

Конспект лекції № 7

Тема № 7. СЕРЕДОВИЩЕ CLIPS. ЗАГАЛЬНА ХАРАКТЕРИСТИКА

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. Вступ до CLIPS. Історія створення CLIPS.
2. Робота з CLIPS. Синтаксис визначень.

Опорні поняття: ознайомитися з експертною оболонкою CLIPS, охарактеризувати основні переваги та недоліки, розглянути основи функціонування CLIPS.

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.
- Експертні системи в міжнародних відносинах / Ю.О.Лунь, А.М. Козел, С.М. Ніколаєв. – Львів% Видавництво Львівської політехніки, 2011. – 196с.

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. Вступ до CLIPS. Історія створення CLIPS.

Спочатку абревіатура **CLIPS** була назвою мови — **C Language Integrated Production System** (тобто мова C, інтегрована із продукційними системами), зручної для розробки баз знань і макетів експертних систем.

Тепер CLIPS являє собою сучасний інструмент, призначений для створення експертних систем (expert system tool). CLIPS складається з інтерактивного середовища — експертної оболонки зі своїм способом подання знань, гнучкої й потужної мови й декількох допоміжних інструментів. Зараз, завдяки добрій волі своїх творців, CLIPS є абсолютно вільно розповсюджуваним програмним продуктом.

Зараз на ринку доступно не так багато експертних оболонок (інструментів, призначених для створення експертних систем). Незважаючи на те, що CLIPS поширюється безкоштовно, він досить успішно конкурує навіть із найвідомішими комерційними проектами. Кількість користувачів CLIPS росте рік у рік. Про це можна судити по активності відвідування сайтів, форумів і конференцій, присвячених CLIPS.

Поява мови CLIPS можна датувати 1984 р., місце народження CLIPS — космічний центр Джонсона NASA. Саме в цей час відділ штучного інтелекту (тепер Software Technology Branch) розробив множину прототипів експертних систем, що використовують сучасне програмне й технічне забезпечення. Однак, незважаючи на великий потенціал експертних систем, не багато хто із цих додатків дійшли до кінцевого споживача. Ця невдача обумовлювалася технологією створення експертних систем, який у той час оперували в NASA. Основні обмеження накладав мову LISP, використовувана як базова мова для розробки експертних систем.

Як головні недоліки мови LISP можна виділити наступні три:

- 1) недостатня адаптованість LISP до широкого кола стандартних комп'ютерів;
- 2) висока ціна технічного й програмного забезпечення, призначеного для роботи з LISP;
- 3) низька здатність інтеграції систем, написаних на LISP, із системами, написаними на інших мовах (виробництво вкладених додатків).

Співробітники відділу штучного інтелекту зрозуміли, що використання традиційних мов програмування, таких як C, усуне більшість виниклих проблем, і відділ почав пошуки виробників і постачальників інструментів для створення експертних систем, що оперують одним із традиційних мов програмування. Незважаючи на те, що число подібних інструментів було досить велике, ціна таких інструментів виявилася досить висока. Крім того, більшість із цих інструментів працювали на дуже невеликому числі платформ, а швидкість їхньої роботи залишала бажати кращого. Стало очевидно, що для одержання інструмента, що задовольняє всім вимогам NASA, необхідна розробка власного засобу для створення експертних систем.

Прототип CLIPS був розроблений навесні 1985 р., деяким більш ніж

за два місяці. Особлива увага була приділена створенню мови, сумісного з мовами, що використовуються в NASA у той момент. Таким чином, синтаксис мови CLIPS був зроблений дуже схожим на синтаксис експертної оболонки ART, розробленою корпорацією Inference.

Незважаючи на те, що ART послужив прообразом, CLIPS розроблявся зовсім без допомоги Inference або доступу до вихідних кодів системи ART.

Завдяки тому, що CLIPS є вільно розповсюджуваним програмним продуктом з доступними вихідними кодами, останнім часом була випущена множина програм і бібліотек, що вдосконалять і доповнює можливості CLIPS. Деякі із цих продуктів є власністю їхніх компаній, що випустили, і призначені для внутрішнього використання або комерційного поширення, інші, як і сам CLIPS, поширюються вільно.

Як самі відомі приклади подібних проектів можна привести DLL/OCX-бібліотеку, що дозволяє використати механізм логічного висновку CLIPS у ваших додатках, FuzzyCLIPS, CLIPS++, CLIPS code generator.

[Продовжити перегляд](#)

2. Робота з CLIPS. Синтаксис визначень.

Для демонстрації прикладів, використовуваних у цій книзі, буде застосовуватися Windows-версія CLIPS 6.2. Незважаючи на повну сумісність із Apple Macintosh й UNIX-версіями, при роботі з даною книгою бажано використати саме Windows- версію середовища CLIPS. Зовнішній вигляд головного вікна CLIPS показаний на рис. 1.

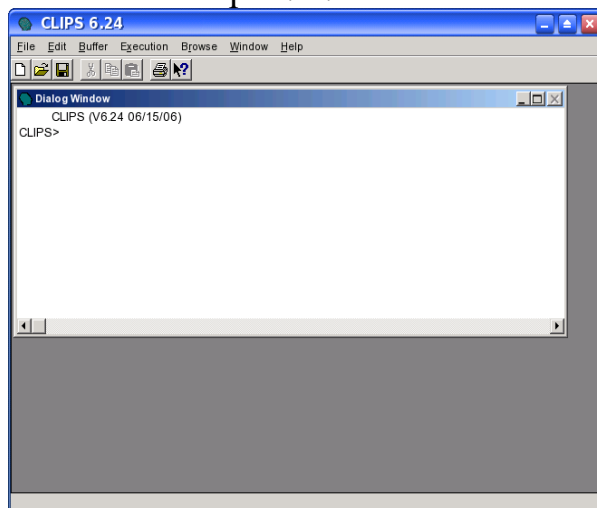


Рис. 1. Головне вікно CLIPS

Windows-версія середовища CLIPS повністю сумісна з базовою специфікацією мови. Уведення команд здійснюється безпосередньо в головне вікно CLIPS. Однак у порівнянні з базовою Windows-версією надає множину додаткових візуальних інструментів (наприклад, менеджери фактів або правил), значно полегшуюче життя розробника експертних систем.

Експертні системи, створені за допомогою CLIPS, можуть бути запуснені трьома основними способами:

- введенням відповідних команд і конструкторів мови безпосередньо в середовище CLIPS;
- використанням інтерактивного віконного інтерфейсу CLIPS (наприклад для версій Windows або Macintosh);
- за допомогою програм-оболонок, що реалізують свій інтерфейс спілкування з користувачем і механізмів, що використовують, знань і логічного висновку CLIPS.

Крім того, CLIPS при запуску дозволяє виконувати командні файли власного формату (ця можливість також доступна за допомогою команди batch). Для реалізації цієї можливості необхідно запустити CLIPS (у нашому випадку це файл CLIPSWin.exe) з одним із трьох наступних аргументів:

- -f<імені-файлу>
- -f2<імена-файли>
- -l<імені-файлу>

Текстовий файл, заданий за допомогою опції -f, повинен містити команди CLIPS.

Якщо заданий файл містить команду exit, то CLIPS завершить свою роботу й користувач повернеться в операційну систему. У випадку якщо команда exit відсутня, то після виконання всіх команд із заданого файлу користувач потрапить у головне вікно CLIPS.

Команди в текстовому файлі повинні бути набрані так само, як якби вони вводилися безпосередньо в командний рядок CLIPS (тобто всі команди повинні бути укладені в круглі дужки й розділені символом переходу на новий рядок).

Опція -f фактично еквівалентна запуску команди batch відразу після запуску CLIPS.

Опція -f2 ідентична -f, але, на відміну від опції -f, вона використає команду batch*. Файл, заданий цією опцією, також виконується після запуску CLIPS, але результати виконання команд не відображаються на екрані.

Опція -l задає текстовий файл, що містить конструктори CLIPS, які відразу запускаються на виконання. Використання цієї опції еквівалентно використанню команди load відразу після запуску CLIPS.

Основним методом спілкування з CLIPS, використовуваним у даному навчальному курсі, є застосування командного рядка. Після появи в головному вікні CLIPS запрошення - CLIPS > - команди користувача можуть вводитися в середовище безпосередньо із клавіатури. Команди можуть бути викликами системних або користувальницьких функцій, конструкторами різних даних CLIPS і т.д. У випадку виклику користувачем деякої функції, вона негайно виконується, і результат її роботи відображається користувачеві.

Для виклику функцій або операцій CLIPS використовує префіксну нотацію - аргументи завжди впливають після імені функції або операції. При виклику конструкторів CLIPS створює новий об'єкт відповідного типу, так чи інакше представляє деякого знання в системі. При введенні в середовище імені створеної раніше глобальної змінної CLIPS відобразить її поточне значення. Введення в середовище деякої константи просто приведе до її негайного відображення в головному вікні CLIPS.

Як приклад уведіть у середовище наступні команди (лістинг 1).

Лістинг 1. Приклад команд CLIPS

```
(+ 3 4)
(defglobal ?*x* = 3)
?*x* red
```

Результат виконання цих команд **наведений** на рис. 2.

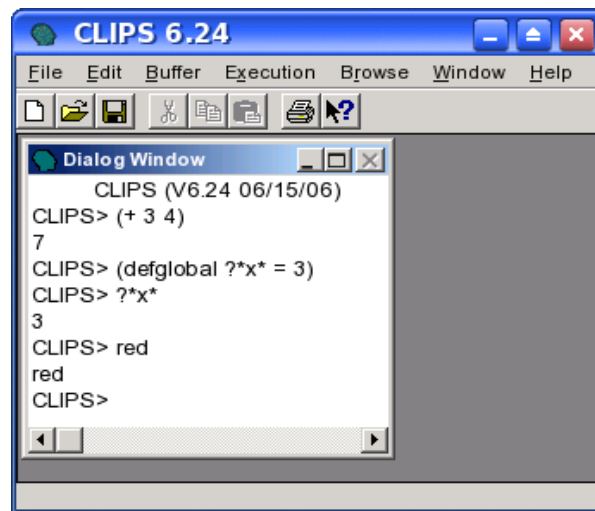


Рис.2. Результат виконання команд лістингу 1

Першою командою даного приклада викликається функція арифметичного додавання двох чисел: 3 й 7. Результат роботи цієї функції - 7 відразу ж відображається на екран.

Після цього, за допомогою конструктора `defglobal`, створюється глобальна змінна `?*x*`, що ініціюється значенням 3. При введенні в командний рядок імені глобальної змінної `?*x*` CLIPS відображає її поточне значення, рівне 3. Останньою командою була уведена якась константа `red`, значення якої було відразу виведене на екран.

Синтаксис визначень

Як базовий синтаксис для визначення конструкцій мови використовується стандартна **БНФ- нотация**. Нижче наведені правила, використовувані для побудови визначень.

Слово або вираження, укладене в кутові дужки, називається нетермінальним символом (наприклад, `<string>`).

Нетермінальний символ вимагає подальшого визначення. Слова або вираження, не ув'язнені в кутові дужки, називаються термінальними символами, і представляють синтаксис описуваної конструкції мови CLIPS.

Термінальні символи (особливо круглі дужки) повинні вводитися в командний рядок саме так, як показано у визначенні. Якщо за нетермінальним символом треба символ `*`, то це означає, що в даному місці може перебувати

список з нуля або більше елементів цього типу. Якщо ж за нетермінальним символом треба +, то в даному місці може перебувати список з одного або більше елементів цього типу. Символи * й +, що зустрічаються самі по собі (не наступні після нетермінальних символів), є термінальними.

Багатокрапки, як горизонтальні, так і вертикальні, також використовується для відображення списку з одного або більше елементів. Елементи, укладені у квадратні дужки (наприклад, [<коментарі>]), є необов'язковими елементами, які можуть входити у визначення. Вертикальна риса, що розділяє два або більше елементи визначення, указує на те, що в конструкції необхідно використати один з перерахованих елементів. Символ ::= використовується для позначення необхідності заміни деякого нетермінального символу. Наприклад, визначення:

<lexeme> ::= <symbol> |

<string> позначає, що нетермінальний символ

<lexeme>, що зустрічається в деякому визначенні, повинен бути замінений або на символ *<symbol>*, або на символ *<string>*.

Пробіли, символи табуляції, переходи на інший рядок використовуються тільки для логічного поділу елементів визначення й ігноруються CLIPS (крім рядків, укладених у подвійні лапки).

Контрольні питання:

1. Що таке CLIPS?
2. Основні переваги CLIPS.
3. Основні недоліки CLIPS.
4. Якими способами можуть бути запущені експертні системи, створені за допомогою CLIPS?
5. Для чого використовуються багатокрапки в CLIPS?
6. Для чого використовуються пробіли та символи табуляції в CLIPS?

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