Programozás nyelvek 3
3. előadás

Prolog feladattípusok

Beszélgetés tanulással – egy párbeszéd:

```
Ki vagy?Anna
Mi a nemed (N/F)?N
Hány éves vagy, Anna?15
Hogy köszöntselek?Szia
Szia, Anna!
Ki vagy?Andrea
Mi a nemed (N/F)?N
Hány éves vagy, Andrea?35
Hogy köszöntselek?
Jó napot, hölgyem!
```

```
Ki vagy?Anna
Szia, Anna!
```

Prolog feladattípusok

Beszélgetés tanulással:

```
párbeszéd ha write("Ki vagy?") és readln(Név)
  és Név<>"" és beszélget(Név) és párbeszéd
  vagy write("Vége") and nl.
beszélget(Név) ha személy(Név,Nem,Kor)
  és válasz(Név,Nem)
  vagy kivagy(Név,Nem,Kor) és
  asserta(személy(Név,Nem,Kor))
  és válasz(Név,Nem).
```

Prolog feladattípusok

Beszélgetés tanulással:

```
válasz(Név,Nem) ha köszönés(Név,Kösz)
  és üdvözöl(Név,Nem,Kösz) és !
  vagy write("Hogy köszöntselek?") és
  readln(Így) és asserta(köszönés(Név,Így))
  és válasz(Név,Nem).
üdvözöl(Név,Nem,Üdv) ha Üdv="" és mit(Nem,Mi)
  és write("Jó napot, ",Mi,"!") és nl
  vagy write(Üdv,", ",Név,"!") és nl.
```

Prolog feladattípusok

Beszélgetés tanulással:

```
mit("F","uram").
mit("N","hölgyem").
kivagy(Név,Nem,Kor) ha
  write("Mi a nemed (N/F)?") és readln(Nem)
  és write("Hány éves vagy, ",Név,"?")
  és readint(Kor) és nl.
```

Prolog rekord típus

Rekord-struktúra::

```
domains
  szemelyleiras=szemely(nev,kor,cim)
predicates
  olvas(szemelyleiras)
clauses
 olvas(szemely(N,K,C)) ha readln(N) és
  readint(K) és readln(C).
goal
 olvas(SZ) és write(SZ) és !
```

Prolog sorozat típus

Sorozat megadás:

- `NIL` vagy `elem.sorozat`
- `[]` vagy `[elem|sorozat]` vagy `[elem,elem,...]`

Példa:

- `1.2.3.NIL`
- `[1|[2|[3|[]]]]` vagy `[1,2,3]`

Eleme művelet megvalósítása:

```
eleme(E,[E,_]).
eleme(E,[_,S]) ha eleme(E,S).
```

Prolog sorozat típus

Hozzáfűz művelet megvalósítása:

```
hozzáfűz([],S,S).
hozzáfűz([E|S1],S,[E,S2]) ha hozzáfűz(S1,S,S2).
```

Létrehoz művelet megvalósítása:

```
létrehoz(S,T) ha valami(E) és nem(eleme(E,T))
  és létrehoz(S,[E|T]) vagy S=T.
```

Elemszám művelet megvalósítása:

```
elemszám(0,[]).
elemszám(X,[_|S]) ha elemszám(Y,S) és X=Y+1.
```

Programozási tételek Prologban

Összegzés:

```
összeg(0,[]).
összeg(X,[E|S]) ha összeg(Y,S) és X=E+Y.
```

Eldöntés:

```
eldönt([E|_]) ha t(E).
eldönt([_|S]) ha eldönt(S).
```

Megjegyzés: ha a sorozat nem bontható részekre (azaz üres), akkor egyik szabály sem alkalmazható, tehát a válasz hamis.

Programozási tételek Prologban

Kiválasztás:

```
kiválaszt(E,[E|_]) ha t(E).
kiválaszt(E,[_|S]) ha kiválaszt(E,S).

kiválaszt(1,[E|_]) ha t(E).
kiválaszt(X,[_|S]) ha kiválaszt(Y,S) és X=Y+1.

kiválaszt(1,E,[E|_]) ha t(E).
kiválaszt(X,E,[_|S]) ha kiválaszt(Y,E,S) és
  X=Y+1.
```

Megjegyzés: Ez egyben a keresés tétel is.

Programozási tételek Prologban

Megszámolás:

```
megszámol(0,[]).
megszámol(A,[E|S]) ha t(E) és megszámol(B,S) és
   A=B+1.
megszámol(A,[_|S]) ha megszámol(A,S).
```

Rövidebb változat, ha lehet zárójelezni:

```
megszámol(0,[]).
megszámol(A,[E|S]) ha megszámol(B,S) és
   (t(E) és A=B+1 vagy A=B).
```

Programozási tételek Prologban

Megszámolás:

Rövidebb változat, ha nem lehet zárójelezni:

```
megszámol(0,[]).
megszámol(A,[E|S]) ha megszámol(B,S) és
  új(A,B,E).
új(A,B,E) ha t(E) és A=B+1 vagy A=B.
```

Programozási tételek Prologban

Maximumkiválasztás:

```
maximum(E,[E]).
maximum(E,[F|S]) ha maximum(G,S) és
  nagyobb(E,F,G).
nagyobb(E,E,G) ha E>=G.
nagyobb(E,F,E) ha E>=F.
```

Programozási tételek Prologban

Kiválogatás:

```
kiválogat([],[]).
kiválogat([E|T],[E|S]) ha t(E) és
  kiválogat(T,S).
kiválogat(T,[_|S]) ha kiválogat(T,S).
```

Szétválogatás:

```
szétválogat([],[],[]).
szétválogat([E|T],U,[E|S]) ha t(E) és
  szétválogat(T,U,S).
szétválogat(T,[E|U],[E|S]) ha
  szétválogat(T,U,S).
```

Programozási tételek Prologban

Összefuttatás:

```
össze([],[],[]).
össze(S,[],S).
össze([],S,S).
össze([X|A],[Y|B],[X|C]) ha X<Y és
  össze(A,[Y|B],C).
össze([X|A],[Y|B],[X|C]) ha X=Y és
  össze(A,B,C).
össze([X|A],[Y|B],[X|C]) ha X>Y és
  össze([X|A],B,C).
```

Prolog rekurzív típus

Fa-struktúra::

```
összeg(F,gyökér(F)).
összeg(F,fa(Bal,Jobb,E)) ha összeg(B,Bal) és
  összeg(J,Jobb) és F=E+B+J.
```

TProlog szimuláció

TProlog újdonságok::

Folyamat létrehozása: `new().`

Folyamat felfüggesztése adott időtartamra: `hold(idő).`

Folyamat feltételre felfüggesztése: `wait()` vagy `wait-for().`

Folyamatok közötti kommunikáció: `send().`

TProlog szimuláció

Feladat:

Egy kikötőben 3 mólónál lehet kikötni (a,b,c). Ismerjük mindegyikhez a bejárattól a mólóig eljutási időt. A kikötés a következő lépésekből áll:

- engedélykérés valamely mólóhoz;
- várakozás az engedélyre;
- a móló megközelítése;
- kirakodás.

TProlog szimuláció

Megoldás:

```
kikötés(X) ha móló(X) és send(kérelem(X)) és
  wait-for(ok) és távolság(X,I) és hold(I) és
  hold(kirakodási idő).
móló(a).  móló(b).  móló(c).
távolság(a,2). távolság(b,3). távolság(c,2).
engedélyezés ha wait-for(kérelem(X)) és
  szabad(X) és hold(1) és send(ok).
szabad(b).     szabad(c).
new(kikötés(Móló),hajó,0,11) és
  new(engedélyezés,parancsnokság)?
```

TProlog szimuláció

Kérdés:

```
new(kikötés(Móló),hajó,0,11) és
  new(engedélyezés,parancsnokság)?
```

Programozás nyelvek 3
3. előadás vége