



# Programozás nyelvek a közoktatásban 2. előadás



# Prolog feladattípusok



Ha valamiből csak egy megoldás kell:

`egy_szülő(X)` ha `szülő(X)` és `!`.

Ha valamiből az összes megoldás kell:

`összes_szülő` ha `szülő(A)` és `write(A, ' ')`  
és `fail` vagy `succeed`.

Magyarázat:

- `fail` – azonosan hamis logikai formula
- `succeed` (néha: `true`) – azonosan igaz logikai formula
- `write` – kiírás





# Prolog feladattípusok



## Adattípusok:

- számkonstansok – 1, -3, 2.7, 3.1e5
- szövegkonstansok – 'István', 'Szent István', istvan
- műveleti jelek – +, -, \*, /, <, =, >
- struktúranevek – neve(istvan)





# Prolog feladattípusok



Adott tulajdonsággal nem rendelkező:

`nem_szülő(A)` ha `valami(A)` és `nem(szülő(A))`.

`valami(A)` ha `szülője(A,_)` vagy `szülője(_,A)`.

Rákérdezés valami egyszerességre:

`egyszeres_szülő(A)` ha `szülő(A)` és  
`nem(többszörös_szülő(A))`.

`többszörös_szülő(A)` ha `szülője(A,X)` és  
`szülője(A,Y)` és `X<>Y`.





# Prolog feladattípusok



Rákérdezés valami összességére:

`összes_szülő(A)` ha `szülő(A)` és  
`nem(vanmás_szülő(A))`.

`vanmás_szülő(A)` ha `szülő(B)` és  $A \neq B$ .

Rákérdezés a legnagyobbra:

`max_korú(Y,K)` ha `kora(Y,K)` és  
`nem(vannagyobb_korú(K))`.

`vannagyobb_korú(K)` ha `kora(_,L)` és  $L > K$ .





# Logikai feladatok



## "Portia ládikája" első feladata:

Portiának van három ládikája (arany, ezüst, ólom). Férjhez akar menni, de intelligens férjet szeretne magának. Ezért mielőtt a kérőnek igent mondana, próbára teszi észbeli képességét.

Elrejt egy képet valamelyik ládikába, és mindegyikre egy-egy állítást ír föl, amely a kép hollétére vonatkozik.

Majd hozzátesz segítségként egy "peremfeltételt", ami alapján már egyértelműen meg lehet találni a helyes választ.





# Logikai feladatok



A konkrét feladvány részletei:

- Az arany ládika felirata: "ebben a ládikában van".
  - Az ezüst ládika felirata: "nem ebben a ládikában van".
  - Az ólom ládika felirata: "nem az arany ládikában van".
- A peremfeltétel: "az állítások közül legfeljebb egy igaz".





# Logikai feladatok



Ugyanez Prolog-ban:

ládán (arany, arany) .

ládán (ezüst, arany) .

ládán (ezüst, ólom) .

ládán (ólom, ezüst) .

ládán (ólom, ólom) .

vagy

ládán (ezüst, X) ha  $X = \text{'arany'}$  vagy  $X = \text{'ólom'}$  .

ládán (ólom, X) ha  $X = \text{'ezüst'}$  vagy  $X = \text{'ólom'}$  .







# Logikai feladatok



A mesének szó szerint megfelelően:

ládán (arany, arany) .

ládán (ezüst, X) ha láda (X) és nem (X='ezüst') .

ládán (ólom, X) ha láda (X) és nem (X='arany') .

ládá (X) ha X='arany' vagy X='ezüst' vagy  
X='ólom' .

Más tények vagy szabályok is lehetnének, pl.:

arany (arany) . ezüst (arany) . ezüst (ólom) .

ólom (ezüst) . ólom (ólom) .

ezüst (X) ha láda (X) és nem (X='ezüst') .

ólom (X) ha láda (X) és nem (X='arany') .





# Logikai feladatok



Ugyanez Prolog-ban:

```
egy_igaz(X) ha ladan(arany,X) és
    nem(ladan(ezüst,X)) és nem(ladan(ólom,X))
vagy ladan(ezüst,X) és
    nem(ladan(ólom,X)) és nem(ladan(arany,X))
vagy ladan(ólom,X) és
    nem(ladan(arany,X)) és nem(ladan(ezüst,X)).
mind_hamis(X) Ha lada(X) és nem(ladan(arany,X))
és nem(ladan(ezüst,X)) és nem(ladan(ólom,X)).
legfeljebb_1_igaz(X) ha egy_igaz(X) vagy
    mind_hamis(X).
```





# Logikai feladatok



Ugyanez a másfajta tényekkel egyszerűbben :

`egy_igaz(X)` ha

`arany(X) és nem(ezüst(X)) és nem(ólom(X))`

`vagy ezüst(X) és nem(ólom(X)) és nem(arany(X))`

`vagy ólom(X) és nem(arany(X)) és nem(ezüst(X)) .`

`mind_hamis(X)` ha `láda(X) és nem(arany(X))`

`és nem(ezüst(X)) és nem(ólom(X)) .`

`legfeljebb_1_igaz(X)` ha `egy_igaz(X)` vagy

`mind_hamis(X) .`



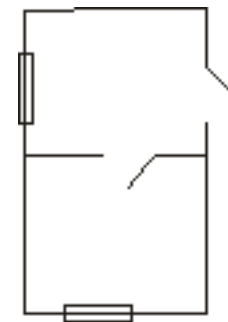


# Lakástervezés



## Lakás szempontok:

- 2 szoba legyen
- az egyikben legyen bejárati ajtó
- mindkettőn legyen ablak
- legyen közöttük átjáró ajtó
- egy falon egy dolog (ajtó vagy ablak) lehet



## Kiegészítő szempontok:

- semmi ne nézzen északra
- a két ablak ne szemben levő falon legyen





# Lakástervezés



## Tények:

irány (észak) .

szemben (észak, dél) .

irány (dél) .

szemben (kelet, nyugat) .

irány (kelet) .

szemben (dél, észak) .

irány (nyugat) .

szemben (nyugat, kelet) .

## Lakás:

lakás (Be, Ajtó, Ablak1, Ablak2) ha

bejáratos\_szoba (Be, Ajtó, Ablak1) és

szoba (Ajtó2, Ablak2) és szemben (Ajtó, Ajtó2)

és nem (szemben (Ablak1, Ablak2)) .





# Lakástervezés



Lakás:

Hatékonysági szempontok – ha a az ajtó az egyik szobából nem ellenkező irányban van a másik szobából, akkor a másik szoba biztos nem jó, felesleges olyanokat vizsgálni:

lakás (Be, Ajtó, Ablak1, Ablak2) ha  
bejáratos\_szoba (Be, Ajtó, Ablak1) és  
szemben (Ajtó, Ajtó2) és szoba (Ajtó2, Ablak2)  
és nem (szemben (Ablak1, Ablak2)) .





# Lakástervezés



## Szobák:

bejáratos\_szoba (Be, Ajtó, Ablak) ha  
szoba (Ajtó, Ablak) és irány(Be)  
és nem(Be=észak) és nem(Be=Ajtó)  
és nem(Be=Ablak) .

szoba (Ajtó, Ablak) ha  
irány(Ajtó) és irány(Ablak)  
és nem(Ablak=észak) és nem(Ajtó=Ablak) .





Programozás nyelvek a  
közoktatásban  
2. előadás vége