

# LOGIKAI FELADVÁNYOK MEGOLDÁSA KORLÁTPROGRAMOZÁSSAL

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

SZÉCHENYI ISTVÁN EGYETEM, GIVK INFORMATIKA TANSZÉK 2018



#### Tartalom



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



## Logikai feladványok



Einstein-féle logikai feladványok

#### Adottak:

- személyek/objektumok
- Tulajdonságok
- Kikötések

Zebra feladatok



## Logikai feladványok



#### "Movies Night":

4 barát moziba megy -> ki hol ül, milyen tulajdonságokkal

**Ing:** zöld piros fekete Kék

**Keresztnév:** Joshua Ryan Nicholas Daniel

**Kedvenc film:** horror vígjáték akció thriller

Nassolnivaló: popkorn chips cracker süti

**Életkor:** 13 év 12 év 14 év 11 év

Joshua szereti a horror filmet.

Joshua az egyik szélen ül.

# Korlátprogramozás



Ivan Sutherland (1963): Sketchpad, a legkorábbi kikötéseket használó rendszer

1980-as években egyre keresettebb módszer lett

1990-es évektől eladható változatok

Optimalizálási és kielégíthetőségi feladatok megoldására

Feladatok megoldása propagációval:

- folyamatosan csökken a változók lehetséges értékkészlete



## Térképszínezés



#### MiniZinc:

- grafikus szerkesztőprogram és nyelv is egyben

Christian Schulte (2005): Gecode megoldóprogram





#### Térképszínezés



#### 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



Movies night:

4x5-ös méret

13 kikötés

Fundrasing dinner:

5x6-os méret

21 kikötés

4 modell a megoldásukra

keresztnév

viselt ruha színe

nyakláncukon található drágakő típusa

életkoruk

fogyasztott koktéljuk

adakozott pénzmennyiség dollárban



#### "Zebra" feladatok



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

#### I. Bináris mátrix logikai operátorokkal:

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

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

#### II. Bináris mátrix relációs operátorokkal:

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

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



#### "Zebra" feladatok



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

#### III. Tömbök:

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

#### IV. Tömbök, where záradékkal:

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

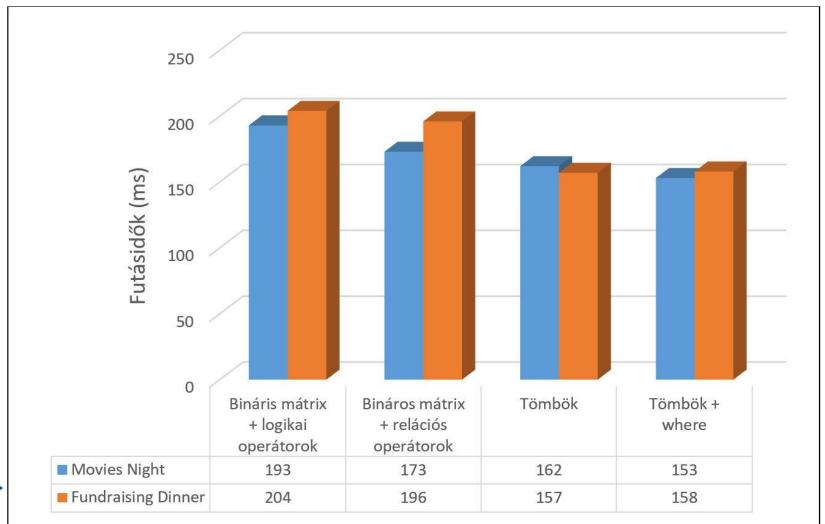
constraint forall(a,b in PEOPLE where a<b)(not(age[a]=sixty /\ dress[b]=black));</pre>



### Teszteredmények I.



#### Könnyű/nehéz feladványok - modellek

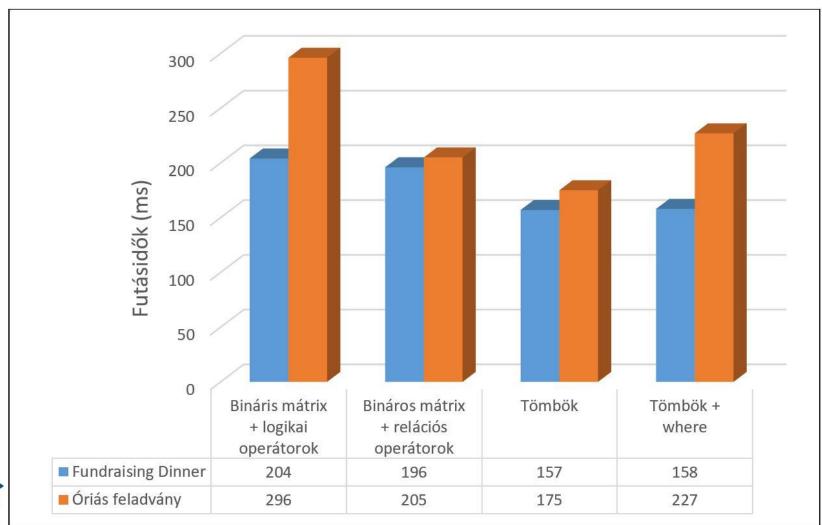




## Teszteredmények II.



Nehéz/óriás feladványok - modellek





## "Gardens" feladat



#### Einsteinnek tulajdonított logikai feladvány

#### Adott:

5 kert

5 tulajdonos

12 termény

Hank	Sam	Paul	Zick	Luke
	1			
	***************************************			HIE WAR
			*	
	Y			



#### "Gardens" feladat



#### Több modell is készült:

Bináris mátrix

$$\sum_{z \in Zoldsegek} termeszt_{z,t} = 3 \qquad \forall t \in Tulaj : tulaj_t = 4$$

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

Logikai operátorok használata

$$tulaj_t = 4 \to \sum_{z \in Zoldsegek} J_{z,t} = 3 \quad \forall t \in Tulaj$$



```
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

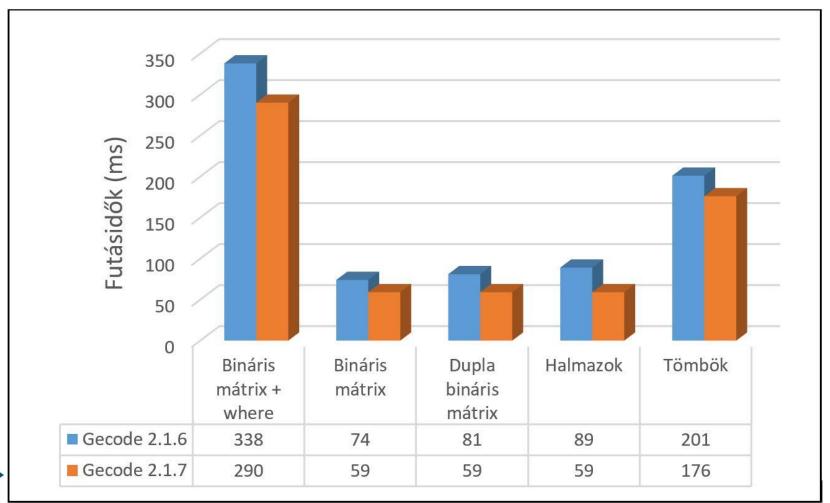
```
constraint alldifferent(tulaj);
```



### Teszteredmények III.



Megoldó verziói – "Gardens" modellek





#### Redundáns megkötések



Elhagyott korlátozások	Movies Night	Fundraising Dinner	Gardens
	(13-ból)	(21-ből)	(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
Egyszerre 5 elhagyása	0 db	0 db	0 db



# Összegzés



- 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.
- Megvizsgáltuk hány redundáns megkötés található a feladatokban.

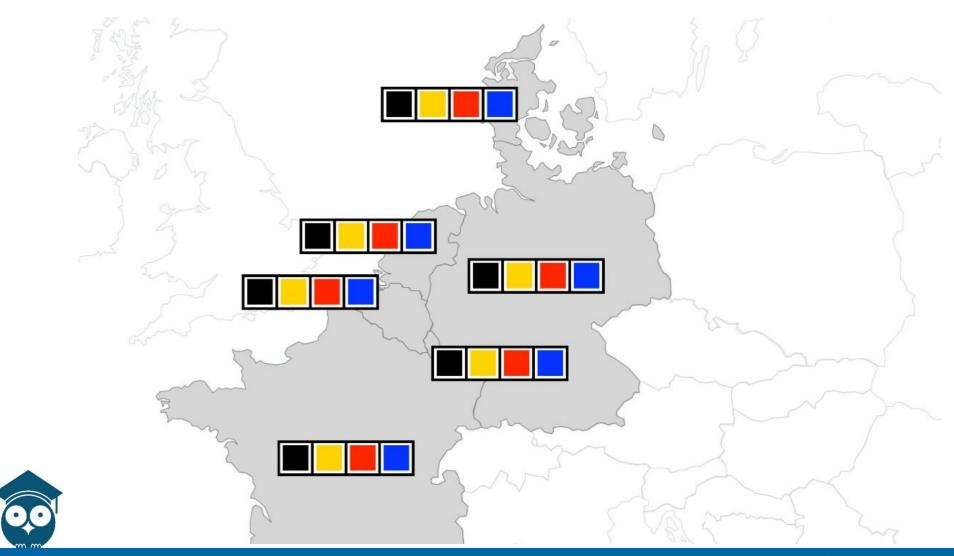


# Köszönjük a figyelmet!



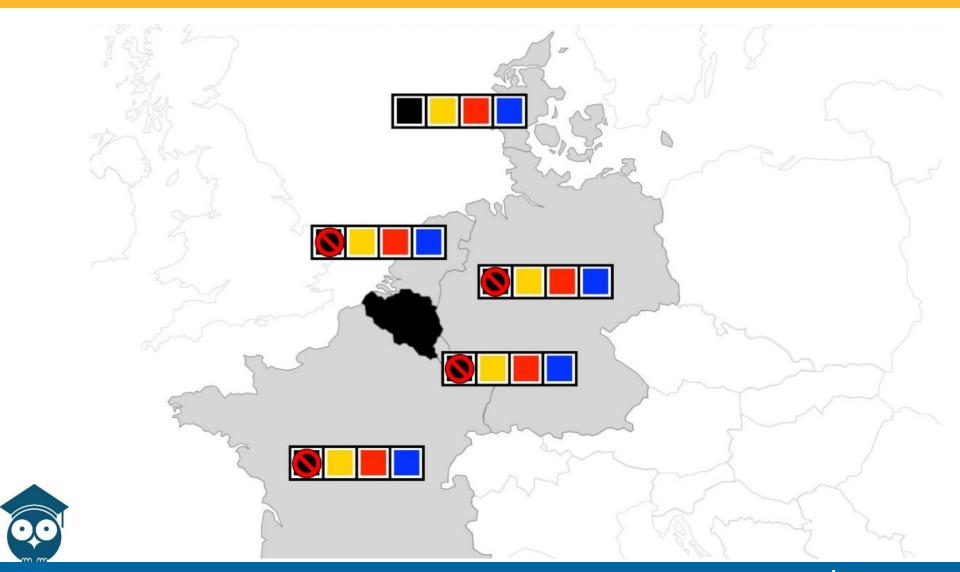
# Propagáció





# Propagáció





# Propagáció



