https://doi.org/10.36007/4782.2023.195

MATEMATIKAI PARAMETRIZÁLT FELADATLAPOK ÉS MEGOLDÓKULCS KÉSZÍTÉSE LaTeX-BEN

Miklós VONTSZEMŰ [1]

Abstrakt

This paper describes how to create mathematical problems with variable parameters in a Latex environment, how to edit such problems, how to transform them into a problem sheet, and how to generate a solution key for the generated problem sheet. The aim was to create generic mathematical worksheets, regardless of subject and difficulty, thus facilitating the work of teachers when preparing assessments or exams. In this article, the creation of a generic worksheet is presented. It also discusses possible problems that may arise when parameterising different tasks. Tasks with a higher capacity and requiring more complex computation can be computed, so that it is feasible to generate a solution key for most of the topics. Thus, a framework is produced which can serve as a good basis for a user to create the problem sheets described in the title.

Keywords: *mathematics, problems, problem sheet, problem solving, LaTeX*

Bevezetés

A cikk kialakulásához az a gondolat vezetett, hogyan lehetne megkönnyíteni matematikai feladatlapok, tesztlapok, vagy vizsgák elkészítését, segítve ezzel a tanárok munkáját. Természetesen nem sablonos tesztlapokról van szó, hanem változókat tartalmazó, automatikusan generálódó feladatok soráról. Míg online feladatoknál ez egyénenként is változhat, a klasszikus papírra írt vizsgáknál általában egy vagy kétféle feladatlapot használnak. Ezért fontos szempont a felhasználás, és gondolnunk kell arra, hogy nem minden tanárnak lenne lehetősége számítógépes termekben tesztet íratni. Ez a törekvés elsősorban azoknak a tanároknak a munkáját hivatott megkönnyíteni, akik hagyományos módon készítenek feladatlapokat, legyen szó írásbeliről vagy vizsgáról. Tanároknak bizonyára ismerős lehet az az érzés, amikor újra el kell készíteni egy feladatlapot, lehetőleg ne pont azokkal a feladatokkal, amik tavaly, vagy az elmúlt években voltak. Még ha van is elkészített feladatlap az adott tárgyhoz vagy témakörhöz – ahogy az a felkészült tanárokra jellemző –, időbe telik kitalálni, átgondolni, átírni egy újabb feladatlapot. Itt jön képbe az az ötlet, hogy ezt ne ilyen módon, word-ben átírva kelljen megtenni, hanem rövidebb időn belül álljon rendelkezésre egy eltérő feladatlap. Ennek megvalósítására a LaTeX környezetre esett a választás. A választás okai a következők voltak: a keretet egyszer kell elkészíteni, az eredményt pdf-ben kapjuk meg, könnyen használhatóak változók (mivel programozás alapú szerkesztő), a kész dokumentum nem igényel további programozási készségeket, egyszerűen generálható, nem utolsó sorban műveletek is végezhetőek, így megoldókulcs generálása is kivitelezhető. A megoldókulcs elsősorban a végeredmény kiszámolását jelenti, de megfelelő kialakítással sok esetben a feladatok részeredményei is generálhatóak a felhasználó igénye szerint

[1] Mgr. Vontszemű Miklós, Matematika Tanszék, SJE Komárom, vontszemum@ujs.sk

Kivitelezés

A bevezetésben említett okokból a feladatlap LaTeX-ben (később TeX) kerül kivitelezésre. A matematikai témakörben különösen előnyös a TeX környezet, tekintettel a képletekre és a matematikai jelek írásának módjára, például a word-el szemben. Ez talán nem is csoda, mivel a fejlesztők maguk is matematikusok és informatikusok [2], így fontos szempont volt a program tervezésénél a matematika kezelése. Ebből is látszik, hogy ez egy jó platform akár arra is, hogy manuálisan készítsünk feladatlapot. Előnye az előbb említett matematika könnyebb kezelése, a hátránya pedig az, hogy bele kell tanulni. Jó hír azonban, hogy az alapvető dokumentum létrehozása és formázása nagyon könnyen és rövid idő alatt elsajátítható [5], ez saját tapasztalatból is elmondható, mivel az egyetemünkön oktatom a TeX-et, mint tantárgyat. A kivitelezés három főbb lépésből tevődik össze:

1. Változók definiálása és használata
2. Létrehozott feladatok kezelése
3. Megoldókulcs generálása

Változók definiálása és használata

Lépjünk is tovább az alapoktól, és nézzük az első lépést ahhoz, hogy parametrizált feladatlapokat tudjunk létrehozni: a változók használatát. Ehhez egy új csomagra lesz szükség, mégpedig a pgf -re (esetlegesen kiegészülve a pgfmath csomaggal).

```
\usepackage{pgf}
```

A változó generálása \pgfmathsetmacro{ }{ } paranccsal történik, és az első zárójelbe a változó neve kerül (parancsként), a másodikba pedig az értéke, jelen esetben egy random intervallum, amelyből véletlenszerűen generálódik majd a változónk, például:

```
\pgfmathsetmacro{\a}{int(random(1,9))}
```

Ez a sor létrehoz egy \a változót, amely értéke 1 és 9 közül kerül kiválasztásra. Lehetőségünk van több változót is felvenni, így egyszerűen parametrizálhatjuk majd feladatainkat, ami annyit tesz, hogy a feladatokban szereplő értékeket immár nem fixen beírjuk, hanem egy intervallumból választjuk ki. A pgf csomag azért is előnyös, mert tartalmaz grafikai részt is, ami egy további tikz nevű csomaggal kibővítve alkalmas lehet függvényeket tartalmazó, illetve geometriai feladatok generálására is, így a változók használata a továbbiakban nem lehet akadály.

Itt adekvát kitérni a parametrizálás nehézségeire és a megfelelő intervallumok megválasztásának fontosságára. Kijelenthető, hogy különböző okokból nem minden feladat parametrizálhtó könnyedén, és valószínűsíthető, hogy akad olyan, ami egyáltalán nem. Az alapműveleteket igénylő feladatok egyszerűen létrehozhatóak. Egy dologra azonban mindig oda kell figyelni, ez pedig a megfelelő intervallum kiválasztása. Akár alapiskolás feladatokról van szó, akár nehezebb szöveges feladatokról, át kell gondolni a feladatok létrehozásánál, milyen megoldásokat szeretnénk kapni. Ebből kell visszavezetni az egyes paraméterek megfelelő intervallumát. Például az alapvető műveleteknél, az összeadásnál és szorzásnál nem okoz gondot a változók kezelése, kivonásnál és osztásnál viszont már előfordulhat. Gondoljunk csak arra, hogy ha kisebb számból vonunk ki nagyobbat, már negatívat kapunk, illetve két véletlen szám közül az egyik nem mindig osztja a másikat maradék nélkül. Tehát figyelembe kell venni, hogy mit szeretnénk kapni. Ha kivonásnál pozitív eredményt szeretnénk, több megoldás is létezik, például az kisebbítendő változó minimum értékét nagyobbra állítjuk, mint a kivonandó maximum értékét. Alternatív lehetőség, hogy a kivonandó változót véletlenszerűen választjuk, a kisebbítendő értékét pedig úgy definiáljuk, hogy a kivonandóéhoz hozzáadunk egy változót. Mindkét esetben elértük a célt. Osztásnál, ha egészeket szeretnénk kapni megoldásul,

használhatjuk a visszafelé haladás elvet: először a megoldást választjuk ki tetszőleges intervallumból, majd ezt megszorozzuk egy másik tetszőleges értékkel, ami az osztó lesz, az így kapott érték pedig az osztandó. Így minden esetben egész eredményeket kapnánk. Ezek a példák egyszerűek ugyan, de jól szemléltetik a változók használatának kezdeti nehézségeit, amire fontos odafigyelni. Minden feladatnál érdemes számba venni az eshetőségeket és a lehetséges eredményeket.
Vegyünk példának egy valószínűségszámítást bevezető szöveges feladatot:

```
Legyen egy dobozban \a fehér, \b piros és \c kék golyó. Egy
golyót találomra kiveszünk, határozzuk meg annak a
valószínűségét, hogy a kivett golyó:
a) fehér
b) piros
c) kék
```

Ahogy az látható, a változókat rendre a, b és c-vel lettek jelölve, és a korábban említett módon,

```
\pgfmathsetmacro{\a}{int(random(1,9))}
\pgfmathsetmacro{\b}{int(random(1,9))}
\pgfmathsetmacro{\c}{int(random(1,9))}
```

értelemszerűen a szöveg előtt lettek generálva. Mivel ez a feladat egy bevezető, szeretnénk, ha a kapott eredmények százalékban egészek lennének. Ekkor a harmadik változót nem generáljuk, hanem kiszámoljuk, úgy, hogy a c legyen 20-(a+b). Így elérve, hogy a három érték összege 20 legyen, ezáltal biztosítva a kerek százalékos eredményt.
Ha egy feladatnál a változó kiszámítása nem mindig egész, de csak az egész részét szeretnénk használni, akkor a \pgfmathtruncatemacro{}{} parancsot alkalmazzuk:

```
\pgfmathtruncatemacro{\x}{1.7}
```

Az így generált x érték 1-et vesz fel, tehát látható, hogy nem kerekítés történik, csupán levágja a tizedesrészt (a kerekítésre más megoldás van), továbbá a tizedes értéket ponttal kell jelezni, mert vesszőnél hibát jelezne.
Ahhoz, hogy minden újra generálásnál ténylegesen új változók kerüljenek megjelenítésre, használjuk a:

```
\pgfmathsetseed{\number\pdfrandomseed}
```

parancsot, így minden generálásnál garantáltak az új értékek.
A \pgffor csomag a \pgf bővítménye és lehetővé teszi a ciklus használatát változókkal:

```
foreach \x in {1,...,5}{}
```

A ciklus annyit tesz, a beállított értékig ismételten végrehajtja a feladatot. Ebben az esetben az x 1-től 5-ig, vagyis ötször hajtja végre a második zárójelbe írt parancsot. Ez a funkció hasznos, például egyszerű feladatoknál, ahol egyszerre többet szeretnénk generálni.
Amint az már említésre került, akadnak feladatok, amik nem parametrizálhatóak, ilyen például az ötös vagy hatos lottó nyerési esélyének összehasonlítása, vagy hasonló feladatok, amelyek eredménye fix érték. Viszont rengeteg olyan feladat található az adott témakörökben, amelyek viszont parametrizálhatóak ezekre ajánlott összpontosítani. A legtöbb matematikai témakörben fellelhetőek típusfeladatok. Az ilyen típusfeladatok általában parametrizálhatóak, mivel egy-egy változó megváltoztatása esetén továbbra is ugyan azt a megoldási mintát követik. Pár példa típusfeladatokra: másodfokú egyenlet, permutáció, négyszög területszámítása stb.
Előfordulhat még, hogy egyazon feladatban túl sok változót szeretnénk parametrizálni. Ezt javasolt elkerülni, és mindig csak a minimálisan szükséges változókkal dolgozni. Ennek a

megoldókulcs generálásánál is fontos szerepe lesz.
Mindezt összefoglalva az alábbiakra kell figyelni a változók megválasztásánál:

- olyan feladatoknál alkalmazni, ahol van értelme
- figyelembe kell venni, hogy milyen tartományban szeretnénk eredményt kapni
- a változókat akár az eredményből visszafejtve megválasztani
- ne legyen logikai hiba (pl. 0-val való osztás stb.)
- ne álljanak elő szélsőséges feladatok (gondolva itt a túl egyszerű és túl nehéz feladatra)
- vizsgánál fontos szempont, akár két eltérő feladatlapnál is, hogy ne legyen nehézségi különbség.

Létrehozott feladatok kezelése

Az előző részben taglaltak alapján elkészíthetőek különböző feladatok, feladattípusok. Egy feladatlap általában több feladatból áll össze. Egy új feladatlap szerkesztésénél nem mindig ugyanazokat a feladatokat szeretnénk látni, ezért szükséges ezeket a feladatokat rendezni és kezelni. Az alábbi megoldás azért is jó, mert a kész feladatok nem vesznek el, mégis a feladatlap mindig tetszőlegesen összeszerkeszthető lesz.
Ha az egyes feladatokat parametrizáltuk, még mindig előfordulhat, hogy jobbnak látnánk egyazon feladattípuson belül más feladatot adni, vagy változtatni a sorrenden, esetleg tetszőleges feladatokból összerakni a feladatlapot. A tetszőleges összerakás érdekében a TeX azon funkcióját használjuk ki, hogy meghívhatóak különböző dokumentumok. Így minden feladat külön fájlba kerül, és ha megfelelően elnevezzük ezeket, akkor könnyedén kiválaszthatjuk a kívánt feladatokat és azok sorrendjét. A változók ezekben a külön fájlokban is generálhatók, így nem kell a fő dokumentumban deklarálnunk a változókat. A méret miatt sem kell aggódni, mert a TeX fájlok általában nagyon kevés helyet foglalnak. Nincs más dolgunk, mint egy új fájlban létrehozni a feladatot. Ez tartalmazza a korábban leírt változók generálását, és magát a szöveget. Fontos, hogy az elnevezés lehet tetszőleges magyar szó is, de ne tartalmazzon ékezetet, különben a fordító nem tudja értelmezni. Célszerű a feladatra utaló elnevezést adni a fájlnak (pl.: 3golyo.tex), és abba a mappába menteni, ahol az eredeti fájlunk van. Az előbbi azért lesz előnyös, ha később létrehozunk több feladatot, könnyen eligazodjunk közöttük. Az utóbbi pedig azért, hogy a fordító gond nélkül megtalálja. Ekkor az

```
\input{3golyo.tex}
```

paranccsal tetszőleges sorrendben meghívhatóak az adott feladatok a fő fájlban. Ezeket a feladatokat akár egy számozott listába is tehetjük. Ez mellett természetesen formázhatjuk az adott feladatlapot, adhatunk címet, írhatunk rá dátumot, elhelyezhetünk a tanuló nevére szolgáló vonalat, akár pontozó táblázatot stb. Ezen megoldás további előnye, hogy ha nem szükséges módosítás, akkor továbbra is működhet úgy, mint egy fix feladatlap változókkal, de ha igény lenne a sorrendváltozásra, vagy feladatcserére, így rövid időn belül megoldható.
Nem utolsó sorban a sorrend változtatást, illetve a feladatok kiválasztását a létrehozott feladatok halmazából is rábízhatjuk a TeX-re. Ennek kivitelezése további gondolkodást igényel, de például egy for ciklus és if feltételek használatával például megvalósítható.

Megoldókulcs generálása

A parametritzálás és a feladatok különböző sorrendje miatt kifejezetten hasznos a javító tanár számára egy megoldókulcs ezekhez a feladatlapokhoz. Ennek a kivitelezése két fő lépésből állt:

1. az adott feladatok tényleges kiszámítása
2. a megoldókulcs generálása

Az első részben nem a legismertebb oldalát használjuk ki a TeX-nek. Ez nem más, mint a számolás. Nem csak megjeleníteni tudunk feladatokat, hanem ki is tudjuk számoltatni, akár a bonyolultabb feladatokat is. Persze az ilyen esetekben új csomagra is szükség lesz. Az alvető számítások elvégezhetőek a pgf csomaggal is, mint például az összeadás, kivonás, szorzás, osztás. Viszont az összetettebb műveleteknél, főleg, amelyek nagyobb kapacitásúak, bizony már korlátokba ütközhetünk.
Mivel a feladatlapok nem kizárólag egy adott matematikai témakörben készülnének, hanem tetszőleges témaköröket lefedve, gondolni kell a különböző esetlegesen felmerülő műveletekre. Ezeket a műveleteket a megoldókulcsnál használnunk kell majd. Szükségünk lehet kerekítésre, gyökszámításra, hatványozásra és így tovább. Itt merült fel először az fp és xfp csomagok

```
\usepackage{xfp}
```

használata, melyek valamennyire képesek voltak ezt kezelni A kerekítést például jobban is kezeli, mint a pgf csomag:

```
\FPeval{\m}{round(\x,2)}
```

Itt a parancs \FPeval-ra változik a megadás viszont hasonlóan a pgf-hez történik. Mindezeket viszont csak egy bizonyos nagyságrendig képes végrehajtani, ugyanis bitkorlát miatt a hozzávetőlegesen {-16000,16000} intervallumon kívül eső értékekkel nem képes dolgozni. Ez gondot jelentett, például az általam főként kidolgozott feladatoknál is, mivel a valószínűségszámítás témakörében faktoriális és binomiális számításokat is igényelnek. Egy faktoriális számításnál derült ki, hogy a 8! már problémát jelent ennek a csomagnak, míg kisebb értékekkel megfelelően számolt. Így további keresés után meglett a megoldás, az xintexpr csomag, melyhez egy terjedelmes használati útmutató is fellelhető az interneten [6] Ennek a csomagnak már nem jelent gondot a nagyobb számítási kapacitás, a 100! kiszámításával és megjelenítésével is könnyedén boldogult. Ez mellett szinte az összes felmerülhető számításra kiterjed, a trigonometria függvényektől az integrálszámításig. Ezáltal további távlatok nyíltak meg ebben a koncepcióban, mivel a feladatok kiszámítása már nem lehet akadály, legalábbis nagyon kicsi rá az esély.
A feladatok kiszámítását célszerű még a megoldókulcsba íratás előtt elvégezni. Ehhez tökéletesen megfelel a már említett \pgfmathsetmacro parancs, ahol az első kapcsoszárójelbe egy új változót írunk, melynek az értékét a második kapcsoszárójel adja. A második zárójelbe kerül a számítás, ahol a változókat „backslash”-el jelöljük, ahogyan azt korábban definiáltuk:

```
\pgfmathsetmacro{\ossz}{\a+\b+\c}
```

Ekkor az új változó az \ossz az a \a, \b, illetve \c változóink összegét veszik fel. Természetesen használhatóak itt különböző műveletek. Ha szükséges a számítási lépéseket külön változókban is kiszámolhatjuk és a megoldásnál használhatjuk az újonnan definiált részeredmények változóit. Az efféle definiálás akkor ajánlott, ha a létrehozott feladatnál nem csak a végeredmény számít, hanem a részeredmények is. Ekkor a megoldókulcsban a részeredmények is könnyedén kiírhatóak a lentebb leírt módon.
A második „probléma” a megoldókulcs generálása volt. Pontosabban annak a helye, hogy az a feladatlap végén szerepeljen, mindazt figyelembevéve, amit a véletlen sorrendről szóló fejezetben esett szó. Tehát függetlenül a feladatok sorrendjétől és azok változóitól, ugyanilyen sorrendben szerepeljenek a kiszámított eredmények egy új hozzáadott lapon. Szerencsére ezt is sikerült megoldani. A nehézség az volt, hogy mivel egyesével kerülnek a létrehozott feladatok meghívásra, ezért mindenképpen abban a fájlban kell kiszámítani az adott feladatot. Viszont nem kerülhetnek kiíratásra, mert úgy már a feladatlap nem használható számonkérésre, ha minden egyes feladat után szerepel a megoldás is. Itt jött képbe a késleltetés, és egy külső ideiglenes fájl létrehozása.

```
\newwrite\tempfile
```

Ez a parancs létrehoz egy ideiglenes fájt, amit tetszés szerint elnevezhetünk, majd ezt a fájlt a dokumentumon belül meg kell nyitni, ilyenkor írhatunk bele, majd a végén be kell zárni:

```
\immediate\openout\tempfile=megoldas.tex
\immediate\write\tempfile{Megoldasok}
\immediate\closeout\tempfile
```

Amint látható, az openout nyitja, a closeout zárja, közte a write ír a fájlba. A parancsok előtt az inmediate, (magyarra fordítva: azonnal) a nevéből is adódóan azonnal hajtja végre ezeket az utasítások, egyébként előfordulhat, hogy ezek késleltetve történnének, ami akár gondot is okozhatna. Azért is megfelelő ez a megoldás, mert a meghívott fájlban is gond nélkül írhatunk, így az ideiglenes megoldások nevű fájlunkba, amit így a végén meghívva egy input paranccsal mindig az utolsó oldalon jeleníthetünk meg. Javasolt még a:

```
\usepackage{morewrites}
```

csomag használata, mivel ez lehetővé teszi, hogy ilyen módon hosszabb szöveget is tudjunk az ideiglenes fájlba írni.

Ennek a megoldásnak vannak korlátjai, például a megoldásokat tartalmazó lapon nem tudunk mindent beállítani, például a számozott listát, vagy az elrendezést. Mindazonáltal ez egy működő keret és tesztelés alapján is elmondható, hogy megfelelően generál véletlen értékeket, és a meghívott sorrendnek megfelelően a feladatlap végén helyesen írja ki a megoldókulcsot is.

Összegzés

Így nézne ki egy általános feladatsor létrehozása, ez lenne a fő dokumentum:

```
\documentclass[12pt]{article}
\usepackage{pgf}                              % változókhoz
\usepackage{fp}                               % egyszerű számítások
\pgfmathsetseed{\number\pdfrandomseed}        % randomizálás
\usepackage{xintexpr}                         % bonyolult számítások
\newwrite\tempfile                            % megoldás
\usepackage{morewrites}                       % bővebb írás
\begin{document}
\immediate\openout\tempfile=megoldas.tex % megoldás nyitása
\immediate\write\tempfile{}                   % megoldásba írás
\input{feladat1.tex}                          % feladat meghívása
\input{feladat2.tex}                          % feladat meghívása
\input{feladat3.tex}                          % feladat meghívása
\immediate\closeout\tempfile                  % megoldás zárása
\newpage                                      % új oldal
\input{megoldas.tex}                          % megoldás kiírása
\end{document}
```

amelyben jelenleg 3 feladat lett meghívva. Az első feladat szemléltetve:

```
\pgfmathsetmacro{\x}{int(random(5,15))}   % változó
\pgfmathsetmacro{\y}{int(random(3,4))}    % változó
Az A és B helységet \x \ km hosszú telefonvezeték köti össze. A
vezetékek valahol meghibásodik. A meghibásodás valószínűsége
egyenletes eloszlású az egész vonalon. Mekkora a valószínűsége,
hogy a hiba A-tol mérve \y \ km-nél távolabb következett be?
\FPeval{\m}{round((\x-\y)/\x*100,2)}      % eredmény kiszámítása
```

```
\immediate\write\tempfile{$\m $\%}        % eredmény beírása
```

Az alap parancsokon túl számos parancs és beállítás megtalálható az overleaf oldalán [4] és a LATEX nem túl röviden dokumentumban magyarul is [1].

Befejezés

A kutatás célja egy általános, változók által generált matematikai feladatokból álló feladatlap és egy hozzá tartozó megoldókulcs generálása volt. Mindezt sikerült kivitelezni LaTeX környezetben, ahol egy fő fájlban meghívhatóak az elkészített feladatok tetszőleges mennyiségben és sorrendben, amihez egy generált megoldókulcs is tartozik. A megoldókulcs opcionális, apró módosítással kivehető. Bármikor létrehozható új feladat is, vagy módosítható egy már létrehozott. Az elkészített feladatbankból a fő fájlban pár lépésben egy teljesen új feladatlap hozható létre, a készítő kedve szerint. Annak, aki nem zárkózik el a LaTeX használatától, nagy segítség lehet ez környezet és az itt bemutatott struktúra a feladatlapok elkészítésében. Erre az alapra épülő további érdekes téma lehet a különböző témakörök és feladattípusok elemzése, és akár egy komplex tananyag készítése.

Irodalomjegyzék

[1] Csárdi, G. (1998) *LATEX nem túl röviden*, https://math.bme.hu/latex/dl/latex69.pdf
[2] Lamport, L. (1994) *A Document Preparation System – LATEX*, ISBN 9780201529838
[3] LaTeX original site https://www.latex-project.org//
[4] Overleaf, Online LaTeX Editor
https://www.overleaf.com/learn/latex/Mathematical_expressions
[5] Udvaros, J. (2023) *Bevezetés a LATEXbe*, ISBN 9788081224591
[6] Xint macro https://ctan.math.washington.edu/tex-archive/macros/generic/xint/xint.pd