

Лекція 9.

Продукційні моделі

Продукцією (продукційним правилом) називають вираз виду:

$$(i); Q; P; A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_k; N,$$

де i – ім'я продукції, у якості котрого може виступати деяка лексема, що відбиває суть даної продукції або її порядковий номер, Q – елемент, що характеризує сферу застосування продукції, $A_1, A_2, \dots, A_n \rightarrow B_1, B_2, \dots, B_k$ – ядро продукції, « \rightarrow » – знак секвенції, A_i – i -та передумова (умова) правила, B_j – j -ий висновок (наслідок, дія) правила, P – умова застосовності ядра продукції, N – постумови продукції.

Продукційні правила читаються в такий спосіб: якщо передумови A_1 та A_2 та ... та A_n є вірними, то виконати дії B_1 та B_2 та ... та B_k .

Сфера застосування – визначає для яких випадків може бути застосована продукція, тобто задає множину елементів для якої продукція є застосовною. Поділ знань на окремі сфери дозволяє заощаджувати час на пошук потрібних знань.

Ядро продукції складається з двох частин:

– *антецедент* (передумова, умови правила) – являє собою комбінацію *умов правила* (припущень про наявність деяких властивостей, що приймають значення істина або хибність з визначеним ступенем вірогідності), з'єднаних логічними зв'язуваннями (ТА, АБО і т. д.), призначену для розпізнання ситуації, коли це правило повинне спрацювати: правило спрацьовує, якщо факти з робочої пам'яті задовольняють умовам передумови правила, після цього правило вважається відпрацьованим. Передумови звичайно бувають подані у формі вектора *об'єкт – атрибут – значення*. Впевненість у вірогідності передумови залежить від того, наскільки достовірною є оцінка умов;

– *консеквент* (висновок, наслідки правила) – містить опис дій, що повинні бути виконані над робочою пам'яттю у випадку виконання відповідних умов.

Іноді використовується й інша термінологія, відповідно до якої передумови називаються *лівою частиною* правила, а дії – *правою*.

Умова застосовності ядра продукції – визначає при якій умові продукція може бути виконана. Звичайна умова застосовності ядра являє собою логічний вираз. Коли вона приймає значення «істина», ядро продукції може бути активізовано. Якщо умова є хибною, то ядро продукції не може бути використано.

Постумови продукції – описують дії і процедури, які необхідно виконати після реалізації наслідків продукції й актуалізуються тільки в тому випадку, якщо ядро продукції реалізувалося.

Продукційна система (production system) складається з *продукційної пам'яті* (production memory) – бази знань у вигляді продукційних правил, *машини логічного виведення*, що послідовно визначає, які продукції можуть бути активовані в залежності від умов, у них що містяться, вибирає одне з застосовних у даній ситуації правил продукцій і виконує дії для обраного правила, а також *робочої пам'яті*, що містить дані (факти), опис мети і проміжні результати, що у сукупності визначають поточний стан проблеми.

Опис поточного стану проблеми, поданий фактами робочої пам'яті, є *зразком*, що зіставляється з умовною частиною продукцій з метою вибору відповідних дій при вирішенні задачі.

Керування системою продукції (механізм виведення) здійснюється за допомогою машини логічного виведення, що виконує дві функції:

- перегляд існуючих фактів з робочої пам'яті і правил з бази знань і додавання (у міру можливості) у робочу пам'ять нових фактів;
- визначення порядку перегляду і застосування правил.

Керування пошуком у продукційній системі здійснюють:

– *за допомогою структури правил*, що у продукційній системі, включаючи розходження між умовою і дією, а також порядок перевірки умов, визначає метод дослідження простору рішень. Оскільки продукційна система перевіряє правила у визначеному порядку, програміст може керувати пошуком через структуру і порядок проходження правил у продукційному наборі;

– *на основі зразків (pattern-directed search): зразок* – опис задачі, що подає поточний стан світу, визначає конфліктну множину і, отже, конкретний шлях пошуку і рішення задачі.

Типи виконання систем продукції виділяють:

- *прямий* (висхідний): пошук йде від лівих частин продукції, тобто перевіряються умови й актуалізуються ті продукції, для яких умови виконуються;
- *зворотний* (спадний): пошук йде від початково заданих висновків, за якими визначаються необхідні для висновків значення умов, що, у свою чергу, ототожнюються з правими частинами ядер продукції у системі.

Чиста продукційна модель не має ніякого механізму виходу з тупикових станів у процесі пошуку; вона просто продовжує працювати доти, поки не будуть вичерпані всі припустимі продукції. Багато практичних реалізацій продукційних систем містять механізми повернення в попередній стан робочої пам'яті.

Машина логічного виведення працює в режимі здійснення циклів «розпізнавання – дія» (цикл «вибрання – виконання», цикл «ситуація – відгук», цикл «ситуація – дія»): вона послідовно в циклі виконує деякі групи задач до виявлення визначених критеріїв, що викликають припинення виконання, при цьому в одному циклі може спрацювати тільки одне правило (див. рис. 2.3).

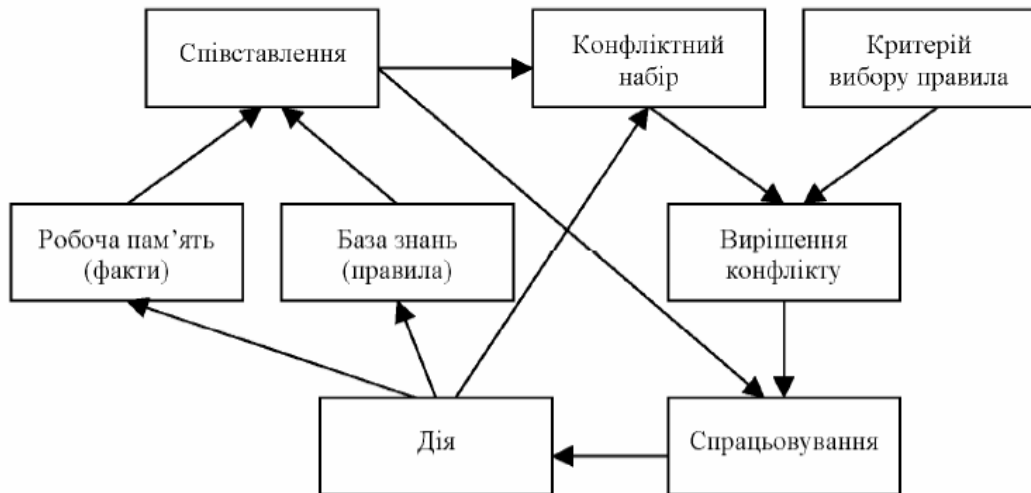


Рисунок 2.3 – Схема циклу роботи механізму виведення

Метод роботи машини логічного виведення містить такі кроки.

1. *Ініціалізація:* робоча пам'ять ініціалізується зразком – початковим описом задачі в у робочій пам'яті.

2. *Цикл «розпізнавання – дія».*

2.1 *Ідентифікація* (зіставлення, узгодження, розпізнавання) – порівняння умов з лівих частин продукцій з фактами робочої пам'яті, що призводить до конкретизації й активізації правил.

Конкретизація правила – сукупність узагальненого формулювання правила і значень використовуваних у ньому змінних.

Активізоване (реалізоване) – правило, всі умови якого задоволено. Всі активізовані правила заносяться до робочого списку правил.

Конфлікти правил виникають у робочому списку правил, якщо одночасно активізується кілька правил.

Конфліктний набір правил (конфліктна множина, conflict set) – набір конкретизованих правил, які активізовані протягом одного циклу обчислень.

Припустимі продукції – продукції, що містяться в конфліктній множині.

2.2 Вирішення конфліктів правил (conflict resolution) – вибір зі сформованого списку заявок (конфліктної множини) єдиного правила, яке має бути виконане в поточній ситуації. Не маючи механізму вирішення конфліктів, продукційна система буде не в змозі ефективно справлятися з відсутністю детермінізму в наборі правил, обробкою виключень і переключенням уваги на визначений стиль розмірковувань. Іншими словами, подання буде страждати відсутністю евристичних здібностей, а керувати функціонуванням такої системи буде досить важко, навіть якщо знання подано цілком коректно.

2.3 *Дія* (запуск правил) – зміна вмісту робочої пам'яті за допомогою послідовного виконання дій, зазначених у правій частині активізованого правила, обраного з робочого списку правил. Після виконання дії, відповідне правило видаляється з робочого списку правил або виконується його релаксація.

Терміну запуск у нейрофізіології (науці про те, як працює нервова система) відповідає термін збудження. У результаті стимуляції окрема нервова клітина випускає електричний сигнал. Подальша стимуляція, наскільки сильною вона б ні була, не може змусити нейрон знову збудитися, поки не пройде якийсь короткий період часу. Цей феномен називається *релаксацією*. Експертні системи, засновані на правилах, створюються з використанням релаксації з метою запобігання тривіальних циклів. Для забезпечення релаксації розроблено різні методи. Наприклад, кожному факту після його введення в робочу пам'ять присвоюється унікальний ідентифікатор, називаний *часовою оцінкою* (timetag). Слідом за тим, як деяке правило запускається під впливом деякого факту, машина логічного виведення не запускає його знову під впливом того ж самого факту, якщо з моменту застосування його часової оцінки пройшло занадто мало часу.

2.4 *Перевірка умов зупинення.* Якщо хоча б одна з умов зупинення задовольняється, тоді закінчити роботу і повернути рішення, якщо воно знайдено, у протилежному випадку – повторити кроки 2.1–2.4.

Умови зупинення виділяють такі:

- переривання роботи користувачем;
- вичерпання всіх правил із продукційної пам'яті;
- виконання деякої умови, якій задовольняє вміст робочої пам'яті (наприклад, поява в ній якогось факту);
- невідповідність вмісту робочої пам'яті ніяким умовам.

Протягом кожного циклу можуть бути активізовані і поміщені в робочий список правил багато правил. Крім того, у робочому списку правил залишаються результати активізації правил від попередніх циклів, якщо не відбувається *деактивізація* цих правил у зв'язку з тим, що їхні ліві частини більше не виконуються. У такий спосіб у ході виконання програми кількість активізованих правил у робочому списку правил змінюється. У залежності від програми раніше активізовані правила можуть завжди залишатися в робочому списку правил, але ніколи не вибиратися для запуску. Аналогічним чином деякі правила можуть ніколи не ставати активізованими. У подібних випадках варто повторно перевіряти призначення цих правил, оскільки або такі правила взагалі не потрібні, або їхні антецеденти неправильно спроектовано.

Використання різних стратегій перебору наявних знань, як правило, досить істотно впливає на характеристики ефективності програми. Ці стратегії визначають, яким чином програма відшукує рішення проблеми в деякому просторі альтернатив. Як правило, не буває так, щоб дані, які має програма роботи з базою знань, дозволяли точно визначити ту область у цьому просторі, де є сенс шукати відповідь. Оскільки стратегія вирішення конфліктів впливає на продуктивність системи в цілому, у більшості програмних систем передбачаються визначені опції для підстроювання цього механізму.

Стратегії вирішення конфліктів правил виділяють такі.

– *Розмаїтість* (рефракція – refraction): після активізації правила воно не може бути запущене знову, поки не зміняться елементи робочої пам'яті, що відповідають його умовам. Найпростіший варіант реалізації цього механізму – видаляти зі списку заявок застосоване раніше правило. Іноді використовується інший варіант: зі списку видаляється правило, активізоване в попередньому циклі, що запобігає зацикленню, але якщо бажано саме повторювати процедуру, то в розпорядження програміста надається функція відновлення, що дозволяє тимчасово придушити механізм, який діє за замовчуванням.

– *Новизна* (recency) – віддає перевагу правилам, умови яких відповідають фактам, доданим у робочу пам'ять останніми. Це дозволяє зосередити пошук на одній лінії суджень. Елементи в робочій пам'яті в таких системах позначаються спеціальним атрибутом часу породження. Це дозволяє системі ра-

нжирувати елементи в списку заявок відповідно до того, як давно введені в робочу пам'ять дані, що використовувалися при зіставленні, а потім пріоритет віддається правилам, що реагують на більш свіжі дані. Ідея полягає в тому, щоб впливати за поточною хвилиною і менше уваги приділяти тим даним, що були давно сформовані. До них можна буде повернутися надалі, якщо поточний ланцюжок розсудів наштотхнеться на яку-небудь перешкоду.

– *Новизна правил*: найбільш нові правила, введені в систему в останню чергу, здобувають за замовчуванням найвищий пріоритет.

– *Старовина правил*: найбільш старі правила, введені в систему в першу чергу, здобувають за замовчуванням найвищий пріоритет.

– *Стратегія глибини* – утілення стратегії новизни даних стосовно правил, що мають однаковий клас опуклості. Правила, обрані в список заявок на підставі даних, що були включені в робочу пам'ять порівняно недавно, розташовуються в цьому списку раніше правил, при виборі яких використані більш старі дані. Таким чином, перевага віддається принципу пошуку в глибину в просторі станів проблеми, тобто правила, що є наслідком більш пізніх змін стану системи, мають визначений пріоритет.

– *Стратегія ширини* – протилежна стратегії глибини і призначається для реалізації пошуку в ширину в просторі станів проблеми. Правила, обрані в список заявок на підставі даних, що були включені в робочу пам'ять порівняно давно, розташовуються в цьому списку раніше правил, при виборі яких використані більш свіжі дані.

– *Стратегія складності* (специфічність – specificity, принцип найбільш довгої умови) – спирається на розуміння «здорового глузду», що частки правила, які відносяться до вузького класу ситуацій, є важливішими загальних правил, що відносяться до широкого класу ситуацій, оскільки частки правила враховують більше інформації про ситуацію, ніж загальні, і полягає у виборі з фронту готових продукцій тієї, у якої складність більше. *Складність правила* визначається кількістю операцій перевірки, які потрібно виконати при аналізі умов даного правила. Одне правило є більш специфічним (складним, конкретним) ніж інше, якщо воно містить більше умов, і відповідно складніше задовольняється, а виходить, відповідає меншій кількості правил у робочій пам'яті і має пріоритет перед більш загальними правилами. Цю стратегію можна ефективно використовувати при роботі з виключеннями з загальних правил, коли знання і самі продукції добре структуровані прив'язкою до типових ситуацій, на яких задане відношення типу «часткове – загальне». Труднощі використання даного принципу полягає в тому, що треба заздалегідь упорядкувати умови за входженням одна до одної за відношенням «часткове – загальне».

– *Стратегія простоти* – протилежна стратегії складності: більш верхнє положення (більший пріоритет) у списку заявок при реалізації цієї стратегії віддається тим правилам, складність яких нижче.

– *LEX-стратегія* – припускає спочатку видалення зі списку заявок усіх правил, що вже були раніше використані. Правила, що залишилися з рівним значенням опуклості, потім впорядковуються за новизною використовуваних даних. Якщо виявиться, що два правила використовують дані однакової новизни, то перевага віддається тому правилу, що втягує в аналіз передумов більше даних. Метод LEX практично ідентичний методу MEA за одним виключенням – у ньому відсутній крок 2, а на кроці 3 порівнюються всі елементи умов конкретизованих правил і зв'язаних з ними елементів робочої пам'яті (перші елементи умов конкретизованих правил, є, як правило, лексемами задач у робочій пам'яті).

– *МЕА-стратегія* (Mean-Ends Analysis) багато в чому аналогічна LEX-стратегії, але при аналізі новизни беруться до уваги тільки перші умови в передумовах правил. Якщо виявиться, що в списку заявок виявилися два претенденти з рівними показниками, то для вибору між ними застосовується механізм LEX-стратегії. Метод MEA включає п'ять кроків.

Крок 1. Виключити з конфліктної множини правил ті правила, що вже були застосовані в попередньому циклі. Якщо після цього множина стала порожньою, припинити процес.

Крок 2. Порівняти новизну елементів у робочій пам'яті, що відповідають першим елементам умов у конкретизованих правилах, які залишилися в конфліктній множині. Пріоритет віддається тим правилам, що звертаються до найновіших елементів робочої пам'яті, – ці правила домінують над іншими. Якщо існує єдине таке правило, то воно вибирається для застосування, після чого процес повинний бути припинений. У протилежному випадку, якщо є кілька домінуючих правил з рівним пріоритетом, вони зберігаються в конфліктній множині, а інші з неї видаляються. Далі виконується перехід до кроку 3.

Крок 3. Упорядкувати конкретизовані правила за новизною інших елементів умов у правилах. Якщо домінує одне правило, воно застосовується і потім процес припиняється. Якщо ж домінують два чи кілька правил, то вони залишаються в конфліктуючій множині, а інші видаляються з неї. Потім виконується перехід до кроку 4.

Крок 4. Якщо за критерієм новизни елементів умов відібрано кілька правил з рівними показниками, то порівнюється інший показник правил – показник специфіки. Перевага віддається тому правилу, застосування якого вимагає перевірки найбільшої кількості умов у робочій пам'яті. Якщо претендентів виявиться кілька, інші видаляються з конфліктної множини і виконується перехід до кроку 5.

Крок 5. Застосовується правило, обране довільним чином з тих, що залишилися у конфліктній множині, і процес припиняється.

Таким чином, стратегія МЕА поєднує в одному методі аналіз таких показників, як повторюваність, новизна і специфіка.

– *Принцип «стопки книг»* (частотна перевага) – заснована на ідеї, що найбільш часто використовувана продукція є найбільш корисною. Готові продукції як би утворюють «стопку», у якій порядок визначається накопиченою частотою використання продукцій у минулому. На самому верху «стопки» знаходиться продукція, що використовувалася частіше усіх. При актуалізації деякого фронту готових продукцій для виконання вибирається та продукція (чи ті продукції при наявності рівнобіжних технічних пристроїв), у якої частота використання є максимальною. Подібний принцип є особливо гарним, коли частота виконання підраховується з урахуванням деякої ситуації, у якій раніше виповнювалася продукція, і це виконання мало позитивну оцінку. При такому зворотному зв'язку метод стопки книг може перетворитися в процедуру, що навчається та адаптується до тих задач, що виникають у зовнішньому середовищі. Принцип стопки книг доцільно застосовувати, якщо продукції є відносно незалежними одна від одної.

– *Принцип метапродукції* – заснований на ідеї введення в систему продукцій спеціальних метапродукцій (метаправил), завданням яких є організація керування в системі продукцій при можливості неоднозначного вибору з фронту готових продукцій.

– *Керування за іменами* – засновано на задаванні для імен продукцій, що входять у деяку систему, певної формальної граматики або іншої процедури, що забезпечує звуження фронту готових продукцій і вибір з нього чергової продукції для виконання.

– *Упорядкування на основі цілей*: правила упорядковуються, принаймні частково, у залежності від того, наскільки швидко вони приводять до досягнення мети. Виконується пошук випадків, ціль яких відповідає поточній ситуації.

– *Принцип пріоритетного вибору* – пов'язаний із уведенням статичних або динамічних пріоритетів на продукції. Статичні пріоритети можуть формуватися апріорі на підставі відомостей про важливість продукційних правил у даній проблемній області. Ці відомості, як правило, являють собою інформацію, що витягається з експерта. Динамічні пріоритети виробляються в процесі функціонування системи продукцій і можуть відбивати, наприклад, такий параметр, як час перебування продукції у фронті готових продукцій. До даного типу керування відноситься задавання послідовності пріоритетів за допомогою спеціальної каузальної семантичної мережі. У цьому випадку задається певний каузальний сценарій, рух за яким визначається виникаючими ситуаціями, і в кожній вершині якого задана функція вибору чергової продукції з фронту готових продукцій.

– *Модель дошки оголошень* (модель класної дошки) – стратегія вирішення складних системних задач із залученням різнорідних джерел знань, взаємодіючих через загальне інформаційне поле – модель керування, що успішно застосовується для вирішення задач, які вимагають координації різних процесів або джерел знання, яка ґрунтується на ідеї спускових функцій (див. рис. 2.4).

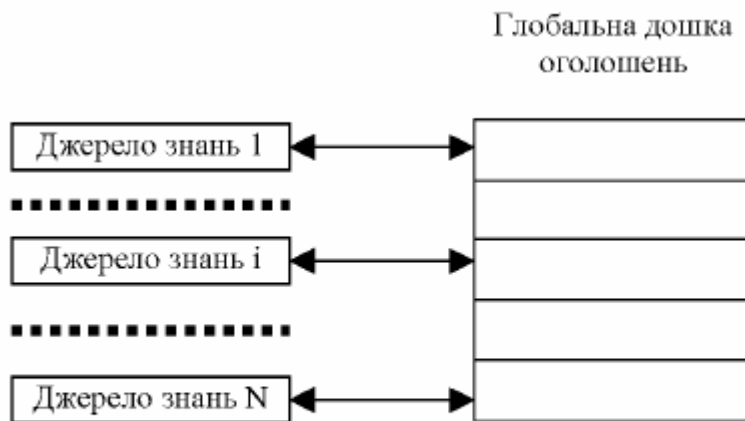


Рисунок 2.4 – Архітектура дошки оголошень

Метою розробки моделі дошки оголошень було створення такої системи, що компенсує обмеженість монотонних виведень і може застосовуватися для вирішення складних проблем за рахунок використання механізму поділу джерел знань та їхніх погоджених дій у загальній робочій пам'яті зі збереженням при цьому такої переваги продукційної системи, як модульність знань.

Дошка оголошень (класна дошка – blackboard) – центральна глобальна база даних, призначена для зв'язку незалежних асинхронних джерел знань.

Як правило, на дошці оголошень виділяють спеціальні поля для формування умов застосовності ядер продукцій, різні для різних сфер застосування продукцій, спеціальні поля для запису результатів спрацьовування продукцій і для запису постумов, якщо вони адресовані іншими продукціям.

Одне з джерел знань, називане *планувальником*, обробляє повідомлення про результати обробки інформації, передані між іншими джерелами знань. Планувальник ранжирує результати діяльності кожного джерела знань

і за допомогою пріоритетної черги забезпечує деякий напрямок вирішення задачі. Якщо жодне з джерел знань не активним, планувальник вирішує, що задача є завершеною, і припиняє свою роботу.

З принципом дошки оголошень може комбінуватися принцип керування за допомогою метапродукцій, оскільки він вимагає перевірки деяких умов, що фіксуються в робочому полі пам'яті, а також інші принципи керування.

Перевагами продукційних моделей є:

- модульність організації знань у вигляді продукційних правил;
- незалежність правил, що виражають самостійні фрагменти знань про предметну область. Важливий аспект у моделюванні продукційних систем – це відсутність синтаксичної взаємодії між продукційними правилами. Правила можуть тільки впливати на активізацію інших правил, змінюючи зразок у робочій пам'яті. Правила не можуть «викликати» інше правило безпосередньо. При цьому вони не можуть установлювати значення змінних в інших продукційних правилах. Область дії змінних цих правил обмежена окремим правилом. Ця синтаксична незалежність сприяє інкрементальній розробці експертних систем шляхом послідовного додавання, видалення або зміни знань (правил) системи;
- простота створення і розуміння окремих правил;
- легкість і природність модифікації знань;

- відділення керуючих знань, забезпечуваних циклом «розпізнання – дія» продукційної системи, від предметних знань, зосереджених безпосередньо в правилах, що дозволяє застосовувати різні керуючі стратегії і легко змінювати базу знань, а також не вимагає зміни коду програми керування і, навпаки, дозволяє змінювати код керуючої частини програми, не торкаючи набір правил виведення;

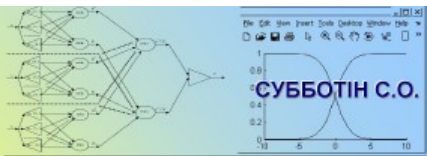
- природний паралелізм у системі продукцій, асинхронність їхньої реалізації, що роблять продукційні системи зручною моделлю обчислень для ЕОМ нової архітектури, у якій ідея паралельності й асинхронності є центральною;

- простота механізму логічного виведення;

- гнучкість у застосуванні знань;

- природна відповідність пошуку в просторі станів: компоненти продукційної системи природно відображаються в логічну структуру пошуку в просторі станів. Послідовні стани робочої пам'яті складають вершини графа простору станів. Правила виведення – набір можливих переходів між станами. Вирішення конфліктів забезпечує вибір переходу (гілки) у просторі станів. Ці правила спрощують виконання, відлагодження і документування методів пошуку.

Недоліками продукційних моделей є: неясність взаємних відношень правил; складність оцінки цілісного образу знань; у край низька ефективність обробки; відмінність від людської структури знань; складність перевірки не-суперечності системи продукцій при великому числі продукцій, що змушує при додаванні нових продукцій витрачати багато часу на перевірку не-суперечності нової системи; недетермінованість (неоднозначність вибору виконуваної продукції з фронту активізованих продукцій), що обумовлює принципові труднощі при перевірці коректності роботи системи (вважається, що якщо в інтелектуальній системі число продукцій досягає тисячі, то мало шансів, що система продукцій у всіх випадках буде правильно функціонувати).



ПОДАННЯ Й ОБРОБКА ЗНАЙЬ У СИСТЕМАХ ШТУЧНОГО ІНТЕЛЕКТУ ТА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

навчальний посібник



Запорізький
національний технічний
університет

2008