

Az MProlog avagy a Magyar Prolog story

Szeredi Péter

`szeredi@cs.bme.hu`

BME Számítástudományi és Információelméleti Tanszék

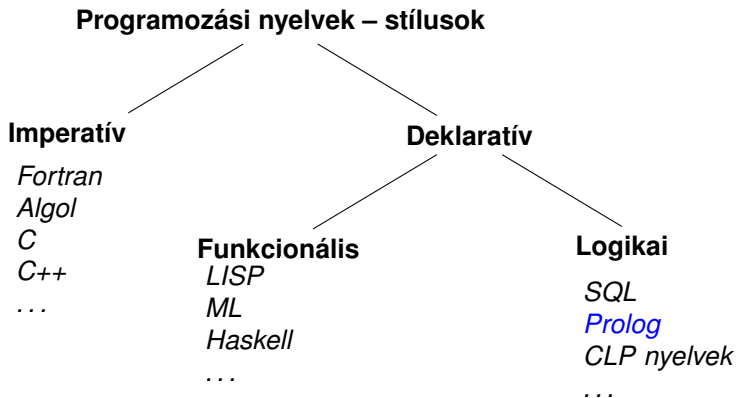
2019. május 16.

I. rész

Röviden a Prolog nyelvről

- 1 Röviden a Prolog nyelvről
- 2 A Prolog kezdetei
- 3 Az MPROLOG rendszer

A Prolog helye a programozási nyelvek világában



Deklaratív programozási nyelvek

- A funkcionális nyelvek alapja a matematika függvényfogalma
- A logikai nyelvek alapja a matematika relációfogalma
- Közös tulajdonságaik
 - A program jelentése egy matematikai állításként olvasható ki (deklaratív szemantika)
 - Matematikai változó *egyetlen* egyelőre ismeretlen értéket jelöl, vö. egyszeres értékadás
- Jelmondat
 - MIT **inkább mint HOGYAN** (WHAT rather than HOW):
a *megoldás módja* (a HOGYAN rész) helyett **inkább** a megoldandó *feladat leírását* (a MIT részt) kell megadni
 - A gyakorlatban mindkét szemponttal foglalkozni kell – kettős szemantika:
 - MIT (milyen feladatot) old meg a program (deklaratív rész)
 - HOGYAN oldja meg a program a feladatot (procedurális rész)

Példa: adott értékű kifejezés előállítása

- Egy érdekes feladat (állítólag a Microsoft interjúban használta):
 - Az 1, 3, 4, 6 számokból a négy alapművelet felhasználásával állítsuk elő a 24 számértéket!
 - Mind a négy számot pontosan egyszer kell felhasználni, tetszőleges sorrendben.
 - Tetszőleges alapműveletek használhatók, tetszőleges zárójelezéssel.

Példa: -23 előállítása az 1, 3, 4, 6 számokból

Egy „brute force” módszer:

- 1 Készítsük el az adott számlista összes permutációját:
 $(1,3,4,6), (1,3,6,4), \dots, (4,1,6,3), \dots$ (24 lehetőség)
- 2 Készítsük el az összes olyan zárójelezési sémát (bináris fát), amellyel az A, B, C, D számokra az xx, yy, zz kétargumentumú műveleteket alkalmazzuk (a sorrendjük megőrzésével):
 $((A \text{ xx } B) \text{ yy } C) \text{ zz } D), (A \text{ xx } (B \text{ yy } C)) \text{ zz } D), \dots$ (5 lehetőség)
- 3 Minden 1-beli permutációt helyettesítsünk be minden 2-beli sémába az A, B, C, D változók helyére:
 $\dots ((4 \text{ xx } (1 \text{ yy } 6)) \text{ zz } 3)) \dots$ ($24 \cdot 5 = 120$ lehetőség)
- 4 A 3-ban kapott minden sémában az xx, yy és zz változókat helyettesítsük be minden lehetséges módon a megengedett műveletekkel (ezeknek a helyettesítéseknek a száma $4 \cdot 4 \cdot 4 = 64$):
 $\dots ((4 * (1 - 6)) - 3)) \dots$ ($64 \cdot 120 = 7680$ lehetőség)
- 5 Értékeljük ki az összes kapott kifejezést, és csak azokat tartjuk meg, amelyek értéke megegyezik az elvárt számértékkel

Prolog program adott értékű kifejezés előállítására

```

| ?- kif([1,3,4,6], [+,-,*,/], -23, Kif).
Kif = (1-6)*4-3 ? ;
Kif = 4*(1-6)-3 ? ;
no

kif(SzamL, MuvL, E, Kif) :- % Az adott SzamL számlistából
                           % az adott MuvL listabeli műveletekkel
                           % felépíthető a Kif kifejezés
                           % amelynek értéke az adott Ertek, ha
    permutation(SzamL, PL), % SzamL egy permutációja PL és
    kif1(PL, MuvL, Kif),    % a PL levéllistából a MuvL műveletekkel
                           % felépíthető a Kif kifejezés és
    catch(Kif == E, _, fail).% Kif számértéke Ertek.

% kif1(L, MuvL, Kif):
kif1([Kif], _, Kif).      % Az egyelemű [Kif] listából Kif építhető fel.
kif1(L, MuvL, Kif) :-    % L-ből a MuvL műveletekkel felépíthető Kif ha
    append(L1, L2, L),    % L előáll mint L1 és L2 összefűzése ahol
    L1 \= [], L2 \= [],   % L1 nem üres lista és L2 nem üres lista, és
    kif1(L1, MuvL, K1),   % L1-ből a MuvL műveletekkel felépíthető K1, és
    kif1(L2, MuvL, K2),   % L2-ből a MuvL műveletekkel felépíthető K2, és
    member(M, MuvL),      % M egy MuvL listabeli művelet, és
    Kif =.. [M,K1,K2].    % Kif egy az M művelettel a K1-ből és K2-ből
                           % felépülő kétargumentumú kifejezés.

```

II. rész

A Prolog kezdetei

- 1 Röviden a Prolog nyelvről
- 2 A Prolog kezdetei
- 3 Az MPROLOG rendszer

A Prolog rövid őstörténete

- A Prolog „ősatyái”: Robert Kowalski (Edinburgh, London) – elmélet; Alain Colmerauer (Marseille) – gyakorlat; Alan Robinson ún. rezolúciós tételbizonyítási módszerére (1965) építve
- 1973: az első publikus Prolog változat, Fortran nyelven
- Marseille-en kívül eleinte csak Edinburgh-ban használták, onnan Gergely Tamás–Németi István–Andréka Hajnal közvetítésével jutott el a rendszer Magyarországra a NIM IGÜSZI Szoftverfejlesztési osztályára (Náray Miklós) – nagyméretű kártyacsomag + csekély dokumentáció (1974?)
- Németi vezetésével a NIM IGÜSZI-ben már több éve foglalkoztunk automatikus tételbizonyítással, programhelyesség-bizonyítással stb., CDL (Compiler Description Language – C.H.A. Koster) környezetben
- A Fortran program hordozása nehezen haladt — kompatibilitási gondok
- 1975. május 19-én kezdődött a NIM IGÜSZI szokásos évi egyhetes „elvonulása” az kerektoi NIM üdülőben, a fő téma: a Prolog nyelv
- Az elvonulásra sikerült elkészítenem egy Prolog interpretert CDL nyelven, David H. D. Warrennek a Marseille Prologról szóló rövid (3 A4-es oldal) prezentációja alapján

A Prolog Magyarországon

- Az elvonuláson a fejlesztők és potenciális alkalmazók is részt vettek
- A működő, házi fejlesztésű Prolog megléte sok kollégát meghihetett:
 - Szóts Miklós: egyszintes panelcsarnokok tervezése
 - Darvas Ferenc: gyógyszeripari alkalmazások
 - Futó Iván: szimulációs alkalmazások, T-Prolog
 - ...
- Az alkalmazások igényei ösztönözték a Prolog továbbfejlesztését: nyomkövetés, visszaléptethető beépített eljárások stb.
- 1-2 éven belül Magyarország a világ legnagyobb Prolog alkalmazója lett (Kowalski 1976-ban, Warren 1977-ben látogatott meg minket, 1978-ban a Salgótarjáni „Logic in Computer Science” konferencián több Prologos vett rész, 1980-ban Debrecenben volt az első nagyobb Prolog konferencia: Prolog Workshop – kb. 60 magyar és 60 külföldi résztvevővel)
- 1981 végén a japánok bejelentették az FGCS (Fifth Generation Computer Systems) projektet, középpontjába a Prolog nyelvet helyezték

III. rész

Az MPROLOG rendszer

- 1 Röviden a Prolog nyelvről
- 2 A Prolog kezdetei
- 3 Az MPROLOG rendszer**

MPROLOG – Magyar Moduláris Prolog

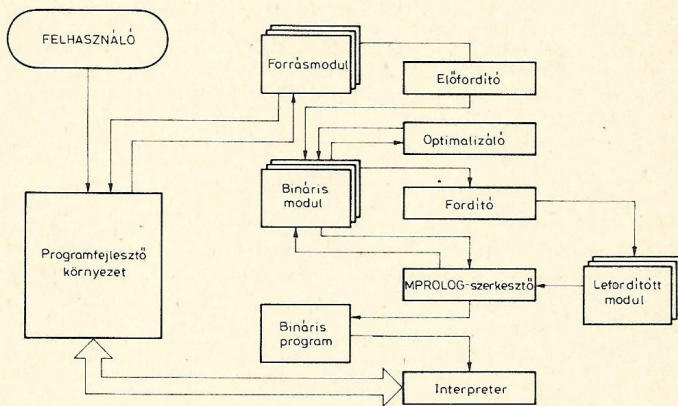
- A 1977 körül kezdődött meg egy új moduláris Prolog rendszer kidolgozása KSH OSZI támogatással a NIM IGÜSZI-ben
- 1980 nyarán demonstráltuk az MPROLOG rendszer SIEMENS BS2000-es változatát Debrecenben a Logic Programming Workshop-on
- Az 1980-as évek elejére a fejlesztés zöme átkerült az SZKI-ba, a Dömölki Bálint vezette Elméleti Laboratóriumba (ELL) – Farkas Zsuzsa, Köves Péter, Sántáné-Tóth Edit főmunkatársakkal
- 1983-ban a SZÁMKI Rendszertechnológiai osztályának Langer Tamás vezette 7-fős csapata csatlakozott az MPROLOG csoporthoz, Langer lett az MPROLOG projekt vezetője

Az MPROLOG megvalósítása

- Az MPROLOG rendszert CDL2-ben ill. a CDL labor segítségével valósítottuk meg, alapvetően interpretált végrehajtásra terveztük.
- Az MPROLOG fő komponensei:
 - előfordító (pretranslator) – egy modult bináris alakra hoz
 - szerkesztő (consolidator) – bináris modulokat programmá rak össze
 - interpreter – bináris programot hajt végre
 - PDSS fejlesztő környezet – a programfejlesztés támogatása
- Később kifejlesztett kiegészítő komponensek:
 - fordítóprogram – gépi kódra fordít
 - optimalizáló – a programot forrásszintű transzformációkkal gyorsítja

Az MPROLOG felépítése

Az MPROLOG-rendszer felépítése



Az MPROLOG mint szoftvertermék

- 1981-ben, a japán 5. generációs projekt bejelentésének idején talán az MPROLOG volt a legfejlettebb ipari (nem-egyetemi) Prolog rendszer
- Előnyei:
 - széles alkalmazási háttér (az első magyar Prolog alkalmazási tapasztalatainak beépítése révén)
 - modern fejlesztési környezet (PDSS)
 - hordozhatóság (CDL)
- Hátrányok:
 - a számítástechnikai embargó miatt nehézkes hordozás
 - a korai Prolog végrehajtási modell alkalmazása (1983-ban publikálta Warren a WAM modellt, amely a mai Prolog megvalósítások alapja)
 - a fordítási modell nem eléggé hatékony volta

Az MPROLOG mint exporttermék

- Az első sikeres eladás 1982 nyarán: Schlumberger, Párizs (DEC VAX)
- Fokozatosan kiépülő disztribútori hálózat (Sipka Júlia, Sci-L)
 - Epsilon, Nyugat Berlin: a CDL fejlesztő cége, önnálló hordozásokat is csináltak
 - Rikei, Japán: széles (16-bites) karakterek támogatása
 - Teclogic, Olaszország
 - Proper s.a., Franciaország
 - Softline Technology, Finnország
- Logicware – az MPROLOG Észak Amerikában
 - 1983-ban alapította Budai Róbert magyar származású üzletember az MPROLOG értékesítésére
 - kb. 20 helyi munkatárs + kiutazó SZKI-s szakemberek
 - az MPROLOG teljes terminológiai átdolgozása – a de facto Prolog szabvány teljes elvetése :-)
 - Google keresés a „Logicware MPROLOG” szövegre – sok találat!
 - komoly árbevétel az SZKI (és munkatársai) számára
 - a 80-as évek végére megszűnik a Logicware cég

Logicware MPROLOG prospektusok

The programmer-friendly artificial intelligence language



When you choose PROLOG, Logicware gives you complete support

When you deal with us, you can be assured of complete support services. These include an ongoing support and maintenance of your product line, the comprehensive documentation and software.

Client Support
Our software engineers will provide you with technical consulting or consultation, a debugger, release production, software problem analysis and other support services. We also provide a "hot line" telephone support of our maintenance and software support from "9-5" to "8" support outside the normal computer hours.

Consulting
At Logicware we provide you with both knowledge engineering and software engineering support to assist you in developing your PROLOG applications.
Working in conjunction with your personnel, our knowledge engineers will help analyze and program the facts and rules applicable to the specific under development. This helps ease the system for maximum performance.

Education
Regularly scheduled education is a feature of Logicware support because we realize the need for education, based on time, quality, cost and materials. This is why we have placed great emphasis on the availability and quality of our education. How does our support fit client needs, with "hands-on" examples and sample programs. This can be achieved at our continuing education program.

Documentation
Our documentation and education material are designed to provide a clear, concise explanation to assist you and to act as a broad an educational resource.

To find out more about MPROLOG and Logicware call or write

LOGICWARE INC.

1750 31st Street
Waltham, MA 02154
Tel: 617-253-2534

5000 Birch Street
The Wood Group
Toronto, Ontario
Canada M3J 2Y5
Tel: 416-242-5631 416-242-5632

11015 Avenue Ferry Road
Suite 205
Northwood Farm (outlying)
Atlanta, GA 30339
Tel: 404-943-9850

1800 Birchmount Way
Suite 800
Toronto, Ontario
Canada M3J 2Y5
Tel: 416-242-5631 416-242-5632

High Technology - December, 1984



With computer programming costs escalating rapidly, more and more industrial establishments and research institutes are turning to the new logic-based programming language called Prolog. And they have discovered that they can save time and money by taking advantage of Logicware's advanced implementation, MPROLOG.

This modular version of Prolog is available exclusively through Logicware, and represents one of the most dynamic breakthroughs in the entire field of Artificial Intelligence.

The PROLOG revolution
Selected as the basis for the Japanese Fifth Generation Computer System Project, Prolog enables designers to describe their application in logical terms for interpretation by the computer. And it enables them to do it clearly and quickly.

— the concept of logic programming can serve to unify all these fields (software engineering, data base, computer architecture and artificial intelligence), and — is the language that bridges knowledge engineering and parallel computation.

— is the language of the ACM, September, 1983

Here's what MPROLOG gives you
— A powerful inference "engine" supporting the capabilities of most expert system shells.
— Multiple procedure methods, often driving reasoning with the rules and facts in the program knowledge base. MPROLOG is fully language with statement handling and extended data and control facilities.
— Interfaces to procedural languages.

— Hierarchical modularity allows structured design and facilitates maintenance of applications.

— Hardware and operating system independence. The same application can run on a PC and on a mainframe.

— High performance with efficient use of resources. Compiling and optimization.
— Program development environment.
— On-line "help."
— Interactive editing and error correction program trace.

The MPROLOG operating environment
MPROLOG operates in a number of computer environments including:

- IBM[®] VM/CMS
- IBM[®] MVS
- Digital Equipment Corporation VAX/VMS[®]
- VAX/VMS (UNIX[®])
- AI Work Station
- Interphase 4.0[™]
- Sp[®] Apollo[™]
- Chaos Project Maniac[®]
- Other vendors under development (include the more popular MPROLOG based work stations, in addition to IBM machines such as the Symbolics 3600.

MPROLOG is supported on both the Logicware VLSI and IBM[®] hardware. IBM[®] is a registered trademark of International Business Machines Corporation. IBM, MVS, VM, and UNIX are trademarks of International Business Machines Corporation. Sp[®] is a trademark of Sparc Systems, Inc.

MPROLOG is an interactive logic programming tool that lets you define your own rules and facts in a declarative and easy-to-use language. It is designed to work on both IBM and non-IBM hardware.

MPROLOG

SYSTEM FEATURES AND COMPONENTS

A rich and powerful language, MPROLOG is an expert-friendly, robust tool for the development, support and maintenance of "knowledge-structure" applications. Hundreds of North America's leading corporations have employed MPROLOG because of each advanced system feature as:

- Full support to assist in program development.
- Advanced logic engine, including native support for arithmetic, string, numeric, and symbolic operations.
- Flexible and cost-saving program generation.
- Expensive (including on-line) debugging facilities.
- Complete on-line storage and retrieval capabilities.
- Real-time response to user requests.
- Full support to assist in program maintenance.
- Advanced logic engine, including native support for arithmetic, string, numeric, and symbolic operations.
- Flexible and cost-saving program generation.
- Expensive (including on-line) debugging facilities.
- Complete on-line storage and retrieval capabilities.
- Real-time response to user requests.
- Full support to assist in program maintenance.

Hierarchical Modularity
MPROLOG facilitates the independent programming and testing of modules by imposing repeatable interfaces between modules, modularity of names, and other Advanced techniques.

Compatibility/Portability
MPROLOG is an interpreted, single logic programming language in powerful hardware and operating system independence. Close system compatibility permits maintenance of a single source for applications sharing a client, production environments.

Performance
Interpreter performance ranges from 500 LIPS on an IBM PC/XT to 30,000 LIPS on an IBM MVS under VLSI/MS. The compiler can provide a 4x performance improvement.

MAJOR SYSTEM COMPONENTS

PDS — Interactive Program Development Support System

- Integrated and unified editor
- Interactive app-point menu facility including backtracking
- Statements are pre-processed
- Selection of syntactic components for pattern matching.
- Real-time editing
- Interactive app-point menu facility including backtracking
- Real-time interaction.
- Debug Facility to analyze logical errors.
- Variable names are preserved.
- Environment can be tailored by operator.
- On-line help.
- Result of previous commands.
- Initialization keys user's profile.
- Accurate operating system commands and output.
- System in MPROLOG as PDS modules can be consolidated with user modules.

Pretranslator

- Converts source modules into an efficient internal form "Binary module" giving listing and cross-reference listing.

Compiler

- Compiles binary modules into directly executable machine code resulting in a significant speed-up.

Consolidator

- Links binary modules and compiled modules into either a new binary module with a qualified runtime or a binary program.

Interpreter

- Executes binary programs.
- Full recursion optimization and success popping (uses space efficiently).
- Large interpreter tables.
- Supports stand alone "working applications".



Built-In Predicates

- More than 200 built-in predicates are used in processing
- Modulus
- Subtraction
- Escape/line
- Modulus
- Comparison
- File
- List
- Remainder
- Expression
- Division
- String
- Output
- Bitwise

Egy Logicware hirdetés

IMPORTANT P.C. BREAKTHROUGH!

Now you can buy a full power mainframe Artificial Intelligence language for your IBM[®] compatible personal computer



Through a major breakthrough in the development of Artificial Intelligence applications, a unique version of Prolog is now available for your personal computer. It is called MPROLOG P-500. MPROLOG P-500 is an advanced, programmer friendly version of Prolog - the computer language selected as the basis for the famous IBM Collaboration Computer System Project.

MPROLOG has been specially designed for:

- Software developers
- All education
- Expert systems research and development
- Capabilities including a comprehensive

How MPROLOG Programming Works
 In MPROLOG, program concepts are STRICTS and RULES expressed in a natural way. The former mechanism for generating operators or arriving at conclusions based on RULES and RULES in both case the language. This language allows you to write potential software users and those that wish traditional algorithmic languages. MPROLOG enhances your computer's ability to work at current state of technology, and every thing data to reducing cable that can be found here: www.logicware.com provide it.

It is programmed with the standard Boston in the capital of Massachusetts and all copies are available. The primary world database that Boston is a city, as mentioned by the actual history for the time being - "capital of Boston Massachusetts" - the language is distinctive and easy to learn.

- High Technology.
- December 1982

MPROLOG P-500 Offers Seven Key Features

- Easy to learn
- Flexible enough to let you do your own expert system
- Supported by Logic Labs, a potential in the computer programming development environment to maintain correspondence
- Designed for the immediate development of AI applications to reduce their complexity and to make a programming team by concurrently develop applications
- Machine independent - Any application written in MPROLOG on one hardware can be transferred to run on any other machine with a fast state implementation of MPROLOG.
- A full featured implementation of PROLOG. It contains over 200 built-in predicates including DECIO (IO control code)

- Runs to offer performance, productivity and portability
- MPROLOG P-500 on your IBM[®] PC or compatible system also runs on the identical configuration of IBM PC, DOS and run on other IBM[®] VAX/VMS, IBM[®] MVS/TSP, DEC[®] VMS/VAX, DEC[®] VAX/VMS, AIX/RS6000, UNIX, and ILLINOIS/UNIX alpha UNIX/REFLEX and other AI work stations.

For further details call the number below or complete and mail this correspondence

LOGICWARE
 Los Angeles • Toronto • Atlanta • Boston
 Customer Service Locations at:
 1-617-547-2393 or
 1-416-665-0022 (In Canada)

Find Out More about MPROLOG P-500

The information, complete and mail this coupon to:

Logicware Inc.
 5000 Chertway
 8700 Forest Park Rd. Suite 208
 Atlanta, GA 30339

Yes, please send me more information on MPROLOG P-500

Name

Title Company

Address

State/Prov. Zip

Post/Postal Code

Computer Operating System Version #

Manufacturer's Make & Model Memory

PI

CIRCLE NO. 38 ON READER SERVICE CARD

pcjs.org

Az MPROLOG utóélete

- ZEXPERT – az osztrák Zentralsparkasse (mai neve: Bank Austria) számára SzKI-ban fejlesztett, MPROLOG-alapú szakértői rendszer
- A ZEXPERT kapcsán a Zentralsparkasse kisebbségi tulajdonosa lett az Elméleti Laboratórium munkatársaiból alapított IQSoft Rt-nek
- A 90-es években az IQSoftban is számos projektben használtuk az MPrologot, ill. más Prolog rendszereket is
- 2000-ben egy ügyfél Németországból jelentkezett, hogy az évezredváltás miatt nem indul az MPROLOG rendszerük. . .

Összefoglalás

- Szakmailag nagy sikerenek tekinthető, hogy csapatunk
 - a Prolog születésétől kezdve jelen volt a nemzetközi porondon
 - időnként vezető szerepet is tudott betölteni
 - számos alkalmazás kifejlesztését tette lehetővé
- Ugyanakkor üzletileg is sikeres volt, ha nem is kiugróan
- Számomra, személyesen: nagy kaland volt!
- Köszönetnyilvánítás
 - a NIM IGÜSZI Számolóközpontból: Csébfalvi Károlynak, Pázmány Bélának, Náray Miklósnak, Bedő Árpádnak, hogy középiskolás koromtól befogadtak;
 - az SZKI-ból: Náray Zsoltnak, Dömölki Bálintnak, hogy a sok nehézség ellenére is megteremtették a K+F munka lehetőségét;
 - minden kedves kollégámnak, akik nélkül soha sem tudtuk volna a felvázolt eredményeket elérni;
 - és végül a családomnak, hogy a Prolog-őrülettel járó nehézségeket, gondokat türelemmel elviselték. . .
- Hivatkozás: Langer Tamás: Emlékeim

<http://itf2.njszt.hu/324rtr4/uploads/Emlekeim-V2.0.pdf>