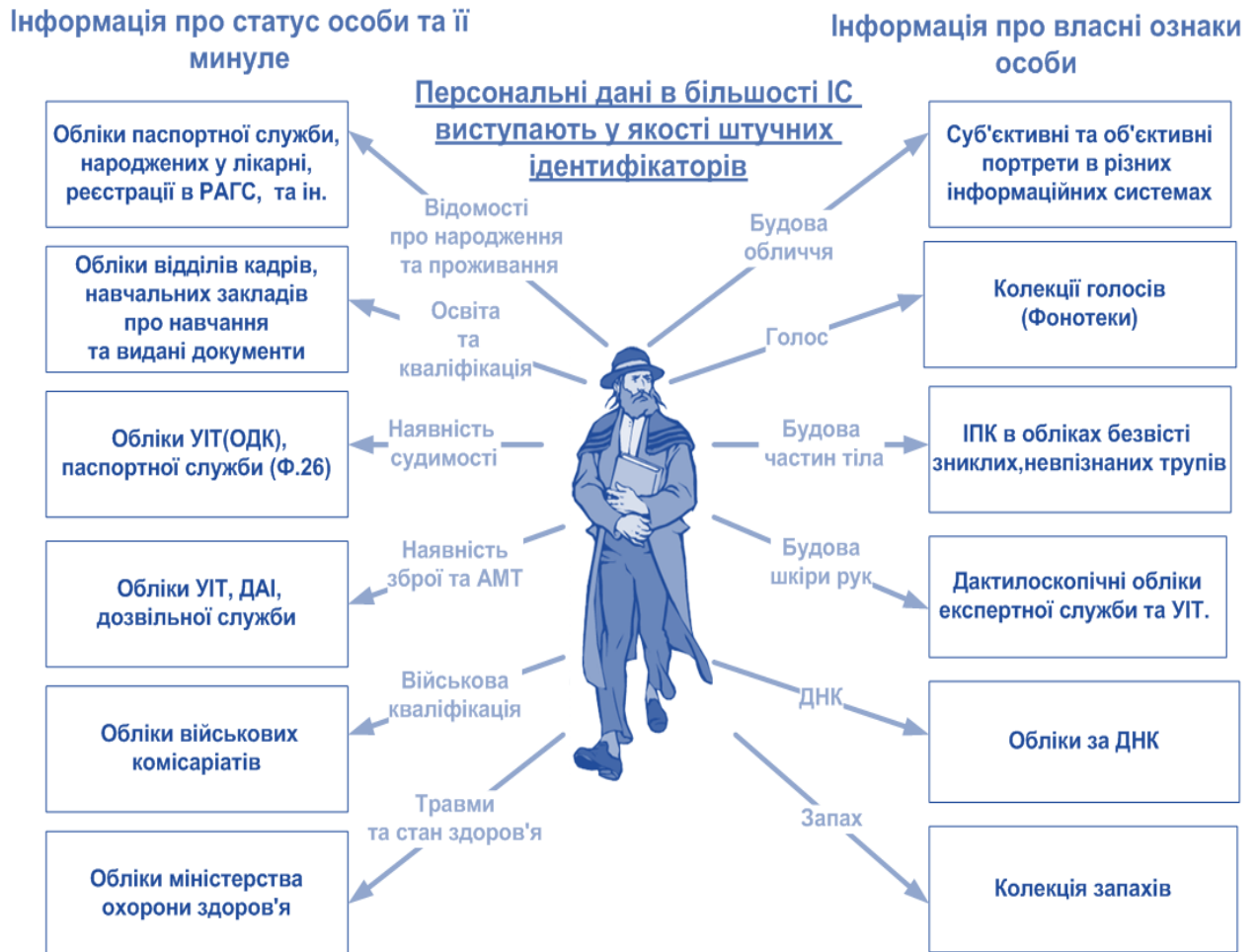
- **Кінцевий користувач** - людина, яка використовує закінчену експертну систему.
- **Коефіцієнт упевненості** - число, яке означає вірогідність або ступінь упевненості, з яким можна вважати даний факт чи правило достовірним або справедливим.
- **Концепт поняття** - включає як власні ознаки, що ідентифікують поняття, так і ознаки, що дозволяють встановити зв'язок даного поняття з іншими (валентні ознаки).
- **Концепції** - перегруповання або узагальнення базових елементів.
- **Механізм виведення** - частина експертної системи, яка містить загальні знання щодо схеми управління розв'язанням задачі.
- **Метазнання** - спеціальним чином організовані знання про знання з метою реалізації процесу їх інтерпретації та планування виведення.
- **Мова подання (representation language)** - комп'ютерна мова, орієнтована на організацію описів об'єктів і ідей, на противагу статичним послідовностям інструкцій або збереженню простих елементів даних.
- **Набуття знань** - це передача потенційного досвіду розв'язання проблеми від певного джерела знань і перетворення його у вигляд, що дозволяє використовувати ці знання.
- **Неявні знання** - підсвідомі знання, що не можуть бути виражені за допомогою мови. Якщо йдеться про комп'ютерні системи, то знання, подані в штучній нейронній системі, нагадують неявні знання, оскільки зазвичай нейронна мережа нездатна безпосередньо пояснити суть знань, які містяться в ній, але могла б набути таку здатність за наявності відповідної програми.
- **Пошук** - організований перегляд простору можливих рішень, що гарантує ефективне віднайдення прийнятного рішення.
- **Правило** - формальний спосіб завдання рекомендацій, директив або стратегій, викладених за допомогою:
 - **ЯКЩО <ПЕРЕДУМОВА> ТО <ВИСНОВОК>**
 - **ЯКЩО <УМОВА> ТО <ДІЯ>.**
- **Предметні знання** - знання з предметної царини.

- **Предметна область або галузь (ПРГ)** - сукупність взаємозалежних відомостей, необхідних і достатніх для розв'язання даної інтелектуальної задачі. Знання про предметну галузь включають описи об'єктів, явищ, фактів, подій, а також відношень між ними.
- **Процедурні знання** (імперативне уявлення) - знання, що задають послідовності дій, які мають бути виконані, та послідовності цілей, які мають бути досягнуті - належать до процедур обробки інформації і методів логічного виведення.
- **Середовище Детерміноване** - середовище, наступний стан якого цілком визначається поточним станом і дією, виконаною агентом.
- **Середовище Повністю спостережне** - середовище, у якому сенсори агента надають йому доступ до повної інформації про стан середовища в кожний момент часу, необхідної для вибору агентом дії.
- **Середовище Стохастичне** - середовище, наступний стан якого цілком не визначається поточним станом і дією, виконаною агентом.
- **Середовище Частково спостережне** - середовище, у якому через сенсори, що створюють шум і є неточними, або через те, що окремих характеристик його стану просто бракує в інформації, отриманій від джерел, агент не може мати доступ до повної інформації про стан середовища в певний момент часу.
- **Стратегії й евристика** - правила поведінки, які дозволяють у конкретній ситуації прийняти рішення про необхідні дії.
- **Твердження і визначення** - засновані на базових елементах і заздалегідь розглядаються як достовірні.
- **Теореми і правила перезапису** - окремий випадок продукційних правил (правил виду «якщо..., то..., інакше...») з цілком визначеними властивостями.
- **Теорія подання знань** - це окрема галузь досліджень, тісно пов'язана з філософією формалізму і когнітивною психологією.
- **Факти** - це інформація, що розглядається як надійна.
- **Джерела знань** - книги, архівні документи, вміст інших баз знань тощо, тобто деякі об'єктивні знання, у формі, доступної для споживача.

Узагальнений алгоритм структурування знань [3,15].

1. *Крок 1. Визначення вхідних і вихідних даних.* Цей крок визначає напрямок руху в полі знань - від вхідних даних до вихідних. Крім того, структура вхідних і вихідних даних істотно впливає на форму та зміст поля знань.
2. *Крок 2. Складання словника термінів і наборів ключових слів.* На цьому кроці на основі протоколів сеансів добування знань виписуються всі значимі слова, що позначають поняття, явища, процеси, предмети, дії, ознаки тощо.
3. *Крок 3. Виявлення об'єктів і понять.* На основі опрацювання словника вибирають значимі для прийняття рішень поняття та їх ознаки. Утворюється повний систематичний підбір термінів у даній ПРГ.
4. *Крок 4. Виявлення зв'язків між поняттями* і визначення «сили зв'язності». Побудова мережі асоціацій.
5. *Крок 5. Виявлення метапонять і деталізація понять.* Аналіз зв'язків дає змогу структурувати поняття, виявляти метапоняття або деталізувати поняття.
6. *Крок 6. Побудова піраміди знань*, тобто ієрархічної градації понять, підіймання якою означає поглиблення понять і підвищення рівня абстракції (узагальненості) понять.
7. *Крок 7. Визначення відношення* як усередині кожного з рівнів піраміди, так і між рівнями. На цьому кроці даються імена тим зв'язкам, що визначені на кроках 4 та 5. Позначаються причинно-наслідкові й інші види відношень.
8. *Крок 8. Визначення стратегій прийняття рішень*, тобто виявлення логічних схем суджень, що пов'язують усі сформовані поняття і відношення в динамічну систему поля знань. Стратегії додають активності знанням і забезпечують перехід від вхідних даних до рішення.

**Приклад структури бази даних (БД),
на підставі якої формується база знань (БЗ) та створюється
інтелектуальна експертна система «Пошук та ідентифікація особи».**



Структура і архітектура експертної системи ^[9622].

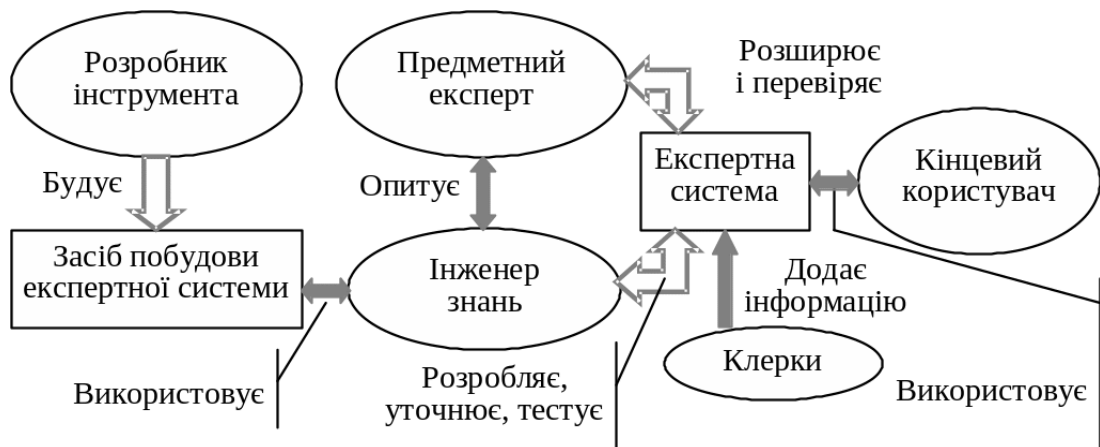


Рис. 1. Схематична структура експертної системи.

Основою експертної системи є сукупність знань, структурована з метою спрощення процесу прийняття рішення експертною системою. Деякі правила експертної системи є *евристичними*, тобто емпіричними правилами чи спрощеннями, що ефективно обмежують процес рішення. Такі правила не піддаються точному математичному або алгоритмічному розв'язанню. *Алгоритмічний метод* гарантує коректне чи оптимальне розв'язання задачі, а *евристичний метод* - дає прийнятне розв'язання в більшості випадків ^[9].

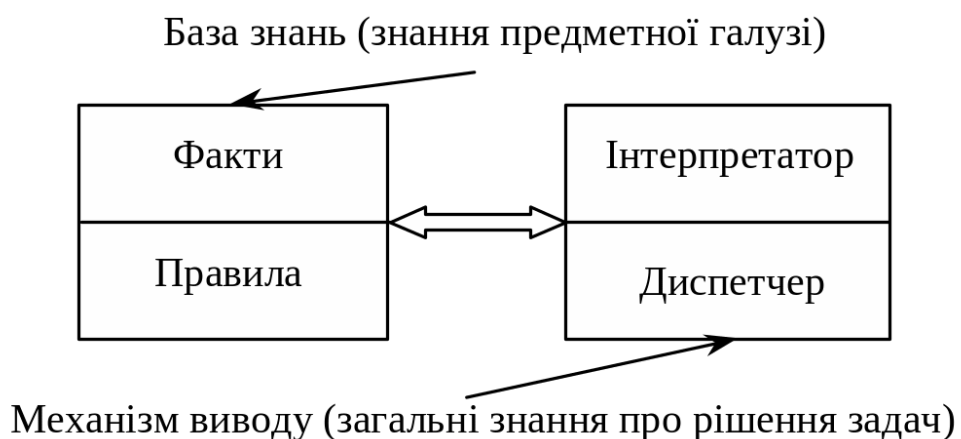


Рис. 2. База знань експертної системи.

Ієрархічна структура підсистем експертної системи та зв'язків між ними наведена на рисунку 3.



Рис. 3. Структура підсистем експертної системи ^[22].

Інші особливості експертних систем, що відрізняють їх від звичайних програм, наведені на рисунку 4.



Рис. 4. Функціональні особливості експертних систем ^[9].

Накопичування досвіду створення експертних систем виявило доцільність створення порожніх експертних систем (оболонок), які не містять знань у області майбутнього застосування і тому дозволяють створити будь-яку експертну систему. Для порожніх експертних систем розробляється план подання знань необхідний для навчання бази знань (БЗ) і вирішення задачі.

Далі з оболонки генерується експертна система, яка відповідає вимогам. Архітектура експертної системи з порожніми оболонками наведена на рис. 5.

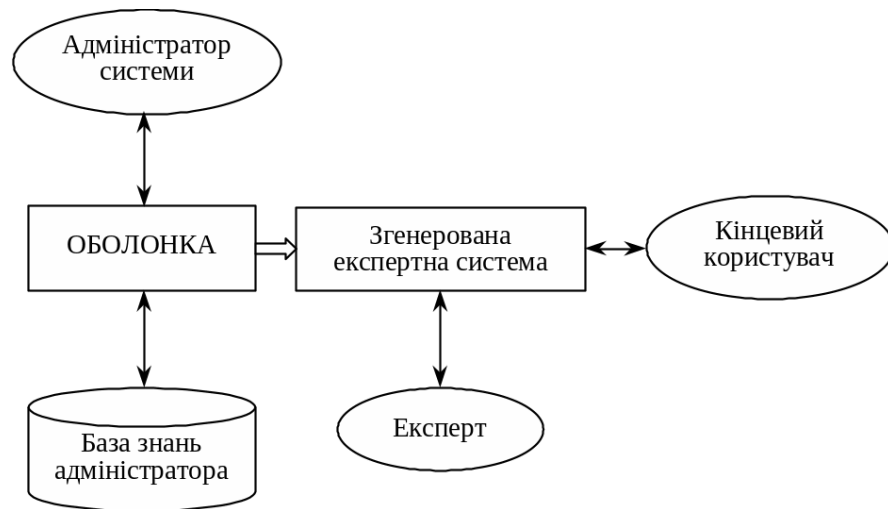


Рис. 5. Архітектура експертної системи з оболонками [22].

Архітектура інтелектуальної інформаційної системи з інтелектуальним інтерфейсом на основі використання баз знань наведена на рисунку 2.

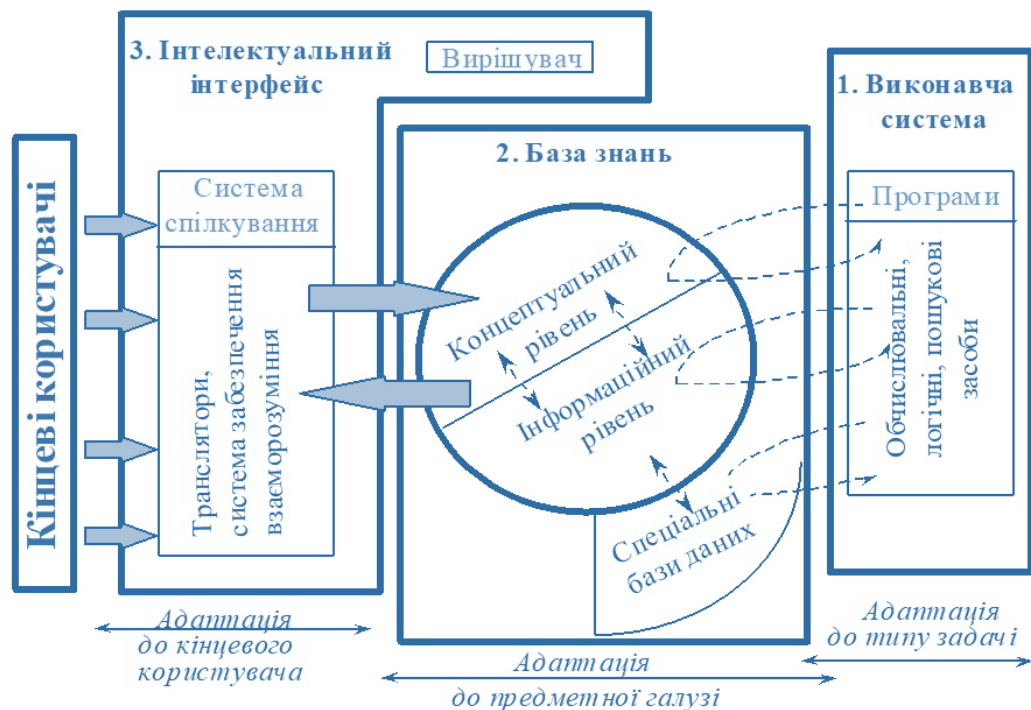


Рис. 6. Архітектура інтелектуальної інформаційної системи [11].

Методи пошуку і отримання знань ^[7].

Методи пошуку знань поділяються на дві великі групи:

- *Методи пошуку явних знань.*
- *Методи пошуку знань прихованих (неявних).*

Методи пошуку явних знань поділяються на дві групи (див. рис.1):

I. Комунікативні.

II. Текстологічні.

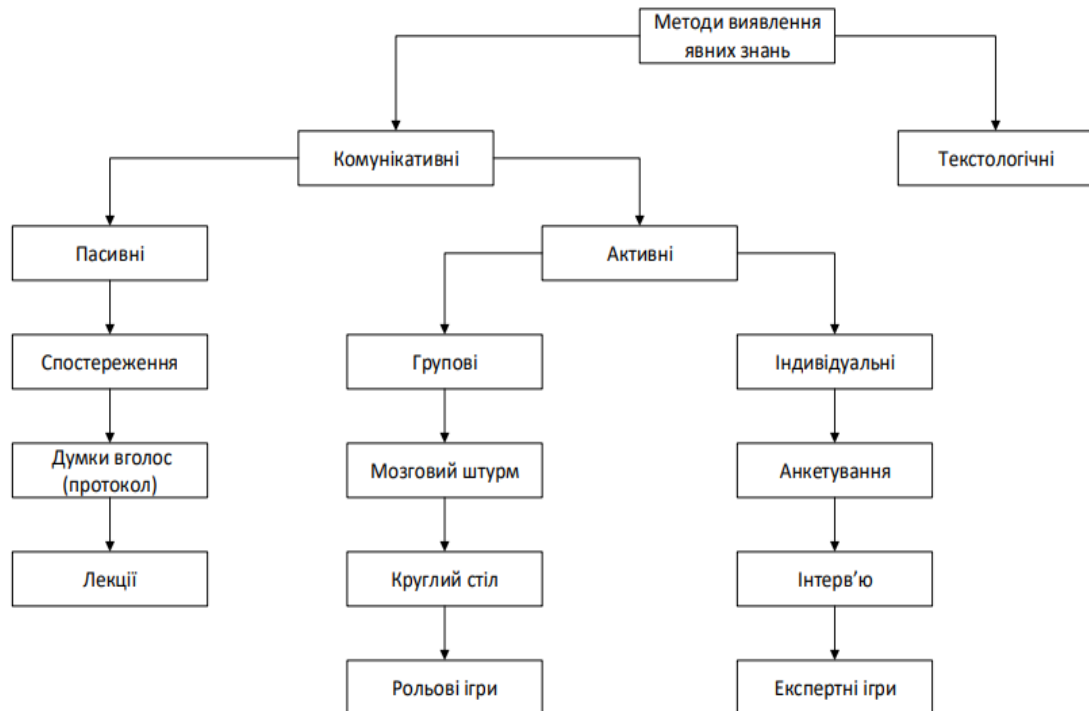


Рис. 1 – Класифікація методів пошуку і отримання явних знань

Методи пошуку неявних знань:

- ❖ *Метод багатовимірного оцінювання* – заснований на методах статистики. Полягає в оцінюванні схожості експертних оцінок певного об'єкта. Результати опрацювання представляються у вигляді точок деякого координатного простору.
- ❖ *Метод метричного оцінювання* – створення нових класів з використанням метричних відстаней. Цей тип обробки даних орієнтовано на максимальне зближення числових значень двох матриць. Важливо нагадати, що даний метод не виявляє нових знань, об'єктів, а виявляє нові простори, а отже його результати необхідно інтерпретувати як відновлену структуру розміщення точок у площині, на координатній площині чи в просторі ^[7].

- ❖ *Метафоричний підхід* – орієнтований на виявлення прихованих знань з практичного досвіду експерта та заснований на порівнянні об'єктів предметної області з абстрактними об'єктами зі світу метафор. У результаті застосування даного методу можливо виявити нові властивості об'єктів, що аналізуються, а також виявити ставлення експерта до цих об'єктів. При застосуванні даного методу експерт виходить за рамки об'єктивності та діє з точки зору свого суб'єктивного уявлення.
- ❖ *Метод репертуарних решіток* – даний метод широко використовуваний у психології. Полягає у заповненні експертом матриці, стовпцями якої є певні будь-які об'єкти, рядками матриці є «конструктори», які представлені біполярними ознаками, що можуть бути звуками, характерною поведінкою об'єкта, кольором і тощо. Іншими словами, конструктор – це така характеристика об'єкта, яку можливо використовувати для розподілу об'єктів на класи. Наприклад «холодний - гарячий», «провідний – непровідний», «мережевий – локальний» тощо.

Всі знання, отримані від експертів, повинні відповідати метрикам цілісності та метрикам експертної системи.

НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни

«ІНЖЕНЕРІЯ ЗНАНЬ І РОЗРОБКА ЕКСПЕРТНИХ СИСТЕМ»

до виконання курсового проекту з дисципліни «Проектування СРЧ»

для студентів спеціальності 122 «Комп'ютерні науки»

усіх форм навчання

Комп'ютерний набір і верстка:

Шамасв Віталій Віталійович

Укладач:

Шамасв Віталій Віталійович

ДВНЗ «Донецький національний технічний університет»

85300, м. Покровськ, пл. Шибанкова, 2.