

Конспект лекції № 6

Тема № 6. ПОДАННЯ ЗНАНЬ

Міжпредметні зв'язки: Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху” та „Інформатика”, «Методологія тестування програмного забезпечення».

Мета лекції: охарактеризувати поняття процесу здобуття знань; зробити огляд структуризації знань предметної області; розглянути методи здобуття знань.

План лекції

1. Поняття подання знань та її принципи.
2. Моделі подання знань.

Опорні поняття: подання знань, принципи подання знань, логічні моделі, продукційні моделі, семантичні мережі, фреймові моделі, модель дошки оголошень.

Інформаційні джерела:

Основна та допоміжна література:

- Федорчук Є.Н. Програмування систем штучного інтелекту. Експертні системи / Є.Н.Федорчук, Вид-во Львівської політехніки, 2012. - 168 с.
- Баклан І.В. Експертні системи. Курс лекцій / Навчальний посібник. - К.: НАУ, 2012. – 132с.

Інтернет ресурси:

- Концепции общей теории информации. Статьи. Наука и техника. [Електрон. ресурс]. – Режим доступу: <http://www.n-t.org/tp/ng/oti.htm>
- Общая теор. информации. [Електрон. ресурс]. – Режим доступу: <http://www.inteltec.ru/publish/articles/textan/ibook.shtml>
- Теория информации. [Електрон. ресурс]. – Режим доступу: <http://www.inftech.webservis.ru/it/information/>

Навчальне обладнання: ТЗН, презентація тощо.

ВИКЛАД МАТЕРІАЛУ ЛЕКЦІЇ

1. Поняття подання знань

В галузі експертних систем подання знань означає не що інше, як систематизовану методику опису на машинному рівні того, що знає людина-експерт, яка спеціалізується в конкретній предметній області. Але помилково вважати, начебто подання знань зводиться до кодування в сенсі, аналогічному шифруванню. Якщо закодувати повідомлення, підставивши деяким регулярним образом замість одних символів інші, то отриманий результат не має нічого спільного з поданням змісту повідомлення в тому сенсі, як це припускається в теорії штучного інтелекту, навіть якщо отриманий код легко сприймається на машинному рівні і його можна зберігати в пам'яті комп'ютера.

Звернемо увагу хоча б на те, що в такому коді зберігається та лексична чи структурна неоднозначність, що притаманна природній людській мові. Наприклад, повідомлення "Відвідування тіточки може бути докучливим" буде настільки ж неоднозначним у закодованому виді, як і на "людській" мові. Переклад цього тексту в машинний код не позбавить нас від того, що це повідомлення можна трактувати і як твердження, що "набридає наносити візити тіточці", і як твердження, що "набридає, коли тіточка наносить візит".

Таким чином, один з парадоксів штучного інтелекту полягає в тому, що багато задач пошуку значеннєвого змісту, що легко вирішуються людиною, дуже важко реалізувати на ЕОМ і навпаки.

Будь-яке спілкування людини зі світом техніки припускає наявність деякого попереднього знання. Якщо, наприклад, хтось береться за пошук несправності в цифровій схемі, то можна припустити, що він володіє певними базовими знаннями з області електротехніки. Немає необхідності підкреслювати, що комп'ютер у чистому вигляді ніяких попередніх знань не має, а тому технічна експертність – набір якостей, що лежать в основі високого рівня роботи людей-фахівців при вирішенні проблем у певній вузькій області, – повинна включати і ці попередні знання.

І нарешті, подання припускає певну організованість знань. Подання знань повинне дозволити отримувати їх у потрібній ситуації за допомогою нескладного і більш-менш природного механізму. Простого переведення інформації (знань) у форму, придатну для збереження на машинних носіях, тут явно недостатньо. Для того щоб можна було досить швидко отримувати ті елементи знань, що найбільш придатні в конкретній ситуації, база знань повинна мати досить розвинуті засоби індексування і контекстної адресації. Тоді програма, що використовує знання, зможе керувати послідовністю застосування певних "елементів" знань навіть не маючи точної інформації про те, як вони зберігаються.

Звичайно, програмний код, виконуваний комп'ютером, повинен відповідати застосовуваній системі позначень, але це не можна вважати занадто серйозним обмеженням. Багато схем подання, на перший погляд, надзвичайно сильно відрізняються, проте виявляються насправді

формально еквівалентними, тобто все, що може бути виражене в одній системі подання, може бути виражене й в іншій

Перш ніж перейти до розгляду конкретних прикладів, давайте уточнимо термінологію, узявши за основу цитати з "класичних" робіт зі штучного інтелекту.

Подання (representation) у роботі вченого Уїнстона визначається як "безліч синтаксичних і семантичних угод, що уможливорює опис предмета". У штучному інтелекті під "*предметом*" розуміється стан деякої проблемної області, наприклад об'єкти в цій області, їх властивості, відношення, які існують між об'єктами. *Опис* (description) "дозволяє використовувати угоди з подання для опису певних предметів". Синтаксис подання специфікує набір правил, що регламентують об'єднання символів для формування виразів мовою подання. Можна говорити про те, що вираз гарно чи погано сформований, тобто про те, наскільки він відповідає цим правилам. Зміст повинні мати лише добре сформовані вирази [28].

Загальноприйнятим в галузі штучного інтелекту є синтаксис у вигляді конструкції предикат-аргумент, що має форму

$$\langle \text{вираз} \rangle ::= \langle \text{предикат} \rangle (\langle \text{аргумент} \rangle, \dots, \langle \text{аргумент} \rangle)$$

У цій конструкції за k -значним предикатом повинні впливати k аргументів. Так, предикат „*at*” може бути двомісним відношенням, у якому перший аргумент виступає як ім'я деякого об'єкта, а в якості другого – його місцезнаходження (наприклад, кімната):

$$at(\text{робот}, \text{кімната } A)$$

Семантика подання специфікує, як повинен інтерпретуватися вираз, побудований відповідно до синтаксичних правил, тобто як з його форми можна отримати певний зміст. Специфікація звичайно виконується присвоєнням змісту окремим символам, а потім індукуванням присвоєння в більш складних виразах. Так, привласнюючи зміст символам „*at*”, робот, кімната A , ми можемо сказати, що вираз

$$at(\text{робот}, \text{кімната } A)$$

означає: робот знаходиться в кімнаті A (але не навпаки – кімната A знаходиться в роботі).

Процес вирішення проблеми, як правило, містить у собі порядок із поданням предметів навколишнього світу і судження про деякі дії. Як уже було відзначено деякі проблеми формуються в термінах вихідного і цільового станів і множини операцій, які можна використовувати при спробах перетворити початковий стан у цільовий. Але тут залишається нез'ясованим питання про те, як можна подавати операції.

2. Моделі подання знань.

Логічні моделі подання знань.

Логічне програмування є порівняно новим напрямком в програмуванні та інформатиці, який оснований на ідеях та методах запозичених із математичної логіки. Саме слово логіка тлумачиться як “наука про закони мислення та його форми” або як “хід міркувань, умовиводів”. Це слово походить від грецького слова “логос”, що з одного боку означає “слово” або “мова”, а з іншого – те, що виражає мова, тобто – мислення. Таким чином логіка має безпосереднє відношення до мови і стикується з граматиною та лінгвістикою.

Логіку створив видатний вчений стародавньої Греції – Арістотель в IV столітті до нашої ери. Він зробив це так вдало, що протягом багатьох століть вона залишалась незмінною. І в наш час розвиток логіки йде шляхом поширення та уточнення її основних положень, але не шляхом їх перегляду, як це сталося з більшістю інших наук.

Лише в XVIII столітті німецький вчений Лейбніц вирішив створити нову логіку, яка стала би “мистецтвом числення”. Він запропонував поставити у відповідність кожному поняттю деякий символ, і зобразити міркування у вигляді деяких обчислень. Однак його ідеї не були сприйняті вченими того часу і не отримали розвитку.

Ідею Лейбніца частково втілював у життя лише ірландський математик Дж.Буль, який в XIX столітті створив алгебру логіки, де діяли закони подібні до законів звичайної алгебри, але символами позначалися не числа, а речення. Мовою булевої алгебри можна було описувати міркування і “обчислювати” їх результати. Але вона охоплювала лише найпростіший тип міркувань. Алгебра логіки Буля на відміну від формальної логіки Арістотеля, отримала назву математичної логіки, оскільки вона використовувала мову та методи математики і була викликана саме потребами математики.

На початку XX ст. ряд вчених показали можливість застосування у техніці логіки висловлювань, яка є одним з розділів математичної логіки. В середині XX ст. було виявлено тісний зв'язок математичної логіки з кібернетикою. З того часу розпочалось широке впровадження математичної логіки в медицині, біології, економіці, техніці і т.ін.

Наприкінці 50-х років розпочались роботи зі створення методу автоматизованого доведення теорем, в результаті яких французькому вченому Робінсону вдалося створити правило виведення, придатне для виконання машиною. Це правило Робінсон назвав резолюцією. Але пройшло ще більш ніж 10 років напруженої роботи багатьох вчених, доки французькі вчені Кольморос і Русель у 1972 р. створили в Марсельському університеті мову програмування, яка була основана на правилі резолюцій. Цю мову вони назвали ПРОЛОГ - мова ПРОграмування за допомогою ЛОГіки.

Перша версія ПРОЛОГу була створена на ФОРТРАНі, працювала дуже повільно і не знайшла широкого розповсюдження. Але ПРОЛОГ наочно продемонстрував великі потенційні можливості щодо реалізації ідей та методів штучного інтелекту, які були високо оцінені вченими різних країн. У 1981 році Міністерство зовнішньої торгівлі і розвитку Японії оголосило, що японський проект створення комп'ютерів п'ятого покоління буде засновуватись саме на методах логічного програмування. Метою проекту було створення систем обробки інформації, що базуються на знаннях, мають можливість спілкування з користувачем за допомогою природної мови або графічних засобів, і можуть надавати експертні консультації при розв'язанні задач високого рівня. На відміну від традиційних, продуктивність Пролог-програм вимірюється за допомогою нового параметра – кількості логічних виведень за секунду. Сьогодні вже створені експериментальні зразки Пролог-машин, продуктивність яких складає більш за 1 Млв./сек.

Формальні мови

Математична логіка є *формальною мовою* в тому сенсі, що у відношенні будь-якої послідовності символів вона дозволяє сказати, чи задовольняє ця послідовність правила конструювання виразів у цій мові (*формули*). Звичайно формальним мовам протиставляються природні, такі як французька і англійська, у яких граматичні правила не є жорсткими. Твердження, що логіка є *численням* з певними синтаксичними правилами логічного висновку, означає, що вплив одних членів виразу на інші залежить лише від *форми виразу* в даною мовою і ні в якому разі не залежить від будь-яких сторонніх ідей чи інтуїтивних припущень. Під *автоматичним формуванням суджень (automatedreasoning)* розуміється поведінка деякої комп'ютерної програми, що будує логічний висновок на підставі певних законів. Так, не можна віднести до класу програм автоматичного формування суджень програму, що моделює підкидання монетки, щоб визначити, чи впливає одна формула з набору інших. В літературі також часто зустрічається термін *автоматична дедукція (automateddeduction)*, рівнозначний за змістом терміну *автоматичне формування суджень*.

[Продовжити перегляд](#)

Фрейми

Термін *фрейм* (від англ. *frame* — "каркас" або "рамка") був запропонований Марвіном Мінським, одним з піонерів ШІ, в 70-і роки для позначення структури знань для сприйняття просторових сцен. Ця модель, як і семантична мережа, має глибоке психологічне обґрунтування.

Фрейм - це абстрактний образ для подання стереотипу об'єкта, поняття або ситуації.

Інтуїтивно зрозуміло, що під абстрактним образом розуміється деяка узагальнена й спрощена модель або структура. Наприклад, проголошення вголос слова "кімната" породжує в почутому образ кімнати: "житлове приміщення із чотирма стінами, підлогою, стелею, вікнами й дверима, площею 6—20 м²". Із цього опису нічого не можна забрати (наприклад, забравши вікна, ми одержимо вже прикомірок, а не кімнату), але в ньому є "дірки" або "слоти"— це незаповнені значення деяких атрибутів — наприклад, кількість вікон, кольори стін, висота стелі, покриття підлоги й ін.

У теорії фреймів такий образ кімнати називається фреймом кімнати. Фреймом також називається й формалізована модель для відображення образу.

Розрізняють фрейми-зразки або прототипи, що зберігаються в базі знань, і фрейми-екземпляри, які створюються для відображення реальних фактичних ситуацій на основі даних, що надходять. Модель фрейму є досить універсальною, оскільки дозволяє відобразити все різноманіття знань про світ через:

- *фрейму-структури*, що використовуються для позначення об'єктів і понять (позика, застава, вексель);
- *фреймів-ролі* (менеджер, касир, клієнт);
- *фрейми-сценарії* (банкрутство, збори акціонерів, святкування іменин);
- *фреймів-ситуацій* (тривога, аварія, робочий режим пристрою) й ін.

Традиційно структура фрейму може бути представлена як список властивостей: (ІМ'Я ФРЕЙМУ:

(ім'я 1-го слоту: значення 1-го слоту),

(ім'я 2-го слоту: значення 2-го слоту),

.....

(ім'я N-го слоту: значення N-го слоту)).

[Продовжити перегляд](#)

Контрольні питання:

1. Поняття та принципи подання знань.
2. Логічні моделі подання знань: загальні положення.
3. Формальні мови.
4. Числення висловлювань.
5. Числення предикатів.
6. Логічне програмування. Мова PROLOG.
7. Принципи резолюцій.
8. Продукційні моделі подання знань: загальні положення.

9. Проектування продукційних експертних систем.
10. Характеристика структурних елементів ЕС продукційного типу.
11. Семантичні мережі як модель подання знань: загальні положення.
12. Класифікація семантичних мереж.
13. Використання семантичних мереж при розробці експертних систем.
14. Фреймові моделі подання знань: загальні положення.
15. Основи теорії фреймів.
16. Основні властивості фреймів.
17. Модель дошки оголошень: загальні положення.
18. Принципи організації систем з дошкою оголошень.
19. Система HEARSAY: загальна характеристика.
20. Використання джерел знань у HEARSAY-II.
21. Оболонка для створення систем з дошкою оголошень HEARSAY-III.

Укладач: _____ Старух А.І., доцент, к.е.н.
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)