

# LOGIKAI FELADVÁNYOK MEGOLDÁSA KORLÁTPROGRAMOZÁSSAL

SZERZŐK: **PAPP ÁDÁM, SÓS NIKOLETT**  
MÉRNÖK INFORMATIKUS BSC., I. ÉVFOLYAM  
TÉMAVEZETŐ: **ŐSZ OLIVÉR**

SZÉCHENYI ISTVÁN EGYETEM, GIVK  
INFORMATIKA TANSZÉK

2018



- ❖ Logikai feladványok
- ❖ Korlátprogramozás
- ❖ Modellezési módszerek
- ❖ Összehasonlító tesztek
- ❖ Redundáns megkötések kiszűrése
- ❖ Összegzés



Einstein-féle logikai feladványok

Zebra feladatok

„Movies Night”:

<b>ing</b>	<i>fekete</i>	<i>kék</i>	<i>zöld</i>	<i>piros</i>
<b>keresztnév</b>	<i>Daniel</i>	<i>Joshua</i>	<i>Nicholas</i>	<i>Ryan</i>
<b>kedvenc film</b>	<i>akció</i>	<i>vígjáték</i>	<i>horror</i>	<i>thriller</i>
<b>nassolnivaló</b>	<i>chips</i>	<i>süti</i>	<i>cracker</i>	<i>popkorn</i>
<b>életkor</b>	<i>11 év</i>	<i>12 év</i>	<i>13 év</i>	<i>14 év</i>



Ivan Sutherland (1963): Sketchpad

1980-as években egyre keresettebb módszer lett

1990-es évektől eladható változatok

MiniZinc:

- grafikus szerkesztőprogram és nyelv is egyben

Christian Schulte (2005): Gecode megoldóprogram

Megoldó működése propagációval:

- folyamatosan csökken a változók lehetséges értékkészlete



## MiniZinc modell

```
%megadjuk milyen értékeket vehetnek fel
```

```
var 1..4 : Belgium;
```

```
var 1..4 : Dánia;
```

```
var 1..4 : Franciaország;
```

```
var 1..4 : Németország;
```

```
var 1..4 : Hollandia;
```

```
var 1..4 : Luxemburg;
```

```
%kikötések
```

```
constraint Belgium != Franciaország;
```

```
constraint Belgium != Németország;
```

```
constraint Belgium != Hollandia;
```

```
constraint Belgium != Luxemburg;
```

```
constraint Dánia != Németország;
```

```
constraint Franciaország != Németország;
```

```
constraint Franciaország != Luxemburg;
```

```
constraint Németország != Hollandia;
```

```
constraint Németország != Luxemburg;
```

```
solve satisfy;
```



# „Zebra” feladatok

Kikötés: A fekete ruhás hölgytől valahol jobbra ül a 60 éves adományozó.

## I. Bináris mátrix logikai operátorokkal:

$$(dress_{black,p} = 1) \rightarrow \left( \sum_{szek \in p+1..db} age_{sixty,szek} = 1 \right) \quad \forall p \in People$$

```
constraint forall(x in PEOPLE)(dress[black, x]=1 -> sum(p in x+1..db)(age[sixty, p])=1);
```

## II. Bináris mátrix relációs operátorokkal:

$$dress_{black,p} \leq \sum_{szek \in p+1..db} age_{sixty,szek} \quad \forall p \in People$$

```
constraint forall(x in PEOPLE)(dress[black, x] <= sum(p in x+1..db)(age[sixty, p]));
```



# „Zebra” feladatok

Kikötés: A fekete ruhás hölgytől valahol jobbra ül a 60 éves adományozó.

## III. Tömbök:

$$\begin{aligned} dress[p] = black \rightarrow count([age[szek] \mid \forall szek \in p + 1..db], sixty) = 1 \quad \forall p \in PEOPLE \quad \wedge \\ age[p] = sixty \rightarrow count([dress[szek] \mid \forall szek \in 1..p - 1], black) = 1 \quad \forall p \in PEOPLE \end{aligned}$$

```
constraint forall(x in PEOPLE)(dress[x]=black -> count([age[p] | p in x+1..db], sixty)=1)
/\ forall(x in PEOPLE)(age[x]=sixty -> count([dress[p] | p in 1..x-1], black)=1);
```

## IV. Tömbök, where záradékkal:

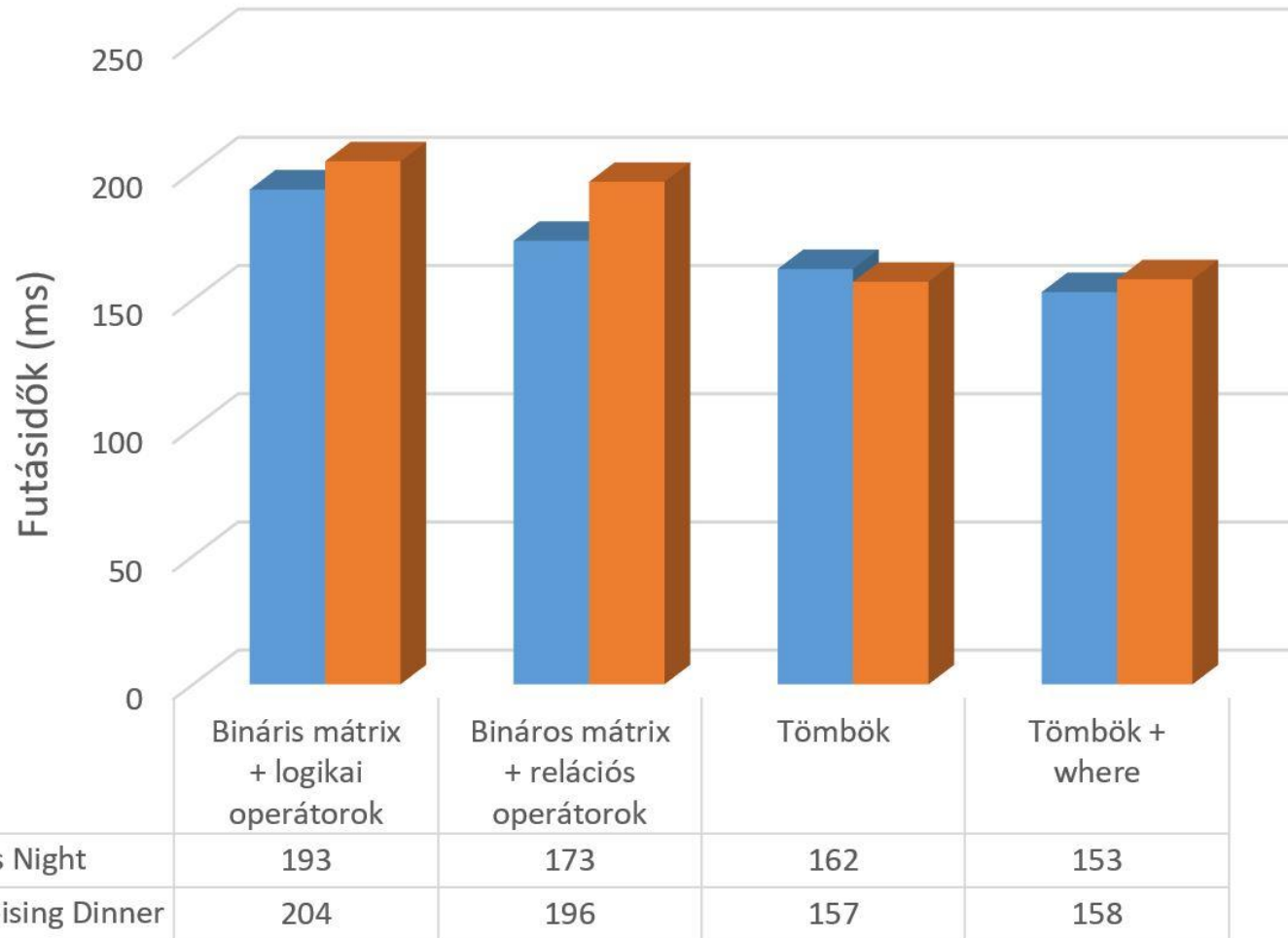
$$\neg(age_a = sixty \wedge dress_b = black) \quad \forall a, b \in PEOPLE \wedge a < b$$

```
constraint forall(a,b in PEOPLE where a<b)(not(age[a]=sixty /\ dress[b]=black));
```



# Teszteredmények I.

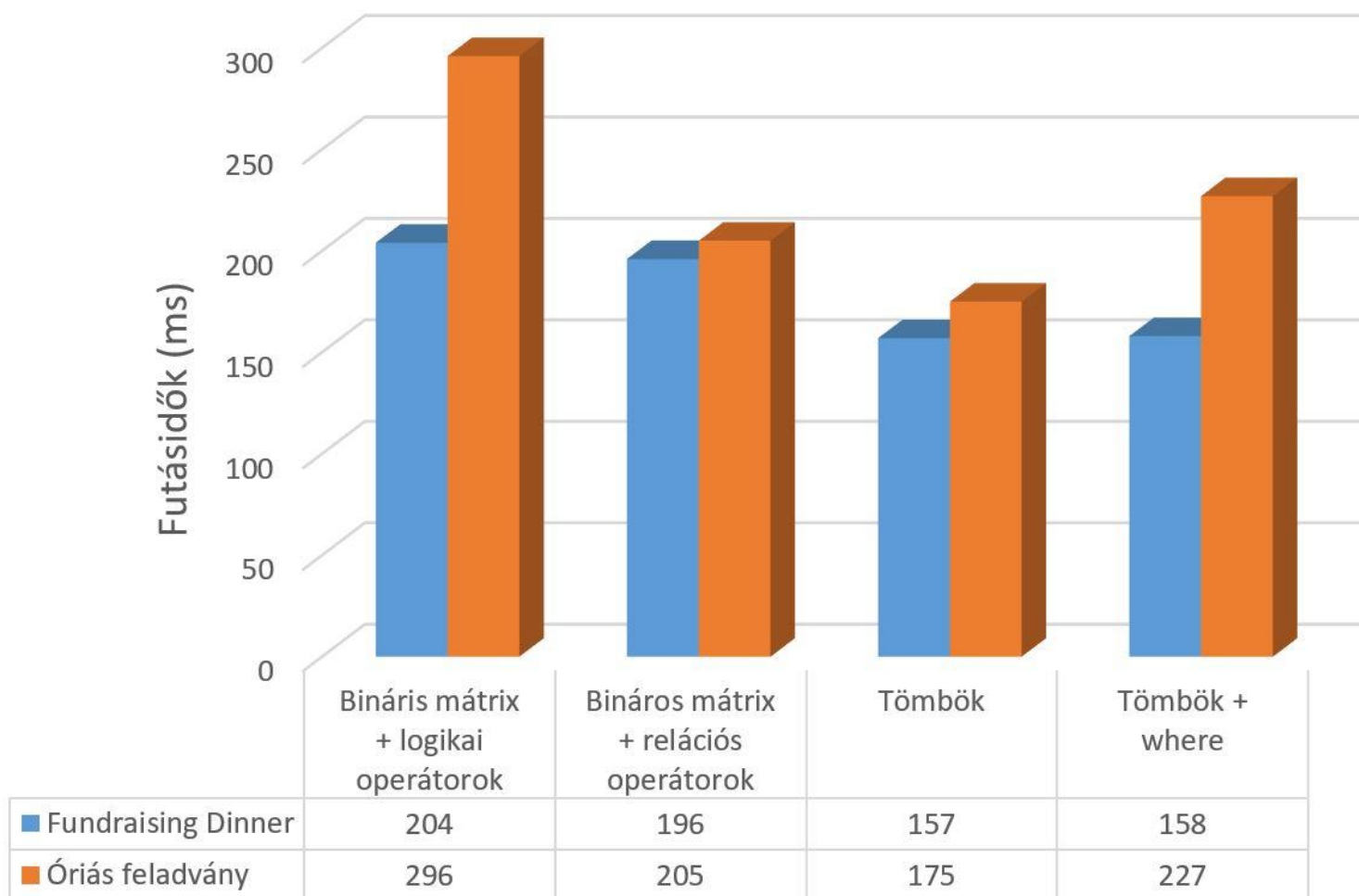
Könnyű/nehéz feladványok - modellek





# Teszteredmények II.

## Nehéz/óriás feladványok - modellek



# „Gardens” feladat





















Einsteinnek tulajdonított logikai feladvány

Adott:

5 kert

5 tulajdonos

12 termény

Hank	Sam	Paul	Zick	Luke
				
				
				
				



# „Gardens” feladat

Több modell is készült:

- Bináris mátrix

$$\sum_{z \in \text{Zoldsegek}} \text{termeszt}_{z,t} = 3 \quad \forall t \in \text{Tulaj} : \text{tulaj}_t = 4$$

```
constraint forall(t in Tulaj where tulaj[t]=4)
(sum(z in Zoldsegek) (termeszt[z,t])=3);
```

- Logikai operátorok használata

$$\text{tulaj}_t = 4 \rightarrow \sum_{z \in \text{Zoldsegek}} J_{z,t} = 3 \quad \forall t \in \text{Tulaj}$$

```
constraint forall(t in Tulaj)(tulaj[t]=4 ->
sum(z in Zoldsegek)(termeszt[z,t])=3);
```



# „Gardens” feladat

- Tulaj hozzárendelési mátrix  
(A kikötések felépítése ugyan az, mint az első verziónál.)

- Termény halmazok

$$I_t = 4 \rightarrow \|Zoldsegek \cap termeszt_t = 3\| \quad \forall t \in Tulaj$$

- Integer mátrix

```
constraint forall(k in Kertek)(tulaj[k]=Hank ->  
count([termeszt[k, n] | n in Noveny], oszirozsa)=0);
```

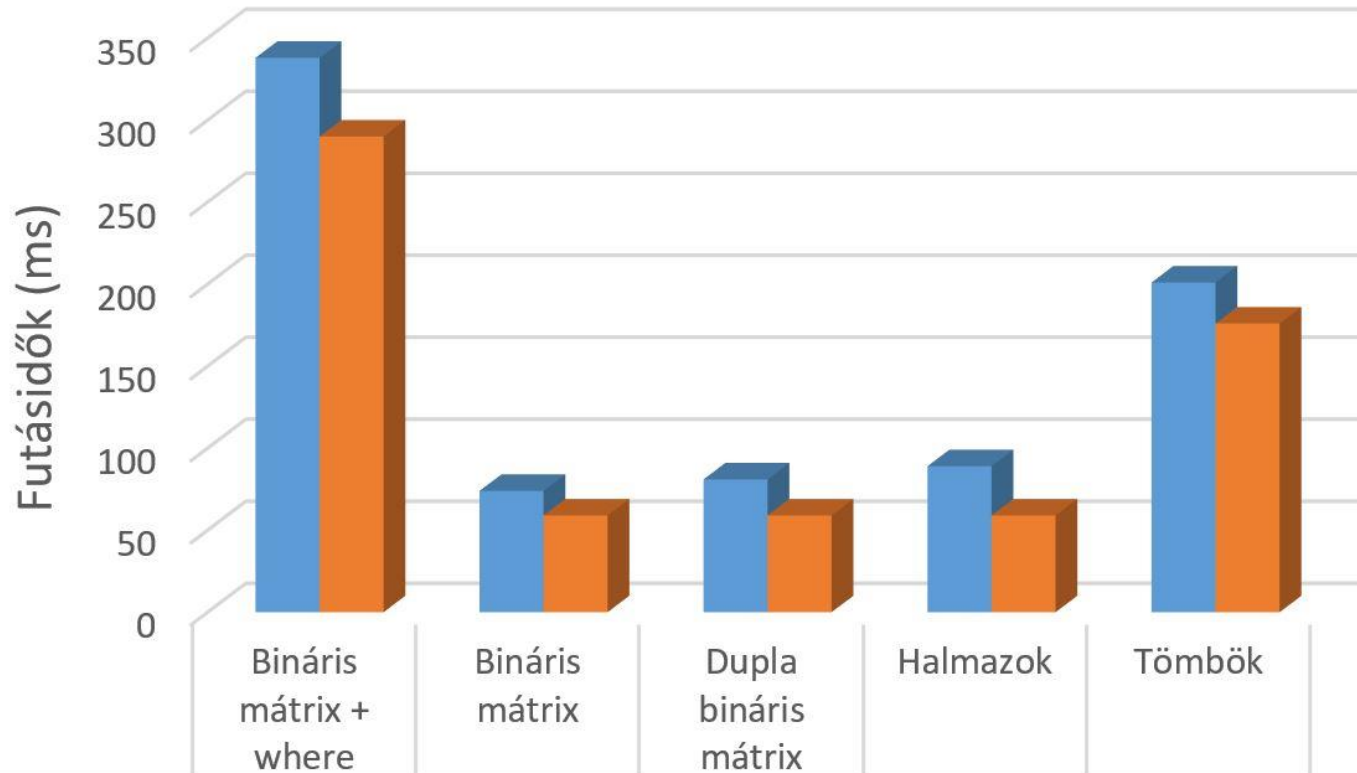
- Tömbökhöz plusz kikötés

```
constraint alldifferent(tulaj);
```



# Teszteredmények III.

Megoldó verziói – „Gardens” modellek



■ Gecode 2.1.6	338	74	81	89	201
■ Gecode 2.1.7	290	59	59	59	176



# Redundáns megkötések

```
constraint if kivesszuk[4]=1 then true else
    forall(p in PEOPLE)(necklace[p]=sapphire <->
        age[p]=fiftyfive)
endif;
```

Elhagyott korlátozások	Movies Night (13-ból)	Fundraising Dinner (21-ből)	Gardens (21-ből)
Egyszerre 1 elhagyása	3 db	7 db	5 db
Egyszerre 2 elhagyása	3 db	15 db	5 db
Egyszerre 3 elhagyása	0 db	13 db	2 db
Egyszerre 4 elhagyása	0 db	4 db	0 db



- ❖ Munkánk során megvizsgáltuk az „Einstein-féle” logikai feladványok szerkezetét és lehetséges megoldásukat.
- ❖ A feladatok általános modellezése végett megismerkedtünk a korlátprogramozás módszereivel.
- ❖ A feladatokat többféle módon modelleztük, és végül teszteltük őket bizonyos szempontok alapján.



Köszönjük a figyelmet!

