Н.М.Дацун
(Донецький державний технічний університет)

ФУНКЦІОНАЛЬНЕ І ЛОГІЧНЕ ПРОГРАМУВАННЯ: СУЧАСНІ ПІДХОДИ ДО КОМП'ЮТЕРНОГО НАВЧАННЯ

Зміна парадигм у інформатиці — це неминучий результат її розвитку. Зараз все більше поширюються парадигми функціонального та логічного програмування. Названі парадигми більше відповідають способові мислення людини порівняно з імперативними мовами, тому мови функціонального та логічного програмування відносять до мов штучного інтелекту. У статті розглядаються сучасні підходи до навчання функціонального та логічного програмування.

Поява японського проекту ЕОМ п'ятого покоління, а також хвилі "відповідних" національних і міжнаціональних програм (МСС, ESPRIT) висунули мови Лісп і, особливо, Пролог до числа найперспективніших мов. Свою роль відіграє поширення мов функціонального й логічного програмування на персональних комп'ютерах і робочих станціях: це системи muLISP, Golden Common Lisp 4.0, Allegro Common Lisp, Common Lisp 4.0, Turbo-Prolog, Prolog-2, Prolog III, Quintus Prolog 3.0, МПролог.

Крім того, за допомогою Ліспу і Прологу створені різноманітні комерційні інтелектуальні продукти: системи евристичного прогнозування, діагностуючі й консультуючі експертні системи, оболонки експертних систем для роботи з невизначеністю у знаннях, системи прийняття рішень, системи інженерії знань.

Мовам функціонального й логічного типу приділяється увага у національних загальноосвітніх програмах, міжнаціональних проектах (ESPRIT, Delta, Race, Comett, Euronet) [1], а також у навчальних планах вищих навчальних закладів.

1. Навчання функціонального програмування

Парадигма функціонального програмування оформилася першою серед перелічених. Тому комп'ютерні навчальні системи (КНС), або автоматизовані навчаючі системи за допомогою ЕОМ (АНС) для мов функціонального типу, з'явились раніше. Абсолютна більшість їх призначена для навчання мови Лісп. Розглянемо чотири групи програмних засобів, які являють собою КНС безпосередньо або виконують частину їх функцій. Це так звані системи-РЕПЕТИТОРи, експерт-

но-навчаючі системи, "асистенти програміста" і системи візуального програмування. Три перші групи підтримують технологію штучного інтелекту, а візуальні системи використовують можливості нових ін формаційних технологій у вигляді мультісередовищ. Далі будемо вживати синоніми "студент" і "той, хто навчається".

Системи-РЕПЕТИТОРи. У [2] АНС програмування класифіко вані за способом подання предмета навчання на три групи: ЛЕКТОРи, АСИСТЕНТИ і РЕПЕТИТОРИ. Характерні риси РЕПЕТИТОРІВ такі: матеріал про предметну галузь подається студентові довільно; головну увагу така АНС приділяє практичній роботі студента під постійним контролем системи; навчання ведеться з обліком індивідуальних особливостей студента.

Серед КНС функціонального програмування найвідоміші роз робки університету Карнегі-Мелона (США): LISP TUTOR [2], PTA [2] і GREATERP І3]. Усі ці системи являють собою інтелектуальні РЕПЕТИТОРИ.

Модель студента цих систем являє собою правила планування і складання програм на Ліспі, а також помилкові варіанти таких правил. Названі КНС використовують відповідну інструментальну систему продукції (у LISP TUTOR це система GREPES, у PTA — PUPS).

Інтерфейс цих систем використує вікна і меню. Текстові редак тори систем-РЕПЕТИТОРів обов'язково структурні. Так, GREATERP і LISP TUTOR мають власні структурні редактори, а PTA використується разом з набором структурних редакторів GNOME. LISP TUTOR має також командну мову, внкористуючи для цього шаблони команд.

Система допомоги систем-РЕПЕТИТОРів призначена для активі зації роботи студента у важкій ситуації. Так, GREATERP може сама побудувати невеликий фрагмент програми, який надає студентові можливість продовжити програмування. Така допомога надається або за проханням студента, або коли він зробив таку кількість помилок, що е максимально припустимою для даного відрізку програми. Крім того, GREATERP забезпечує "миттєвий" зворотний зв'язок, тобто реагує на помилковий елемент програми відразу, а не після завер шення завдання всієї функції або їх сукупності.

Підказки в РЕПЕТИТОРАХ мають вигляд запитань і нагадувань про ту мету, що переслідується на даному етапі: так, PTA мае ге нератор підказок, а LISP TUTOR генерує їх природною мовою.

Реалізації КНС, що були розглянуті, виконані у середовищах систем програмування Ліспу: GREATERP для VAX725 і IBM PC, LISP TUTOR для VAX725, PTA для MicroVAX.

Експертно-навчаючі системи (ЕНС). Система FLEX [4] являє собою ЕНС супроводжувального типу. Вона забезпечує користувачеві Ліспу такі послуги, що відповідають його рівневі знань. Модель студента дозволяє системі коригувати свою реакцію на поведінку останнього у критичних ситуаціях. Тому ця КНС належить до систем адаптивного навчання.

"Асистенти програміста". У [5] серед інструментальних систем автоматизації програмування виділено групу програмних засобів типу "асистент програміста" (АП). Серед ідей і концепцій АП назвемо ті, що відповідають вимогам КНС: АП містить у собі семантичну модель програми, що розроблюється, та базу знань про способи роз робки програм; АП виявляє й виправляє більшість помилок синтаксичного характеру і немотивовані семантичні помилки; АП використовує структурні текстові редактори; засоби налагодження таких систем здатні встановлювати джерело виникнення помилки; АП при виявленні помилок на етапі виконання програми визиває користувача у ролі підпрограми для реакції і продовження виконання програми; АП високою мірою інтерактивний.

До КНС функціонального програмування типу АП належать системи SCENT [2, 6], DWIM [5], KBEmacs [7], які реалізовані у середовищах Ліспу.

Система SCENT (університет Саскачевана, Канада) являє собою інтелектуальний "асистент програміста", який використовує ментальні моделі рекурсії. Інтерфейс студента має систему вікон, меню, а також засоби інтерактивного генерування повідомлень.

Роль КНС відіграють також інші системи типу АП, які є частиною інтегрованих середовищ Ліспу. Так, система DWIM — це інтерактивна програма INTERLISPy. DWIM вміє виправляти триві альні помилки, які допустили при роботі з редакторами і файловою системою INTERLISPy, а також може пояснити користувачеві деякі помилки та ситуації. DWIM працює **з** інтерфейсом користувача подіб но до мови Smalltalk.

KBEmacs являє собою прототип системи типу АП. Знання про мову Лісп в цій інтелектуальній системі подані у вигляді кліше програм на алгоритмічній мові високого рівня. Інтерфейс **з** користу вачем виконано на мові ключових слів.

Системи візуального програмування. Візуальні методи програмування пропонують більш природний шлях для вираження алгоритмів і структур даних, ніж це роблять лінійні мови. Серед систем, що забезпечують візуальну взаємодію, мову Лісп підтримують Iconlisp, Tinkertoy і VICON [8].

Iconlisp є візуальним розширенням мови Лісп. Ікона тут являє собою трійку, < *Pic Log, Name* >, *де Pic — це графічний образ, Log* Лісп-список, Name —ідентифікатор. Iconlisp розрізнює три класи ікон: об'єктів, переходів (Log-компонента містить програми або їх частини) і функцій (Log-компонента може являти собою або базові примітиви, або функції, що визначені користувачем). Iconlisp дозволяє користувачеві не турбуватися про розстановку дужок, а також програмувати як у природному логічному порядку, так і у порядку, зворотному логічній послідовності.

Tinkertoy — це графічний інтерфейс для мови Лісп. Ікони Тіткertoy являють собою фрагменти програм, які припустимо об'єднувати у структури. Після введення з клавіатури Лісп-виразу система формує для нього іконічне уявлення, з яким потім можна маніпулювати. На відміну від Iconlisp, система Tinkertoy не проводить безпосереднє обчислювання іконічної структури, що сформована, а попередньо перетворює її у Лісп-код. Однак іконічний образ складних структур даних Ліспу, побудований за допомогою Tinkertoy, більш відповідає машинному поданню програми, ніж лінійний текст. Останнє особливо важливе при навчанні Ліспу.

VICON являє собою інтерактивне середовище програмування для маніпуляції з іконічними структурами. До звичайного визначення ікони тут додається поняття відношення ікон: це е покажчик від однієї ікони до іншої або до впорядкованої множини ікон (підтримка об'єктно-орієнтованого підходу). Операції над іконами візуалізуються на екрані дісплею та інтерпретуються у Лісп-код.

Відзначимо, що така реалізація Ліспу, як InterLISP-D, є вдалим середовищем для реалізації КНС візуального програмування завдяки розвиненим засобам машинної графіки. Так, ікони можливо реалізувати на верхньому рівні середовища в InterLISP-D, базу даних InterLISP-D використати як базу даних візуальної системи, а спільний редактор ікон може вміщувати редактор графічних образів, редактор логічної структури системи ікон, структурний Лісп-редактор, текстовий редактор InterLISP-D.

2. **Навчання логічного програмування**

Навчання логічного програмування традиційно виконується за три етапи. Перший крок — це навчання основ математичної логіки, далі слідує навчання програмування **у** декларативному стилі і нарешті вивчається мова типу Пролог. Для кожного з перелічених етапів зроблені інтелектуальні КНС. Крім того, **у** ролі КНС використую ся графічні системи налагодження.

Навчання основ математичної логіки. Логічне програмування "у вузькому розумінні слова" (хорнівське, прологівське) як мате матичну основу використовує логіку числення предикатів першого прядку. Навчанню логіки першого порядку призначені інтелекту- альні комп'ютерні системи [9] і "ІПЛОГ" [10]. Вони призначені для перевірки перетворювання формул логіки і вивчення застосування теорем логіки.

Названі системи мають продукційні бази знань. Система [10] використовує модель студента, що базується на адаптивному навчанні. Система [9] вміщує чотири моделі: автоматичну (що реалізує авто- матично перевірку розв'язання всієї задачі); автоматичну перевірку одного кроку; автоматичну перевірку кожного кроку через застосу вання правил, що використовує студент; інтерактивну. Ці системи працюють у режимі діалогу природною мовою.

Навчання програмування у декларативному стилі Навчанню формалізації предметних галузей у вигляді ствердженъ природною мовою призначені системи, що розроблені в Інституті кібернетики їм. Глушкова АН України: САКІО [11] і ПАЛЕВАС [12].

САКІО являє собою експериментальну версію ЕНС. У цій системі інтегровані різні інформаційні технології: продукції для уяв лення основної бази знань, таблиці рішень для відображення методики навчання, реляційна база даних для інформаційно-довідкової служби, електронні відомості для подання моделі студента.

Система ПАЛЕВАС являє собою подальший розвиток робіт у технології адаптивних навчаючих систем. Модель студента фіксує 6 типів помилок при виконанні останнього завдання і 5 типів помилок на рівні виконання студентом всього уроку. Інтерфейс з тим, хто навчається, використовує систему вікон, підказок, екранний редактор, також виконує графічну інтерпретацію задачі.

Системи САКІЮ та ПАЛЕВАС реалізовані у MS DOS для IBM PC з використанням середовищ Прологу.

Навчання програмування на мові Пролог. Навчання граматики мови Пролог на відміну від навчання граматиці будь-якої імперативної мови вважається порівняно нескладним [13]. Найпростішою навчальною системою може бути розширення Прологу, як в середовищі APPE [14]. Тут звичайний транслятор Прологу для діагностики семантичних помилок розширено інтелектуальним діагностичним механізмом. Пролог-програми в APPE класифікуються за допомогою цього механізму на 5 категорій відповідно до помилок студента.

Однак для досягнення певного рівня майстерності у програмуванні на мові Пролог знань лише синтаксису недостатньо: основні труднощі викликає побудова логічних програм. Інтелектуальна навчальна система Prolog Tutor [13] виявляє логічні помилки в програмах студентів і дає рекомендації щодо їх усунення.

Системи візуального програмування. При навчанні мови Пролог можливо використання систем графічного налагодження і трасування. До систем графічного налагодження Прологу належать системи TPM-Prolog [15] і ПролоГраф [15]. У TPM-Prolog (ф. Expert Systtem Int. Ltd) програми постають у вигляді дерева І/АБО, вершини якого навантажені. На цьому дереві фіксується історія виконання Пролог-програми. В інструментальній системі ПролоГраф (ІК ім. Глушкова) Пролог-програма подається не у лінійній формі, а у вигляді Р-графу. Інформація про налагодження також візуалізується на Р-графі, який постає як повне дерево рішення. На цьому графі різними кольорами подається різний стан цілей, а також стан Пролог-програми (відсічення, рекурсія) і Пролог-системи. Реалізовано інтеграцію ПролоГрафу з системами CS-Prolog та TP-Пролог у вигляді додаткової компоненти до цих середовищ.

Мова Пролог має властивість паралелізму, що дозволяє використовувати моделі паралельної обробки логічних програм. Тому при навчанні програмування на мовах паралельного Прологу важливе місце належить тлумаченню тих процесів, що відбуваються. У роботі [16] описується графічний засіб для трасування паралельних Пролог-програм, що дозволяє продемонструвати студентові роботу програми крок за кроком.

Таким чином, КНС функціонального і логічного програмування являють собою інтелектуальні програмні засоби. Важливою рисою цих систем є те, що їх реалізації можуть бути виконані у природному

середовищі цільової мови. Крім того, сучасні реалізації Ліспу та Прологу дозволяють інтегрувати у технології штучного інтелекту (для такої предметної галузі як програмування) системи, що засновуються на знаннях, з когнітивною графікою.

Список літератури

1. EURIT90: A European Conference on Technology and Education // Cornput. and Educ. — 1991. — 16. —N 1. P.l — 132.
2. Мельников И.А., Монкус В.В., Тамм Б.Г. Обзор и анализ зарубежных компьютерных обучающих систем в области прог раммирования // Прикладная информатика. —М., 1984 —Вып 15. — С. 131—153.
3. Андерсон Дж.Р., Рейзер Б.Дж. Учитель ЛИСПа // Реаль ность и прогнозы искусственного интеллекта. —М., 1987. — С.24 — 27.
4. Стефанюк В.Л., Алексеева Е.Ф. К созданию интеллектуальной операционной системы // Алгоритмы обработки экспериментальных данных. —М., 1978. —С. 115—130.
5. Чаплинскас А.А. Принципы конструирования проблемно-ориентированных систем. — Вильнюс, 1988. — 176 с.
6. Bhuiyan S.H., Greer J.E., McCalla G.I. Mental models of recursion and their use in the SCENTprogramming advisor // Lect. Notes Artif. Intell. — 1990. — 444. — P. 135—144.
7. Тепанди Я.Я. Прикладные системы искусственного интеллекта и автоматизация построения программного обеспечения // Прог раммирование. — 1989. — N 1. — С.61—69.
8. Хлебцевич Г.Е., Цыганкова С.В. Визуальный стиль прог раммирования: понятия и возможности // Программирование . — 1990. — N 4. — С.68 — 79.
9. Dafa Li. Intelligent CAI course in the first-order logic // Lect. Notes Computer. Sci. — 1990. — 438. — P.67 — 72.
10. Анацкий Н.И., Левин Н.А., Поспелова Л.Я. Экспертная обучающая система "ИПИЛОГ" // Искусственный интеллект-90: Докл. Всесоюз. конф. — Минск, 1990. — Т.1. — С.190—192.
11. Золотопуп О.Н., Колос В.В. Обучение построению логических программ в системе САКИО // Компьютерные техно логии обучения. — К., 1989. — С.5—8.

12. Колос ВВ., Кудрявцева СП., Сахно А .А. ПАЛЕВАС — система обучения основам логического программирования (фор мализация предметных областей) // Интеллектуализация компь ютерных технологий обучения. — К., 1993. — С.50—53.

13. Lee M C. Designing an intelligent Prolog tutor // Lect. Notes Comput. Sci. — 1990. — 438. — P.420—431.

14. Lee N. An augmented Prolog programming environment for tutoring application // IEA/AIE'90. — New York, 1990. — P.898—906.

15. Галаган Н.И., Румянцев Ю.И., Адеев В.В. и др. Инстру ментальная система логического программирования на основе графических способов визуализации процессов программирования, анализа и отладки // УСимМ. — 1991. — N 5/6. — С.42—50.

16. Geshke U., Schmidt J. A graphical tool for tracing concurrent Prolog program // Res. Int. — 1991. — 1. — С.29—53.