



Programozás nyelvek a
közoktatásban
3. előadás

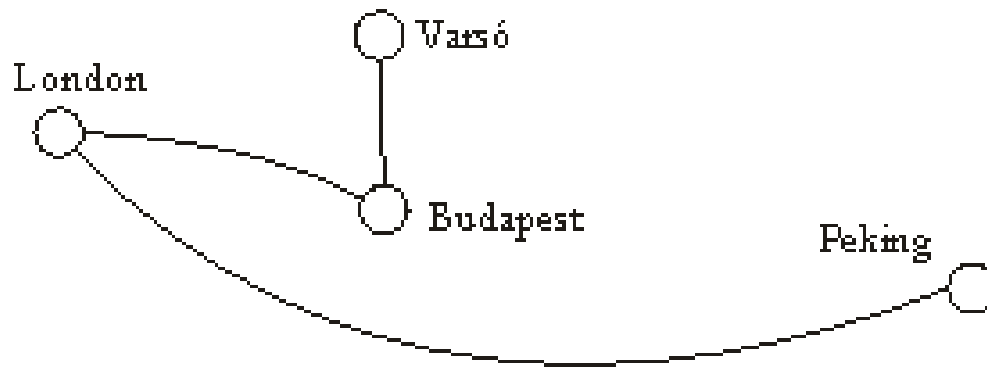


Prolog feladattípusok



Feladat: ismerünk városok közötti repülőgépjáratokat, amelyek alapján el kellene döntenünk, hogy egy városból eljuthatunk-e egy másik városba, esetleg átszállásokkal.

Felhasználható tudás: a repülőgépjáratok oda-vissza közlekednek.





Prolog feladattípusok



A térkép tárolása tényekkel és szabályokkal (a szimmetria miatt):

`út (budapest, varsó) .`

`út (budapest, london) .`

`út (london, peking) .`

`van_út (X, Y) ha út (X, Y) vagy út (Y, X) .`





Prolog feladattípusok



Megoldási ötlet: vagy van közvetlen járat, vagy megpróbálunk átszállással eljutni (mint az őse szabálynál).

mehetünk (X, Y) ha $\text{van_út}(X, Y)$ vagy
 $\text{van_út}(X, Z)$ és $\text{mehetünk}(Z, Y)$.

Probléma: végtelen ciklusba kerülünk.

Az oka: a repülőgép hálózatot leíró gráf irányítatlan. (Az ősoket leíró gráf irányított és körmentes volt.)

Megoldás: kellenének változók azon városok tárolására, ahol már jártunk.





Prolog feladattípusok



Megoldási elv: a Prolog program képes futás közben változtatni önmagán, új tényeket, szabályokat vehetünk fel, korábbiakat törölhetünk.

Az ehhez kapcsolódó műveletek:

`assert(tény vagy szabály).`

`asserta(tény vagy szabály).`

`assertz(tény vagy szabály).`

`retract(tény vagy szabály feje).`

`retractall(tény vagy szabály feje).`





Prolog feladattípusok



Megoldási ötlet: vagy van közvetlen járat, vagy megpróbálunk átszállással eljutni (mint az őse szabálynál).

```
mehetünk(X, Y) ha van_út(X, Y) vagy
    van_út(X, Z) és szabad(Z) és mehetünk(Z, Y) .
szabad(Z) ha voltunk(Z) és ! és fail
    vagy asserta(voltunk(Z)) .
```

Ha az adatbázisban már van `voltunk(Z)` tény, akkor oda ne menjünk még egyszer; ha nincs, akkor vegyük fel és menjünk is el oda!





Prolog feladattípusok



Újabb problémák:

- A "varsói" kitérőt megteesszük.
- Másodszori kérdésre nem kapunk választ.

Megoldás egy főprogram, amely kitakarítja a korábbi információkat és a kezdő városba visszalépést nem engedi:

kérdés (X, Y) ha retractall(voltunk(_)) és
assert(voltunk(X)) és mehetünk(X, Y) .





Prolog feladattípusok



Adott tények megszámlolása (gyerekek száma):

szülője (anna, antal) .

szülője (anna, andrea) .

szülője (andrea, andrás) .

darab (X, A) ha szülője (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)) .

Ez csak akkor működik, ha a gyerekek neve különböző!

Egyes Prolog verziókban:

$A \text{ is } C+1$





Prolog feladattípusok



Adott tények összegzése (termékek árát tároljuk):

ára (kávé, 1000) .

ára (tea, 800) .

ára (kakaó, 1000) .

összeg(A) ha ára(X,B) és szabad(X) és
összeg(C) és $A=C+B$ vagy $A=0$.

szabad(X) ha volt(X) és ! és fail
vagy `asserta(volt(X))` .

Ez csak akkor működik, ha a termékek neve különböző!

Egyes Prolog verziókban:

`A is C+B`





Prolog feladattípusok



Adott tények számának maximuma (legtöbb gyerekes szülő):

szülője(anna, antal) .

szülője(anna, andrea) .

szülője(andrea, andrás) .

darab(X, A) ha szülője(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))` .

legtöbbb(X) ha `darab(X, A)` és `nem(vantöbbb(A))` .

`vantöbbb(A)` ha `darab(_, B)` és $B>A$.

Egyes Prolog verziókban:

`A is C+1`





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.
```





Programozás nyelvek a
közoktatásban
3. előadás vége