## Korlátprogramozás bevezető

A feladatunkat úgynevezett korlátprogramozással oldottuk meg. Ennek a módszernek a legkorábbi verziója – a Sketchpad – 1963-ra tehető és [Ivan Sutherland](https://en.wikipedia.org/wiki/Ivan_Sutherland) nevéhez fűződik. Az 1980-as évektől egyre keresettebb lett, és mivel a logikai programozást szerették volna kiterjeszteni, így sok helyen korlát-logikai programozásként hivatkoznak rá. Az első praktikus verziókat – amiket üzleti célokra alkottak és már eladásra is bocsájtottak ­– az 1990-es években készítették.

A projektünkben a MiniZinc nevű grafikus szerkesztő programot használtuk – maga nyelvet is így hívják– , aminek a fordítója az mzn2fzn, amely FlatZinc-re fordítja a szövegünket. Ezt a fájlt adja tovább a megoldónak, ami végül kiadja a megoldást. Legtöbbször ebből mi a Gecode-ot használtuk. A fő alkotója Christian Schulte volt, de számos más kutató és programozó is dolgozott rajta. A munkát 2002-ben kezdték meg, 2005 decemberében adták ki az első verziót, és onnantól kezdve több évben is aranyérmes lett a kategóriájában. Ezen kívül több megoldót is mellékeltek a programhoz, ilyen például a Gurobi, a Chuffed, amiknek speciel ugyanaz a céljuk és a működésük végeredménye is. Tehát a MiniZinc önmagában nem tudná elvégezni a rá bízott munkát, csak segédprogramok segítségével képes erre, ám a feladatokat automatikusan adja ki a részegységeknek.

A megoldó működését legkönnyebben egy feladaton keresztül lehet szemléltetni. A feladványt a szakirodalomban „négyszín tételként” szokták említeni. Adottak bizonyos országok és ezeket úgy kell kiszínezni adott számú színnel, hogy a szomszédos területek ne legyenek azonosak.

A modell elején megadjuk hány színt szeretnénk használni és külön megjegyezzük, hogy a megoldás során is ezeket vegyék fel az egységek. A szomszédsági mátrixot – ami egyben a feltételek listája is itt esetünkben – kikötésekben adjuk meg. A legprimitívebb mód erre az, hogy leírjuk páronként a szomszédokat, amiknek más és más értéket kell felvenniük, azaz két egymás mellett lévő rész nem lehet egyenlő tulajdonságú.



Térkép színezés

# MiniZinc modell

**%adatok**

**int: maxSzin = 4;**

**%megadjuk milyen értékeket vehetnek fel**

**var 1..maxSzin: Belgium;**

**var 1..maxSzin: Dánia;**

**var 1..maxSzin: Franciaország;**

MiniZinc logó

**var 1..maxSzin: Németország;**

**var 1..maxSzin: Hollandia;**

**var 1..maxSzin: Luxemburg;**

**%kikötések**

**constraint Belgium != Franciaország;**

**constraint Belgium != Németország;**

**constraint Belgium != Hollandia;**

**constraint Belgium != Luxemburg;**

**constraint Dánia != Németország;**

**constraint Franciaország != Németország;**

Gecode logó

**constraint Franciaország != Luxemburg;**

**constraint Németország != Hollandia;**

**constraint Németország != Luxemburg;**

**solve satisfy;**

A megoldó úgy dolgozik, hogy először értéket ad egy tetszőlegesen kiválasztott országnak. A szomszédos területeknél kizárja azt a lehetőséget, amit már felhasználtunk, így már csak a megmaradt színekből választhat. Sorban megy végig a kód az egységeken és amellett, hogy ellenőrizzük nem szegtük-e meg valamelyik korlátozást, a változók értékkészletét csökkentjük, és ezt addig ismételjük amíg még ezt meg tudjuk tenni. Ezt a műveletet propagációnak nevezzük. Természetesen nem csak egy megoldásunk lesz a végén, hiszen, ha például a piros és a fekete színt felcseréljük egymással, akkor az már másik megoldásnak fog számítani. A korlátprogramozásnak ez egy hatalmas előnye, hogy az összes lehetséges megoldást kiadja nekünk, nem pedig csak egyet.

<http://www.constraint.org/en/history.html>

<http://www.gecode.org/presentations/Gecode%202011.pdf>

<https://en.wikipedia.org/wiki/Gecode>