A PROLOG programozási nyelv

I. Története

1972 – Marseille (PROgramming in LOGic)

1975 – NIM IGU:SZI

1978 – PROLOG fordító

1982 – MPROLOG, TPROLOG

1985 – TC PROLOG

198x – Turbo PROLOG

1984 – mikroPROLOG

1989 – CS PROLOG

II. Jellemzői

Amatőr, illetve professzionális, logikai

Program = több fordítási egység

1 fordítási egység = sok programegység (tény vagy szabály)

Nincs változó, csak paraméterek

A program egy logikai formula

Rekurzió, logikai műveletek, mintaillesztés

III. A program elemei

Tények (pl. szuloje(a,b).)

Szabályok (pl. nagyszuloje(X,Y) ha szuloje(X,Z) és szuloje(Z,Y))

Kérdések (pl. szuloje(a,X)?) - mintaillesztés mindenféle lehetséges kérdésre

Irásmód:

kulcsszavas – ha (if) és (and) vagy (or)

írásjeles – :- , ;

IV. A Turbo PROLOG program felépítése

trace

domains

típusdefiníciók (pl. sorozat=symbol*)
(elemi típusok: symbol, integer, real, char, string, felsorolás)

predicates

szabály és ténydeklarációk (pl. szuloje(symbol,symbol)
osszefuttat(lista,lista,lista))

database

szabály és ténydeklarációk

goal

kérdések

clauses

tények és szabályok

V. Elemi utasítások, programszerkezet

Elágazás

szuloje(X,Y) ha apja(X,Y) vagy anyja(X,Y).

más megoldás:

szuloje(X,Y) ha apja(X,Y).

szuloje(X,Y) ha anyja (X,Y).

Rekurzió

ose(X,Y) ha szuloje(X,Y) vagy szuloje(X,Z) és ose(Z,Y).

Akármi jel: _

szulo(X) ha szuloje(X,_).

Ha csak egyetlen megoldás kell valamiből:

egy_x(A) ha x(A) és !.

A ! egyéb használata: hiányzó feltételek pótlása

absz(X,Y) ha X>=0 és X=Y vagy –X=Y.

absz(5,–5)-re igazat ad, absz(5,X)-re a –5 is megoldás, mert az X<0 feltétel hiányzik az egyik ágról.

Helyesen:

absz(X,Y) ha X>=0 és ! és X=Y vagy –X=Y.

(MPROLOG: X is Y)

Öszes megoldás megadása: fail, write, nl, (succeed – MPROLOG)

osszes_x ha x(A) és write(A," ") és fail vagy nl.

Adott tulajdonsággal nem rendelkező:

Nem_x(A,B) ha valami(A) és valami(B) és nem(x(A,B)).

Tagadás

(pl.: not(X) ha X és ! és fail vagy . – csak MPROLOG, mert ott lehet metafeltétel a paraméter – X csak futáskor kap értéket)

Rakérdezés valami egyszerességére:

egyszeres_x(A) ha x(A) és nem(tobbszoros_x(A)).
tobbszoros_x(A) ha x(A) és x(B) és A<>B.

Rakérdezés valami összességére:

osszes_x(A) ha x(_,B) és nem(nincs_x(A,B)).
nincs_x(A,B) ha nem(x(A,B)).

Rákérdezés a legnagyobbra:

max_x(Y) ha x(Y) és nem(nagyobb_x(Y)).
nagyobb_x(Y) ha x(Z) és Z>Y.

Adatbáziskezelés: asserta, assertz, retract, retractall.

Ez tanulás, több, mint az adattárolás, mert nem csak tényt, hanem új szabályt is felvehetünk ezen szabályok paramétereként.

ut(budapest,kijev).
ut(budapest,london).
ut(london,hongkong).

van_ut(X,Y) ha ut(x,y) vagy ut(y,x).

mehetunk(X,Y) ha van_ut(X,Y) vagy van_ut(X,Z) és szabad(Z) és mehetunk(Z,Y).

szabad(Z) ha voltunk(Z) és ! és fail vagy asserta(voltunk(Z)) vagy töröl(voltunk(Z)).)

Adott tények megszámolása:

darab_x(A) ha x(B) és szabad(B) és darab_x(C) és A=C+1 vagy A=0.
szabad(B) ha volt(B) és ! és fail vagy asserta(volt(B)).

Adott tények összegzése:

összeg_x(A) ha x(B,Y) és szabad(B) és összeg_x(C) és A=C+Y vagy A=0.

A legtöbb adott tulajdonsággal rendelkező:

maxdarab_x(Y) ha darab_x(Y,A) és nem(több_x(A)).

több_x(A) ha darab_x(Z,B) és B>A.

Öszetett példa: lakástervezés

lakas(BE,AJTO1,ABLAK1,ABLAK2) ha bejaratos(BE,AJTO1,ABLAK1) és szemben(AJTO1,AJTO2) és szoba(AJTO2,ABLAK2) és not (szemben(ABLAK1, ABLAK2)).

bejaratos(BE,AJ,AB) ha szoba(AJ,AB) és irany(BE) és not (BE=eszak) és not (BE=AJ) és not (BE=AB).

szoba(AJ,AB) ha irany(AJ) és irány(AB) és not (AB=eszak) és not (AJ=AB).

irany(eszak). szemben(eszak,del).
irany(del). szemben(kelet,nyugat).
irany(kelet). szemben(del,eszak).
irany(nyugat). szemben(nyugat,kelet).)

Operátor deklaráció:

operator(nev,irany,prioritás)

VI. Öszetett adattípusok és alkalmazásai

Rekord-struktúra:

domains szemelyleiras=szemely(nev,kor,cim)

predicates olvas(szemelyleiras)

clauses olvas(szemely(N,K,C)) if readln(N) and readint(K) and readln(C).

goal olvas(SZ) and write(SZ) and !

Sorozat megadás: NIL vagy elem.sorozat
[] vagy [elem!sorozat] vagy [elem,elem,...]

Műveletek:

eleme(ELEM,ELEM._).
eleme(ELEM,_.SOROZAT) ha eleme(ELEM,SOROZAT).

hozzafuz(NIL,SOROZAT,SOROZAT).
hozzafuz(E1.S1,SOROZAT,E1.S2) ha hozzafuz(S1,SOROZAT,S2).

osszefuttat(NIL,NIL,NIL).
osszefuttat(_,NIL,_).
osszefuttat(NIL,_,_).
osszefuttat(X.A,Y.B,X.C) ha X<Y és osszefuttat(A,Y.B,C).
osszefuttat(X.A,Y.B,X.C) ha X=Y és osszefuttat(A,B,C).
osszefuttat(X.A,Y.B,Y.C) ha X>Y és osszefuttat(X.A,B,C).

kulonbseg(X,Y,Z) ha ...

Összetett példa: Portia ládikái

VII. Programozási tételek

1. Összegzés
2. Eldöntés
3. Kiválasztás (index, érték)
4. Megszámlálás
5. Maximumkiválasztás
6. Kiválogatás

VIII. Hatékonyság: a visszalépések korlátozása

Kizáró feltételek vizsgálata:

szuloje(X,Y) ha anyja(X,Y) és ! vagy apja(X,Y).

Felesleges megoldások elhagyása:

eleme(X,X._) ha !.
eleme(X,_.Y) ha eleme(X,Y).

IX. Egyéb PROLOG változatok

1. Szimulációs TPROLOG
2. Párhuzamos CSPROLOG
3. Moduláris MPROLOG