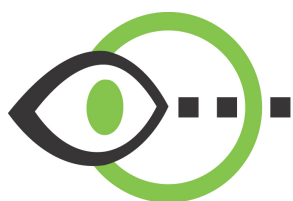


Akadálymentes dokumentum szerkesztése matematikai tartalmakhoz
– Alapszintű kiadvány

Készült az
Informatika a Látássérültekért Alapítvány
és a
„Grapheel Limited”
közreműködésével.

A kiadvány az
Emberi Erőforrások Minisztériuma
és a
Fogyatékos Személyek Esélyegyenlőségéért Közhasznú Nonprofit Kft.
támogatásával készült.

2018



„Informatika a
látássérültekért”
Alapítvány



EMBERI ERŐFORRÁSOK
MINISZTERIUMA



Fogyatékos Személyek
Esélyegyenlőségéért
Közhasznú Nonprofit Kft.

Előszó

A matematikai ismeretek elsajátításához elengedhetetlen az önálló olvasás és írás képessége, ezért az oktatást és a tanulást segítő szakirodalom akadálymentessége kiemelkedően fontos szempont a tananyagok elkészítése és közlése során. A matematikai szaknyelv a vizualításra nagymértékben támaszkodó jelölésrendszere miatt a látássérült emberek gyakran nem tudnak hozzáférni az írásos tananyagokhoz. A tanár és a diák közötti információcserére a saját, nem szabványos jelölésrendszer kialakítása a leggyakoribb módszer. Ez azonban sok hátránnyal jár, például a tanárnak minden látássérült diáknak külön-külön meg kell tanítania az egyedi jelöléseket. Az Informatika a Látássérültekért Alapítvány tevékenységének köszönhetően a matematikai olvasást magyar nyelven elérhetővé tévő eszközök száma jelentősen nőtt az utóbbi időben. A JAWS for Windows a 17.0, honosított verziótól kezdve támogatja a HTML dokumentumokba ágyazott MathML szabvány szerint készült tartalmak verbalizálását, illetve ennek szabványos matematikai Braille változatát is. A DEX a 2.7-es verziótól szintén támogatja a MathML szabvány alapján készült dokumentumok Braille konverzióját, valamint a LaTeX dokumentumok meghangosított formájú változatának előállítását.

(A JAWS for Windows programról a <http://www.infoalap.hu/jaws/>, a DEX-ről pedig a <http://www.infoalap.hu/dex/> oldalon található részletes információ.)

A fejlesztések következménye, hogy a szakirodalom önálló olvasása lehetővé vált a vak és gyengénlátó emberek számára. Azonban a célközönség körében az olvasást lehetővé tévő standard számítógépes leíró nyelvek elterjedéséhez, illetve akadálymentes szakirodalom készítéséhez elengedhetetlen az írásképesség fejlesztése is.

Az Informatika a Látássérültekért Alapítvány azt a célt tűzte ki, hogy támogassa az Oktatókutató és Fejlesztő Intézet által kibocsátott Nemzeti Alaptantervben foglalt matematikai tárgyú témakörök elsajátítását a LaTeX leírónyelv széles körű elterjedésének, mindennapos használatának elősegítésével. Jelen kiadvány a szabványos és hozzáférhető matematikai szövegszerkesztés alapismereteivel kívánja megismertetni az alábbi célcsoportok tagjait, ezzel is hozzájárulva a gördülékenyebb adaptációs folyamathoz. Bízunk benne, hogy

- a tananyag-akadálymentesítéssel foglalkozó módszertani központok és csoportok munkatársai
- a speciális iskolákban tanuló illetve az integrált oktatásban résztvevő látássérült diákok és tanáraik
- a közoktatásból már kikerült, érdeklődő látássérült emberek és segítők

egyaránt hasznos ismereteket sajátíthatnak el, melyek megkönnyítik a matematikai szakirodalom írását, szerkesztését illetve olvasását.

A dokumentum négy részre tagolódik. Az első szekció a LaTeX használatához szükséges általános bevezetést tartalmazza. Az ezt követő részben a LaTeX a Nemzeti Alaptanterv alapján készült, matematikai témakörök szerinti használata kerül bemutatásra. A harmadik és negyedik részben további hasznos, haladó információk ismertetésére kerül sor, melyek a kiadvány készítéséhez használt forrásokkal és további tananyagokkal egészülnek ki.

Tartalomjegyzék

Előszó	i
I. A matematikai szövegszerkesztés alapismeretei	1
Bevezető	2
1. A LaTeX bemutatása	3
1.1. Mi is az a LaTeX?	3
1.2. A LaTeX működése: Utasítások és paraméterek	4
1.3. „Hozzávalók” a LaTeX-hez	4
1.3.1. A dokumentum létrehozása	4
1.3.2. A dokumentum felépítése	5
A dokumentumosztály	5
A címblokk	6
A LaTeX csomagok használata	6
A tartalom szerkezete	6
Láthatatlan tinta – a felhasználói megjegyzések	7
1.3.3. LaTeX-ből PDF – avagy, hogyan tegyük a forrásdokumentumot olvashatóvá?	7
Grafikus felület használatával	8
Parancssor használatával	8
1.4. A leíró nyelv előnyei	8
1.5. A leíró nyelv hátrányai	9
1.6. Magyar nyelvű LaTeX dokumentum előállítás	10
2. Általános célú tartalom szerkesztése	12
II. Matematikai tartalom szerkesztése	17
A matematikai fejezetek felépítése	18
Az utasítások és a paraméterek osztályozása	18
3. Halmazok és kombinatorika	19
3.1. Jellemzők	19
3.2. Leíró utasítások	20
3.3. Példák	21
3.3.1. Vizuális megjelenés	21
3.3.2. LaTeX forrás	22

4. Számтан és algebra	23
4.1. Jellemzők	23
4.2. Leíró utasítások	23
4.3. Példák	26
4.3.1. Vizuális megjelenés	26
4.3.2. LaTeX forrás	26
5. Függvények: az analízis elemei	28
5.1. Jellemzők	28
5.2. Leíró utasítások	28
5.3. Példák	30
5.3.1. Vizuális megjelenés	30
5.3.2. LaTeX forrás	30
6. Geometria	31
6.1. Jellemzők	31
6.2. Leíró utasítások	31
6.3. Példák	33
6.3.1. Vizuális megjelenés	33
6.3.2. LaTeX forrás	34
7. Statisztika, valószínűségszámítás és matematikai logika	35
7.1. Jellemzők	35
7.2. Leíró utasítások	35
7.3. Példák	36
7.3.1. Vizuális megjelenés	36
7.3.2. LaTeX forrás	37
8. Egyenletek, egyenletrendszerek, egyenlőtlenségek	38
8.1. Jellemzők	38
8.2. Leíró utasítások	38
8.3. Példák	41
8.3.1. Vizuális megjelenés	41
8.3.2. LaTeX forrás	41
9. Hatvány, gyök, logaritmus	42
9.1. Jellemzők	42
9.2. Leíró utasítások	42
9.3. Példák	43
9.3.1. Vizuális megjelenés	43
9.3.2. LaTeX forrás	43
10. Trigonometria és vektorszámítás	45
10.1. Jellemzők	45
10.2. Leíró utasítások	45
10.3. Példák	46

10.3.1. Vizuális megjelenés	46
10.3.2. LaTeX forrás	46
11.Sorozatok, sorok és integrálok	47
11.1. Jellemzők	47
11.2. Leíró utasítások	48
11.3. Példák	48
11.3.1. Vizuális megjelenés	48
11.3.2. LaTeX forrás	49
12.Gráfok és koordinátageometria	50
12.1. Jellemzők	50
12.2. Leíró utasítások	50
12.3. Példák	52
12.3.1. Vizuális megjelenés	52
12.3.2. LaTeX forrás	53
III. További hasznos tudnivalók	54
A gyakorlati tapasztalatok összegzése	55
13.A tartalom szerkesztésén túl	56
13.1. TeX forrásszerkesztő programok	56
13.2. Tudnivalók a TeX csomagok használatáról	56
13.3. A forrásfordítás technikai és problémái	59
IV. Források és további referenciák	61
Források listája	62
14.Hasznos olvasnivalók	63

I. rész

**A matematikai szövegszerkesztés
alapismeretei**

Bevezető

Az első részben bemutatásra kerülnek a LaTeX (ejtsd LéTech) alapfogalmai, használatának előnyei, hátrányai illetve a dokumentumkészítéshez szükséges elemek. Az alapokkal megismerkedve, a második részben kezdődik el a tantervspecifikus LaTeX-használat ismertetése. A kiadvány írása során a szerző célja az volt, hogy ne „professzori szinten” ossza meg az ismereteket a tankönyv-átíró személyekkel, a tanulni vágyó diákokkal, tanárokkal, családokkal, hanem sokkal inkább egy átlagos, esetleg vak felhasználó szemszögéből fogalmazta meg a tudnivalókat. A kiadvány a végén felsorolt forrásokkal ellentétben, igyekezett könnyen befogadható, követhető ismertetőt közreadni érthető, hétköznapi formában.

1. fejezet

A LaTeX bemutatása

A matematika ismeretek átadása – mint minden olyan tudományos tárgyé, amely ismerethalmazának elsajátításához szükség van a matematikai szaknyelv használatára – nehézségekbe ütközik, amikor a látásra nem hagyatkozható emberek számára szükséges elvont fogalmakat a közmegegyezésen alapuló jelölésrendszer alkalmazásával kívánjuk kommunikálni. A tudományos információközlés eszköze a matematikai nyelv (például egyenletek), illetve ezen nyelv grafikai megjelenítése (például diagramok, gráfok). A képi megjelenítéstől tekintünk el egy pillanatra, és gondoljuk végig, hogy a matematikai nyelv milyen írásjeleket, jelöléseket használ. Gyakran előfordulnak kétszintes kifejezések, ilyenek például a törtek, ahol a nevező a számláló alá kerül. Ezen kívül léteznek még a kitevők illetve az alsó index jelölései. Ezek megjelenítése önmagában is nagy kihívás, mivel a számítógép csak sorról sorra tud haladni és feldolgozni az információt. Míg a táblán vagy a papíron könnyen, első ránézésre láthatóak a síkban szabadon felírt számok, jelölések, addig azok számára, akik Braille-kijelzőre illetve beszédszintetizátoros képernyőolvasó programokra vannak utalva, a kétdimenziós jelöléseket át kell alakítani egydimenziós változatra. Minden jelnek egy sorban, egymást követve kell megjelennie.

1.1. Mi is az a LaTeX?

A LaTeX (ejtsd LéTech) olyan számítógépes leíró nyelv, amely a fenti követelménynek felel meg, lehetővé teszi a kétdimenziós matematikai jelölések egydimenziós leírását. Így törtek esetén a nevező nem a számláló alatt, hanem azt követően jelenik meg, olyan formában, amit a képernyőolvasó program megért, ezáltal lehetővé válik a matematikai anyagok írása és olvasása a vak és gyengénlátó emberek számára is. A LaTeX az összes olyan jel leírását, ami nem a magyar ABC része (például a gyökjel, az integrál jele), a magyar ABC illetve utasítások segítségével biztosítja. Szövegesen, sorról-sorra, a billentyűzet alapbillentyűinek használatával szerkeszthető a kívánt dokumentum, a végeredmény mégis olyan leírás, amely a matematikai jelöléseket szokásos formájukban jeleníti meg. A LaTeX segítségével az egydimenziós bevitellel megadott matematikai kifejezések oly módon jeleníthetők meg, amit a látó emberek a megszokott módon el tudnak olvasni. A végeredmény gyakran PDF formátumban áll elő. A szerkesztés szempontjából lényegtelen, de érdekes információ, hogy a LaTeX a TeX szerkesztő része, és azért ejtendő Tech-nek, mivel az ötlet onnan származik, hogy könnyen és hatékonyan lehessen technikai jellegű dokumentumokat számítógépes környezetben létrehozni, módosítani.

1.2. A LaTeX működése: Utasítások és paraméterek

A LaTeX jellemzője, hogy a billentyűzetről, könnyen elérhető módon lehet technikai jellegű szöveget szerkeszteni anélkül, hogy bármiféle speciális szimbólumot kellene beszúrni, formázni. Minden dokumentum négyféle elemből épülhet fel:

- folyó szöveg
- utasítások
- kötelező paraméterek
- opcionális paraméterek.

A folyó szöveg a szokványos szövegszerkesztőkben (például Microsoft Word) megszokott módon írható, a fordító – az a program, ami a LaTeX forrásból például PDF formátumú, a jelöléseket az elfogadott módon megjelenítő változatot állítja elő – ezt nem alakítja át. A LaTeX utasítások mindig a `\` karakterrel kezdődnek. Az utasításokat nagy körültekintéssel kell leírni, különben a forrás fordításakor hibát jelez a program. Az utasításoknak egy, kettő vagy több paramétere lehet, illetve léteznek paraméter nélküli parancsok is. A paraméter azt mondja meg, hogy az utasítást milyen elemen kell végre hajtani. Vagyis az utasítás-paraméter kapcsolat azt fejezi ki, hogy mivel mit kell csinálni. A kötelező paramétereket `{*}` hullámos vagy más néven kapcsos zárójelek közé kell írni, amint azt a csillag jelölés is mutatja. Az opcionális paraméterek szögletes zárójelek közé írandóak, vagyis `[*]`. Összegezve, egy LaTeX utasítássor az alábbi módon néz ki:

```
\utasítás[opcionális paraméter 1, opcionális paraméter 2, opcionális paraméter 3]
{kötelező paraméter 1}{kötelező paraméter 2}{kötelező paraméter 3}
```

A paraméter nélküli utasítások általában egy darab szimbólumot tartalmaznak, például a görög alfa karaktert. Az egyparaméteres utasítások általában azt határozzák meg, hogy a paraméterrel mit kell csinálni, például aláhúzni vagy kék színben megjeleníteni. A kétparaméteres utasítások azt határozzák meg, hogy például az első paraméter tartalmát a második paraméter tartalma fölé kell írni. Ez lehet egy tört számlálója és nevezője vagy egy matematikai szimbólum fölé írt másik jel.

A `\begin` utasítás valaminek a kezdetét jelzi. A `{document}` paraméter megadása azt jelöli, hogy hol kezdődik a dokumentum. Ennek a párja a `\end` utasítás, ami valaminek a végét jelzi; például az `\end{document}` utasítás a dokumentum végét jelöli.

1.3. „Hozzávalók” a LaTeX-hez

A LaTeX dokumentumok összeállítása előtt érdemes megismerkedni a „hozzávalókkal”. Bár a LaTeX nagyon hasznos eszköz, elsőre kissé bonyolultnak, összetettnek tűnhet. Két fő szempontot szükséges figyelembe venni. Az egyik, hogy milyen módon lehet egy dokumentumot létrehozni, a másik, hogy hogyan szerkeszthető a tartalom. Egy hasonlattal élve, hogyan kell papírt és ceruzát készíteni, illetve hogyan kell azokkal írni.

1.3.1. A dokumentum létrehozása

Gondoljuk végig, mi kell ahhoz, hogy egy matematikai kifejezést vagy akár csak egy mondatot leírjunk! Kezdetnek egy papír és egy toll is elegendő. Először is, vegyük a LaTeX „papírt”. Ez nem

más, mint egy .tex kiterjesztésű egyszerű szövegfájl. Ehhez első lépésként indítsuk el a Jegyzetömb programot. Válasszuk ki a *Fájl* menüpontban a *Mentés* opciót, majd a fájlnev mezőbe írjuk be, hogy „matekfüzet.tex”, majd kattintsunk a *Mentés* gombra. Az utóbbi műveletsor eredményeképp a „papír” most már rendelkezésre áll, de mivel lehet írni rá? Számítógépes „ceruzával vagy tollal”, vagyis egy szövegszerkesztővel. Szövegszerkesztő programból sokféle létezik - ezek közül néhány a kiadvány végén felsorolásra kerül - de egyenlőre használjuk a Windows rendszer részét képező Jegyzetömböt. A szövegszerkesztőben megnyitva a .tex kiterjesztésű fájlt, már meg is van a „papír” és a „toll”, amivel írni lehet. Szükség van még „tintára” is. Hiába tudunk szépen írni, ha a „toll” nem fog. Ez a LaTeX esetében sincs másképpen. Ahhoz, hogy a fájlba írt szöveg értelmezhető, olvasható legyen, fordítóprogramra is szükség van. A LaTeX forrásdokumentumok csak akkor válnak olvashatóvá, ha valamelyik TeX rendszer már telepítve van a számítógépen. Windows rendszeren a **MiKTeX** vagy a **TeX Live** az ajánlott illetve megszokott rendszer. Ezt telepítsük, és egyenlőre ne is foglalkozzunk többet vele! A haladó felhasználók kedvük szerint konfigurálhatják a rendszert, de ezzel egyenlőre ne törődjünk! Összefoglalva, egy .tex kiterjesztésű fájlt nyissunk meg egy szövegszerkesztőben, és máris készen állunk a dokumentum írására, illetve átalakítására a látó emberek számára megszokott formába, amennyiben egy fordítóprogram is rendelkezésre áll.

1.3.2. A dokumentum felépítése

Előfordulhat, hogy hosszabb dokumentumot kívánunk írni, azaz nem csak egy „papírlapra”, hanem egy „egész füzetre” van szükség. A TeX dokumentumokhoz szükséges szerkezet felépítése két részre osztható. Az egyik rész a „füzet borítója”, a másik a „füzet lapjai”. A „füzet lapjaira” írhatjuk a tényleges tartalmat. Ezt a szakaszt, a

```
\begin{document}
tartalom
\end{document}
```

dokumentum kezdő és -záró sorok közé kell írni. Bármi, ami ezen kívül helyezkedik el, a „füzet borítójának” számít, és nem lesz olvasható a dokumentum előállítás után. A „füzet borítójának előlapja” is sok hasznos információt tartalmazhat, amit külön meg kell adni. (A hátlapal nem kell foglalkozni.) A dokumentum kezdősora előtti rész a szövegfájlban több részre osztható. Ilyen például

- a dokumentumosztály – a „füzet” stílusa
- a címblokk – a „füzet” címe és szerzője
- a LaTeX csomagok – a felhasználható eszközök tárháza
- a felhasználó által meghatározott utasítások – ez már a haladó szint
- a dokumentum külalakja, a formázási szabályok, stb.

A dokumentumosztály

Amikor füzetet vásárolunk, eldönthetjük, hogy kis- vagy nagyméretűt szeretnénk, vonalast vagy koc-kásat, esetleg álló vagy fekvő tájolásút. Ennek megfelelően, a fájl elején meg kell határozni, hogy milyen stílusú füzetben dolgozunk. Egyszerűbb dokumentumokhoz a

```
\documentclass{article}
```

utasítássor teljesen megfelelő.

A címblokk

A papíralapú kiadványokhoz hasonlóan, érdemes a dokumentumoknak címet adni, esetleg még a szerzőt és a keletkezési dátumot is feltüntetni. Ezeket a dokumentumkezdő sor előtt lehet megadni. A következő utasítások használhatóak:

```
\title{*}  
\author{*}  
\date{*}
```

A fentiekkel – sorrendben – a dokumentum címe, a szerzője és a keletkezés dátuma adható meg. A dokumentumkezdő sor után, tehát már a tartalmon belül, a címblokk megjelenítéséhez a `\maketitle` utasítás szükséges. Ez az adott információt beszúrja az utasítás helyén.

A LaTeX csomagok használata

Képzeld el, hogy olyan „borítót” használunk, aminek az elején egy „tolltartó” van kialakítva, amibe „színes ceruzákat” tehetünk. Ez lehetővé teszi, hogy bizonyos szövegeket pirossal írjunk, esetleg zölddel húzzunk alá. Ahogy a színes íróeszközökkel, a LaTeX csomagokkal is szebb megjelenésű dokumentumokat lehet létrehozni. Tekintsük úgy, hogy a LaTeX „tolltartója” vásárláskor „egy kék és egy fekete tollat” tartalmaz. A fordítóprogram az ezekkel a „tollakkal” írt parancsokat egyéb eszközök nélkül tudja értelmezni. Azonban ennél sokszor több – esetleg csak általunk vagy ritkán használt – eszközre (például „betűsablon”, „filctoll”) lehet szükség. Ezért fejlesztenek különféle csomagokat, amiket tetszés szerint lehet betölteni és használni. A dokumentumkezdő sor elé, ha a

```
\usepackage{*}
```

utasítás kerül, és a csillag helyére a kívánt csomag nevét írjuk, akkor máris elhelyeztünk egy új „ceruzát” a „tolltartóba”. Ezzel az alap utasításkészlet máris kibővült, most már ezeket is értelmezni tudja a fordító. Az új csomagok új utasításokat tartalmaznak, amelyekkel speciális feladatok hajthatóak végre.

Példák a gyakori csomagokra:

- `\usepackage{amsmath}` – matematikai szimbólumok betöltésére szolgál,
- `\usepackage{babel}` – nyelvi csomagok betöltésére szolgál,
- `\usepackage{graphicx}` – képek kezelését teszi lehetővé.

A csomagok betöltése során opcionálisan egy csomagrész is megadható, például

```
\usepackage[magyar]{babel}
```

, ami a nyelvi csomagok közül kizárólag a magyar nyelvre vonatkozó eszköztárat tölti be. A csomagok kezeléséről részletesebben jelen kiadvány végén lesz szó.

A tartalom szerkezete

A matematikafüzet tipikusan négyzetrácsos, közismertebb nevén kockás. Szöveget inkább vonalas füzetbe szokás írni, hangjegyeket kottafüzet lapjaira, geometriafeladatokat pedig sima oldalakra. Minden tantárgyhoz adott egy füzetstílus, környezet. A dokumentumkezdő- és a dokumentumzáró sor közé írt tartalom a folyó szöveg és a TeX utasítások összessége. Azonban létezik egy érdekes és hasznos lehetőség! Az úgynevezett LaTeX környezetek lehetővé teszik, hogy ne csak a dokumentumosztályban

megadottak alapján legyen formázva a szöveg, lokálisan is módosíthatóak annak tulajdonságai. Vagyis, nem csak a borítón lehet egyetemesen megadni, hogy kockás- vagy vonalas füzetbe történjen az írás, de lapról lapra is meg lehet változtatni ezt. Egy környezet megadhatja, hogy most éppen matematikai szöveget írunk vagy szöveges magyarázatot. A környezetek használatával megadható, hogy táblázat, lista vagy kép következzen. A környezet megfelelő használata határozza meg, hogy milyen jellegű tartalom következik. Ez többek között azért fontos, mert a matematikai képleteket, jeleket általában más betűtípussal, más helyközzel jelenítik meg, mint a folyó szöveget. Matematikai utasításokat kizárólag matematikai környezetben lehet írni, illetve táblázatelemeket csak az arra kijelölt környezetben lehet szerkeszteni. Amennyiben nem így járunk el, a „tinta” nem fog fogni a papíron. Az alábbi lista a leggyakoribb matematikai környezeteket határozza meg:

- `$$` – sorbeli matematikai kifejezés,
- `$$$` – új sorban megjelenő matematikai kifejezés,
- `\[*\]` – úgynevezett „kirakatban” megjelenő matematikai kifejezés (pont mint előbb),
- `\begin{equation}*\end{equation}` – egyenlet írására szolgál.

A táblázatokat, a képeket és a listákat is a megfelelő környezetek közé kell határolni. Ezek közül a leggyakoribbak:

- `\begin{table}*\end{table}` – táblázat,
- `\begin{figure}*\end{figure}` – grafikus elem,
- `\begin{enumerate}*\end{enumerate}` – számozott lista.

A környezetek egy része bemutatásra kerül a következő fejezetben.

Láthatatlan tinta – a felhasználói megjegyzések

Az utasításokon és azok paraméterein kívül arra is van lehetőség, hogy megjegyzéseket írjunk a forrásba anélkül, hogy az a lefordított változatban megjelenne. Minden olyan sor, amely a `%` karakterrel kezdődik, csak a LaTeX forrásban jelenik meg, a PDF fájlban nem. Ez akkor hasznos, amikor saját magunknak kívánunk emlékeztetőt bejegyezni. Például:

```
%Itt kezdődik a példák leírása.
```

1.3.3. LaTeX-ből PDF – avagy, hogyan tegyük a forrásdokumentumot olvashatóvá?

A LaTeX forrásdokumentumok fordítása vizuálisan könnyebben áttekinthető, például PDF formátumra alapesetben nem nehéz feladat. Speciális esetekben – például az angoltól eltérő nyelvek használatakor – bonyolultabb a helyzet, erről a későbbi fejezetekben lesz szó. Tegyük fel, hogy előállítottunk egy minimális tartalmú, de szabványos `.tex` dokumentumot. Ekkor kétféle módon lehet PDF dokumentumot előállítani, grafikus felület használatával vagy parancssori módban.

Grafikus felület használatával

Lehetőségünk van szövegszerkesztő használatával, gombnyomásra PDF-et előállítani a LaTeX forrásból, abban az esetben, ha erre a szövegszerkesztő fel van készítve. A Windows-ban található Jegyzet-tömb önmagában nem képes erre a feladatra. Azonban, amint azt látni fogjuk a harmadik részében ennek a tananyagnak, a WinEdt és több más hasznos LaTeX szövegszerkesztő is rendelkezésünkre áll. Ezek hasonló kényelmi funkciókkal rendelkeznek. Vegyük például ezt a dokumentumot, amit éppen szerkesztünk. A WinEdt 8.0 programot elindítva, és például „matematikahazifeladat.tex” fájlt megnyitva, a teljes forráskód olvasható LaTeX leíró nyelven. Ha ezt PDF formátumra kívánjuk alakítani, a *TeX* menüpont alatt, a *PDF* opciót kell keresnünk. Ennek almenüjében találjuk a *PDF TeXify* lehetőséget. Ezt kiválasztva megkezdődik a LaTeX-PDF konverzió. Láthatjuk, hogy az opcióhoz billentyűparancs, a Ctrl + Shift + P is tartozik.

Parancssor használatával

Tegyük fel, hogy a fent említett, LaTeX fordításra képes szövegszerkesztők nem állnak rendelkezésre. A Jegyzet-tömböt megnyitva, majd a LaTeX forráskódot mentve is megoldható a PDF előállítása. A parancssori ablakot megnyitva, a .tex fájl könyvtárába kell navigálni, majd a *pdflatex fájl-név.tex* parancs segítségével, tehát a pdflatex.exe futtatásával le kell fordítani a forrásfájlt. Lépésről lépésre ez így hajtható végre:

- A Windows + R billentyűparanccsal nyissuk meg a Futtatás ablakot, majd a *cmd* paranccsal indítsuk el a parancssort.
- Navigáljunk a LaTeX forrásfájlt tartalmazó könyvtárba:
(például `cd D:\felhasználó\dokumentumok\Matematika\`)
- A könyvtárba navigálva, írjuk be a *pdflatex file.tex* parancsot.

Abban az esetben, ha hiba nélkül sikerült megírni a dokumentumot, az eredmény egy PDF fájl, illetve néhány segédfájl lesz. Ezek a forrásfájl könyvtárában találhatóak meg. Hiba esetén a PDF nem, vagy csak részben áll elő. Ezeket a hibákat – a dokumentum sorszámával együtt – a pdflatex program általában jelzi.

Megjegyzés: A forrás fordítása során gyakran találkozhatunk segédfájlokkal. Ezek közül a .log, .bbl, .toc kiterjesztésű fájlok a leggyakoribbak. Ezekre akkor lehet szükség, amikor a program metaadatokat használ. Például, a .toc olyan ideiglenes adatokat tartalmaz, amelyeket a program akkor használ, amikor a dokumentum tartalomjegyzéket is tartalmaz. Hasonlóan, a .bbl fájl a referencialista ideiglenes adatait gyűjti ki. A LaTeX forrás első fordítása során jönnek létre ezek az ideiglenes fájlok, ezért sokszor kétszer- háromszor is le kell fordítani a dokumentumot, amíg a PDF fájlban minden adat helyesen megjelenik. A .log fájl abban az esetben lehet hasznos, ha fordítási hibát észlelünk. Ez a fájl tartalmazza a hibaplót.

1.4. A leíró nyelv előnyei

Szakanyagok írása és olvasása közben sok olyan szimbólummal lehet találkozni, amelyek összeállítása bonyolultabban történik. Ennek az a fő oka, hogy a billentyűzetre nem férnek rá a matematikai, technikai jelöléseket tartalmazó billentyűk. Ezért WYSIWYG szerkesztő esetén (például Microsoft

Word) ezeket a speciális karaktereket általában menüből vagy billentyűkombináció lenyomásával lehet előhívni. Ez sok kattintással, lapozgatással esetleg táblázatok végigböngészésével jár.

Ezzel szemben a LaTeX egyik nagy előnye, hogy az összes speciális karaktert le lehet írni utasításokkal, a billentyűzetten található billentyűk segítségével. Ez írás közben is nagy segítség, mivel nincs szükség grafikus, olykor egérhasználat nélkül nem hozzáférhető felületek böngészésére. Olvasáskor könnyebben megbirkóznak a képernyőolvasó programok a feldolgozással. Nincs olyan speciális karakter, amit a képernyőolvasó program nem ismer.

Másrészről, a LaTeX széles körben elterjedt, szabványos leíró nyelv. Ha a tanár technikai tartalmú dokumentumot ír, majd megosztja azt a diákokkal, kollégákkal, akkor azt ők is azonnal érteni fogják, számukra is ismert lesz a használt nyelv. Könnyen tudnak dolgozni a dokumentummal, szerkeszthetik, javíthatják azt. Nincs szükség egyedi jelölőrendszerre, amit csak az azt kifejlesztő emberek értenek.

A szabványos nyelveket, amelyek adott szabályokra épülnek, algoritmusok segítségével fel lehet dolgozni. Így a LaTeX forrásdokumentumból könnyen előállható a látó emberek számára jobban megfelelő PDF formátum, ami a szimbólumok és formázási utasítások vizuális megfelelőjét tartalmazza. Lehetőség van az utasítások halmazának hallható formátumba történő konvertálására is. Erre szolgál a DEX-ben a 2.7 verziótól használt LaTeX Access (<http://latex-access.sourceforge.net/>) modul is.

1.5. A leíró nyelv hátrányai

A LaTeX használata sok szempontból előnyös, de – mint mindennek – ennek az eszköznek is megvannak a hátrányai. Három fő nehézséggel kell megküzdeni abban az esetben, ha nem vagyunk járatosak programozásban, magyar nyelven szeretnénk leíró nyelvet alkalmazni, és a LaTeX az erre kiszemelt eszköz.

Először is – mint a legtöbb leíró nyelv esetén – az angol nyelv ismerete előnyös lehet. A legtöbb utasítás angol kifejezés, ami általában megegyezik a szimbólum megnevezésével, annak rövidítésével illetve a fogalom első szótagjával. Például a szorzásjel megjelenítésére szolgáló „times” kifejezés angolul szorzást jelent. Ezen kívül, a „frac” utasítás az angol „fraction” kifejezésből származik, ami törtet jelent. Látható, hogy az angol nyelvet ismerő felhasználók előnyben vannak, mivel sok parancs, utasítás a nyelvismeret alapján is kitalálható, könnyen megtanulható.

Másodszor, aki valaha tanult programozni, az tudja, hogy egy hiányzó pont vagy rossz helyre írt zárójel sok problémát okozhat. Ez a leíró nyelvek – különösen a LaTeX – esetében sincs másképp. A fordító nem tolerálja a hibákat. A fordítás során, a programtól függően segítséget kaphatunk, hogy pontosan melyik sorban és milyen jellegű hibát kell orvosolni, mielőtt a PDF fájl elkészülhet. A hiba elhárítása azonban nem minden esetben egyszerű. Sokszor a hibás sor száma nincs megadva, vagy a program ismeretlen hibát ír ki, illetve az első sorban jelez gondot, miközben valójában egy sokkal későbbi sor valamelyik utasításába ágyazott másik utasítás második paraméterének zárójele maradt le gépelés közben. A különféle szövegszerkesztők más és más szinten kezelik a hibákat, illetve jelzik azok helyzetét. Idővel és egyre növekvő tapasztalattal rutinosan lehet hibát keresni. Jó módszer a dokumentum végét jelző utasítást mindig egy-egy bekezdéssel lejjebb helyezni, így kiszűrve, hogy hol lehet a hiba, amikor annak helye nem egyértelmű. A hibakeresés sok időbe telhet, különösen abban az esetben, ha a használt fordítóprogram látás nélkül nehézkesen kezelhető.

A harmadik nagy hátránya a LaTeX-nek az, hogy a komplexebb technikai jellegű szövegek szerkesztésének elsajátítása nagy energiabefektetést igényel. Időt és gyakorlást igénylő feladat a dokumentumok szerkezetét megismerni és megérteni, az utasításokat helyesen használni, új utasításokat meghatározni,

felfedezni a szükséges betölthető csomagokat, a hibákat kiszűrni, stb.

1.6. Magyar nyelvű LaTeX dokumentum előállítás

Aki programozott már, az tudja, hogy a magyar ékezetes betűk helyes megjelenítése a legtöbb esetben gondot okoz. Ennek az az oka, hogy azok a számítástechnikai eszközök, amelyek szöveg megjelenítését is támogatják, az Egyesült Államokban jelentek meg először. Az angol nyelv nem tartalmaz ékezetes betűket, ezért eleinte a különböző programok szerzői nem is gondoltak a „világ többi részére”. A számítástechnika széles körű elterjedésével ez a probléma megoldódott, de a legtöbb esetben a magyar ékezetes betűk alapértelmezetten nem jelennek meg, ahhoz különböző beállításokat kell beleírni a forrásba.

Vegyük az alábbi példát:

```
\documentclass[a4paper,11pt]{article}
\begin{document}
Szia Imre!
\end{document}
```

Ha lefordítjuk, a PDF dokumentumban helyesen jelenik meg a szöveg. Cseréljük ki az „Imre” szót „Gézá”-ra! Azt fogjuk tapasztalni, hogy a lefordított fájlban csak Gza fog szerepelni!

Ennek az az oka, hogy ékezetes karakter szerepel benne! Amikor a forrásszöveg fordításra kerül, akkor a leírt karakterek számokká konvertálódnak. Az angol ABC betűi minden esetben ugyanazon számok lesznek, míg az ékezetes betűk értéke függ a használt kódolástól. Ezért a dokumentumba bele kell írni, hogy mi szerint történik a fájl kódolása. Ezt a `\usepackage[utf8]{inputenc}` utasítás használatával lehet megtenni. Azt, hogy az eredményfájlban magyar karakterek jelenjenek meg, a `\usepackage[T1]{fontenc}` utasítással lehet elérni.

Ha a fenti példát kiegészítjük,

```
\documentclass[a4paper,11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\begin{document}
Szia Géza!
\end{document}
```

akkor a fordítást követően a „Géza” szó is helyesen jelenik meg.

Ahhoz, hogy a dokumentum megfeleljen a magyar helyesírás szabályainak, az ékezhelyes megjelenítésen kívül szükség van arra is, hogy a sorvégi elválasztás megfelelő legyen, a fejezetek számozása a magyar szabályok szerint jelenjen meg, stb. Ezeket a beállításokat a `\usepackage[magyar]{babel}` utasítás beillesztésével lehet megadni.

Az elméleti feltételeken túl, a tapasztalat azt mutatja, hogy minden szempontból helyes magyar nyelvű PDF dokumentumokat csak az alábbi lépések hozzáadásával lehet elérni. Először is, adjuk meg, hogy a dokumentum használja a „lmodern” csomagot, vagyis a fejléceket az alábbi módon bővítjük ki:

```

\documentclass[a4paper,11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\begin{document}
Árvíztűrő tükörfúrógép
\end{document}

```

Ez azt eredményezi, hogy nem csak vizuálisan, de képernyőolvasó programmal felolvastatva is helyesen jelenik meg az összes ékezetes karakter a PDF dokumentumban.

Ahhoz, hogy a kimenet valóban ilyen legyen, a legtöbb esetben nem elegendő az alapértelmezetten használt `pdflatex.exe` program segítségével elvégezni a fordítást, hanem más módszert kell alkalmazni. Ez a módszer három különféle fordítóprogram használatát jelenti, amelyek a MiKTeX programcsomaggal települnek. Ebben az esetben a fordítás három lépést tartalmaz.

1. `latex fájlnev.tex` – ez a „`latex`” nevű programot indítja el, és egy `.dvi` fájlt generál.
2. `dvips fájlnev.dvi` – ez a „`dvips`” nevű programot indítja el, és egy `.ps` fájlt generál.
3. `ps2pdf fájlnev.ps` – ez a „`ps2pdf`” nevű programot indítja el, és egy `.pdf` fájlt generál.

A fordítás technikáiról további részletek a III. részben találhatóak. Végezetül érdemes megjegyezni, hogy minden fordítóprogram kicsit másképpen működik. Az eltérő működésre jellemző példa, hogy míg képek beillesztése esetén a **pdflatex.exe** program elfogadja a `.jpg` kiterjesztésű fájlokat is, addig a **latex.exe** program csak az `.eps` formátummal boldogul. A `\includegraphics{fájlnev}` utasítás esetén a **latex.exe** fordító a fájlnev kiterjesztését nem várja el a fájlnev után, a **pdflatex.exe** viszont igen. (A képekről bővebben a második dokumentumrészben írunk.) Összegezve, magyar nyelvű dokumentumok előállítására még tapasztalt felhasználók számára is sok problémával járhat.

2. fejezet

Általános célú tartalom szerkesztése

Ebben a fejezetben röviden bemutatásra kerül néhány általános célú szerkesztési lépés.

Miért is van erre szükség? A matematikai tartalom – egyenlet, definíció, stb. – általában nem önállóan jelenik meg, hanem leíró elemeket tartalmazó dokumentum részeként.

A leíró részeket egyéb szövegszerkesztők (például Microsoft Word) használatakor megszokott módon érdemes formázni, tagolni, stb.

A folyó szöveg tagolásának egyik eszköze a bekezdésekre bontás. Ezt a forrásban egy üres sor kihagyásával vagy a `\par` utasítás beírásával lehet megtenni. Új sort a `\\` kifejezéssel lehet beszúrni.

Félkövér formázást a `\textbf{Félkövér szöveg}` vagy a

```
\begin{bf}
Félkövér szöveg
\end{bf}
```

utasítással lehet létrehozni.

Dőlt betűs formázást a `\textit{Dőlt betűs szöveg}` vagy a

```
\begin{it}
Dőlt betűs szöveg
\end{it}
```

utasítással lehet létrehozni.

Aláhúzott formázást a `\underline{Aláhúzott szöveg}` utasítással lehet létrehozni.

A fentieket lehet kombinálni is, például a `\textit{\textbf{Dőlt betűs félkövér}}` utasítással ***Dőlt betűs félkövér*** szöveg hozható létre.

A szöveg alapértelmezetten sorkizárt igazítással jelenik meg.

Középre igazított szöveget a

```
\begin{center}
Középre igazított szöveg
\end{center}
```

környezet megadásával lehet beállítani.

Jobbra igazított szöveget a

```
\begin{flushright}
Jobbra igazított szöveg
\end{flushright}
```

környezet megadásával lehet beállítani.

Balra igazított szöveget a

```
\begin{flushleft}
Balra igazított szöveg
\end{flushleft}
```

környezet megadásával lehet beállítani.

Felsorolást az `\itemize` környezet beállításával lehet létrehozni:

```
\begin{itemize}
\item Első elem
\item Második elem
\end{itemize}
```

Az utasítás eredménye az alábbi lista lesz:

- Első elem
- Második elem

Többszintű listát az `\itemize` környezetek egymásba ágyazásával lehet létrehozni:

```
\begin{itemize}
\item Első beágyazott lista
\begin {itemize}
\item Első beágyazott lista első elem
\item Első beágyazott lista második elem
\end{itemize}
\item Második elem
\end{itemize}
```

Az utasítás eredménye az alábbi többszintű lista lesz:

- Első beágyazott lista
 - Első beágyazott lista első elem
 - Első beágyazott lista második elem
- Második elem

Számozott listát hasonló módon, az `\enumerate` környezet beállításával lehet létrehozni:

```

\begin{enumerate}
\item Első számozott elem
\item Második számozott elem
\end{enumerate}

```

Az utasítás eredménye az alábbi lista lesz:

1. Első számozott elem
2. Második számozott elem

Többszintű számozott listát az `\enumerate` környezetek egymásba ágyazásával lehet létrehozni:

```

\begin{enumerate}
\item Első beágyazott számozott lista
\begin {enumerate}
\item Első beágyazott számozott lista első elem
\item Első beágyazott számozott lista második elem
\end{enumerate}
\item Második beágyazott számozott lista
\begin {enumerate}
\item Második beágyazott számozott lista első elem
\item Második beágyazott számozott lista második elem
\end{enumerate}
\end{enumerate}

```

Az utasítás eredménye az alábbi többszintű lista lesz:

1. Első beágyazott számozott lista
 - a) Első beágyazott számozott lista első elem
 - b) Első beágyazott számozott lista második elem
2. Második beágyazott számozott lista
 - a) Második beágyazott számozott lista első elem
 - b) Második beágyazott számozott lista második elem

A példában az első szint arab számokkal, a második pedig kisbetűkkel van jelölve. Ráadásul a második beágyazott lista elemeinél újrameződik a jelölés. Az egyes szintek számozását a `\renewcommand{\labelenumX}{YY}` utasítással lehet megváltoztatni, ahol X a megváltoztatni kívánt szint sorszáma római számmal, YY pedig az új érték.

Ha az előző példa elé beillesztésre kerül a `\renewcommand{\labelenumii}{\arabic{enumii}.}` utasítás, akkor a második szint is arab számokkal lesz jelölve:

1. Első beágyazott számozott lista
 1. Első beágyazott számozott lista első elem
 2. Első beágyazott számozott lista második elem
2. Második beágyazott számozott lista

1. Második beágyazott számozott lista első elem
2. Második beágyazott számozott lista második elem

Többszintű számozás beállításához a

`\renewcommand{\labelenumii}{\arabic{enumi}.\arabic{enumii}.}` utasítás szükséges:

1. Első beágyazott számozott lista
 - 1.1. Első beágyazott számozott lista első elem
 - 1.2. Első beágyazott számozott lista második elem
2. Második beágyazott számozott lista
 - 2.1. Második beágyazott számozott lista első elem
 - 2.2. Második beágyazott számozott lista második elem

A szintek számozásához az alábbi értékek használhatóak:

- `\alph` – Kisbetűk
- `\Alph` – Nagybetűk
- `\arabic` – Arab számok
- `\roman` – Kisbetűs római számok
- `\Roman` – Nagybetűs római számok

Az oldalak számozása automatikusan, arab számokkal történik.

Ha nincs szükség oldalszámozásra, akkor a `\begin{document}` után a `\pagenumbering{gobble}` utasítást kell elhelyezni. Római számokkal történő oldalszámozás a `\pagenumbering{roman}` utasítással adható meg. Az oldalszámozás ki- és bekapcsolása, illetve a formátum változtatása a dokumentumon belül is végrehajtható.

A dokumentumok fejezetekre, alfejezetekre bontása – címsorokkal megjelölése –, az alábbi utasításokkal történhet:

- `\section{section}` – 1. szintű címsor
- `\subsection{subsection}` – 2. szintű címsor
- `\subsubsection{subsubsection}` – 3. szintű címsor
- `\paragraph{paragraph}` – 4. szintű címsor
- `\subparagraph{subparagraph}` – 5. szintű címsor számok

A `\section{section}`, a `\subsection{subsection}` és a `\subsubsection{subsubsection}` utasítás eredménye alapértelmezetten sorszámozva jelenik meg. Ha sorszám nélküli címkézést szeretnénk, akkor azt a `\section*{section}`, a `\subsection*{subsection}` és a `\subsubsection*{subsubsection}` utasításokkal tehetjük meg.

A `\paragraph{paragraph}` és a `\subparagraph{subparagraph}` alapértelmezetten sorszámozás nélküli megjelenítést hoz létre. A sorszámozott megjelenítés az első előfordulás előtt elhelyezett `\setcounter{secnumdepth}{4}` illetve `\setcounter{secnumdepth}{5}` utasítások beírásával valósíthatók meg, például:

`\setcounter{secnumdepth}{4}`

`\paragraph{paragraph}`

Ha a dokumentum **report** vagy **book** osztályú (például `\documentclass{report}`), akkor a -1. illetve 0. szintű `\part{part}` illetve `\chapter{chapter}` utasítások is használhatóak tagolásra.

Tartalomjegyzék a `\tableofcontents` utasítással helyezhető el. Következő utasításként érdemes a `\clearpage` megadni, hogy a tartalom új oldalra kerüljön. A

`\pagenumbering{gobble}`

`\tableofcontents`

`\clearpage`

`\pagenumbering{arabic}`

utasítássorozattal lehet megoldani, hogy a tartalomjegyzék ne legyen oldal számozva, illetve a számozás az első tartalmi oldalon kezdődjön.

A tartalomjegyzék alapértelmezetten háromszintű. A szintek számát a `\tableofcontents` utasítás elé írt `\setcounter{tocdepth}{x}` utasítás használatával lehet módosítani, ahol x a kívánt szintszámot jelöli.

A tartalomjegyzékben alapértelmezetten nem szerepelnek a `\section*{section}`, a `\subsection*{subsection}` és a `\subsubsection*{subsubsection}`, stb. utasításokkal létrehozott elemek. Ezek szerepeltetése a

`\section*{sectiontitle}`

`\addcontentsline{toc}{section}{sectiontitle}`

jellegű utasításpárokkal oldható meg.

II. rész

Matematikai tartalom szerkesztése

A matematikai fejezetek felépítése

Az előző részben arról volt szó, hogy miért, esetleg miért ne, illetve hogyan használjuk a LaTeX adta lehetőségeket a matematikai szövegek szerkesztése során. Ez a rész a Nemzeti Alaptanterv 5.-12. évfolyamának követelményeire épül. Mivel az évfolyamok ismétlődő tananyagot is tartalmaznak, illetve olyan ismeretet, amely matematikailag magasabb fokú koncepciókat mutat be azonos készletű matematikai szimbólumok használatával, ezért az alábbi fejezetek is matematikai témák szerint kerülnek ismertetésre. A főbb csoportok három részre oszlanak. Az első szekcióban a tananyagra legjellemzőbb LaTeX szintaktikát emeljük ki, mivel a gyakorlatban ezekre van a legnagyobb szükség. A használatához – ahol ez indokolt – megjegyzéseket fűzünk. A második szekcióban bővebb, de közel sem teljes körű listát mutatunk be a témához tartozó utasításokról. Végül minden fejezet harmadik része egy-egy témaspecifikus példát tartalmaz.

Az utasítások és a paraméterek osztályozása

A matematikai környezetekben használt utasításokat osztályozni lehet aszerint, hogy hány paramétert tartalmaznak. Ebből a szempontból három típust különböztethetünk meg:

- nullad rangú – nem tartozik hozzá paraméter, csak folyó szöveg előzi meg illetve követi az utasítást. A folyó szöveget „abc123” karaktersorozattal jelöljük.
- elsőrangú – egy paraméter tartozik az utasításhoz. Ezt jelöli az „arg1” változó.
- másodrangú – két paraméter tartozik az utasításhoz. Ezt jelölik az „arg1” és az „arg2” változók.

Például az olyan szimbólumok, mint az összeadás jele (+) vagy a szorzás jele (·) nullad rangú utasítás, mivel ezeket számok, változók előzik meg, vagy követik őket. Például, $abc123 + abc123$ vagy $abc123 \cdot abc123$ formában adhatóak meg. A $\sim\{arg1\}$ vagy a $\vec\{arg1\}$ utasítások elsőrangúak, mivel egy paraméter tartozik hozzájuk. Ez azt fejezi ki, hogy a paraméterrel mit kell tenni az utasításnak. Jelen esetben hatványként kell megjeleníteni $abc123^{arg1}$, vagy $arg1$ formában. Végezetül, másodrangú utasítás a törtszámok megjelenítése, ahol a számlálót és a nevezőt is meg kell adni. Például, a $\frac{arg1}{arg2}$ vagy az $\{arg1\}\over\{arg2\}$ utasítás két paramétert tartalmaz, és az eredményük $\frac{arg1}{arg2}$ formában jelenik meg.

Tételezzük fel, hogy $abc123$ most x értéket vesz fel, míg $bcd234$ az y értéket kapja. Ezen kívül az $arg1$ paraméter a 27 számértékkel rendelkezik, míg az $arg2$ paraméter a 9 értéket hordozza. Ekkor felírható a következő kifejezés:

$$x^y + y \cdot \frac{27}{9} + \vec{x} + \frac{\vec{y}}{27}.$$

Ennek a forrásszöveg szerinti megfelelője az alábbi:

```
$$ x^y + y \cdot \frac{27}{9} + \vec{x} + \frac{\vec{y}}{27} . $$
```

A fenti példa azt is mutatja, hogy az utasítások paraméterei lehetnek folyószöveg-elemek illetve beágyazott utasítások is, lásd a törtbe ágyazott vektort a számlálóban. Az alábbi fejezetekben szereplő táblázatok az utasítások típusa szerint kerültek csoportosításra.

3. fejezet

Halmazok és kombinatorika

3.1. Jellemzők

A halmazok és a kombinatorika tárgyalása során az alapvető halmazműveleti szimbólumok kerülnek bemutatásra. Ezekon kívül a táblázatok szerkesztésével kapcsolatos tudnivalókra lesz szükség. A középiskola felsőbb évfolyamainak tanítása során hasznos lehet a binomiális együttható ismerete is. A halmazok leírása során kiemelten fontos a metszet, az unió, a részhalmaz, a komplementer és az üres halmaz jelölése. Ezeket sorrendben a \cap (\cap), a \cup (\cup), a \subset (\subset), a \complement (\complement) illetve az \emptyset (\emptyset) utasításokkal jelöljük. Ha például A és B két halmaz, akkor az $A \cup B$ vagy a $B \subset A$ kifejezések a két halmaz közötti viszonyt írják le.

Halmazok, csoportok tagjainak felsorolása során gyakran lehet szükség táblázatokra. A táblázatok megjelenítéséhez az $\&$ és a \backslash karaktereket kell a leggyakrabban használni. Előbbi az oszlopokat választja el, utóbbi új sort kezd. A táblázatot a „tabu” környezetbe kell beágyazni. Legegyszerűbb formában ez az alábbi módon jelenik meg:

A	2	4	6	8	10
B	1	3	5	7	9

A fenti táblázat LaTeX forrása az alábbi:

```
\begin{tabu}{c|ccccc}
% A táblázat kezdete. A második argumentumban a függőleges vonal azt jelzi,
% hogy a cellákat követően elválasztóvonal is szükséges.
A c (center) jelzi, hogy a cella tartalma középre zárt. A vonalrajzolás opcionális,
% de mindenképpen itt kell megadni az oszlopok számát.
$A$ & 2 & 4 & 6 & 8 & 10 \\ \hline
% Itt kerül megadásra az első sort. Az & karakter jelzi az oszlop végét.
$B$ & 1 & 3 & 5 & 7 & 9 \\
% A \hline jelzi, hogy vízszintes elválasztóvonal is szükséges.
\end{tabu}
```

Megjegyzés: Az oszlopok tartalmának igazítása nem csak középre, hanem balra (l) illetve jobbra (r) is történhet (a jelölések az angol *left*, illetve *right* szavakból erednek). Számtalan táblázattípus létezik a LaTeX környezetekben, amelyek tetszés szerint formázhatóak, azonban a halmazok tagjainak leírásához az itt megjelenített tabulátorkörnyezet megfelelő megoldást biztosít.

A kombinatorika tanulása során gyakran kerül elő az $\binom{arg1}{arg2}$ (ejtsd: „arg1 alatt az arg2”) jelölés, amelynek használatával leírható az alábbi összefüggés:

$$\binom{5}{3} = \frac{5!}{3! \cdot (5-3)!}$$

Utóbbi kiemelt képlet LaTeX forrása a következő:

`$$ \binom{5}{3} = \frac{5!}{3! \cdot (5-3)!} . $$`

3.2. Leíró utasítások

Utasítás	Vizuális megjelenés	Megnevezés
<code>\in</code>	\in	elem
<code>\cap</code>	\cap	metszet
<code>\cup</code>	\cup	unió
<code>\complement</code>	\complement	komplement
<code>\setminus</code>	\setminus	halmazkülönbség
<code>\emptyset</code>	\emptyset	üres halmaz
<code>\varnothing</code>	\varnothing	üres halmaz
<code>\subset</code>	\subset	részhalmaza
<code>\supset</code>	\supset	befoglalja
<code>\subseteq</code>	\subseteq	részhalmaza vagy egyenlő
<code>\supseteq</code>	\supseteq	befoglalja vagy egyenlő
<code>\nsubseteq</code>	$\not\subseteq$	sem részhalmaza, sem egyenlő
<code>\nsupseteq</code>	$\not\supseteq$	sem befoglalja, sem egyenlő
<code>\subsetneq</code>	\subsetneq	részhalmaza, de nem egyenlő
<code>\supsetneq</code>	\supsetneq	befoglalja, de nem egyenlő
<code>\vartriangleleft</code>	\triangleleft	normál alcsoportja
<code>\vartriangleright</code>	\triangleright	normál alcsoportként tartalmazza
<code>\trianglelefteq</code>	\trianglelefteq	normál alcsoportja vagy egyenlő
<code>\trianglerighteq</code>	\trianglerighteq	normál alcsoportként tartalmazza vagy egyenlő
<code>\ntriangleleft</code>	$\not\triangleleft$	nem normál alcsoportja
<code>\ntriangleright</code>	$\not\triangleright$	nem tartalmazza normál alcsoportként
<code>\ntrianglelefteq</code>	$\not\trianglelefteq$	nem normál alcsoportja vagy egyenlő
<code>\ntrianglerighteq</code>	$\not\trianglerighteq$	nem tartalmazza normál alcsoportként vagy egyenlő

Utasítás	Vizuális megjelenés	Megnevezés
<code>\Subset</code>	\Subset	dupla részhalmaz
<code>\Supset</code>	\Supset	dupla befoglaló halmaz
<code>\Cap</code>	\Cap	dupla metszet
<code>\Cup</code>	\Cup	dupla unió

3.1. táblázat. Nullad rangú utasítások és szimbólumok a halmazműveletek leírásához.

Középiskolai tanulmányaink során találkozhatunk nevezetes számhalmazokkal. Ezeket a tankönyvek általában `\mathbb{*}` „betűtípussal” jelölik, ahol a `*` helyére a számhalmaz angol nevének kezdőbetűjét szokták írni. A következő táblázat ezeket foglalja össze:

Utasítás	Vizuális megjelenés	Megnevezés
<code>\mathbb{N}</code>	\mathbb{N}	természetes számok
<code>\mathbb{Z}</code>	\mathbb{Z}	egész számok
<code>\mathbb{Q}</code>	\mathbb{Q}	racióális számok
<code>\mathbb{R}</code>	\mathbb{R}	valós számok
<code>\mathbb{C}</code>	\mathbb{C}	komplex számok

Megjegyzés: A `\emptyset` (\emptyset), illetve a `\varnothing` (\varnothing) szimbólumok között csak formázási eltérés van. Előbbi inkább matematikai egyenletkörnyezetben megszokott, míg utóbbi folyó szövegben használatos.

Megjegyzés 2: Egy halmaz komplementer halmaza esetén, a \mathbb{C} jelölés helyett gyakran találkozhatunk olyannal is, ahol az adott halmaz felett egy vonal van, például \overline{H} (ejtsd: H felülvonás). Előbbi a tudományos cikkekben, utóbbi inkább a középiskolás tankönyvekben fordul elő leginkább.

Megjegyzés 3: Az itt felsorolt szimbólumok halmaza közül érdemes kiemelni a `\Cap` és a `\Cup` utasításokat. Ezek a dupla unió és a dupla metszet fogalmát határozzák meg. Matematikai képletben az alábbi módon használhatóak ezek a jelek:

$$A \cup B := \{a \cup b \mid a \in A, b \in B\}.$$

Hasonlóan, a dupla metszet így használható:

$$A \cap B := \{a \cap b \mid a \in A, b \in B\}.$$

Utóbbiakat csupán megemlítettük, a normál csoportok és alcsoportok jelöléseivel együtt, ugyanis nem része a középiskolai tanulmányoknak.

3.3. Példák

3.3.1. Vizuális megjelenés

Ha P olyan halmaz amely a prímszámok első 9 tagját foglalja magába, illetve O a páratlan számok első 9 tagjának halmaza (lásd táblázat), akkor adja meg az alábbi halmazműveletek eredményét.

1. $P \cup O$,
2. $O \cap P$.

P	2	3	5	7	11	13	17	19	23
O	1	3	5	7	9	11	13	15	17

3.3.2. LaTeX forrás

Ha P olyan halmaz amely a prímszámok első 9 tagját foglalja magába, illetve O a páratlan számok első 9 tagjának halmaza (lásd táblázat), akkor adja meg az alábbi halmazműveletek eredményét.

```

\begin{enumerate}
\item  $P \cup O$  ,
\item  $O \cap P$  .
\end{enumerate}
\begin{center}
\begin{tabu}{c|cccccccc}
 $P$  & 2 & 3 & 5 & 7 & 11 & 13 & 17 & 19 & 23 \\ \hline
 $O$  & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\
\end{tabu}
\end{center}

```

4. fejezet

Számтан és algebra

4.1. Jellemzők

Talán a számtani fogalmak, az aritmetikai műveletek illetve az algebra az, ami a legnagyobb figyelmet érdemli a tanítás és a tanulás során. A témában elsajátított fogalmak végigkísérik a tanulót az évek során. Szövegszerkesztési szempontból szerencsés, hogy a legtöbb szükséges matematikai kifejezés, szimbólum könnyen és intuitív módon írható le a LaTeX dokumentumban. Jellemzően a $+$, $-$, $=$ jelekre lesz szükség (például $2 + 3 - 1 = 4$) az összeadás, a kivonás és az egyenlőség jelöléséhez, ugyanúgy, mint más szövegszerkesztő használata esetén. A szorzás kicsit bonyolultabb, a `\times`, illetve a `\cdot` utasítás a \times és a \cdot szorzásjeleket írják le. Az osztásra használható a `\div`, azaz a \div utasítás. A becslésre tipikusan a `\approx` (\approx), a `\sim` (\sim) vagy a `\simeq` (\simeq) utasítások alkalmazhatóak. Ezek az utasítások mind nullad rangú parancsként értelmezhetőek.

Törtekre használhatjuk a

```
\frac{5}{6}
{5}\over{6}
```

utasításokat (legtöbbször inkább az elsőt használjuk, mint az utóbbit). Ezek másodrangú parancsok, amelyek két paramétert tartalmaznak. Hatványozni, gyököt vonni az alábbi utasításokkal lehet:

- `a^{n}` – a alapú hatvány, n kitevővel,
- `\sqrt[m]{b}` – b -ből m -edik gyököt vonunk, ahol m a gyökvonás rangját jelzi az opcionális paraméterben.

A fentiek alapján a `\sqrt[3]{8}` utasítással a $\sqrt[3]{8}$ kifejezés adható meg, míg a `2^{3}` parancs a 2^3 kifejezést jeleníti meg.

Egyenlőtlenségek esetén a `\le` (\leq), illetve a `\ge` (\geq) utasítások használhatóak. Az itt kiemelt illetve egyéb szimbólumok következő részben részletesen, táblázat formájában bemutatásra kerülnek.

4.2. Leíró utasítások

A 4.1., a 4.2., illetve a 4.3. táblázatban szereplő szimbólumok közül sokat már az általános iskolában megismernek a tanulók, és ezeket használják a további tanulmányaik során, más matematikai fogalmak elsajátításakor.

Utasítás	Vizuális megjelenés	Megnevezés
+	+	plusz
-	-	mínusz
<	<	kisebb
=	=	egyenlő
>	>	nagyobb
((bal zárójel
))	jobb zárójel
[[nyitó szögletes zárójel
]]	záró szögletes zárójel
\{	{	kezdő kapcsos zárójel
\}	}	záró kapcsos zárójel
\pm	±	plusz mínusz
\mp	∓	mínusz plusz
\div	÷	osztva
\slash	/	törtvonás
\times	×	keresztmetszorzat
\cdot	·	skalár szorzat
\colon	:	hányados
\infty	∞	végtelen
\mid		osztja
\nmid	∤	nem osztja
\%	%	százalék
\		abszolút érték
\propto	∝	arányul
\simeq	≈	aszimptotikus egyenlő
\cong	≅	megközelítőleg egyenlő
\ncong	≇	sem megközelítőleg sem ténylegesen nem egyenlő
\approx	≈	közel egyenlő
\thickapprox	≈	közel egyenlő (félkövér)
\approxeq	≈	közel egyenlő vagy egyenlő
\asymp	∞	egyenértékű
\ne	≠	nem egyenlő
\neq	≠	nem egyenlő
\equiv	≡	azonos

Utasítás	Vizuális megjelenés	Megnevezés
<code>\le</code>	\leq	kisebb egyenlő
<code>\leq</code>	\leq	kisebb vagy egyenlő
<code>\ge</code>	\geq	nagyobb egyenlő
<code>\geq</code>	\geq	nagyobb vagy egyenlő
<code>\leqq</code>	\leqq	egyenlőségjel felett kisebb mint
<code>\geqq</code>	\geqq	egyenlőségjel felett nagyobb mint
<code>\lneqq</code>	\lneqq	kisebb, de nem egyenlő
<code>\gneqq</code>	\gneqq	nagyobb, de nem egyenlő
<code>\ll</code>	\ll	sokkal kisebb mint
<code>\gg</code>	\gg	sokkal nagyobb mint
<code>\lll</code>	\lll	nagyon sokkal kisebb mint
<code>\ggg</code>	\ggg	nagyon sokkal nagyobb mint
<code>\nless</code>	\nless	nem kisebb mint
<code>\ngtr</code>	\ngtr	nem nagyobb mint
<code>\nleq</code>	\nleq	sem egyenlő, sem kisebb mint
<code>\ngeq</code>	\ngeq	sem egyenlő, sem nagyobb mint
<code>\lesssim</code>	\lesssim	egyenértékű vagy kisebb mint
<code>\gtrsim</code>	\gtrsim	egyenértékű vagy nagyobb mint
<code>\lessgtr</code>	\lessgtr	kisebb vagy nagyobb mint
<code>\gtrless</code>	\gtrless	nagyobb vagy kisebb mint
<code>\lesseqgtr</code>	\lesseqgtr	kisebb egyenlő, vagy nagyobb mint
<code>\gtreqless</code>	\gtreqless	nagyobb egyenlő, vagy kisebb mint

4.1. táblázat. Nullad rangú utasítások számtani szövegek szerkesztéséhez.

Utasítás	Vizuális megjelenés (példák)	Megnevezés
<code>\^{arg1}</code>	3^8	3 a 8. hatványon
<code>_{arg1}</code>	3_2	3 alsó index 2
<code>\sqrt{arg1}</code>	$\sqrt{4}$	négyzetgyök 4
<code>\sqrt[opc-arg1]{arg1}</code>	$\sqrt[3]{8}$	köbgyök 8

4.2. táblázat. Elsőrangú utasítások számtani szövegek szerkesztéséhez.

Megjegyzés: Mivel a gyökvonásnak egy kötelező argumentuma, paramétere van, így ezt elsőrangú utasításnak tekintjük. Figyeljük meg azonban, ha az opcionális argumentum kitöltésre kerül, azzal a gyökvonás jele fölé írt érték határozható meg, például köbgyök, ötödgyök stb.

Utasítás	Vizuális megjelenés (példák)	Megnevezés
$\{\arg1\}\overline{\{\arg2\}}$	$\frac{4}{5}$	négy per öt
$\frac{\{\arg1\}}{\{\arg2\}}$	$\frac{2}{3}$	kétharmad
$\stackrel{\{\arg1\}}{\{\arg2\}}$	$3 < 5$	3 halmozott jelölés, felül kettő, alul kisebb mint 5

4.3. táblázat. Másodrangú utasítások számtani szövegek szerkesztéséhez.

Megjegyzés: A 8. évfolyam felett nagyrészt az eddigi szimbólumokat lehet használni a haladó fogalmak ismertetése során.

4.3. Példák

Az alábbi egyszerű számtani példák a gyakorlati használatot mutatják be. Először a megjelenő szöveg olvasható, majd annak LaTeX megfelelője.

4.3.1. Vizuális megjelenés

Egyszerűsítsük a következő törteket és határozzuk meg a viszonyukat.

$$\begin{aligned} \frac{216}{32} & \stackrel{?}{=} \frac{256}{42} \\ \frac{6^3}{2^5} & \stackrel{?}{=} 2^8 \cdot 6 \cdot 7 \\ \frac{2^3 \cdot 3^3}{2^5} & \stackrel{?}{=} \frac{2^8}{2 \cdot 3 \cdot 7} \\ 2^{-2} \cdot 3^3 & \stackrel{?}{=} \frac{2^7}{3 \cdot 7} \\ \frac{3^3}{4} & \stackrel{?}{=} \frac{2^7}{3 \cdot 7} \\ \frac{27}{4} & \stackrel{?}{=} \frac{128}{21} \\ \frac{567}{84} & > \frac{512}{84} \end{aligned}$$

4.3.2. LaTeX forrás

Egyszerűsítsük a következő törteket és határozzuk meg a viszonyukat.

```
\begin{eqnarray}
\frac{216}{32} & \stackrel{?}{=} & \frac{256}{42} \\
\frac{6^3}{2^5} & \stackrel{?}{=} & \{ 2^8 \cdot 6 \cdot 7 \} \\
\frac{2^3 \cdot 3^3}{2^5} & \stackrel{?}{=} & \frac{2^8}{2 \cdot 3 \cdot 7} \\
2^{-2} \cdot 3^3 & \stackrel{?}{=} & \frac{2^7}{3 \cdot 7} \\
\frac{3^3}{4} & \stackrel{?}{=} & \frac{2^7}{3 \cdot 7} \\
\frac{27}{4} & \stackrel{?}{=} & \frac{128}{21} \\
\frac{567}{84} & > & \frac{512}{84}
\end{eqnarray}
```

```
\frac{27}{4} &\stackrel{?}{=} &\frac{128}{21} \\
\frac{567}{84} &> &\frac{512}{84}
\end{eqnarray}
```

Megjegyzés 1: A $\stackrel{x}{y}$ utasítás egymás fölé írt, halmozott relációs jelek írását teszi lehetővé.

5. fejezet

Függvények: az analízis elemei

5.1. Jellemzők

A függvények szerves részét képezik a matematikának. A tanulók már az oktatás korai szakaszában megismerkednek sok új, analízissel kapcsolatos fogalommal. Ez a későbbi tanulásaik során is fontos szerepet kap, amikor kémiából vagy más természettudományi tantárgyból a változók összefüggéseit sajátítják el. Az analízis fogalmához külön LaTeX utasítások, -szimbólumok tartoznak. A leggyakrabban használtak ebben a fejezetben kerülnek bemutatásra, a következő szekció pedig bővebb táblázatot tartalmaz a LaTeX elemekről. Az egyenes arányosság tanulása során gyakran használt utasítás és szimbólum a `\propto` (\propto). Ez a jel a $F \propto x$ példával demonstrálható.

A hozzárendelés, a függvény, a növekedés, a fogyás, az értelmezési tartomány és az értékkészlet fogalmának megismerése során hasznos lehet a `\mapsto` (\mapsto), a `\to` (\rightarrow), az `\in` (\in) és a `\forall` (\forall) karakterek használata.

Hasznos lehet a sorozatok tárgyalása során a minta folytatása pontozott értékkel (`\ldots` (\dots) vagy `\cdots` (\cdots)). Minimum, maximum illetve függvénybe írt függvény esetén a `\min` (\min), a `\max` (\max), valamint a `\circ` (\circ) utasítások lehetnek hasznosak. Ezekkel például az $f \circ g$ kompozit függvény egyszerűen leírható.

A következő részben bemutatásra kerül, hogy a LaTeX milyen lehetőségeket biztosít a nyilak különféle megjelenítésére. Ezek közül jellemzően az alábbi négyet szokás használni.

- `\mapsto` (\mapsto) – Értékkészlet hozzárendelése függvényhez.
- `\to` (\rightarrow) – Ha egy mennyiség tart adott értékhez. Például $x \rightarrow \infty$.
- `\Rightarrow` (\Rightarrow) – Folyamat leírása. Ez jellemzően az „ebből következik” kifejezés leírására használatos.
- `\Leftrightarrow` (\Leftrightarrow) – Ha a két oldalnak egyaránt teljesülnie kell. Matematikailag kifejezve „akkor és csak akkor, ha”.

5.2. Leíró utasítások

Az alábbi példán keresztül egy olyan meghatározás kerül bemutatásra, amely matematikai elméletek bizonyítása során lehet fontos.

Utasítás	Vizuális megjelenés	Megnevezés
<code>\between</code>	$\textcircled{\smile}$	között
<code>\forall</code>	\forall	minden
<code>\exists</code>	\exists	létezik
<code>\nexists</code>	\nexists	nem létezik
<code>\in</code>	\in	eleme
<code>\therefore</code>	\therefore	ezért
<code>\because</code>	\because	mert
<code>\colon</code>	$:$	hányados
<code>\propto</code>	\propto	arányul
<code>\infty</code>	∞	végtelen
<code>\mid</code>	$ $	osztja
<code>\nmid</code>	\nmid	nem osztja
<code>\doteq</code>	\doteq	limeszhez közelít
<code>\vdots</code>	\vdots	függőleges pont pont pont
<code>\cdots</code>	\cdots	középvonalú vízszintes pont pont pont
<code>\ddots</code>	\ddots	jobbra lefelé átlós pont pont pont
<code>\nvdash</code>	\nvdash	nem bizonyítja
<code>\mapsto</code>	\mapsto	érték hozzárendelő jobbra nyíl
<code>\Rightarrow</code>	\Rightarrow	ebből következik (jobbra nyíl)
<code>\rightarrow</code>	\rightarrow	jobbra nyíl
<code>\to</code>	\rightarrow	tart (valamihez)
<code>\leftrightarrow</code>	\leftrightarrow	balra-jobbra nyíl
<code>\Leftrightarrow</code>	\Leftrightarrow	„akkor és csak akkor, ha” nyíl
<code>\longrightarrow</code>	\longrightarrow	jobbra nyíl
<code>\longleftrightarrow</code>	\longleftrightarrow	hosszú balra-jobbra nyíl
<code>\Longleftrightarrow</code>	\Longleftrightarrow	hosszú „akkor és csak akkor, ha” nyíl
<code>\min</code>	\min	minimum
<code>\max</code>	\max	maximum
<code>\lceil</code>	\lceil	felfelé kerekít
<code>\rceil</code>	\rceil	felkerekítés zár
<code>\lfloor</code>	\lfloor	lefelé kerekít
<code>\rfloor</code>	\rfloor	lekerekítés zár

5.1. táblázat. Nullad rangú utasítások függvényanalízis tartalmú szövegek szerkesztéséhez.

5.3. Példák

5.3.1. Vizuális megjelenés

Legyen adott egy $(a_n)_{n \in \mathbb{N}}$ valós számsorozat. Azt mondjuk, hogy az $(a_n)_{n \in \mathbb{N}}$ sorozat konvergens, ha létezik olyan α valós szám úgy, hogy

$$\forall \varepsilon > 0 \quad \exists \nu_\varepsilon \in \mathbb{N}: \quad \forall n > \nu_\varepsilon \Rightarrow |a_n - \alpha| < \varepsilon.$$

5.3.2. LaTeX forrás

Legyen adott egy $(a_n)_{n \in \mathbb{N}}$ (valós) számsorozat.

Azt mondjuk, hogy az $(a_n)_{n \in \mathbb{N}}$ sorozat konvergens,

ha létezik olyan α valós szám úgy, hogy

```
\begin{equation*}
\forall \varepsilon > 0 \quad \exists \nu_{\varepsilon} \in \mathbb{N}: \quad
\forall n > \nu_{\varepsilon} \Rightarrow |a_n - \alpha| < \varepsilon.
\end{equation*}
```

6. fejezet

Geometria

6.1. Jellemzők

A geometriai ismeretek tanulása során számos grafikai fogalmat kell elsajátítani. Ezek egy részét némi kreativitással adaptálni lehet leíró nyelven is. Nem szabványos megoldás, de a képi illusztráció helyett az olyan fogalmakat, mint a pont, az egyenes, a szakasz és a félegyenes ki lehet fejezni a `\cdot`, a `\circ` illetve a `\hline` utasítások használatával. Az előbbi utasítások jelentése – sorrendben – középső pont, kör illetve vízszintes vonal. Például egy szakasz jelölése megadható egy vízszintes vonal elé és után írt pont vagy kis kör használatával. A megoldás ugyan nem tökéletes – pláne nem szabványos –, de a tanulást elősegítheti, mivel a grafikus elemek a képernyőolvasó program által feldolgozható utasításokkal kerülnek helyettesítésre. Egyéb fogalmak esetén, például merőlegesség, párhuzamosság, szögfajták az alábbi táblázatban található szimbólumok nyújthatnak segítséget. Az olyan fogalmakat, mint a távolság, a szakaszfelező merőleges, a szögfelező, a síkidom, a sokszög, a kör, a test, a csúcs, az él, a lap, a szög, a gömb, a terület, a felszín, a testek hálójaja, a térfogat, a magasság és a tengelyes tükrözés, a „Gráfok és koordinátageometria” című fejezetben bemutatott módon a legegyszerűbb megadni.

6.2. Leíró utasítások

Utasítás	Vizuális megjelenés	Megnevezés
<code>\angle</code>	\sphericalangle	szög
<code>\measuredangle</code>	\sphericalangle	mért szög
<code>\sphericalangle</code>	\sphericalangle	térszög
<code>\parallel</code>	\parallel	párhuzamos
<code>\nparallel</code>	\nparallel	nem párhuzamos
<code>\Doteq</code>	\doteq	mértanilag egyenlő
<code>\Bumpeq</code>	\bumpeq	mértanilag egyenértékű

6.1. táblázat. Nullad rangú utasítások geometriai szimbólumok beviteléhez.

Megjegyzés: A mértanilag egyenlő (\doteq) és a mértanilag egyenértékű (\bumpeq) szimbólumok esetén fontos, hogy az utasítást nagy kezdőbetűvel írjuk. Ellenkező esetben a nem kívánt (\doteq) valamint (\simeq) jeleket

idézzük elő.

A matematikai tárgyak esetén idővel elengedhetetlen a görög betűk megismerése. A görög betűk jellemzően a geometria tárgykörében fordulnak elő, ezért érdemes megismerni az `\alpha` (α), a `\beta` (β) és a `\pi` (π) szimbólumokkal a kör kerülete, területe illetve a szögjelölések kapcsán. Az alapvető szögfüggvények és vektorjelölések LaTeX megfelelőjét is érdemes megismerni. A szélesebb körű vektor- és trigonometriai jelölések az erre vonatkozó, tematikus fejezetben bővebben is kifejtésre kerülnek, azonban a görög ABC jelölései itt találhatóak meg teljes körűen.

Utasítás	Vizuális megjelenés	Megnevezés
<code>\sin</code>	sin	szinusz
<code>\cos</code>	cos	koszinusz
<code>\tan</code>	tan	tangens
<code>\alpha</code>	α	alfa
<code>\beta</code>	β	béta
<code>\gamma</code>	γ	gamma
<code>\delta</code>	δ	delta
<code>\epsilon</code>	ϵ	epszilon
<code>\zeta</code>	ζ	zéta
<code>\eta</code>	η	éta
<code>\theta</code>	θ	théta
<code>\iota</code>	ι	ióta
<code>\kappa</code>	κ	kappa
<code>\lambda</code>	λ	lambda
<code>\mu</code>	μ	mú
<code>\nu</code>	ν	nú
<code>\xi</code>	ξ	kszí
<code>\omicron</code>	\omicron	omikron
<code>\pi</code>	π	pí
<code>\rho</code>	ρ	ró
<code>\sigma</code>	σ	szigma
<code>\tau</code>	τ	tau
<code>\upsilon</code>	υ	üpszilon
<code>\phi</code>	ϕ	fi
<code>\psi</code>	ψ	pszí
<code>\chi</code>	χ	khí
<code>\omega</code>	ω	ómega

6.2. táblázat. Nullad rangú utasítások a trigonometriai függvények leírásához illetve a görög ABC kisbetűs karaktereinek beviteléhez.

A görög betűk nagybetűs változata az esetek egy részében az utasítások első betűjének nagybetűs használatával adható meg, például a Δ a `\Delta` utasítással írható le. Azok a nagybetűk, amelyek képe megegyezik valamelyik latin betű képével (alfa A, béta B, epsilon E, dzéta Z, éta H, jóta I, kapa K, mú M, nú N, omikron O, ró P, tau T, hí X) a megfelelő betű leírásával adhatóak meg. Bár nem gyakori karakter, de érdemes megjegyezni, hogy a kis omikron görög szimbólum képe nagyon hasonlít a latin kis o karakterre, ezért ez alapértelmezetten nincs definiálva a LaTeX-ben. A problémát – ahogy ebben a dokumentumban is tesszük – a `\newcommand{\omicron}{o}` makróval lehet áthidalni. A LaTeX makrók olyan személyre szabott utasítások, amelyeknél a felhasználó az első argumentumban adja meg a kívánt utasítás alakját, a másodikban pedig a megjelenítendő szöveget.

Megjegyzés: Bár matematikailag a trigonometriai függvényeknek van argumentuma, a leíró nyelv szempontjából ezek az utasítások nullad rangú parancsoknak tekinthetőek. A `\sin(2x)` példa is mutatja, hogy az utasítás után normál zárójelek közé írandó a paraméter, nincs szükség kapcsos zárójel használatára – `\sin{2x}` –, ahogy az elvart az elsőrangú utasításoknál például `\vec{vektor}` (*vektor*). Ezen kívül érdemes megjegyezni, hogy a kotangens függvényre nem lett definiálva külön LaTeX utasítás. E célból a dokumentumszerzők különféle makrókat (személyre szabott utasításokat) definiálhatnak, illetve kevésbé ismert csomagokat importálhatnak. Ilyen gyakran előforduló makró például a `\newcommand{\cot}{cot}` vagy a `\newcommand{\ctg}{ctg}`.

A felsőbb évfolyamokon tanított fogalmakat, komplex mértani feladatokat, geometriai szerkesztéseket képformátumban – ellátva a megfelelő képleírással – illetve a diagram domborított megfelelőjével célszerű tanítani. Az ehhez szükséges LaTeX környezetet lásd a „Gráfok és koordinátageometria” című 12. fejezetben.

6.3. Példák

6.3.1. Vizuális megjelenés

Vezesse le a pitagoraszai azonosságot

$$\sin^2 \theta + \cos^2 \theta = 1, \quad (6.1)$$

a kör egyenletéből kiindulva

$$x^2 + y^2 = r^2 \quad (6.2)$$

Használja a polárkoordináta adta lehetőségeket.

Megoldás:

$$x^2 + y^2 = r^2 \quad (6.3)$$

$$(r \cos \theta)^2 + (r \sin \theta)^2 = r^2 \quad (6.4)$$

$$r^2 \cos^2 \theta + r^2 \sin^2 \theta = r^2 \quad (6.5)$$

$$\cos^2 \theta + \sin^2 \theta = 1 \quad (6.6)$$

Figyeljük meg! A (6.5) egyenletet r^2 -tel osztjuk, hogy a várt eredményt elérjük.

6.3.2. LaTeX forrás

Vezesse le a pitagoraszi azonosságot

```
\begin{equation}
\sin^2 \theta + \cos^2 \theta = 1,
\end{equation}
```

a kör egyenletéből kiindulva

```
\begin{equation}
x^2 + y^2 = r^2
\end{equation}
```

Használja a polárkoordináta adta lehetőségeket.

```
\textbf{Megoldás:}
\begin{eqnarray}
x^2 + y^2 &=& r^2 \\
(r \cos \theta)^2 + (r \sin \theta)^2 &=& r^2 \\
r^2 \cos^2 \theta + r^2 \sin^2 \theta &=& r^2 \label{eq:r2} \\
\cos^2 \theta + \sin^2 \theta &=& 1
\end{eqnarray}
```

Figyeljük meg! A `\eqref{eq:r2}` egyenletet r^2 -tel osztjuk, hogy a várt eredményt elérjük.

7. fejezet

Statisztika, valószínűségyszámítás és matematikai logika

7.1. Jellemzők

Az átlag, a biztos esemény, a lehetetlen esemény és a valószínűség fogalmainak tanítása során ismét a szimbólumok új csoportjával találkozunk. Az átlagszámítás jelölésére leggyakrabban a `\bar{x}` elsőrangú utasítás használatos, ami egy vonalat jelenít meg a kívánt mennyiség felett, például (\bar{x}) . Ezen kívül a bal és a jobb hegyes zárójelet is lehet használni erre a célra. Ezek nullad rangú utasítások. Például az x halmaz elemeinek átlaga leírható a `\langle x \rangle` utasításpárral. A valószínűségyszámítási tanulmányok során gyakran felbukkan a `\Pr` (Pr) illetve a `\vert` ($|$) jelölés is, például $\Pr(A|B)$. A $\Pr(A)$ jelölés az A esemény valószínűségét – angolul probability – jelöli P karakterrel, míg a példában szereplő, a két esemény közötti vonal azok feltételeességét jelzi. A matematikai logika ismertetése során többször előfordul a `\wedge` (\wedge), a `\vee` (\vee), a `\vDash` (\vDash) valamint az `\nvDash` jelölés. Ezek sorrendben a logikai vagy, az és, az igaz és a hamis kifejezéseknek felelnek meg. Végül a `\not` utasítás a halmazelméletből ismert komplementerhez hasonlóan használható a kifejezések negálásához, például `\not\supset` ($\not\supset$). A következő részben bemutatott utasítások az informatikai tartalmak oktatásakor is hasznosak lehetnek.

7.2. Leíró utasítások

Utasítás	Vizuális megjelenés	Megnevezés
<code>\Pr</code>	Pr	valószínűség jele
<code>\mid</code>		feltételes valószínűség
<code>\vert</code>		feltételes valószínűség (formázható)
<code>\perp</code>	\perp	függetlenség
<code>\langle</code>	\langle	átlag kezdete
<code>\rangle</code>	\rangle	átlag vége

Utasítás	Vizuális megjelenés	Megnevezés
<code>\wedge</code>	\wedge	logikai és
<code>\vee</code>	\vee	logikai vagy
<code>\veebar</code>	\veebar	kizáró vagy
<code>\barwedge</code>	$\bar{\wedge}$	nem és
<code>\vDash</code>	\vDash	igaz
<code>\nvDash</code>	\nvDash	nem igaz
<code>\lnot</code>	\neg	nem
<code>\not</code>	/	nem
<code>\neg</code>	\neg	negáció

7.1. táblázat. Nullad rangú utasítások logikai kifejezések leírásához.

Megjegyzés: Bár látvány tekintetében a `\mid` valamint a `\vert` utasítások hasonló vizuális kimenetet produkálnak, az előbbi fix magasságú. Ebben az értelemben a második utasítás rugalmasabb, például tört kifejezések esetén is megfelelő lesz a szimbólum magassága. A magasság beállítását a LaTeX automatikusan elvégzi.

7.3. Példák

7.3.1. Vizuális megjelenés

Egy fizika kísérlet során az x mennyiség megállapítására 100 mérést végzünk el. Ekkor $n = 100$, továbbá jelölje n_k az azonos méréseknek a számát. Az átlag eredmény kiszámításának ismeretében

$$\langle x \rangle = \bar{x} = \sum_{k=1}^{100} \frac{x_k \cdot n_k}{n}.$$

Néhány szóban magyarázzuk el a standard eltérés fogalmának jelentését. Ennek képlete:

$$\sigma^2(x) = \sum_{k=1}^{100} \frac{(x_k - \bar{x})^2}{n}.$$

Végül a \wedge, \vee, \neg szimbólumok (logikai műveleti jelek), valamint az alábbi kifejezések használatával, legalább kettő igaz állítást írjon le.

- esik az eső,
- szivárvány keletkezik,
- süt a nap,
- rossz az idő,
- erős a szél,
- erős a forgalom.

7.3.2. LaTeX forrás

Egy fizika kísérlet során az x mennyiség megállapítására 100 mérést végzünk el. Ekkor $n=100$, továbbá jelölje n_k az azonos méréseknek a számát.

Az átlag eredmény kiszámításának ismeretében

$$\langle x \rangle = \bar{x} = \sum_{k=1}^{100} \frac{x_k \cdot n_k}{n}.$$

Néhány szóban magyarázzuk el a standard eltérés fogalmának jelentését.

Ennek képlete:

$$\sigma^2(x) = \sum_{k=1}^{100} \frac{(x_k - \bar{x})^2 \cdot n_k}{n}.$$

Végül a \wedge , \vee , \neg szimbólumok (logikai műveleti jelek),

valamint az alábbi kifejezések használatával, legalább kettő igaz állítást írjon le.

```
\begin{itemize}
```

```
\item esik az eső,
```

```
\item szivárvány keletkezik,
```

```
\item süt a nap,
```

```
\item rossz az idő,
```

```
\item erős a szél,
```

```
\item erős a forgalom.
```

```
\end{itemize}
```

8. fejezet

Egyenletek, egyenletrendszerek, egyenlőtlenségek

8.1. Jellemzők

A mérlegelv nemcsak a matematikában, de az élet egyéb területein is jól használható módszer. Az egyenletek, egyenlőtlenségek ismerete a kezdetektől fogva szerves része a matematika illetve egyéb természettudományos tantárgyak oktatásának, azonban jelen leírás nem sok újdonságot tartalmaz ebben a témakörben. A legtöbb általános- illetve középiskolai szintű egyenlet a már ismertetett illetve az ezután következő fejezetekben tárgyalt szimbólumokkal és utasításokkal könnyen leírható. Ennek ellenére, amikor több egyenletet szükséges egybefűzni, az egyenletrendszer körültekintő leírásához szükség van néhány új ismeretre. Leggyakrabban az `&` jelre lesz szükségünk, illetve a `\\` utasításra. Előbbi a helyes sorkizárásért felelős, míg utóbbi – a dupla visszafelé dőlő per jel – az új sort jelöli. Ezen kívül a `\label` és a `\ref` utasításokra is szükség lehet. Ezek használatát írja le a következő fejezetet.

8.2. Leíró utasítások

Egy egyenletrendszerben, ahol vizuális szempontból előnyös, ha az egyenlet bal és jobb oldala mindig azonos középponttól kezdődik, a legtöbb esetben az egyenlőséget jelző szimbólumtól, szükség van helyes sorkizárásra, illetve sortörésre. A következő egyenletrendszerben ezeknek a szimbólumoknak a helyes használata kerül bemutatásra, a sorok alatt, vagy végén elhelyezett megjegyzésekkel kiegészítve.

```
\begin{eqnarray} % Az egyenletrendszer kezdetének jelölése.
```

```
x^2 + 4x + 4 & = & 0 \\
```

```
% Az & karakter azt az utasítást adja a LaTeX fordítónak, hogy az egymás alá írt soroknak azonos helyen legyen a középpontja. Az ilyen módon három részre osztott egyenletnél mindig az egyenlőségjel kerül középre. Az egyenletek jobb- és baloldala minden sorban ehhez lesz igazítva.
```

```
A sorvégi \\ jelölés a sortörést jelzi.
```

```
(x+2)(x+2) & = & 0 \\
```

```
(x+2)^2 & = & 0\\
```

% Az egyenletrendszer második sorában szintén megadásra került, hogy az = jel legyen az igazítási pont, illetve sortörés is meg lett adva.

```
|x + 2| & = & 0 \label{kulcs} \\\
```

% Utóbbi sorban az egyenletrendszer negyedik egyenletét címkével jelöltük meg. A „kulcs” paraméter szabadon választható kifejezés. Erre a címkére a dokumentum tetszőleges helyén vissza lehet utalni, ahogy azt később látni fogjuk.

```
x & = & -2
```

% Ebben a sorban már nincs szükség a sortörésre, így a végén nem is található \\ jelzés.

```
\end{eqnarray} % Az egyenletrendszer végének jelölése.
```

A fenti egyenletrendszer leírása a következő módon jelenik meg:

$$x^2 + 4x + 4 = 0 \tag{8.1}$$

$$(x + 2)(x + 2) = 0 \tag{8.2}$$

$$(x + 2)^2 = 0 \tag{8.3}$$

$$|x + 2| = 0 \tag{8.4}$$

$$x = -2 \tag{8.5}$$

Egyenletek levezetése során nem ritka a „A példa egyenletrendszer \ref{kulcs} számú egyenletében, megfigyelhetjük...” kifejezés használata. Figyeljük meg, hogy a címkézett egyenletre a \ref{kulcs} utasítással lehet hivatkozni. A referencia és a címke {kulcs} paraméterének meg kell egyeznie. A nagyobb terjedelmű dokumentumokban, ahol számos egyenlet fordulhat elő, ez a funkció igen hasznos lehet. A LaTeX fordításkor automatikusan beszámozza a hivatkozott egyenleteket, így nem probléma, ha a részletesebb magyarázat érdekében utólag kerülnek hivatkozások beszúrásra.

Időnként nemcsak a dokumentum, de egy egyenlet leírása is hosszú lehet az egyenletrendszeren belül. Ebben az esetben a \left és a \right utasításokat érdemes használni, különösen akkor, ha a hosszú kifejezést zárójelek közé kell írni. Például:

```
\begin{eqnarray}
H.E & = & \left( \frac{x}{3} + 2x + \text{hosszú egyenlet eleje } + 2z \right) \\
& + & \left. 3x^2 - 42 + \text{ egyenlet vége } + 5x \right) \\
\end{eqnarray}
```

Nézzük, pontosan mi is történik ebben az esetben. Adott egy hosszú egyenlet, amit két sorban kívánunk megjeleníteni, úgy, hogy az egyenlet bal oldalán kizárólag a H.E elnevezés szerepeljen, és a két sor az egyenlőség-, illetve a plusz jelhez legyen igazítva. A \left és a \right páros a művelet bal- illetve jobb határolóját jelzi. Az az érdekessége ennek az utasításnak, hogy a határoló bal- illetve jobb zárójelek

méretét a bennük szereplő kifejezés legmagasabb tagjához igazítja. Így a törtszámok nem lesznek magasabbak, mint a zárójel. Fontos, hogy sortörés esetén, a sor befejezése előtt be kell zárni a határoló utasítást. Akkor is, ha a következő sorban folytatjuk a zárójelbe írt kifejezést. Erre szolgál a `\left` és a `\right` utasítás. Ezek nem látható módon zárják és kezdik a zárójelezést az új sorban. Ezzel a technikával az egyenlet a következő módon jelenik meg:

$$H.E = \left(\frac{x}{3} + 2x + \text{hosszú egyenlet eleje} + 2z + \right. \\ \left. + 3x^2 - 42 + \text{hosszú egyenlet vége} + 5x \right)$$

A legfontosabb utasításokat és azok használatát az alábbi táblázat foglalja össze:

Utasítás	Jelentés
<code>&</code>	Sor igazítása az adott pozícióhoz.
<code>\\</code>	Új sor.
<code>\label{arg1}</code>	Az adott sor címkézése. Elsőrangú utasítás, ahol a paraméter a címke által használt kulcsszó.
<code>\ref{arg1}</code>	A címkézett sorra történő hivatkozás. Elsőrangú utasítás, ahol a paraméter a referencia által használt kulcsszó.
<code>\eqref{arg1}</code>	A címkézett sorra történő hivatkozás zárójel közé zárva. Elsőrangú utasítás, ahol a paraméter a referencia által használt kulcsszó.
<code>\left</code>	Nullad rangú utasítás, bal határolóértéket jelez. Írható (, [, illetve egyéb karakterek elé is. Az utasítást követő . karakter láthatatlan határolót hoz létre sortörés esetén.
<code>\right</code>	Nullad rangú utasítás, jobb határolóértéket jelez. Írható),], illetve egyéb karakterek elé is. Az utasítást követő . karakter láthatatlan határolót hoz létre sortörés esetén.
<code>\nonumber</code>	Lokálisan kikapcsolja a sor számozását. (Az adott egyenlet nem lesz számozva, a következő viszont már igen.)
<code>\notag</code>	Lokálisan kikapcsolja a sor számozását. (Az adott egyenlet nem lesz számozva, a következő viszont már igen.)

8.1. táblázat. Utasítások egyenletrendszerek leírásához.

8.3. Példák

Csoportosítsuk, majd oldjuk meg az alábbi másodfokú egyenletet!

$$\frac{x}{2} - 1 = -3x^2 + 7,5x + 42 - \frac{11}{2}x + 12x^2 - 9,5x - 30 - 7x^2 - 3 \quad (8.6)$$

$$\begin{aligned} 0 &= 12x^2 - 7x^2 - 3x^2 + \\ &+ \left(7,5x - \frac{11}{2}x - 9,5x + \right. \\ &\left. + 42 - 30 - 3 \right) - \frac{x}{2} + 1 \end{aligned} \quad (8.7)$$

$$0 = 2x^2 - 8x - 10 \quad (8.8)$$

$$= (2x + 2)(x - 5) \quad (8.9)$$

$$\text{Vagyis } x_1 = -1, \text{ vagy } x_2 = 5. \quad (8.10)$$

Az egyenlet (8.6) csoportosításával kapjuk meg az egyenlet (8.8) sorába írt kifejezést, amely már könnyen egyszerűsíthető.

8.3.1. Vizuális megjelenés

8.3.2. LaTeX forrás

Csoportosítsuk, majd oldjuk meg az alábbi másodfokú egyenletet!

```
\begin{eqnarray}
\frac{x}{2} - 1 &=& -3x^2 + \text{MathReal}{7.5} x + 42 - \frac{11}{2}x
+ 12 x^2 - \text{MathReal}{9.5}x - 30 - 7x^2 - 3 \label{e:1} \\
0 &=& 12 x^2 - 7x^2 - 3x^2 + \nonumber \\
&+& \left( \text{MathReal}{7.5}x - \frac{11}{2}x - \text{MathReal}{9.5}x + \right. \notag \\
&+& \left. 42 - 30 - 3 \phantom{\frac{11}{2}} \right) - \frac{x}{2} + 1 \\
0 &=& 2x^2 - 8 x - 10 \label{e:2} \\
&=& (2x+2)(x-5) \\
&& \text{Vagyis } x_1 = -1 \text{, vagy } x_2 = 5.
\end{eqnarray}
```

Megjegyzés: A fenti LaTeX kódban megjelent egy ismeretlen parancs, mégpedig a `\MathReal{*}` kifejezés. A `*` helyébe angol írásmódú tizedes törteket behelyettesítve megkapjuk a „magyar” tizedes törteket. Ezen azt értjük, míg az angol nyelvterületen a tizedespont az elterjedt, addig Magyarországon tizedesvesszőt használunk az egész rész és a tizedesjegyek elválasztására.

9. fejezet

Hatvány, gyök, logaritmus

9.1. Jellemzők

Az algebrai kifejezések kapcsán érintőlegesen már tárgyaltuk a hatványok és a gyökvonás által használt utasításokat, de nem tértünk ki részletesen ezek ismertetésére. Hatványokat azok paraméterének hossza alapján kétféleképpen is írhatunk. Ebben a fejezetben arra is kitérünk, hogy utasításokat be is lehet ágyazni más utasításokba. Például, hatványba írt tört is előfordulhat, ami felső évfolyamokon hasznos szintaktikai kifejezés lehet. Hatványozás, gyökvonás, exponenciális függvények valamint azok inverze esetén az alábbi utasítások a leggyakoribbak. A `^{\arg1}` utasítás a hatvány jelzésére, a `\sqrt[opc-arg]{arg1}` a gyökvonásra szolgál. Előfordul még az `\exp`, az `\ln` és a `\log` utasítás is. Utóbbi három utasításnak nincs paramétere (azaz nullad rangú utasítás), a függvénynév után írt zárójel közé vagy zárójel nélküli térbe írhatjuk a paramétert, vagyis a kifejezést, számot, amire a függvény vonatkozik. Például `\exp(\ln x) = x = e^{\ln x}` a következőképpen jelenik meg: $\exp(\ln x) = x = e^{\ln x}$.

9.2. Leíró utasítások

Az 5.-10. évfolyam között nem sok újdonsággal találkozunk a hatványozáson és négyzetgyök kifejezésein túl. A 11.-12. évfolyam során, a racionális hatványok illetve a logaritmus fogalmának bevezetése indokolja a következő utasítások használatát.

Utasítás	Vizuális megjelenés (példák)	Megnevezés
<code>\sqrt[opc-arg]{arg1}</code>	$\sqrt[n]{b}$	n-edik gyök alatt b
<code>^{\frac{arg1}{arg2}}</code>	$x^{\frac{1}{2}}$	négyzetgyök x (törtkitevős alakban)
<code>e^{\arg1}</code>	e^x	e az x-edik hatványon
<code>\exp</code>	$\exp(x)$	exponenciális
<code>\ln</code>	$\ln x$	természetes alapú logaritmus
<code>\log</code>	$\log x$	logaritmus

9.1. táblázat. Nullad- és elsőrangú utasítások hatványok, exponenciális és logaritmusos függvények szerkesztéséhez.

Amennyiben le szeretnénk írni az „ a alapú logaritmus b ” kifejezést, akkor ezt a `\log_{a}{b}` paranccsal tehetjük meg. Érdekes még a hatványok szintaktikáját alaposabban is megvizsgálni. Abban az esetben, amennyiben a hatvány argumentuma mindösszesen egy karakterből áll, például x^2 , a hatványt írhatjuk közvetlenül az azt jelző szimbólum után. Azonban, ha már kettő vagy több karakterről van szó, ezt a paramétert jelző zárójelek közé kell írni, például x^{2y} , ami `x^{2y}` formában írható. Ha ezt mellőzzük, a képlet x^2y formában jelenik meg, helytelenül. Ezért érdemes mindig használni a zárójeleket, még egykarakteres paraméter esetén is. Fontos, hogy a nullad rangú utasítások, például a végtelen jele (∞) egy karakternek számítanak.

9.3. Példák

9.3.1. Vizuális megjelenés

Oldja meg az alábbi összeadást az exponenciális függvény segítségével, majd az eredményt $c \sqrt[a]{x^b}$ formában határozza meg, ha $y = 8x$.

$$\ln y^{\frac{2}{3}} + \ln x^{\frac{3}{4}} \quad (9.1)$$

Megoldás: Tegyük fel, hogy

$$k = c \sqrt[a]{x^b}. \quad (9.2)$$

Ebben az esetben,

$$\ln k = \ln(y^{\frac{2}{3}} \cdot x^{\frac{3}{4}}) \quad (9.3)$$

$$\ln k = \ln(8^{2/3} x^{2/3} \cdot x^{3/4}) \quad (9.4)$$

$$\ln k = \ln(\sqrt[3]{64} x^{\frac{2}{3} + \frac{3}{4}}) \quad (9.5)$$

$$k = \exp(\ln 4 x^{17/12}) \quad (9.6)$$

$$c \sqrt[a]{x^b} = 4 x^{17/12} = 4 \sqrt[12]{x^{17}}. \quad (9.7)$$

Végeredményben $a = 12$, $b = 17$ és $c = 4$.

9.3.2. LaTeX forrás

Oldja meg az alábbi összeadást az exponenciális függvény segítségével, majd az eredményt `\, \sqrt[a]{x^b}` formában határozza meg, ha $y=8x$.

```
\begin{equation}
```

```
\ln y^{\frac{2}{3}} + \ln x^{\frac{3}{4}}
```

```
\end{equation}
```

```
\textbf{Megoldás:} Tegyük fel, hogy
```

```
\begin{equation}
```

```
k = c \, \sqrt[a]{x^b} \, , .
```

```
\end{equation}
```

```
Ebben az esetben,
```

```
\begin{eqnarray}
```

```
\ln k&=&\ln (y^{\frac{2}{3}} \cdot x^{\frac{3}{4}}) \\
```

```
\ln k &=&\ln (8^{\frac{2}{3}} \cdot x^{\frac{2}{3}} \cdot x^{\frac{3}{4}}) \\
```



```

\ln k &=& \ln (\sqrt[3]{64}\, x^{\frac{2}{3}+\frac{3}{4}}) \\\
k &=& \exp(\ln 4\, x^{17/12}) \\\
c\, \sqrt[a]{x^b} &=& 4\, x^{17/12} = 4\, \sqrt[12]{x^{17}} \, \, .
\end{eqnarray}

```

Végeredményben $a=12$, $b=17$ és $c=4$.

10. fejezet

Trigonometria és vektorszámítás

10.1. Jellemzők

A középiskolai tanulmányok során – különösen emelt szinten – szükség lehet a vektorjelölések és a bonyolultabb trigonometriai függvények akadálymentes leírására. A geometriához és a trigonometriához szükséges alapvető függvényjelölések – például a szinusz vagy a koszinusz – már bemutatásra kerültek a „Geometria” című fejezetben. Ezekon kívül érdemes áttekinteni a hiperbolikus függvényeket, az inverz trigonometriai képleteket illetve a vektorok különféle jelölését is. Utóbbiak közül jellemzően az elsőrangú utasításokat – például a \vec{x} (\vec{x}) és a \hat{x} (\hat{x}) – szokás használni. Vektorok szorzására a \cdot (\cdot) illetve a \times (\times) jelek használatosak. A skalár szorzat például $\vec{a} \cdot \vec{b}$ formában írható le, míg a keresztszorzat $\vec{x} \times \vec{y}$ alakban.

10.2. Leíró utasítások

Utasítás	Vizuális megjelenés	Megnevezés
\sin	sin	szinusz
\arcsin	arcsin	arkusz szinusz
\sinh	sinh	szinusz hiperbolikus
\cos	cos	koszinusz
\arccos	arccos	arkusz koszinusz
\cosh	cosh	koszinusz hiperbolikus
\tan	tan	tangens
\arctan	arctan	arkusz tangens
\tanh	tanh	tangens hiperbolikus

10.1. táblázat. Nullad rangú utasítások haladó trigonometriai függvények jelöléséhez.

Utasítás	Vizuális megjelenés (példák)	Megnevezés
<code>\cdot</code>	\cdot	skalár szorzat
<code>\times</code>	\times	kereszt szorzat
<code>\hat{arg1}</code>	\hat{x}	normál vektor jelölése (ejtsd: x kalap)
<code>\vec{arg1}</code>	\vec{x}	vektor
<code>\overline{arg1}</code>	\bar{x}	vektor jelölés
<code>\mathbf{arg1}</code>	\mathbf{a}	matematikai félkövér

10.2. táblázat. Nullad- és elsőrangú utasítások vektorszámítás leírásához.

Megjegyzés: A vektorokat szokás félkövér betűtípussal is jelölni. Ehhez a matematikai formázási utasítások közül a `\mathbf{}` az egyik leggyakrabban használt kifejezés.

10.3. Példák

10.3.1. Vizuális megjelenés

A hiperbolikus szinusz és koszinusz matematikai meghatározásai segítségével (lásd lejjebb) határozza meg a két függvény hányadosát:

$$\frac{\sinh(x)}{\cosh(x)},$$

ahol

$$\sinh(x) = \frac{e^x - e^{-x}}{2}, \text{ valamint } \cosh(x) = \frac{e^x + e^{-x}}{2}.$$

10.3.2. LaTeX forrás

A hiperbolikus szinusz és koszinusz matematikai meghatározásai segítségével (lásd lejjebb) határozza meg a két függvény hányadosát:

```
\begin{equation*}
\frac{\sinh(x)}{\cosh(x)},
\end{equation*}
```

ahol

```
\begin{equation*}
\sinh(x) = \frac{e^x - e^{-x}}{2}, \text{ valamint } \cosh(x) =
\frac{e^x + e^{-x}}{2}.
\end{equation*}
```

11. fejezet

Sorozatok, sorok és integrálok

11.1. Jellemzők

Bár a sorozatok, a sorok, az integrálás és a differenciálszámítás valamint ezek jelölései csak a középiskola végén illetve a felsőoktatási tanulmányok során szerepelnek, érdemes megismerkedni a legfontosabb jelölésekkel. Ehhez a témakörhöz nem tartozik sok új szimbólum, viszont az utasítások szintaktikáját elemezve érdekes helyzet áll fenn. Nem egyértelmű, hogy az utasítások többségét nullad- vagy másodrangúnak lehet-e tekinteni, a fordító az utasítás után két paramétert vár-e, vagy az mindössze egy szimbólumot szúr be; a határértékek az alsó és a felső index utasítás paraméterének tekinthetőek-e. A szöveg abban az esetben is hiba nélkül lefordul, ha csak az f karaktert írjuk le az `\int` utasítással, határértékek nélkül. A gyakorlatban viszont a szumma és az integrál esetén olyan jelekről beszélünk, amelyek egy alsó és egy felső határértéket várnak.

Érdekesség, hogy a határértékek másként jelennek meg a dokumentumban soron belüli, illetve külön sorban szereplő képlet esetén. Abban az esetben, ha a határértéket tartalmazó matematikai kifejezést a szövegbe írjuk, akkor az indexek a szimbólum után jelennek meg. Külön sorba írt képletek esetén a határértékek a szimbólum fölött illetve alatt jelennek meg. Például, a szumma a `\sum_{[alsó határ értéke]}^{[felső határ értéke]}` kifejezéssel adható meg, ami folyó szövegbe írva $\sum_{n=1}^{\infty} n^2$ formában kerül megjelenítésre. Ugyanez a kifejezés egyenletkörnyezetben így jelenik meg:

$$\sum_{n=1}^{\infty} n^2.$$

A `\limits` utasítás használatával megoldható, hogy folyó szövegben is a szimbólum fölött illetve alatt jelenjenek meg a határértékek. Például a `\sum\limits_{n=1}^{\infty}` utasítássorozat hatására a kifejezés $\sum_{n=1}^{\infty} n^2$ formában jelenik meg.

Gyakran használt jelölések még a `\lim_{arg1}` – például $\lim_{n \rightarrow 0} a(n)$ –, illetve az integrál jelölésére az `\int`, például $\int x dx$ illetve $\int_a^b y^2 dy$. Bizonyos szempontból az integrál egy sornak, összegnek tekinthető, bemutatása ezért történik ebben a fejezetben. Az integráláson kívül a deriválással kapcsolatos legfontosabb jelöléseket is érdemes megemlíteni. A `\dot{arg1}` és a `\ddot{arg1}` utasítások használatosak deriválás esetén. Például $\ddot{x}(t) = \frac{d^2x}{dt^2}$. Az egybetűs változat egy pontot jelenít meg az argumentumban szereplő karakter felett, míg a dupla d betűvel kezdődő parancs két pontot illeszt oda. Az egybetűs jelölést gyakran az időfüggő mennyiségek elsőfokú deriválása során szokás használni, például $\dot{x}(t) = \frac{dx}{dt}$. A dupla pont – a feljebb bemutatott módon – másodfokú idő deriváltak esetén használatos.

11.2. Leíró utasítások

Utasítás	Vizuális megjelenés (példák)	Megnevezés
<code>\sum_{arg1}^{arg2}</code>	\sum_a^b	szumma
<code>\prod_{arg1}^{arg2}</code>	\prod_a^b	produktum
<code>\int_{arg1}^{arg2}</code>	\int_a^b	integrál
<code>\lim_{arg1}</code>	$\lim_{x \rightarrow 0}$	limesz
<code>\dot{arg1}</code>	\dot{x}	pont (deriváláshoz)
<code>\ddot{arg1}</code>	\ddot{x}	dupla pont (deriváláshoz)
<code>\partial</code>	∂	parciális differenciál
<code>\prime</code>	x'	vonás (deriváláshoz)

11.1. táblázat. Utasítások határértékek, sorok és integrálok, differenciálszámítások leírásához.

A következő táblázatban a `\limits` utasítás képletekre gyakorolt hatásait mutatjuk be:

Utasítás	Vizuális megjelenés (példák)
<code>\sum\limits_{arg1}^{arg2}</code>	\sum_a^b
<code>\prod\limits_{arg1}^{arg2}</code>	\prod_a^b
<code>\int\limits_{arg1}^{arg2}</code>	\int_a^b
<code>\lim\limits_{arg1}</code>	$\lim_{x \rightarrow 0}$

Megjegyzés: A középiskolai tanulmányokhoz szükséges utasításokon kívül a LaTeX széles körben támogatja a különféle egyéb integráljelöléseket, például a hurok- illetve a térintegrál megjelenítését.

11.3. Példák

11.3.1. Vizuális megjelenés

Oldja meg az alábbi kifejezést a szumma kibontásával, és az integrálok kiszámításával!

$$y = \sum_{n=1}^3 \int_n^4 x^2 dx$$

Megoldás:

$$\begin{aligned} y &= \sum_{n=1}^3 \int_n^4 x^2 dx \\ &= \sum_{n=1}^3 \frac{x^3}{3} \Big|_n^4 \\ &= \frac{x^3}{3} \Big|_1^4 + \frac{x^3}{3} \Big|_2^4 + \frac{x^3}{3} \Big|_3^4 . \end{aligned}$$

11.3.2. LaTeX forrás

Oldja meg az alábbi kifejezést a szumma kibontásával, és az integrálok kiszámításával!

```
\begin{equation}\nonumber
y = \sum_{n=1}^3 \int_n^4 x^2 dx
\end{equation}
\textbf{Megoldás:}
\begin{eqnarray}\nonumber
y &=& \sum_{n=1}^3 \int_n^4 x^2 dx \\\nonumber
&=& \sum_{n=1}^3 \left. \frac{x^3}{3} \right|_n^4 \\\nonumber
&=& \left. \frac{x^3}{3} \right|_1^4 + \left. \frac{x^3}{3} \right|_2^4 + \left. \frac{x^3}{3} \right|_3^4 , .
\end{eqnarray}
```

12. fejezet

Gráfok és koordinátageometria

12.1. Jellemzők

Bár a LaTeX nagyon hatékony eszköz, nem várható el ettől a leíró nyelvtől sem, hogy szövegesen, összetett utasítások halmazán keresztül egész gráfokat, koordináta-rendszereket lehessen látható (olvasható) módon összeállítani, megszerkeszteni. Haladó felhasználók számára lehetőség van speciális csomagok használatával különféle rajzolási feladatokat parancsokon keresztül elvégezni. Például, lehetőség van Feynmann- illetve különféle téridő diagramok készítésére utasításokon keresztül, melyek vonalakat, téglalapokat, köröket illetve hullámos vonalakat jelenítenek meg a megfelelő utasítás és paraméter segítségével. Az általános- és középiskolai tananyag akadálymentesítése kapcsán egyetlen jellemző LaTeX környezetet lehet megemlíteni, a kép környezet. Bár ez a tapintható gráfok mellett csak kiegészítő lehetőség és nem helyettesítő megoldás, a környezet segítségével képformátumú fájlokat lehet beszúrni a LaTeX forrásba, amelyhez szöveges leírást is lehet fűzni.

12.2. Leíró utasítások

Bár a függvények grafikus formában történő megadása korábban is előfordulhat a tanulmányok során, jellemzően az utolsó két évfolyam alatt fektetnek nagyobb hangsúlyt a képi megjelenítésre. Ahhoz, hogy egy gráf tartalmát szövegesen is meg lehessen jeleníteni a technikai képleírás szabályait, gyakorlatát figyelembe véve, a következő LaTeX környezetet kell alkalmazni:

```
\begin{center}
%A középre igazítás kezdetének jelölése.
\begin{figure}[h]
%A környezetben a [h] opcionális, és arra utal, hogy a kép pontosan ezen a helyen
jelenjen meg. A {figure} környezet a képre vonatkozó utasítások halmazát jelöli.
\includegraphics[width=5cm]{fájl_név}
%Az opcionális paraméter a kép megjelenített szélességét adja meg, majd a kötelező
paraméter a képfájl nevét.
\caption{Ide kerül a kép leírása.}
%Itt lehet részletes, a látássérült olvasóknak szánt képleírást adni a gráfról.
\end{figure}
%A képi környezet végének jelölése.
\end{center}
```

%A középre igazított rész végének jelölése.

Figyeljük meg, hogy a fájlnev után nincs kiterjesztés! Egyes LaTeX fordítók használatakor .jpg, .png vagy más ismert kiterjesztésű fájlt is be lehet illeszteni. Bizonyos esetekben azonban, amikor a LaTeX fordító a PDF fájlt próbálja előállítani, a kép .eps formátumú verzióját igényli. Ez magyar nyelvű dokumentumok parancssorból történő fordításakor is előfordulhat. Ebben az esetben csak a kiterjesztés nélküli .eps fájlokat lehet feldolgozni. A **pdflatex.exe** fordító a jpg kiterjesztésű fájlokat fogadja el, míg a **latex.exe** csak az .eps kiterjesztésűeket kezeli.

A kép méreteit is meg lehet adni, illetve azt, hogy a szöveghasábon belül hol helyezkedjen el, milyen széles legyen a grafika. A LaTeX úgynevezett lebegő objektumoknak értelmezi a képeket és a táblázatokat. Ez azt jelenti, hogy a képet az általa optimálisnak tartott helyre illeszti be, így az nem mindig oda kerül, ahol az utasítás a szövegben előfordul. Ezért szükség lehet a [h], a [t] illetve a [b] opcionális paraméterekre, amelyek azt adják meg, hogy a kép pontosan a megadott helyen, a lap tetején vagy az alján szerepeljen. Leírást is be lehet szűrni a kép alá a már ismertetett módon.

Az alábbi első táblázat a lebegő objektumok (**figure**) elhelyezésére, míg a második a kép méretezésére szolgáló opcionális paramétereket összegzi (a `\includegraphics` utasításhoz):

Opcionális paraméter	Jelentése
[t]	A lap tetején kerül a kép elhelyezésre (top).
[h]	Amennyiben technikailag lehetséges, pontosan a megadott helyen kerül a kép elhelyezésre (here).
[b]	A lap alján kerül a kép elhelyezésre (bottom).

12.1. táblázat. Opcionális paraméterek a lebegő környezet pozicionálásához.

Opcionális paraméter	Jelentése
[width=3cm]	A kép szélessége centiméterben megadva.
[height=2cm]	A kép magassága centiméterben megadva.
<code>\textwidth</code>	A kép a szöveg szélességéhez kerül méretezésre.
<code>\columnwidth</code>	Kéthasábos dokumentum esetén a kép a hasáb szélességéhez kerül méretezésre.

12.2. táblázat. Opcionális paraméterek a képek méretezéséhez.

Megjegyzés: Abban az esetben, ha a kép szélességét határozzuk meg, nincs szükség a magasság beállítására. Ezt a LaTeX automatikusan kiszámolja az oldalarány megőrzésével.

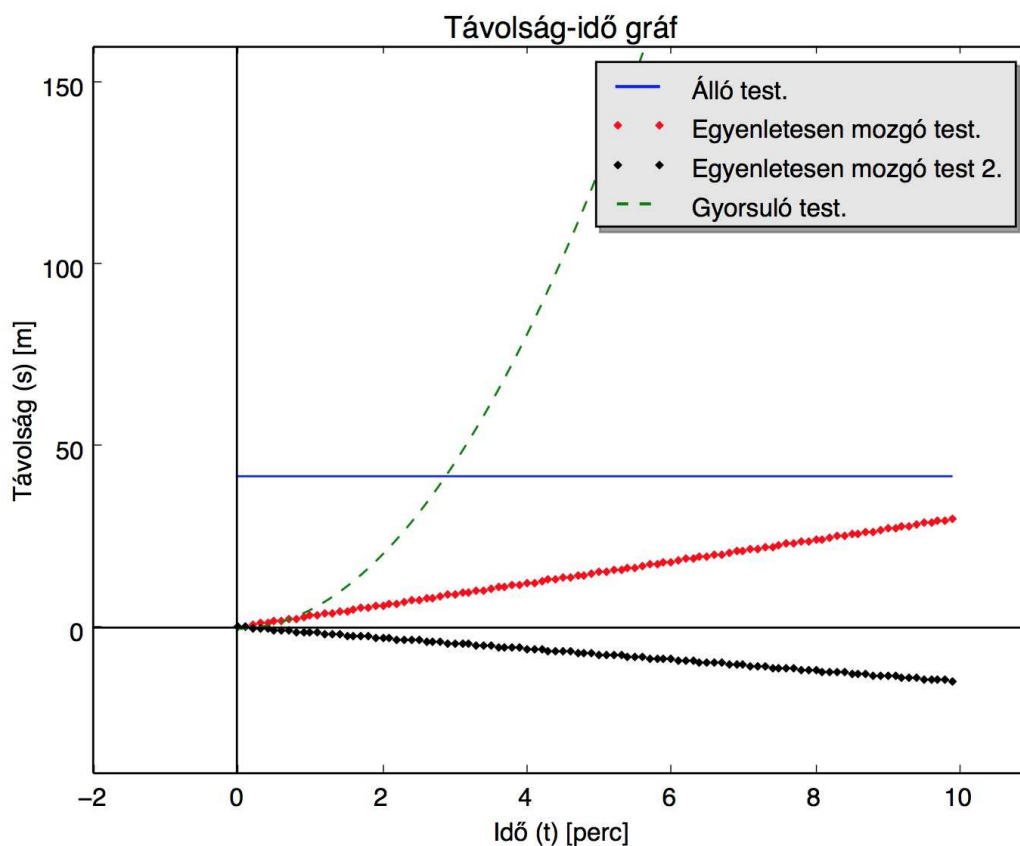
Fontos még a `\textwidth`, illetve `\columnwidth` helyes használata a szöveg hasábkra bontásához. Amennyiben az oldal egyhasábos szöveget tartalmaz, ami 15 cm széles, a `0.8\textwidth` használatával egy 12 centiméter széles képet szűrhatunk be. A szöveg szélessége előtt álló együttható adja meg az ábra és szöveg szélességének arányát. Ez a módszer azért is rugalmas megoldás, mert a margó változtatása

során a program a képet automatikusan hozzáigazítja a szöveg szélességéhez. Figyeljük meg, hogy az angol nyelv szabályai szerint tizedespont és nem tizedesvessző szerepel a tört együttható értékében! Két- vagy többhasábos dokumentum esetén, a `\columnwidth` utasítással érhetjük el, hogy a kép ne lógjon át a következő hasábra. Ez a parancs automatikusan a hasáb szélességét veszi figyelembe.

12.3. Példák

12.3.1. Vizuális megjelenés

Jellemezze a 12.1. ábrán bemutatott gráfon látható négy test mozgását a fizika törvényei alapján.



12.1. ábra. A képen a „Távolság-Idő gráf” cím olvasható. Az x-tengelyen az idő fut -2 perctől 11 percre, 2 perces skálafelosztásban. Az y-tengelyen a távolság méterben van megadva -40 méter és 160 méter között. A gráf négy görbét tartalmaz. Kék színnel egy vízszintes folytonos vonal látható, körülbelül 40 méternél. Egy piros pontozott vonal átlósan felfelé halad. Ennek meredeksége nagyjából 3 m/perc. Egy fekete pontozott vonal átlósan lefelé halad, 1,5 m/perc meredekséggel. Végül a zöld szaggatott vonal egy parabolikus görbét mutat pozitív együtthatóval.

12.3.2. LaTeX forrás

Jellemezze a `\ref{fig:graph1}`. ábrán bemutatott gráfon látható négy test mozgását a fizika törvényei alapján.

```
\begin{center}
\begin{figure}[h]
\includegraphics[width=16cm]{graph1.jpg}
\caption{A képen a „Távolság-Idő gráf” cím olvasható. Az x-tengelyen az idő fut -2 perctől 11 percig, 2 perces skálafelosztásban. Az y-tengelyen a távolság méterben van megadva -40 méter és 160 méter között. A gráf négy görbét tartalmaz. Kék színnel egy vízszintes folytonos vonal látható, körülbelül 40 méternél. Egy piros pontozott vonal átlósan felfelé halad. Ennek meredeksége nagyjából 3 m/perc. Egy fekete pontozott vonal átlósan lefelé halad, 1,5 m/perc meredekséggel. Végül a zöld szaggatott vonal egy parabolikus görbét mutat pozitív együtthatóval.}
\label{fig:graph1}
\end{figure}
\end{center}
```

III. rész

További hasznos tudnivalók

A gyakorlati tapasztalatok összegzése

Ebben a részben a gyakorlati tapasztalatok összegzése történik meg. Hasznos tudni, hogy melyek azok a csomagok, amelyeket érdemes gyakran használni; ezeket hogyan lehet telepíteni, milyen előnyeik illetve hátrányaik vannak. Eddig azt feltételeztük, hogy a Windows részét képező Jegyzettömb programmal történik a forrásfájl kezelésére, de valójában erre a feladatra sokkal kifinomultabb LaTeX szerkesztők állnak rendelkezésre. Végül arról is érdemes egy pár szót ejteni, hogy a PDF előállítása mivel jár, mik a lehetséges megoldási módszerek, amikor a TeX forrásfájl fordítása közben hiba lép fel.

13. fejezet

A tartalom szerkesztésén túl

13.1. TeX forrásszerkesztő programok

A HTML dokumentumok szerkesztéséhez hasonlóan, a szerzők a legtöbb esetben nem manuálisan írják meg a forráskódot, hanem sablonokat, erre specializálódott szövegszerkesztőket használnak. Az ilyen szerkesztőprogramok kiegészítő lehetőségeket tartalmaznak, amelyek lehetővé teszik a listakörnyezetek, táblázatok, képek beszúrását a menüsorból, grafikus gombokon vagy menüelemeken keresztül. A LaTeX utasítások beillesztésre kerülnek a forrásszövegbe, a felhasználónak csak a helyjelölő karakter(ek)e)t kell lecserélnie az aktuális paraméter értékére. Például, a táblázat teljes szerkezetét egy gombnyomással be lehet illeszteni, majd a cellák értékének helyén található * szimbólumokat át kell írni számra, karakterláncra, stb.

Ezen kívül a szerkesztőprogramok olyan beépített funkciókkal rendelkeznek, amelyek a szavak megszámlálásakor figyelmen kívül hagyják az utasításokat, illetve az egyéb metaszövegeket. A forrás fordítása sokszor automatikus, vagy egy billentyűparancs aktiválásával elérhető. Látó szerzők számára sokszor színelőemelés is jelzi az adott hiba helyzetét, jellegét.

Számos szerkesztőprogram létezik, ezek a legtöbb operációs rendszeren elérhetőek, például:

- TeXStudio (<https://www.texstudio.org/>) – nem akadálymentes, de a látó felhasználók körében széleskörűen elterjedt,
- Notepad++ (<https://notepad-plus-plus.org/>) – képernyőolvasó programmal is könnyen kezelhető,
- WinEdt (<https://www.winedt.org/>) – képernyőolvasó programmal is könnyen kezelhető.

Ezen kívül online LaTeX szerkesztőprogramok is elérhetőek, például az Overleaf (<https://www.overleaf.com/>) vagy a ShareLaTeX (<https://www.sharelatex.com/>) szolgáltatások. Ezeknek a szolgáltatásoknak az a nagy előnye, hogy a csomagok kezelése a szolgáltatóra hárul, illetve több szerző egyszerre dolgozhat egy dokumentumon. Azonban ezek a megoldások sokszor nagyon nehezen, vagy egyáltalán nem használhatóak képernyőolvasó programok segítségével.

13.2. Tudnivalók a TeX csomagok használatáról

A TeX disztribúciókhoz letölthető csomagok ismertetéséhez valós példát veszünk alapul, az ehhez a dokumentumhoz szükséges .tex fájl fejlécének tartalmát. Az alábbiakban sorról sorra bemutatjuk,

hogy milyen csomagokra és miért van szükség, illetve ezeket hogyan érdemes csoportosítani. Végül szó lesz néhány, a csomagok kezelésével kapcsolatos hasznos tudnivalóról is.

A csomagokat a dokumentumosztály meghatározása után, a dokumentum törzse elé szokás elhelyezni, a `\documentclass` és a `\begin{document}` sorok között. Az alábbi táblázat első oszlopa egy az egyben tartalmazza jelen kiadvány összes csomagját (ugyanilyen sorrendben), a második oszlopban pedig a hozzá kapcsolódó jelentéseket (esetleg példákat) találunk.

Csomag neve	Mire való (példák)
<code>%--packages--</code>	Itt jelezzük, hogy a következőkben csomagokat kívánunk betölteni.
<code>%+++language specific+++</code>	Itt nyelv- és karakterkódolás specifikus csomagokat töltünk be.
<code>\usepackage[utf8]{inputenc}</code>	A beviteli karakterkódolást UTF8 szabványra állítjuk.
<code>\usepackage[T1]{fontenc}</code>	A kimeneti karakterkódolást a T1 csoportra állítjuk latin ékezetes betűkkel.
<code>\usepackage[magyar]{babel}</code>	A magyar nyelv szabályai szerint kívánjuk a dokumentumot előállítani. A babel csomag segítségével több nyelv elválasztási, mondatok közötti szünethosszának szabályait adhatjuk meg. A magyar opcionális paraméter felel azért is, hogy a kulcsszavak, például a fejezetek címe, a tartalomjegyzék és a táblázatok magyar nyelven jelenjen meg a dokumentumban.
<code>%+++floating objects, tables+++</code>	Táblázatokra és képekre, úgynevezett lebegő objektumokra vonatkozó utasítások.
<code>\usepackage{graphicx}</code>	Képek betöltését teszi lehetővé.
<code>\usepackage{float}</code>	Képek, táblázatok pozicionálását engedi rugalmasabb utasításokon keresztül.
<code>\usepackage{longtable}</code>	Több oldalon átívelő táblázatok szerkesztését teszi lehetővé.
<code>\usepackage{tabu}</code>	Egy adott táblázatstílushoz szükséges csomag.
<code>\usepackage{array}</code>	Matematikai mátrixok, táblázatok, egyenletrendszerekhez szükséges.
<code>\usepackage{makecell}</code>	Táblázat fejléceinek formázásához szükséges utasítások tárháza.
<code>\usepackage{caption}</code>	Táblázatok és képek aláírásait kezeli rugalmasabb utasításokon keresztül.
<code>%+++Mathematics specific+++</code>	Matematikai tartalmakhoz szükséges utasítások csomagjai.

Csomag neve	Mire való (példák)
<code>\usepackage{amsmath}</code> <code>\usepackage{amssymb}</code> <code>\usepackage{amsfonts}</code>	Az American Mathematical Society (AMS) hivatalos LaTeX csomagjai. Ezek a legtöbb matematikai szimbólumot, betűtípust tartalmazzák, illetve sok hasznos matematikai környezetet definiálnak.
<code>\usepackage{mathtools}</code>	Az AMS csomagokból esetlegesen hiányzó utasításokat bővíti ki.
<code>\usepackage{cancel}</code>	Lehetővé teszi a matematikai kifejezések áthúzását, például törtek egyszerűsítése során.
<code>%\usepackage{MnSymbol}</code> <code>%\usepackage{kmath}</code> <code>%\usepackage{kpfonts}</code> <code>%\usepackage{mathabx}</code> <code>%\usepackage{mathdesign}</code> <code>%\usepackage{oz}</code> <code>%\usepackage{pxfonts,txfonts}</code> <code>%\usepackage{placeins}</code>	A kommentekkel ellátott utasítások olyan csomagokat töltenek be, amelyek további matematikai karakterek megjelenítésére képesek a dokumentáció szerint. Azonban, ezek felülírhatják az ams csomagok utasításait, ezzel problémákat idézve elő, például nem vagy rosszul megjelenő karaktereket.
<code>%+++format packages+++</code>	Dokumentum külalakját formázó csomagok.
<code>\usepackage[margin=2cm]{geometry}</code>	A margó szabályozását teszi lehetővé.
<code>\usepackage{lmodern}</code>	Ékezhelyes dokumentumok előállítása során lehet szükség erre a csomagra.
<code>\usepackage{setspace}</code>	A térközök rugalmasabb kezeléséért felelős modul.
<code>%+++else+++</code>	Más, esetleges csomagok használata.
<code>\usepackage{hyperref}</code>	Hiperhivatkozások megjelenítését teszi lehetővé.
<code>%---macro definitions---</code>	Az alábbi rész nem csomagokat tölt be, azonban a fejléc arra is használható, hogy saját magunk által definiált utasításokat tudjunk használni. Például, hosszú, de gyakran használt csomag esetén rövidebb megfelelőt hozhatunk létre.
<code>\newcommand{\bs}{\textbackslash}</code>	A <code>\</code> karakternek <code>\textbackslash</code> a parancsát rövidíti a <code>\bs</code> parancsra.

13.1. táblázat. Jelen kiadvány összes csomagja és azok jelentése.

Ahogy az feljebb látható, a másodrangú `\newcommand` utasításban az első paraméterben adható meg a személyre szabott utasítás, majd a második argumentum szolgál annak az eredeti utasításnak megadására, amit át kívánunk nevezni. Így akár arra is lehetőség van, hogy utasításokat magyarosítsunk, például a `\times` helyettesíthető lenne a `\szorzas` utasítással. Ez azonban nem ajánlott.

Mindezen kívül érdemes azt tudnunk, hogy a csomagok betöltési sorrendje fontos tényező lehet. Előfordul, hogy egy csomagban meghatározott utasítás felülírja egy másik csomagban található azonos nevű parancs jelentését. Ezért olvashatjuk webes dokumentációkban, online fórumokon, hogy adott csoma-

gokat ajánlott a fejléc végére beilleszteni. Ilyen például a hyperref csomag, ami a hiperhivatkozásokat képes megjeleníteni a PDF dokumentumokban a fordítást követően.

Ezen kívül érdemes tudni, hogy a telepített TeX környezetek általában beépített csomagkezelőkkel rendelkeznek, amelyekkel új csomagokat lehet telepíteni, törölni, frissíteni, stb.

13.3. A forrásfordítás technikái és problémái

Az elektromechanikus számítógépek idején, a felmelegedésre hajlamos gépek gyakran vonzották a rovarokat, amelyek adott esetben tönkre is tehetek a reléket, így akadályozva a program futását. Innen ered a *bug* angol eredetű kifejezés, ami bogarat jelent. Ahogy annak idején, a programokból most is el kell távolítani a bogarakat, ez olyan folyamat, amit debugolásnak neveznek. A LaTeX használata esetén is ugyanez a helyzet. A .tex forrás PDF dokumentummá alakítása során, ha egy utasítás nem megfelelően van megírva, a fordítóprogram hibát jelez. Optimális esetben a hiba helye – a forrásfájlbeli sor száma – is megadásra kerül. Nyilvánvaló hibák esetén – például lemaradt zárójel – ezek könnyen kiküszöbölhetőek.

Azonban az is előfordul, hogy a hiba helyeként a fordítóprogram az első sort jelzi, pedig meggyőződünk arról, hogy az helyes. Ilyen esetben érdemes a szövegszerkesztő beállításait átnézni. Bizonyos esetekben a szövegszerkesztők egy úgynevezett BOM karaktert helyeznek el a dokumentum elején. Ezt a jelölőkaraktert a LaTeX fordítók általában nem tolerálják, ezért el kell távolítani azt. Ez a Jegyzetömb program használata esetén gyakran előfordul.

Az is előfordulhat, hogy a fordítóprogram nem ad semmilyen utalást arra, hogy melyik sorban van a hiba. Ebben az esetben érdemes a következő módszert alkalmazni. Helyezzük át a dokumentum végét jelző utasítást (`\end{document}`) az első bekezdés utáni sorba. Ha nincs hiba, hozzuk a dokumentum végét egy bekezdéssel lejjebb, majd folytassuk ezt az eljárást addig, amíg nem találjuk meg a hibát tartalmazó részt.

Érdemes azt is megemlíteni, hogy a fordítási hibákon kívül figyelmeztetések is léteznek. Ezek olyan hibát jeleznek, amelyek nem akadályozzák a PDF dokumentum előállítását, azonban a formázással gondok adódhatnak a vizuális megjelenítés során. Gyakori példa, hogy egy egyenletet hosszan írunk sortörés nélkül, így az túlfut a margón.

Fordításkor figyeljük meg, hogy nem csak a PDF fájl áll elő, de több kiegészítő állomány is létrejön. Ilyen fájlok például a .aux, .bbl, .blg, .log, stb. dokumentumok. Ezek különféle metaadatokat tartalmaznak. Vegyük például a .log állományt, ami a fordítási napló üzeneteit tárolja. Ezek sokszor szükségtelen fájloknak tűnnek, de a fordítóprogram ezekből is dolgozik. Tartalomjegyzéket is tartalmazó dokumentum esetén a **pdflatex.exe** program az első fordítás során kigyűjti a fejezetek, alfejezetek címét és oldalszámát, majd ezeket a .toc fájlba írja. A második fordítási menet során a program ebből a fájlból olvassa ki a szükséges metaadatokat, amelyek alapján a PDF kívánt helyén elhelyezi a tartalomjegyzéket. Hasonlóképpen, amikor egy publikációban vagy egy tankönyvben más műből idézünk, egy referencialistát, más néven bibliográfiát kell létrehozni. Ahhoz, hogy helyesen jelenjenek meg a referenciaszámozások illetve a dokumentum végén található bibliográfia, három lépésben kell lefordítani a dokumentumot. Első lépésben a **pdflatex.exe** program kigyűjti a referenciakulcsokat, és kiegészítő fájlba írja ezeket. Ezt követően a **bibtex.exe** nevű program futtatása szükséges, ami ezeket a kulcsokat a referenciacímekkel párosítja, majd a megfelelő segédfájlba írja. Ezt követően még egyszer le kell futtatni a **pdflatex.exe** programot, ami a végleges PDF dokumentumot előállítja, ezúttal már helyesen megjelenő referenciaszámozással. Ha csak egyszer történik meg a dokumentum lefordítása, akkor

sok esetben kérdőjel karakterek jelennek meg – helytelenül – a referencia helyén. Megfigyelhető, hogy a helyes kimeneti dokumentum előállításához sokszor több lépésen keresztül történik meg. Ezek általában billentyűparancs vagy menüpont használatával hajthatók végre a grafikus szövegszerkesztőkkel. Ez megkönnyíti a felhasználók dolgát, de időnként problémát is okozhat.

Több esetben a **pdflatex.exe** program nem rendelkezik olyan funkciókkal, amikre szükség van. A magyar nyelvű dokumentumok esetén a pdflatex.exe helytelenül állítja elő az ékezetes karakterek képernyőolvasó programmal feldolgozható alakját. Így esetenként a latex.exe programmal kell a fordítást elvégezni, ami .dvi formátumú kimeneti fájlt állít elő. Ezeket további programok segítségével lehet PDF állománnyá alakítani, ahogy az az első fejezetben részletesen bemutatásra került.

Végezetül, érdemes a .sty stílusfájlokról is szót ejteni. Ezeket a fájlokat általában kiadók készítik, és sablonként szolgálnak a dokumentum külalakjához. Vannak fordítók, amelyek ezeket a fájlokat akkor is megtalálják, ha a forrásfájlokat tartalmazó mappába helyezük. Erre jó példa a MacTeX nevű disztribúció Macintosh operációs rendszerekre. Azonban a tapasztalat azt mutatja, hogy a Windows-os MikTeX rendszer alapértelmezetten ezt nem kezeli, és a .sty fájlokat az adott rendszer elérési útvonalán kell tárolni. A stílusfájl használata túlmutat ennek a tananyagának a kitűzött célján, de jól illusztrálja, hogy nincs egyetlen standard fordítási program, rutin, ami minden esetben használható. A feladattól függően más és más módszert kell alkalmazni. Azonban a legtöbb esetben a LaTeX szövegszerkesztő program sok hasznos funkcióval rendelkezik, ami a munkát megkönnyíti.

IV. rész

Források és további referenciák

Források listája

Ebben a fejezetben olyan listát adunk közre, amely további hasznos segítséget nyújthat a LaTeX használatának elsajátítása során, illetve amelyeket a szerző is használt jelen kiadvány elkészítése során.

Megjegyzés: Az alábbi források használatán kívül javasoljuk az internet, illetve az internetes keresők használatát. A világhálón számos fórum, cikk, stb. található, amely a dokumentum szerkesztése közben felmerült problémákat és azok megoldását tárgyalja. Ne feledjük, viszonylag kevés olyan probléma fordul elő, ami valaki másnál még nem merült fel!

14. fejezet

Hasznos olvasnivalók

- **LaTeX 69 percben:** <https://www.szabilinux.hu/forditasok/lkh/konyvek/lkh.html>
- **BME math LaTeX:** <https://math.bme.hu/latex/>
- **Unikód szimbólumok konvertálása LaTeX utasítássá:**
https://www.johndcook.com/unicode_latex.html
- **A LaTeX Access projekt (angol nyelvű weboldal):** <http://latex-access.sourceforge.net>
- **ELTE LaTeX tananyag:** <http://people.inf.elte.hu/alagi/elte/bevinfo/latex.htm>
- **Példa TeX dokumentum és tananyag:**
<http://titan.physx.u-szeged.hu/gaspara/labor2/doksi/latex.tex>
- **Szerkesztés LaTeX-ben:** https://hu.wikibooks.org/wiki/Szerkesztések_LaTeXben