

**GÁBOR DÉNES**  
**FŐISKOLA**

**A MATEMATIKA ELEKTRONIKUS  
DOKUMENTÁLÁSA:**  
*MATHML, OPENMATH VAGY T<sub>E</sub>X?*

**sorszám: 682/1999**

**GAJÁRSZKI LÁSZLÓ**

**TATABÁNYA**

**1999**



# TARTALOMJEGYZÉK

<b>TARTALOMJEGYZÉK .....</b>	<b>3</b>
<b>ELŐSZÓ .....</b>	<b>6</b>
<b>I. RÉSZ: BEVEZETŐ .....</b>	<b>8</b>
1. A matematika és lejegyzése .....	8
2. A leírónyelvek története.....	9
2.1. Normál leírónyelvek .....	9
2.1.1. A T <sub>E</sub> X története .....	9
2.1.2. Egyéb nyelvek .....	10
2.2. Hipertextes leírónyelvek.....	10
2.2.1. A HTML története.....	10
2.2.2. A MathML története.....	11
2.2.3. Az OpenMath története.....	12
2.3. A HTML korlátai .....	12
2.3.1. Megjelenítési akadályok.....	13
2.3.2. Kódolási gondok .....	14
2.4. A matematikai leírónyelvek követelményei .....	14
2.5. A matematikai leírónyelvek céljai .....	16
<b>II. RÉSZ: ALAPOK .....</b>	<b>18</b>
1. A MathML alapjai .....	18
1.1. Általános áttekintés.....	18
1.1.1. A MathML elemek rendszerezése.....	19
1.1.2. Kifejezés-fák és token-elemek.....	20
1.1.3. Megjelenítő jelölés .....	22
1.1.4. Tartalmi jelölés.....	23
1.1.5. A megjelenítés és tartalom ötvözése .....	24
1.2. Néhány példa .....	25
1.2.1. Megjelenítő jelölés példái.....	25
1.2.2. Tartalmi jelölés példái .....	27
1.2.3. Példák a jelölések ötvözésére .....	29
1.3. A MathML szintaxisa és nyelvtana .....	31
1.3.1. Bevezető az XML szintaxisába .....	32
1.3.2. MathML attribútumok értékei.....	32
1.3.3. Üres helyek használata .....	33
2. Az OpenMath áttekintése .....	34
3. A T <sub>E</sub> X áttekintése .....	37
<b>III. RÉSZ: RÉSZLETES ÖSSZEHASONLÍTÁS .....</b>	<b>40</b>
1. A MathML megjelenítő jelölése .....	40
1.1. A megjelenítő elemekről általában.....	40
1.2. Megjelenítő token-elemek .....	41
1.2.1. A token-elemek attribútumai.....	42
1.2.2. Azonosítók .....	43
1.2.3. Számok .....	45
1.2.4. Operátorok .....	47
1.2.5. Szövegek.....	53
1.3. Stílusok .....	56
1.4. Üres helyek kihagyása .....	60
1.5. Normál szövegektől eltérő szerkezetek .....	64
1.5.1. Kifejezések csoportba rendezése .....	64

1.5.2. Törtek .....	65
1.5.3. Gyökös kifejezések .....	67
1.5.4. Zárójelek .....	68
1.6. Indexek és ékezetek.....	70
1.6.1. Indexek .....	71
1.6.2. Ékezetek a kifejezések körül .....	73
1.7. Táblák és mátrixok.....	76
1.7.1. Táblák, mátrixok felépítése .....	76
1.7.2. Egyenletek illesztése .....	81
1.8. Egyéb elemek.....	84
1.8.1. Hibakezelés.....	84
1.8.2. Vezérlő szerkezetek .....	86
2. A tartalmi jelölés .....	87
2.1. A tartalmi jelölés elemeiről általában .....	87
2.2. Tokenelemek.....	89
2.2.1. Számok .....	89
2.2.2. Azonosítók .....	91
2.3. Tartalmi elemek csoportba rendezése .....	92
2.4. Deklarációk .....	93
2.5. Aritmetika, algebra, logika .....	97
2.5.1. Alapműveletek .....	97
2.5.2. Fontosabb aritmetikai, algebrai műveletek .....	99
2.5.3. Fontosabb aritmetikai, algebrai függvények .....	103
2.5.4. Logikai műveletek.....	105
2.5.5. Relációk.....	109
2.6. Függvénytan .....	113
2.6.1. Felhasználói függvények .....	113
2.6.2. A $\lambda$ függvény .....	115
2.6.3. Függvényekkel végzett műveletek .....	116
2.7. Integrál- és differenciálszámítás .....	117
2.7.1. Integrálszámítás.....	117
2.7.2. Differenciálszámítás .....	120
2.8. Halmazelmélet.....	121
2.8.1. Halmazok megadása .....	121
2.8.2. Halmazműveletek .....	124
2.9. Sorozatok .....	126
2.10. Trigonometria .....	130
2.11. Statisztika .....	131
2.12. Lineáris algebra .....	133
2.12.1. Vektorok.....	133
2.12.2. Mátrixok .....	134
2.12.3. Mátrixműveletek .....	135
2.13. Szemantikai elemek .....	137
3. A jelölések ötvözése.....	138
3.1. Miért használunk kétféle jelölést?.....	138
3.2. A két jelölésfajta összekapcsolása .....	139
3.2.1. Megjelenítő elemek a tartalmi elemek között.....	139
3.2.2. Tartalmi elemek a megjelenítő elemek között.....	140
3.3. Makrók készítése .....	141
4. Az interfész.....	142
4.1. A MathML HTML-be ágyazása.....	142
4.1.1. A $\langle \text{math} \rangle$ elem.....	142
4.1.2. Egy egyszerű MathML példa .....	145
4.2. Az OpenMath HTML-be ágyazása.....	146
4.2.1. Az $\langle \text{OMOBJ} \rangle$ elem.....	146
4.2.2. Egy egyszerű OpenMath példa.....	147
4.3. A $\text{T}_{\text{E}}\text{X}$ kapcsolatai .....	148

## IV. RÉSZ: JÖVŐKÉP .....149

1. A MathML jövője .....	149
1.1. Stíluslapok .....	149
1.2. Makrók .....	150
1.3. Egyenletrendszerek illesztése .....	151
1.4. Az interfész hiányosságai .....	151
2. Az OpenMath jövője .....	152
2.1. A Content Dictionaryk előnye.....	152
2.2. Hátrányok .....	152
3. Összegzés.....	153
<b>A FÜGGELÉK: SPECIÁLIS MATEMATIKAI JELEK .....</b>	<b>155</b>
1. Általában a speciális jelekről .....	155
2. A speciális jelek részletesen .....	156
2.1. MathML specifikus jelek .....	156
2.3. Betűk .....	157
2.3.1. Görög betűk.....	157
2.3.2. Gót betűk .....	158
2.3.3. Duplaszárú betűk.....	159
2.3.4. Kalligrafikus betűk .....	159
2.4. ASCII karakterek .....	160
2.5. Matematikai ékezetek .....	161
2.6. Nyilak.....	162
2.7. Különféle műveleti és relációs jelek .....	166
2.8. Egyéb jelek .....	180
<b>B FÜGGELÉK: AZ OPENMATH KÓDFORMÁI .....</b>	<b>185</b>
<b>C FÜGGELÉK: AZ OPENMATH CD-I .....</b>	<b>187</b>
<b>D FÜGGELÉK: MATEMATIKAI SZOFTVEREK .....</b>	<b>189</b>
<b>E FÜGGELÉK: A CD MELLÉKLET TARTALMA.....</b>	<b>190</b>
<b>IRODALOMJEGYZÉK.....</b>	<b>191</b>
<b>TÁRGYMUTATÓ .....</b>	<b>192</b>

## ELŐSZÓ

Diplomamunkámban a matematikai művek elektronikus úton történő dokumentálásának lehetőségeit szeretném bemutatni.

A lehetőségek közül a hármat választottam ki, a matematika szakterületén igen elterjedt *TEX* szerkesztőnyelvet és a még újnak számító *MathML* és *OpenMath* leírónyelveket szeretném összehasonlítani.

Mivel a *MathML* (*Mathematical Markup Language*) és az *OpenMath* a matematikai területen dolgozók körében sem túlzottan ismert nyelv, ezért a nagyobb területet átfogó *MathML* struktúráján keresztül szeretném megismertetni a dokumentálás lehetőségeit, összehasonlítva a hasonló célt szolgáló *TEX* és *OpenMath* eszközeivel.

Mivel a két utóbbi leírónyelv elég frissnek tekinthető, ezért még angolul sem található róla sok szakirodalom, így főképp a *World Wide Web Consortium* által 1998 elején kiadott *MathML* ajánlására és az 1999 elején kiadott *OpenMath* szabványra támaszkodva tervezem megírni szakdolgozatomat.

A téma részletes ismertetése előtt egy áttekintő összefoglalást kívánok írni a fent említett három matematikai leíró nyelvről, melyben kitérek e nyelvek rövid történetére, beleértve a témámnak választott *MathML* nyelvet is.

Témám tényleges kidolgozását a leíró nyelvek alapjainak bemutatásával kezdem. Ezután a nyelvek konkrét elemeit hasonlítom össze. Szó lesz az alapelemekről, a matematikai jelek lejegyzésének formáiról, a képletek jelölésrendszeréről, valamint az összetett matematikai fogalmak (mátrixok, táblák) megjelenítéséről.

Befejezésül a három nyelv jövőben történő alkalmazásának, valamint továbbfejlesztésének lehetőségeit szeretném bemutatni.

Dolgozatom függelékében ismertetem a leggyakoribb, matematikai jelek *MathML* és *T<sub>E</sub>X* megfelelőit, valamint megadok néhány forrást, ahol a bemutatott leíró nyelvekkel kapcsolatos szoftverekről találhatunk információkat.

Dolgozatomhoz egy CD-t is mellékelek, melyen néhány jól használható matematikai szerkesztőszoftver és egyéb segédprogram, valamint munkám alapját szolgáló eredeti dokumentációk találhatók.

Úgy gondolom, hogy mivel a *MathML*-ről és az *OpenMath*-ról magyar nyelven még nem jelent meg semmilyen szakirodalom, a nyelvek bemutatásával úttörő munkát vállalok. A téma hosszabb távon nagy lehetőségeket rejt, munkám fontossá válhat a hazai érdeklődők körében.

Ebben a dokumentumban feltételezek némi Internetes alapismeretet, valamint a HTML nyelv felhasználói szintű ismeretét.

# **I. RÉSZ: BEVEZETŐ**

## **1. A matematika és lejegyzése**

A matematika leglátványosabb ismertetőjegye az összetett, kétdimenziós képletek nagyszámú használata. Természetesen magát a matematikát és lejegyzését nem lehet egy és ugyanak a dolognak tekinteni. A matematikai elméletek jelölésüktől függetlenül léteznek, ezek csak leírják azokat. Azonban a jelentést és a jelölést csak egy hajsza választja el, így a matematika erejének egy részét képzi a képletes formákban való megjelenítés, mely az elemzésen alapul. A kihívás a matematika lejegyzésében az, hogy a jelölésrendszert és a tartalmat lehetőleg egyszerre jelenítsük meg a dokumentumokban, amelyek képesek az elektronikus médiumokon való kapcsolattartásra és a gyakorlatban lehetővé teszik a nyomtatott megjelenítést.

A matematikai jelölésrendszer állandóan alakul, ahogy az emberiség folytatja az egyre újabb, kifejező elméletek felfedezését. Még az elkoptatott, közhelyes számtani jelölések is keresztülmentek érdekes stílusváltozásokon. A modern matematikai jelölésrendszer századok finomításának eredménye, így a jelölési szokások – a kiváló minőségű nyomdai fejlődésnek köszönhetően – meglehetősen bonyolultak lettek. A legegyszerűbb példa erre a változókat jelölő betűk, melyeket tényleges számértékek helyett használunk, ma dőlt betűkkel jelöljük:  $a$ . Hasonlóan, az operátorok (+, -, ·, /) körüli üres helyek jelentősen különbözik a szövegektől, így jelölve az operátorok elsőbbségét. Számos könyvet szenteltek a matematikai jelek nyomdai formáinak és szokásainak leírására, a felső és alsó indexektől a zárójelek

méretének megválasztásának szabályaiig, főként az egyes matematikai területek speciális jelöléseinek leírására.

A matematikai jelölésrendszer, és általában a nyomtatott szöveg vezeti a szemet, a nyomtatott kifejezéseket sokkal könnyebb olvasni és megérteni. Habár természetesnek vesszük, szokások százaira építünk, mint például a bekezdések, a nagybetűk, a betűtípusok, valamint a szakaszok tízes-számrendszerű jelölése, amit ebben a dokumentumban is használok. Ugyanilyen fontosak ezek a szokások az elektronikus médiában, ahol ráadásul meg kell küzdeni a képernyőn való megjelenítés nehézségeivel is.

A matematika elektronikus dokumentálása tehát több mint útkeresés a tradicionális matematikai jelölések elektronikus úton történő fizikai megjelenítésénél. A világháló egy gyökeres változást mutat be az ismeretanyag tárolásának alapjaiban, egy változás, melyben a kölcsönös összekapcsolódás játszik főszerepet. Mindinkább fontossá válik utat találni a matematikával való összeköttetésre, mely elősegíti az automatikus feldolgozást, kereséseket, indexeléseket, valamint a matematikai alkalmazások és összefüggések újrafelhasználását. A kommunikációtechnika e fejlődésével lehetőség nyílik kiszélesíteni képességünket a matematikai tudásunk bemutatásához, kódolásához, végül megosztani azt egymással, így megérteni egymást.

## **2. A leírónyelvek története**

### **2.1. Normál leírónyelvek**

#### **2.1.1. A T<sub>E</sub>X története**

A T<sub>E</sub>X-et *Donald E. Knuth*, a világhírű matematikus fejlesztette ki a hatvanas évek közepén, mivel egyik könyvének

megjelentetése akadályokba ütközött, és az akkori szövegszedő programokat nem találta kielégítőnek. Így nekilátott egy olyan program megírásához, amellyel egy egyszerű számítógép segítségével, nyomdai eszközök nélkül könnyedén lehet matematikai képletekkel, formulákkal tarkított szöveget nyomdai minőségben előállítani.

A  $\text{T}_{\text{E}}\text{X}$  név a  $\tau\epsilon\chi$  szó görög betűs latin átírata (ezért kiejtése is „tech” lett), mely a görög „művészet” szó kezdete. Az „E” betű ejtettsége pedig a program lehetőségeire utal.

A hatvanas években készült programot azóta természetesen továbbfejlesztették, több platformra, különféle fejlesztők által elkészültek változatai, de alapfilozófiája nem változott, alapkövetelményeit minden verzió ugyanúgy teljesíti, mint az első változat. Egy érdekesség: Knuth kívánságának megfelelően a  $\text{T}_{\text{E}}\text{X}$ -változatok verziószáma mindig a  $\pi$ -hez tart.

Diplomamunkámban a Amerikai Matematikai Társaság (*American Mathematical Society*) által kifejlesztett  $\text{AMS-T}_{\text{E}}\text{X}$  változatot vettem alapul.

### **2.1.2. Egyéb nyelvek**

Természetesen számos szövegalapú nyelv létezik a  $\text{T}_{\text{E}}\text{X}$ -en kívül – mint például a *Mathematica*, vagy a *Maple* – azonban ezek szűkebb körben ismertek, általában csak az elkészített munkák lementésére használatosak, ezért ezek bemutatása nem része szakdolgozatomnak.

## **2.2. Hipertextes leírónyelvek**

### **2.2.1. A HTML története**

A HTML fejlesztésekor először kitűzött cél a megjelenítőktől teljesen független, a dokumentum szerkezetére koncentráló

leírásmód volt. Ezért először csak olyan formázó utasítások kaptak benne helyet, amelyek az adott szövegrész dokumentumban elfoglalt helyét határozták meg (címsorok, listák, stb.). A módszer előnye – hogy nem kell törődni a kinézettel, azt minden megjelenítő maga alakítja ki, csak a szerkezetet kell megadnunk – egyben a legnagyobb hátrányává vált: a készítők nem voltak hajlandóak lemondani a dokumentumformázás lehetőségéről, ezért egyre több fizikai formázó utasítás jelent meg a nyelvben. Napjainkra a böngészők már az esetek legnagyobb részében ezeket a formázásokat nagyjából ugyanúgy jelenítik meg az adott HTML forrásból.

A matematika kódolása elektronikus feladatokhoz és kommunikációhoz, mint láttuk, régebbi téma, mint a Web. Kezdetétől, a Web úgy mutatkozott be, mint egy nagyon hatékony információs rendszer, mely nyitott minden különálló egyén előtt. Bár kezdetben a Web tudósok által tudósoknak kitalált és elkészített rendszer volt, mégis a HTML meglehetősen behatárolt eszközökkel rendelkezik matematikai kifejezések leírásához. Napjainkban ezért a matematikusok a szöveges elemeket kénytelenek grafikai elemekkel (GIF, JPG) vegyíteni, amit bonyolult és időigényes elkészíteni.

### **2.2.2. A MathML története**

A *World Wide Web Consortium* (a továbbiakban *W3C*) felismerte, hogy komoly gondot okoz, hogy a rendszer hiányt szenved a tudományos kommunikációt elősegítő eszközökből. *Dave Raggett* 1994-ben csatolt egy indítványt a HTML 3.0-s munkaváltozatához, melyben egy „ős” matematikai leírónyelvet fogalmazott meg, de ezt az indítványt 1995-ben a darmstadti *WWW Conference* alkalmával visszautasították. 1995 novemberében a *Wolfram Research* képviselői bemutattak a *W3C*-nek egy indítványt a standard matematikai leírónyelv

elkészítéséhez, mely az akkoriban már kialakulóban lévő XML-en alapult. 1996 májusában a *Digital Library Initiative* campaign-urbanai találkozóján több érdekeltre nagy hatással volt a beadvány. A találkozó után megalakult a matematikai leíró nyelvvel foglalkozó bizottság, a *HTML Math Editorial Review Board*. Az ezt követő években a bizottság megerősödött, majd 1997 márciusában formálisan átalakult a W3C matematikai munkacsoportjává.

### **2.2.3. Az OpenMath története**

A MathML-hez hasonlóan az XML nyelven alapuló OpenMath kifejlesztése 1993-ban indult. Alapvetően európai kezdeményezés, az Európai Unió (EU) is támogatja a fejlesztését.

Több évre visszanyúló szakmai találkozók során alakították a szakemberek a nyelvet, mígnem 1996-ban megalakult az *OpenMath Society*, mely kiadta az OpenMath első specifikációját. 1997-ben az EU Negyedik Keretprogramjának (*European Union Fourth Framework Programme*) hatására az OpenMath az információtechnológiában az EU egyik szabványává vált.

## **2.3. A HTML korlátai**

A tudományos elektronikus kommunikáció hatékony eszköze iránt óriási az igény. Egyre inkább a kutatók, tudósok, mérnökök, oktatók, tanulók és technikusok saját maguk használják munkájukban és bíznak meg az elektronikus kommunikációban. Ugyanakkor a képalapú rendszerek, melyek jelenleg túlsúlyban vannak a tudományos jelölésben, meglehetősen primitívek és nem kielégítőek. A dokumentumok minősége szegényes, a létrehozásuk nehézkes, a képekben tárolt matematikai információk nem alkalmasak keresésre és indexelésre, valamint más alkalmazásokban való felhasználásra.

A matematikai kommunikáció HTML-lel való együttes használatakor a következő két probléma vetődik fel:

### **2.3.1. Megjelenítési akadályok**

Nézzük a következő egyenletet:  $2^{2^x} = 10$ . Ez az egyenlet ugyan a dokumentum része, de nem szöveg, (jelen esetben) a Word szövegszerkesztő egyenletszerkesztőjével készült; a környező szöveghez lett illesztve, 12 pontos betűnagysággal. Természetesen egy másik rendszerben, más beállítások mellett túl nagy vagy túl kicsi lehetne, nem beszélve arról, hogy jelen formájában nem is tudjuk beépíteni pl. egy HTML-dokumentumba (hiába tud a Word HTML formátumban menteni, az egyenletet ekkor is valamilyen grafikus állományként kapjuk meg).

A másik gond, hogy az egyenlet képe fehér háttérrel készült. Ha egy másik rendszeren más háttérszín van beállítva, akkor az egyenlet körül, a kép határáig fehér marad a háttér. Nézzük most a másodfokú egyenlet megoldó képletét:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  Ez a kép –

alaphelyzetben – egy böngészőben a normál szövegsor aljához lenne igazítva, ami – matematikai képletről lévén szó – nem „szabályos”. Természetesen a HTML rendelkezik eszközökkel a soron belüli függőleges illesztésre, de ekkor megint jelentkezhet az előbb felvetett gondunk: a kép mérete fix, a környezet betűmérete beállítástól függően változhat.

A képalapú egyenletek általában rosszabbul láthatóak, nehezebben olvashatók, érthetőek meg, mint a környező szöveg a böngésző ablakában. Sőt, nyomtatott formában ezek a gondok hatványozódhatnak. Az egyenletek képének felbontása átlagosan 70 dpi, míg a környező szöveg 300 dpi. Ez a minőségi egyenlőtlenség sok felhasználó számára elfogadhatatlan.

### **2.3.2. Kódolási gondok**

Próbáljunk egy képalapú egyenletet tartalmazó oldalon keresni, pl. az „ $=10$ ” részt a fenti első egyenletből. Hasonló helyzet: próbáljunk egy részt kimásolni az egyenletből a vágólapra, és beilleszteni egy másik alkalmazásba. Képalapú rendszereket használva ezeket a mindennapos igényeket nem tudjuk teljesíteni. Bár a HTML `<IMG>` elem „ALT=...” paraméterének használata a dokumentum forrásában segíthet, egyértelmű, hogy egy interaktívabb Web dokumentumnak gondoskodnia kell egy igényesebb interfészről a böngésző és a matematikai jelölésrendszer között. Ide tartozik az is, hogy egy kép megjelenítéséhez nagyobb sáv szélességre van szükség. Egy leírónyelv-alapú rendszer esetén a dokumentum megjelenítése előtt a „fordítás” a felhasználó gépén történik. Egy képlet leírónyelvi megvalósítása jellemzően kisebb méretű és jobban tömöríthető, mint egy képi egyenlet esetén.

### **2.4. A matematikai leírónyelvek követelményei**

Valamilyen leíró nyelv tervezéskor elengedhetetlen megfontolni a potenciális felhasználók szükségleteit. A matematika esetén a felhasználók széles spektrumát kell lefedni, kezdve az oktatástól a kutatásig, beleértve az üzleti felhasználást is.

Az oktatás óriási és fontos csoport, hiszen lehetőséget kell adni a tudományos anyagok Weben történő megjelenítéséhez. Ugyanakkor az oktatók nem rendelkeznek elegendő idővel és megfelelő eszközzel, és komolyan akadályozzák őket a Web dokumentumok készítési technikái. A tanulóknak és a tanároknak matematikai szövegek gyors és egyszerű elkészítéséhez könnyen megtanulható, olcsó segédeszközökre van szükségük.

Az elektronikus könyvek egy másik fajtája a világháló használatának, amely potenciálisan nagyon fontos az oktatásban.

Egyre többször találkozhatunk az Interneten keresztül történő oktatással. Az elektronikus könyveknek mindig naprakésznek kell lenniük, és lehetővé kell tenniük a szöveg és a tudományos szoftver vagy grafika közötti kommunikációt, biztosítani kell a visszakereshetőséget.

Az egyetemi szintű kutatás óriási mennyiségű tudományos anyagot termel. Ezeket az ismertanyagokat egyre gyakrabban valamilyen adatbázisban tárolják. Ezen adatok közzététele az egyetemi kiadványok elkészítésének igen magas ára miatt nem mindig lehetséges. Ezért egyre több intézmény preferálja a kiadás és terjesztés elektronikus úton történő megvalósítását, így egyre több kiadvány jelenik meg a Weben, mint például a *Mathematical Reviews*<sup>1</sup> és a *Zentralblatt für Mathematik*<sup>2</sup> is.

A kutatásokhoz alkalmazkodva a matematikai leírónyelv használata elősegíti a nagy dokumentum-gyűjtemények kezelését és felhasználását, ahol nagyon fontos az automatikus keresés és indexelés. Szintén fontos dolog a már létező, archív adatok, így pl. a nagy tömegű T<sub>E</sub>X és hasonló dokumentumok átkonvertálása az új követelményeknek megfelelően. Ezért az egyetemi kutatás létfontosságú eleme az archív ismeretek kezelhetősége is.

A tudósokhoz hasonlóan a mérnökök is nagy számban használnak technikai dokumentációkat mindennapi munkájuk során, szükségük van kísérleti eredményeik, számítógépes szimulációik lejegyzésére, valamint számításaik ellenőrzésére. A matematikai leírónyelv ezek könnyű rögzítését, olvasását, és sokrétű felhasználását teszi lehetővé.

Felvetődik egy új lehetőség is: a látássérült emberek számára is elérhetővé válna a tudományos szakirodalom a már manapság is használt szoftverek (pl. hangos böngészők) segítségével.

---

<sup>1</sup> Címe: <http://www.ams.org/mathweb/MRInfo/MR-Home.html>

<sup>2</sup> Címe: <http://www.zblmath.fiz-karlsruhe.de/zbl>

A könyvkiadói piac is lehetőséget lát a matematika „Webesítésében” a könyvnyomtatástól az interaktív szövegeken át az egyetemi kiadványokig. A kiadóknak szükségük van egy rendszerre, melynek segítségével a világhálón megjelentethetik magas szintű kiadványaikat széleskörű üzleti felhasználásra, természetesen egy olyan módszer segítségével, mely alkalmazkodik jelenlegi, általában hipertextes, rendszereikhez.

## **2.5. A matematikai leírónyelvek céljai**

A matematikai leírónyelvek segítségével

- matematikai anyagokat kódolhatunk tudományos kommunikációhoz és oktatáshoz minden szinten;
- a kódolás során, együttesen határozhatjuk meg a matematikai jelölést és a jelentést;
- konvertálási lehetőségünk lenne más matematikai formákról, illetve formákra, mind megjelenés, mind jelentés szempontjából. A kimeneti formák a következők lehetnek:
  - grafikus megjelenítés,
  - beszédszintetizáció,
  - számítógépes algebrai rendszerek bemenete,
  - más matematikai leírónyelvek (mint pl. a  $\text{T}_{\text{E}}\text{X}$ ),
  - egyszerű szöveges megjelenítés,
  - nyomtatott forma, beleértve akár a Braille-kiadványokat is.

Természetesen elfogadjuk, hogy a konvertálások során adatok veszhetnek el a folyamatok közben.

- adatokat adhatunk át más alkalmazásoknak;
- hatékonyan böngészhetünk bonyolult kifejezésekben is;
- gondoskodhatunk a kiterjeszthetőségről;
- megfelelő sablonokat tervezhetünk és más matematikai szerkesztési technikákat használhatunk;

- emberi olvasásra alkalmas dokumentumokat készíthetünk;
- számítógépes programok által könnyen legenerálható és végrehajtható kódokat készíthetünk.

Egy széles körben használható matematikai leírónyelvnek minimális funkcionalitását a következő gondolatok jelentik:

- A HTML oldalakba illesztett egyenletek az ismertebb böngészőkben a platformnak megfelelő legmagasabb színvonalúvá teszik dokumentumainkat az olvasó és a szerző kívánalmainak megfelelően.
- Egyenleteket tartalmazó HTML dokumentumaink a nyomtatótól függő legmagasabb szintű nyomtatott anyagokat tesznek lehetővé.
- A Web-oldalakon található egyenletek egyszerűen átadhatóknak más alkalmazásoknak az egér és a böngésző segítségével (pl. a *drag-and-drop* technikával).
- Az egyenletszerkesztők és -konvertálók továbbfejleszthetők, hogy lehetőséget adjanak az adott leírónyelvi formátumban való mentésre is.

Ezek a célok előrevetítik az egyes beágyazott elemek – úgymint Java vagy ActiveX alkalmazások – az adott matematikai leírónyelvre való cseréjét is, valamint magával hozza a közreműködést a böngészőket és más szoftvertermékeket forgalmazó és gyártó cégekkel.

## II. RÉSZ: ALAPOK

Ebben a részben bemutatom az általam kiválasztott három leírónyelv alapgondolatait. A következő részben pedig a MathML elemeit alapul véve összehasonlítom azokat a T<sub>E</sub>X és az OpenMath elemeivel.

### 1. A MathML alapjai

#### 1.1. Általános áttekintés

A matematikai leírónyelvek definiálásánál a kihívás alapja az volt, hogy készítsünk egy olyan leírónyelvet, amely a matematikai jelöléseket és a matematikai jelentéstartalmat egyszerre képes megjeleníteni.

A matematikai jelölés és a matematikai tartalom közötti kapcsolat hajszálnyi és mély. Formális szinten a matematikai logika eredményei zavaros kérdéseket vetnek fel a szimbolikus logikai rendszer és a jelenségek összefüggését tekintve. Mélyebb gondolati szinten, bárki, aki matematikai jelölőrendszert használ, tudja, hogy egy jó jelölésrendszer kiválasztása sokat jelent; a jelölésrendszer szimbolikus struktúrája felveti a logikai struktúrát is. Például a deriváltak Leibnitz-féle jelölése előrebecsátja a láncszabály alkalmazását a törtek jelképes áthúzásával:

$$\frac{\partial f}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial f}{\partial t}$$

A matematikusok és a tanárok nagyon jól értik ezt, véleményük szerint a jelölésrendszer kiválasztása kiemeli a kulcsát egy probléma szemléletének, és elrejtí vagy leszűkíti a tárgyhoz nem tartozó szempontokat. Mindennapos dolog a matematikában és a tudományokban leírni olyan dolgokat, amelyek technikailag bármi

mást is jelenthetnek. A hosszú gyakorlat azt mutatja, erre ténylegesen szükség is van felsőbb szinteken.

Más megközelítésben, a matematikai jelölést egy matematikai objektum precíz leírására használják. A matematikai jelölés bámulatos pontosságra képes, és amikor óvatosan használjuk, látszólag félreérthetetlen. Ez a félreérthetlenség lehetővé teszi a matematikai objektumok leírását úgy, hogy azokat olyan szoftveralkalmazásokban felhasználhassuk, mint egy számítógépes algebrai rendszer vagy egy hanggenerátor. Olyan szituációkban, ahol az alkalmazások közti kommunikáció a legfontosabb, a vizuális megjelenés általában kis szerepet játszik.

A MathML lehetőséget ad arra, hogy a szerzők kódolhassák mind a megjelenítendő jeleket, amelyek bemutatják a matematikai objektumokat, mind magukat az objektumok matematikai struktúráját. Sőt, keverhetik is a kétféle kódot.

#### **1.1.1. A MathML elemek rendszerezése**

Minden MathML elem a következő három csoport valamelyikéhez tartozik: *megjelenítő elemek*, *tartalmi elemek*, vagy *interfész elemek*.

A megjelenítő elemek leírják a matematikai jelrendszer struktúráját. Tipikus példája az **<mrow>** elem, mely a vízszintesen egymás után írt karaktereket jelzi, valamint az **<msup>** elem, mely a felső index jelölésére szolgál. Általános szabály, hogy minden megjelenítő elem megfelel egy egyszerű jelölési sémának, mint például a sor, a felső- és alsóindex, stb.

A tartalmi elemek a matematikai objektumokat írják le közvetlenül, szemben a jelöléssel, amivel azokat csak megjelenítjük. Tipikus példája a **<plus/>** elem, mely általában a valós számok összeadását írja le, valamint a **<vector>** elem, mely a lineáris algebra vektorait ábrázolja. Minden ilyen elem egy pontosan meghatározott matematikai fogalomhoz kapcsolható.

Minden MathML elem vagy megjelenítő, vagy tartalmi elem, kivéve a **<math>** elemet, mely a legfelső szinten helyezkedik el és interfész elemnek nevezzük. Egyik funkciója, hogy továbbadja a paramétereket egy MathML processzornak, és így befolyással van a teljes kifejezésre. A másik szerepe kommunikálni a böngészővel, amely megjeleníti a MathML kifejezéseket, így integrálja azokat a HTML környezetbe.

### **1.1.2. Kifejezés-fák és token-elemek**

Mind a megjelenítési, mind a tartalmi kifejezések számos formális tulajdonságra bonthatók. Mindkét esetben a legtöbb kifejezés természetesen darabokra vagy részkifejezésekre bontható. Például az  $(a+b)^2$  felbontható az  $(a+b)$  alapkifejezésre és egy leíró kifejezésre, amely jelen esetben egyetlen szám (2). Ráadásul, mint ahogy ez a példa is mutatja, a részkifejezés maga is tovább bontható. Természetesen, a felbontás befejeződik az oszthatatlan elemeknél, mint például a számok, a betűk és más szimbólumjelek.

Példánkban a felső index a hatványozás tipikus jelölése, mely jelölés megköveteli az alap és a hatványkitevő meglétét. Ez nem véletlen, a matematikai jelölés, mint egy általános szabály, tükrözi a matematikai objektumok alapját képző logikai struktúrát.

A matematikai objektumok és jelölések önmagához visszatérő természete erősen visszatükröződik a MathML-ben. A legtöbb megjelenítési és tartalmi elem magába foglal néhány más MathML alkotóelemet, melyekből az eredeti objektum rekurzívan felépült. Az eredeti elrendezést általában *szülő* sémának nevezzük, az alkotóelemeket pedig *utód* sémának. Általánosabban a MathML kifejezéseket tekinthetjük fának, ahol minden csomópont egy MathML elemnek felel meg, a „szülőcsoportok” alatt található az ő „utódaik”, a fa levelei pedig a tovább már nem bontható elemek

jelei vagy egységei. (Dolgozatomban az objektumorientált programozás terminológiáját használom.)

A legtöbb levél egy MathML kifejezés-fában vagy *üres elem*, vagy *token-elem* (*jelzéselem*). Az üres elem közvetlenül a MathML-ben található szimbólumot jeleníti meg, mint például a **<plus/>** tartalmi elem. A MathML token-elem az egyetlen olyan elem, mely megenged közvetlen karakteres adattárolást. A karakteres adatok ASCII karakterekből és MathML speciális jelekből állnak. Ezeket a jeleket tipikusan nem-ASCII karakterek jelzésére használjuk, leírásuk *escape-szekvenciával* lehetséges, a következő (HTML-ből származó) módon: *&név*; . Példák: „ $\alpha$ ” - *&alpha*;, „ $\rightarrow$ ” - *&rightarrow*;, „ $\Sigma$ ” - *&sum*; . Harmadik csoportja a leveleknek a *magyarázó elem*, melyet az adatok nem MathML alakú állapotának megtartására használhatjuk.

A legfontosabb megjelenítő token-elemek az **<mi>**, az **<mn>**, és az **<mo>**, az azonosítók („*mathematical identifier*”), számok („*mathematical number*”) és műveleti jelek („*mathematical operator*”) megjelenítéséhez. Ezek különböző formában jelenítik meg a vonatkozó objektumokat: a számokat általában egyenesen álló betűtípussal jelöljük (2), az azonosítókat dőlttel (*x*), a műveleti jeleket pedig szélesebb szóközzel ( + ). A tartalmi jelek között két token-elem van csak, a **<ci>** és a **<cn>**, melyek azonosítók („*content identifier*”) és számok („*content number*”) jelölésére használjuk. A tartalmi jelölésben az általánosan használt függvények és műveleti jelek elemeinek elkülönítéséről gondoskodik. Például az **<fn>** elem a felhasználó által definiált kifejezések jelöléséről gondoskodik.

A MathML elemeknek általában van *kezdő* és *záró* tagjuk, melyek körbeveszik a tartalmukat. Token-elemek esetén a tartalom karakteres adat, a többi esetben valamilyen utód-elem. Harmadik kategória az üres elem, mely nem kíván tartalmat, és egyszerűen a **<elemnév/>** formával jelöljük. Egy példa erre a **<plus/>** elem.

Visszatérve az előző példához  $((a+b)^2)$ , lássuk, hogy épül fel a kifejezés a gyakorlatban (az egyes elemekről bővebben később szólok). A példa megjelenítési elemek felhasználásával:

```
<msup>
  <mfenced>
    <mrow>
      <mi>a</mi>
      <mo>+</mo>
      <mi>b</mi>
    </mrow>
  </mfenced>
  <mn>2</mn>
</msup>
```

Tartalmi elemekkel a megoldás így néz ki:

```
<apply>
  <power/>
  <apply>
    <plus/>
    <ci>a</ci>
    <ci>b</ci>
  </apply>
  <cn>2</cn>
</apply>
```

### 1.1.3. Megjelenítő jelölés

A MathML megjelenítő jelölésnek 28 eleme van, melyeknek több mint 50 attribútumuk lehet. A legtöbb elem összhangban van a szerkezeti sémával, amely tartalmaz más megjelenítő elemet is. Minden szerkezeti séma összhangban van egy kétdimenziós jelölőrendszerrel, úgymint a felső- és alsóindexek, törtek és táblázatok. Ráadásul a fent említett három token-elemen kívül (**<mi>**, **<mn>**, **<mo>**) van néhány kevésbé ismert megjelenítő elem is. A fennmaradó néhány megjelenítő elem üres elem, melyeket főként sorba rendezéssel kapcsolatban használunk.

A szerkezeti séma több osztályba sorolható. Az elemek egyik csoportja az írásformákkal áll kapcsolatban, és olyan elemeket tartalmaz, mint az **<msub>**, az **<munder>**, és az **<mmultiscripts>**. Egy másik csoport általánosan leírja a szerkezetet, mint például az **<mrow>**, az **<mstyle>**, és az **<mfrac>**. A harmadik csoport a táblázatokkal áll kapcsolatban. A

**<maction>** elem önmagában egy külön kategória, a jelöléssel kapcsolatos vezérléseket jelöli.

A legtöbb szerkezeti séma fontos része, hogy lényeges a utód sémák sorrendje. Például, egy **<mfrac>** elem első utóda a számláló, a második a nevező. Amikor ki akarunk emelni egy MathML elemet, mint például az **<mfrac>**-ot, szükség van az utódaira a megfelelő sorrendben. Ekkor, mint argumentumokra hivatkozunk, és tudjuk, hogy az **<mfrac>** elem egy konstruktor.

#### 1.1.4. Tartalmi jelölés

A tartalmi jelölésnek kb. 75 eleme van, közel egy tucat attribútummal. A legfőbb elemek az üres elemek, melyek számos műveleti jelből, relációból és függvényből állnak, például: **<partialdiff/>**, **<leq/>**, **<tan/>**. Egy másik csoport tagjait, mint a **<matrix>**-ot és **<set>**-et, számtalan matematikai adattípus kódolására használjuk. A harmadik csoportba olyan elemek tartoznak, mint az **<apply>**, melyekkel új matematikai objektumokat tudunk készíteni a már meglévőkből.

Az **<apply>** elem talán a legfontosabb a tartalmi elemek között. Használatával argumentumok csoportjából függvényt alkothatunk. Az utód sémák sorrendje itt szintén fontos, az első utód jelzi a függvényt, a többi pedig a függvény argumentumait, az előre meghatározott sorrendben. Jegyezzük meg, hogy az így létrehozott függvényben mindig előbb jelöljük a függvényt és csak utána az argumentumokat. Nézzük példaképpen az  $a-b$  kivonást:

```
<apply>
  <minus/>
  <ci>a</ci>
  <ci>b</ci>
</apply>
```

Számos függvényhez és művelethez szükség van egy vagy több mennyiség előre történő meghatározására. Példaként említjük a határozott integrált, melynél előre definiálni kell az integrál határértékeit. Erre a mennyiségi sémákat használjuk, mint például

a **<bvar>**-t és a **<lowlimit>**-et. Ezeket olyan műveletekkel együtt használjuk, mint a **<diff/>** és az **<int/>**.

A **<declare>** különösen fontos a tartalmi jelölések között. Ezen elem segítségével egy változónak előre deklarálhatjuk a típusát egy bizonyos értékkel. Természetesen a vizuális megjelenítés nem vesz tudomást a deklarációról, akkor használjuk, ha egy kifejezést ki akarunk értékelni valamilyen matematikai alkalmazás segítségével.

### **1.1.5. A megjelenítés és tartalom ötvözése**

Különböző jelölések különböző feladatokra alkalmasak. Az adatok legjobban talán az egyszerű megjelenítő jelöléssel adhatók vissza, de a szerző által kitalált szemantikus információ így nem érzékeltethető. Ezzel szemben néhány matematikai alkalmazás és oktatási segédeszköz valószínűleg a tartalom-alapú formát fogja választani. Az alkalmazások többsége azonban a két véglet között helyezkedik el. Ezek számára a legalkalmasabb a két jelölési forma ötvözése.

A jelölési formák ötvözésének szabályai abból az általános elvből származnak, hogy a kevert jelölés csak ott használandó, ahol van értelme. A tartalmi jelölés beágyazása a megjelenítő jelölésbe alapján véve azt jelenti, hogy valamilyen tartalmi résznek szemantikai jelentéssel kell bírnia, és nem szükséges más argumentum vagy mennyiség pontosabb meghatározása. A megjelenítési jelölés tartalmi jelölésbe való beágyazása azt jelenti, hogy a megjelenítési jelölésnek tartalmaznia kell egy jelölőelemet úgy, hogy azt tovább nem bontható egységnek tekintjük, mint például egy változót vagy függvényt.

Egy másik lehetőség, a **<semantics>** elem használata. Ez az elem a MathML kifejezéseket különféle magyarázatokra kötelezi. Mindkét esetben (a megjelenítő jelölés keverése a tartalmival, vagy fordítva) az alapul vett jelölési rendszer érvényességét és

jelentését szélesíthetjük ki a hozzáfűzött, másik jelölési rendszerben elkészített megjegyzéssel.

## 1.2. Néhány példa

### 1.2.1. Megjelenítő jelölés példái

A kifejezés:  $x^2 + 4x + 4 = 0$

A MathML kód:

```
<mrow>
  <mrow>
    <msup>
      <mi>x</mi>
      <mn>2</mn>
    </msup>
    <mo>+</mo>
    <mrow>
      <mn>4</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>x</mi>
    </mrow>
    <mo>+</mo>
    <mn>4</mn>
  </mrow>
  <mo>=</mo>
  <mn>0</mn>
</mrow>
```

Figyeljük meg az **<mrow>** elem használatát. Kifejezések készítésekor nagyban megkönnyíti pl. a szóközők kezelését vizuális vagy hangos megjelenítéskor.

A kifejezés:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

A MathML kód:

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
        <mrow>
          <msup>
            <mi>b</mi>
            <mn>2</mn>
          </msup>
          <mo>-</mo>
          <mi>a</mi>
        </mrow>
      </msqrt>
    </mrow>
    <mn>2</mn>
  </mfrac>
</mrow>
```

```

<mo>-</mo>
<mrow>
  <mn>4</mn>
  <mo>&InvisibleTimes;</mo>
  <mi>a</mi>
  <mo>&InvisibleTimes;</mo>
  <mi>c</mi>
</mrow>
</msqrt>
</mrow>
<mrow>
  <mn>2</mn>
  <mo>&InvisibleTimes;</mo>
  <mi>a</mi>
</mrow>
</mfrac>
</mrow>

```

Megfigyelhető, hogy a  $\pm$  jel egy speciális műveleti jel, a `&PlusMinus`; segítségével adható meg. A MathML széleskörű listával rendelkezik a matematikai szimbólumok leírásához, melyeket az *A Függelékben* (155. oldal) találhatunk meg részletesen. Mivel a MathML-nek alkalmasnak kell lennie hangos megjelenítésre is, ezért célszerű ezeket a speciális jeleket mindig korrekt módon használni. A hangos megjelenítésnél fontos lehet az olyan, a vizuális megjelenítésnél látszólag felesleges jelek használata, mint pl. az `&InvisibleTimes;`:

```

<mrow>
  <mi>z</mi>
  <mfenced>
    <mrow>
      <mi>x</mi>
      <mo>+</mo>
      <mi>y</mi>
    </mrow>
  </mfenced>
</mrow>

```

A fenti kódot „z-szer x és y összege” formában olvashatunk, de az `&InvisibleTimes;` műveleti jel hiányában a hangos megjelenítés folyamán ez nem lesz egyértelmű. Ezért a z-t tartalmazó sor után be kell iktatni egy `<mo>&InvisibleTimes;</mo>` sort is. Hasonló módon kell használni a `&dd;` műveleti jelet, mely a „differentiál d” megfelelője, és sajátos határoló üres helyekkel jelenik meg. Tehát, ha a szerzők megfelelő gondossággal használják ezeket a speciális jeleket, az általuk készített dokumentumok sokkal szélesebb körben elérhetőek lesznek.

A kifejezés:  $A = \begin{bmatrix} x & y \\ z & w \end{bmatrix}$

A MathML kód:

```
<mrow>
  <mi>A</mi>
  <mo>=</mo>
  <mfenced open="[" close="]">
    <table>
      <mtr>
        <td><mi>x</mi></td>
        <td><mi>y</mi></td>
      </mtr>
      <mtr>
        <td><mi>z</mi></td>
        <td><mi>w</mi></td>
      </mtr>
    </table>
  </mfenced>
</mrow>
```

A legtöbb elemnek számos attribútuma van, amely a tartalom módosítására szolgál. Például az **<mfenced>** elem attribútumai a határoló jelek jellegét szabályozza.

### 1.2.2. Tartalmi jelölés példái

A kifejezés:  $x^2 + 4x + 4 = 0$

A MathML kód:

```
<reln>
  <eq/>
  <apply>
    <plus/>
    <apply>
      <power/>
      <ci>x</ci>
      <cn>2</cn>
    </apply>
    <apply>
      <times/>
      <cn>4</cn>
      <ci>x</ci>
    </apply>
    <cn>4</cn>
  </apply>
  <cn>0</cn>
</reln>
```

Jegyezzük meg, hogy a **<reln>** elem helyett általában használható az **<apply>** elem is, kivéve akkor, ha a műveleti jel valamilyen tényleges relációt ír le.

A kifejezés:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

A MathML kód:

```
<reln>
  <eq/>
  <ci>x</ci>
  <apply>
    <divide/>
    <apply>
      <fn>
        <mo>&PlusMinus;</mo>
      </fn>
      <apply>
        <minus/>
        <ci>b</ci>
      </apply>
    </apply>
  </root>
  <apply>
    <minus/>
    <apply>
      <power/>
      <ci>b</ci>
      <cn>2</cn>
    </apply>
  </apply>
  <apply>
    <times/>
    <cn>4</cn>
    <ci>a</ci>
    <ci>c</ci>
  </apply>
  </apply>
  <cn>2</cn>
</apply>
</reln>
```

A MathML tartalmi jelölőrendszerének nincsen eleme a  $\pm$  műveletre, ezért vezettük be az **<fn>** elemmel, amelyben deklaráltuk, hogy valójában tartalmilag mire is akarjuk használni. Ez egy egyszerű példa, hogyan ötvözhethjük a tartalmi jelölést a megjelenítő jelöléssel.

A kifejezés:  $A = \begin{bmatrix} x & y \\ z & w \end{bmatrix}$

A MathML kód:

```
<reln>
  <eq/>
  <ci>A</ci>
  <matrix>
    <matrixrow>
      <ci>x</ci>
      <ci>y</ci>
    </matrixrow>
    <matrixrow>
      <ci>z</ci>
```

```

        <ci>w</ci>
      </matrixrow>
    </matrix>
  </reln>

```

Jegyezzük meg, hogy a **<matrix>** elem tartalmazza a zárójeleket, tehát azokat nem kell külön deklarálnunk. Mint látjuk, ez teljesen eltér a megjelenítési jelölés **<mtable>** elemétől, ahol kívánalmaink szerint kódolnunk kellett a zárójelezést is.

### 1.2.3. Példák a jelölések ötvözésére

A kifejezés:  $\int_0^t \frac{dx}{x}$

A MathML kód:

```

<semantics>
  <mrow>
    <msubsup>
      <mo>&int;</mo>
      <mn>0</mn>
      <mi>t</mi>
    </msubsup>
    <mfrac>
      <mrow>
        <mo>&dd;</mo>
        <mi>x</mi>
      </mrow>
      <mi>x</mi>
    </mfrac>
  </mrow>
  <annotation-xml encoding="MathML-Content">
    <apply>
      <int/>
      <bvar>
        <ci>x</ci>
      </bvar>
      <lowlimit>
        <cn>0</cn>
      </lowlimit>
      <uplimit>
        <ci>t</ci>
      </uplimit>
    </apply>
    <divide/>
    <cn>1</cn>
    <ci>x</ci>
  </apply>
</annotation-xml>
</semantics>

```

Ebben a példában a **<semantics>** elemet használtuk jelezve, hogy a megjelenítő jelöléshez tartozik egy szemantikai megjegyzés, ami viszont már tartalmi jelölésű. A **<semantics>** elemnek első utóda az a kifejezés, amihez a megjegyzést fogjuk

fűzni, a következő pedig maga a megjegyzés. Arra nincs megkötés, hogy a megjegyzés milyen fajtájú legyen, így lehet TeX kódrészlet vagy valamilyen számítógépes algebra bemenete is. Egyedül annyit kell tennünk, hogy az **<annotation>** vagy **<annotation-xml>** elem *encoding* attribútumában le kell írunk a megjegyzés típusát.

Egy másik módja a **<semantics>** elem használatának, mikor a tartalmi kód mellé megjegyzésként megadunk egy megjelenítési formát. Ebben az esetben a fenti példa MathML-kódja a következő:

```
<semantics>
  <apply>
    <int/>
    <bvar>
      <ci>x</ci>
    </bvar>
    <lowlimit>
      <cn>0</cn>
    </lowlimit>
    <uplimit>
      <ci>t</ci>
    </uplimit>
    <apply>
      <divide/>
      <cn>1</cn>
      <ci>x</ci>
    </apply>
  </apply>
  <annotation-xml encoding="MathML-Presentation">
    <mrow>
      <msubsup>
        <mo>&int;</mo>
        <mn>0</mn>
        <mi>t</mi>
      </msubsup>
      <mfrac>
        <mrow>
          <mo>&dd;</mo>
          <mi>x</mi>
        </mrow>
        <mi>x</mi>
      </mfrac>
    </mrow>
  </annotation-xml>
</semantics>
```

Ez a megjegyzési forma akkor hasznos, ha más formában szeretnénk megjeleníteni a tartalmi jelölésben leírt kifejezést, mint ahogy az automatikusan létrejönne, példánkban az  $\frac{1}{x}dx$  helyett a

$\frac{dx}{x}$  formában. Természetesen itt vigyáznunk kell arra, hogy a

megjelenítőnek (pl. a különböző böngészők) nem kötelessége a megjegyzést figyelembe venni, így ezt a módszert nem lehet megjelenítő-függetlennek tekinteni.

### **1.3. A MathML szintaxisa és nyelvtana**

A MathML egy XML (*Extensive Markup Language*) alkalmazás, mint ilyen, szintaxisa az XML szintaxisán alapul, nyelvtanát pedig a MathML DTD-ben (*Document Type Definition*) specifikálták. Más szavakkal a használt elemek, attribútumok, műveletek nyelvtana az XML nyelv specifikációjában<sup>3</sup> lett definiálva, a MathML elemek és attribútumok neveit, melyek egymásba is ágyazhatók, a MathML DTD szabályozza.<sup>4</sup>

Természetesen a MathML is megad néhány szintaktikai és nyelvtani szabályt, melyet az öröklött XML szabályok mellett alkalmaznunk kell. Ezek a szabályok lehetővé teszik, hogy jóval több információt kódolhassunk, mint amennyit a szabvány XML lehetővé tenne (pl.: sokkal több elemet használhatunk egy lényegesen komplexebb DTD-vel).

Alapjában véve kétféle plusz MathML nyelvtani és szintaktikai szabály létezik. Az egyik fajta magába foglalja az attribútum értékek ismérveit. Például: nem lehetséges az egyszerű XML-ben megkövetelni, hogy egy attribútum érték pozitív egész legyen. A másik fajta szabályok az utód elemek részletesebb korlátjait írják le (pl. az elrendezésben), mint ahogy az a DTD-ben szerepel. Például: nem lehetséges az XML-ben meghatározni, hogy az első utódot egyféle módon értelmezzük, míg a másikat másféleképpen.

---

<sup>3</sup> Lásd: [www.w3.org/TR/PR-XML](http://www.w3.org/TR/PR-XML)

<sup>4</sup> Az itt felsorolt specifikációk nem képezik könyvünk részét, ezeket a [www.w3.org](http://www.w3.org) honlapon lehet megtekinteni.

### 1.3.1. Bevezető az XML szintaxisába

Mivel a MathML egy XML alkalmazás, a MathML specifikáció az XML terminológiáját használja a leírására. Röviden, az XML adatok Unicode karakterekből (melyek tartalmazzák a normál ASCII karaktereket is), speciális jelekből (mint a `&nbsp;`, melyek általában „kiterjesztett karaktereket” jelölnek), valamint elemekből (mint a `<mi fontstyle="normal"> x </mi>`) állnak. Az elemek tartalmát nyitó és záró tagok közé zárjuk, akárcsak a HTML-ben. Vannak üres elemek is, mint a `<plus/>`, melyek mindig `/>`-re végződnek, ezzel jelezve, hogy az elemnek nincs tartalma és záró tagja. A nyitó tagok tartalmazhatnak névvel jelölt paramétereket, attribútumokat (mint a `fontstyle="normal"` a fenti példában).

Mivel az XML kis- és nagybetű-érzékeny, a MathML elemek és attribútumok is azok, ezért a MathML mindet kisbetűvel definiálja.

### 1.3.2. MathML attribútumok értékei

A legtöbb MathML elemnek szüksége van számos utód elem specifikálására és/vagy egyéb többletjelentést hozzákapcsolására az utódhoz a megadott pozícióban. Ezek megadása MathML specifikus, nincs leírva az eredeti XML szintaxisban és nyelvtanban. Amikor egy adott MathML elem utóda a tárgya ezeknek a többletfeltételeknek, akkor általában *argumentumnak* nevezzük őket, így hangsúlyozva ezek speciális MathML használatát.

Az XML specifikációnak megfelelően az elemek attribútumainak a következő formákban adhatunk értéket:

`attribútumnév = "érték"`

vagy

`attribútumnév = 'érték'`

ahol az egyenlőségjel oldalairól a szóközök elhagyhatók. Megjegyzendő, hogy a kétféle idézőjel nem keverhető és nem ágyazható egymásba. Mindezek mellett, ha a `"`, `'` és `&` jeleket

mégis használni kívánjuk az attribútumokban, erre ad lehetőséget ezen speciális jelek escape-szekvenciás használata („&quot;”, „&apos;” és „&amp;”).

Az attribútumok felsorolásánál megadhatjuk azok egységét, a HTML-nél már megszokott egységjelekkel (de ez általában nem kötelező): *em* (horizontális hossz), *ex* (vertikális hossz), *px* (pixel), *in* (hüvelyk), *cm* (centiméter), *mm* (milliméter), *pt* (pont = 1/72 hüvelyk), *pc* (pika = 12 pont), % (százalék).

Az egyes attribútumok alapértékeit az elemek leírásánál minden esetben megadjuk. Ha az attribútum érték öröklött („*inherited*”), akkor ebben az esetben az **<mstyle>** elemnél megadott attribútum értékeket veszi a MathML megjelenítő alapul. Ha az attribútum értéke *automatikusnak* („*automatic*”) lett megadva, akkor a MathML megjelenítő az eddig expliciten meghatározott értékekből számolja ki azt a környezet figyelembevételével.

### 1.3.3. Üres helyek használata

A MathML engedélyezi a üres helyek („*whitespaces*”, „*blanks*”: szóköz, tabulátor, újsor, kocszi-vissza karakterek) használatát a token-elemeken kívül. Ezeket a határoló üres helyeket a MathML megjelenítő levágja, tehát nem fognak látszani.

Például a `<mo> ( </mo>` megfelel a `<mo>(</mo>` alaknak és a

```
<mtext>
  1.
  axióma:
</mtext>
```

megegyezik a `<mtext>1. axióma:</mtext>`-tel

Ezért, ha mindenképpen üres helyet kell megjelenítenünk, akkor használjuk a HTML-ből ismert `&nbsp;` speciális karaktert. Hasonlítsuk össze következő két alakot:

<code>&lt;mtext&gt;</code>	<code>&lt;mtext&gt;</code>
1.	&nbsp;&1.&NewLine;
axióma:	&nbsp;&axióma:
<code>&lt;/mtext&gt;</code>	<code>&lt;/mtext&gt;</code>

Az első változatban nincs üres hely az „1.” előtt, az „1.” és az „axióma:” között egy szóköz van, és szintén nincs az „axióma:” után. A második esetben egy szóközzel kezdődik a szöveg, ezt követi az „1.”, majd új sorban szóközzel kezdve folytatódik, és az „axióma:”-val végződik, ami után nincs üres hely.

## **2. Az OpenMath áttekintése**

Hasonlóan a MathML-hez, az OpenMath kifejlesztői is az XML nyelvet vették alapul saját leírónyelvük megalkotásához. Az OpenMath azonban csak a MathML-ben megismert tartalmi elemek területét fedi le, és inkább programozási nyelv jellegű, ezért elsősorban matematikai alkalmazások során használható elsősorban.

Az egyik legnagyobb különbség a két nyelv között az, hogy az OpenMath sokkal kevesebb elemet használ, a MathML-ben megtalálható elemek többségét attribútumként adja meg. A nyelv kis- és nagybetű-érzékeny: az elemek nevét nagybetűvel, az argumentumokat kisbetűvel, értékeiket idézőjelek között (és nem aposztrófok között!) kell megadni.

A másik nagy különbség a *Content Dictionary* (CD), melyben az elemeknél megadott attribútumok – valójában valamilyen matematikai műveletek – vannak meghatározva. Több CD-t is használhatunk, nevüket az egyes elemek attribútumaként (`cd="..."`) kell megadni.

Hasonlóan a MathML-hez, az OpenMath is egy fastruktúrán alapul. A leveleket *alapobjektumoknak* nevezzük, a következők lehetnek:

- *egészek,*
- *lebegőpontos számok,*

- *szimbólumok* (a *Content Dictionary*ban meghatározott „elem”, név szerint hivatkozhatunk rá, a névben csak betűk, számok és `_` jel lehet, betű-érzékeny.),
- *változók* (a programozási nyelvekhez hasonlóan használhatjuk őket, csak valamilyen OpenMath alkalmazás esetén van értelme használatuknak; nevük speciális jeleket is tartalmazhat, többféle formátumban, pl:  $\alpha = +A7E- = \text{hex. } 3B1$ ),
- *sztringek* (Unicode karakterekből),
- *bájtömbök* (mint a sztringek, azzal a különbséggel, hogy a sztringeknek értelmezhetőnek kell lenniük, a bájtömböknek nem).

Az OpenMath-ban van négy összetett objektum is:

- *alkalmazás* (*application*: két részből áll, az első rész a *fejobjektum*, a második az *argumentumobjektum*, ez az objektum általában megfelel a matematikai függvényeknek),
- *kapocs* (*binding*: minimum két részből áll, az első a kötő objektum, a többi az *argumentumobjektum*, jellemző példa rá a *lambda*-függvény),
- *tulajdonság* (*attribution*: olyan szimbólum és valamilyen más objektum párok, ahol a szimbólum lesz az adott objektum attribútuma),
- *hiba* (*error*: nincs közvetlen matematikai jelentése, alkalmazásokkal történő feldolgozások közben történő hibák kiértékelésére szolgál).

Fontos tudni, hogy az OpenMath elemek közül csak az **<OMS/>**, az **<OMV/>** és az **<OMF/>** elemnek van saját attribútuma. Az **<OMS/>** elemnél meg kell adni a *cd* attribútumban, hogy melyik CD-ben található az adott szimbólum, valamint a *name* attribútumban azt, hogy mi a neve a műveletnek. Az **<OMV/>** elemnek csak *name* attribútuma van, itt adjuk meg a

változó nevét. Ezen kívül az **<OMF/>** elemnél a lebegőpontos szám számrendszerét kell megadni, amit a *dec* vagy a *hex* attribútum jelöl.

Érdekessége a nyelvnek, hogy egyes funkciókat – mint pl. a zárójelezés, vagy az indexek leírása – sajátágosan, egy **<OMATTR>** elemben kell megadnunk:

$x_a$

```
<OMA>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="style"/>
      <OMSTR> subscripted <OMSTR>
    </OMATP>
    <OMV name="x"/>
  </OMATTR>
  <OMV name="a"/>
</OMA>
```

$f(x,y)$

```
<OMA>
  <OMV name="f"/>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="left"/>
      <OMSTR>(</OMSTR>
      <OMS cd="presentation" name="right"/>
      <OMSTR>)</OMSTR>
    </OMATP>
    <OMA>
      <OMS cd="linalg" name="vector"/>
      <OMV name="x"/>
      <OMV name="y"/>
    </OMA>
  </OMATTR>
</OMA>
```

Az OpenMath szabvány kétféle kódolást ismer, az egyik az XML típusú (ebben a munkámban ezt ismertetem), a másik bájtflowam típusú, mely az XML kódolás egyes alkotórészeinek (bájtokkal) való leírása (rövid összefoglaló a *B Függelékben* (185. oldal) található).

Lássunk egy egyszerű példát:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

```
<OMOBJ>
  <OMA>
    <OMS cd="relation" name="eq"/>
    <OMV name="x"/>
    <OMA>
      <OMS cd="arith" name="divide"/>
      <OMA>
        <OMS cd="arith" name="plusminus"/>
```

```

<OMA>
  <OMS cd = "arith" name="unary_minus"/>
  <OMV name="b"/>
</OMA>
<OMA>
  <OMS cd="arith" name="root"/>
  <OMA>
    <OMS cd = "arith" name="minus"/>
    <OMA>
      <OMS cd="arith" name="power"/>
      <OMV name="b"/>
      <OMI>2</OMI>
    </OMA>
  </OMA>
  <OMS cd = "arith" name="times"/>
  <OMA>
    <OMS cd = "arith" name="times"/>
    <OMI>4</OMI>
    <OMV name="a"/>
  </OMA>
  <OMV name="c"/>
</OMA>
</OMA>
<OMA>
  <OMI> 2 </OMI>
</OMA>
</OMA>
<OMA>
  <OMS cd = "arith" name="times"/>
  <OMI>2</OMI>
  <OMV name="a"/>
</OMA>
</OMA>
</OMA>
</OMOBJ>

```

### 3. A TEX áttekintése

Külsőleg a TEX nem hasonlít sem a MathML-re, sem az OpenMath-ra, melyek a HTML-ből az XML-be átöröklött hipertextes struktúrára és elemekre épülnek. A TEX-ben a matematikai kifejezések leírását, formázását speciális parancsokkal oldjuk meg.

Dolgozatomban az *AMS-TEX* matematikai módban használatos parancsaival foglalkozom csak. A matematikai módnak két formája van, a *szövegközi* (be és kikapcsolása a „\$” paranccsal történik), valamint a *kiemelt* (be és kikapcsolása a „\$\$” paranccsal történik). A kiemelt matematikai mód abban különbözik a szövegközitől, hogy ezt külön bekezdésként, középre igazítva jeleníti meg a TEX.

A T<sub>E</sub>X nyelvben egy adott szöveg formázását parancsokkal végezhetjük, melyeket a kezdő „\” karakter jelzi. Ezek a parancsok két részre oszthatóak:

- ◆ ha a „\” jel után betű áll (az angol ábécé kis és nagybetűi), akkor azt az első nem betű karakterig próbálja feldolgozni a program; itt figyelniünk kell arra, hogy ha egy ilyen parancs után szóközöt teszünk, akkor az nem fog megjelenni, mert a T<sub>E</sub>X azt csak határoló jelnek tekinti, nem megjelenítendő karakternek – ennek feloldására használjuk a „\ ” (egy „\” és egy szóköz karakter) parancsot. Pl.: Ez a \TeX\ logo. eredménye: Ez a T<sub>E</sub>X logo.
- ◆ ha a „\” jel után nem betű áll, akkor csak a jel utáni karakter értelmeződik parancsként. Pl.: Ez egy \"o. eredménye: Ez egy ő.

Mint látjuk, vigyázni kell a szóköz használatával, különösen matematikai módban, hiszen az ott beírt szóközöket figyelmen kívül hagyja, mert ilyenkor a matematikában használatos helykihagyásokkal dolgozik a rendszer.

A másik fontos tudnivaló, hogy a T<sub>E</sub>X a matematikai kifejezésekben szereplő betűket a matematikai szabályoknak megfelelően dőlten jeleníti meg. Matematikai módban az egyes zárójelezett kifejezések nem választódnak el.

Külön említést érdemel az *i* és a *j* betű, melyeknek már van ékezetük. Ha tehát az „\’i” parancsot használjuk, az nem egy hosszú í betűt eredményez, hanem egy vesszős *i*-t (*ï*). Ilyenkor használjuk a „\i” és „\j” parancsokat, tehát a hosszú í „\’i” lesz. Itt érdemes megjegyezni, hogy ilyen esetekben, amikor a parancs egyetlen betűből áll, akkor érdemes használni a „{ }” csoportképző karaktereket, különben hibajelzést kapunk: Pl.: „vigy\’azat, v{\’i}rus!” eredménye: Vigyázat, vírus! Itt a „v\’\irus” hiba lenne,

mert nincs olyan parancs, hogy „\irus”, a „v\’\i rus” (szóközzel) pedig vírus lenne.

A  $\text{T}_{\text{E}}\text{X}$  parancsai kis- és nagybetű-érzékenyek, tehát a „\gamma” ( $\gamma$ ) és a „\Gamma” ( $\Gamma$ ) nem ugyanazt jelenti.

A speciális jelek összefoglalása az *A Függelékben* (155. oldal) található.

Néha szükség van egyes parancsokban azok mértékeinek megadására. Ezekre a következő mértékegységeket használhatjuk (mindegyik kétbetűs rövidítésű): *mm* (milliméter), *cm* (centiméter), *pt* (pont), *sp* (osztott pont), *bp* (nagy pont), *dd* (didot), *pc* (pika), *cc* (ciceró), *in* (hüvelyk), *em* (az aktuális betűkészlet „M”-ének szélessége), *ex* (az aktuális betűkészlet „x”-ének magassága). Néhány példa a megadásukra:

6 cm, 20pt, 0.31246in, .23 cc, -5,34pc, 0pt

A jól ismert példánk:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

## III. RÉSZ: RÉSZLETES ÖSSZEHAISONLÍTÁS

### 1. A MathML megjelenítő jelölése

Mivel az OpenMath nem tartalmaz megjelenítő elemeket, ezért ezt a nyelvet csak a tartalmi elemekkel hasonlítom össze.

A példákban először mindig a MathML forma található, majd pedig a T<sub>E</sub>X példa.

#### 1.1. A megjelenítő elemekről általában

A megjelenítő elemek a tradicionális matematika *konstruktorai*. Belőlük építhetjük fel az igen sokféle matematikai szimbólumokat és az igen bonyolult matematikai kifejezéseket is. A leíró nyelvek tulajdonságait figyelembe véve a megjelenítő elemek alkalmasak hangos megjelenítésre is (bár akad néhány attribútum, melyet ezen a módon nem lehet megjeleníteni), de elsősorban a tradicionálisabb, vizuális megjelenítésre készültek.

Mivel a MathML egyik legfontosabb felhasználási területe a HTML dokumentumokba való beágyazás, nem árt néhány szót szólni erről. Bár a HTML-nek is van alapszintű logikai váza (mint például a fejrész – főszöveg elválasztása, bekezdések használata, stb.), de ez (sajnos) csak ajánlás, és nem szabvány. A MathML teljes mértékben eleget tesz ezeknek az ajánlásoknak, de ennél sokkal mélyebb logikai vázzal rendelkezik.

A megjelenítő elemekkel fejezhetjük ki a matematikai jelölésrendszer szintaktikai struktúráját, ugyanúgy, mint más magas szintű szövegalapú dokumentációkban a címetek, szakaszokat, bekezdéseket. Ezért például, egy olyan egyszerű kifejezéseket, mint az  $x + a/b$  sem egyetlen **<mrow>** elemmel

(amely a vízszintesen egymás mellé írandó karaktersorozatok megjelenítésére szolgál) írjuk le, hanem több egymásba ágyazottal, ezzel is jelezve az részkifejezésekhez kapcsolódó logikai jelentéstöbbletet. TEX-ben erre a  $\{ \}$  zárójel-pár szolgál.

```
<mrow>
  <mi>x</mi>
  <mo>+</mo>
  <mrow>
    <mi>a</mi>
    <mo>/</mo>
    <mi>b</mi>
  </mrow>
</mrow>

x+{a/b}
```

Hasonlóan, a felsőindex nem csak a megjelenítendő karaktert jelenti, hanem az összefüggést a hatványkifejezés bázisával. Ez a struktúra egy jobb megjelenítési lehetőségről gondoskodik, és a megjelenítő rendszertől való nagyobb függetlenséget biztosít (pl. képernyőméret, stb.).

Létezik néhány különleges karakter a MathML-ben, melyeket egyes operátorok vagy más jelek helyett használunk, ilyenek pl. a `&DifferentialD;`, az `&ExponentialE;`, az `&ImaginaryI;` vagy a `&InvisibleTimes;`, az `&ApplyFunction;`, az `&InvisibleComma;`. Látszólag ezekre az matematikai jelekre nem lenne szükség, de ha logikai oldalról közelítünk meg egy kifejezést, pl. az  $f(x)$  kifejezést, akkor az ugyan láthatatlan `&ApplyFunction;` sokkal érthetőbbé teszi a dolgot, hogy itt valójában egy függvényről van szó.

## 1.2. Megjelenítő token-elemek

A token-elemek nulla vagy több karakter sorozatából vagy speciális jelekből állnak. Ezek egy része láthatatlan, csak a struktúra felépítésében van szerepük.

A MathML-ben karakterek és speciális jelek csak token-elemek részeként állhatnak, ez alól egyetlen kivétel az üres helyek valamelyik formája, erről azonban a megjelenítés során nem

veszünk tudomást. Egy másik kivétel a **<malignmark/>** elem, mely szerepelhet token-elemek részeként (lásd később).

### 1.2.1. A token-elemek attribútumai

A token-elemek (az **<mspace/>** kivételével) megkaphatják a következő attribútumokat: *fontsize* (betűméret), *fontweight* (betűvastagság), *fontstyle* (dőlt-e), *fontfamily* (betűtípus), *color* (betűszín), valamint a *class*, *style*, *id* attribútumokat, melyek a stíluslapokkal kapcsolatban használunk. Ha szükségünk van karakterformázásra, akkor a fenti attribútumok helyett inkább használjunk stíluslapokat, hiszen azok segítségével sokkal összetettebb formázási lehetőségeink vannak.

Mivel a MathML kifejezéseket általában valamilyen szövegalapú dokumentumba ágyazzuk be, így a mi MathML kifejezéseink öröklök a befogadó szöveg beállításait. Ez alól két kivétel van, az **<mn>** és az **<mi>**, ahol az öröklött beállítások mellett a *fontstyle* más alapértékeket vesz fel, az **<mn>**-nél normál, az **<mi>**-nél dőlt stílusú lesz.

Ha egy attribútumnak megadott érték nem létezik, akkor a megjelenítő megpróbálja a megadott értékhez legközelebb eső, létező értékkel megjeleníteni az elemet.

A T<sub>E</sub>X-nél az egyes parancsoknál nincs lehetőségünk a fenti paraméterek beállítására, ebben a nyelvben erre külön parancsokat használunk.

A betűtípus kiválasztására a következő parancsok közül választhatunk: **\rm** (roman - normál talpas betű), **\sl** (slanted - dőlt), **\it** (italic - kurzív), **\bf** (boldface - kövér), **\tt** (typewriter type - írógép típus).

A betűtípusok méretét is külön parancsokkal szabályozhatjuk, meglehetősen behatárolt keretek között, „gyárilag” a következő parancsok állnak rendelkezésünkre: **\tenrm** (10-es nagyságú normál betű), **\sevenrm** (7-es nagyságú normál betű), **\fiverm**

(5-ös nagyságú normál betű), **\tenit** (10-es nagyságú kurzív betű). Természetesen, ez általában nem elegendő. Ilyenkor a rendelkezésre álló betűtípus-állományokból készíthetünk saját betűméretező parancsot a **\font** parancs segítségével:

```
\font\nyolcas=cmr8  
8 pontos {\nyolcas antikva}
```

A fenti példa alapján a „8 pontos antikva” szöveg jelenik meg. A betűtípus-állományok nevei jelzik a betűtípus tulajdonságait: *b* (félkövér), *sl* (dőlt), *csc* (kiskapitális), *bx* (kövér), *ti* (kurzív), *ss* (sans serif – talpatlan betű); ezek összetetten is állhatnak: *cmbxti10*, ami megfelel a kövér, dőlt 10 pontos nagyságú betűnek (a *cm* a *computer modern* rövidítése).

A T<sub>E</sub>X-ben nincs lehetőség színek megadására.

Stílusokról lásd a „III/1.3. Stílusok” alfejezetet (56. oldal)!

### 1.2.2. Azonosítók

Néhány kivételtől eltekintve (felső- és alsóindexek), a **<mi>** elem segítségével jelenítjük meg az azonosítókat (változók, függvények, szimbolikus állandók).

Az előző szakaszban már megismertedtünk a token-elemek attribútumaival, de ismételten jegyezzük meg, hogy ennél az elemnél a *fontstyle* alapértelmezett értéke „*italic*”, azaz a betű dőlt nem vastagított.

A T<sub>E</sub>X matematikai módjában automatikusan dőlt betűs megjelenítésre vált, így erre nincs külön parancs, bár egyes függvényekre, szimbolikus állandók megjelenítésére léteznek parancsok.

Példák:

```
<mi> x </mi>  
<mi> D </mi>  
<mi> sin </mi>
```

`<mi></mi>`

Ez utóbbi példa is mutatja, hogy „üresen” is használhatjuk ezt az elemet, jelezve, hogy ide egy azonosító kíváncozna.

Ugyanez T<sub>E</sub>X-ben (természetesen az utolsó példának itt nincs értelme):

`x`

`D`

`\sin`

Az azonosítók körébe tartoznak a függvények is, melyeknél azonban használnunk kell az `&ApplyFunction;` (rövid neve `&af;`) műveleti jelet:

`\sin x`

```
<mrow>
  <mi> sin </mi>
  <mo> &ApplyFunction; </mo>
  <mi> x </mi>
</mrow>

\sin x
```

Az egyes kifejezésekhez tartozó szöveget is általában a **`<mi>`** elemmel jelöljük (egyes esetekben célszerű lehet a *fontstyle* attribútumot beállítani):

`1+...+n`

```
<mrow>
  <mn> 1 </mn>
  <mo> + </mo>
  <mi> ... </mi>
  <mo> + </mo>
  <mi> n </mi>
</mrow>

1+\ldots+n
```

Hasonlóképpen jelöljük a szimbolikus állandókat is:

`$\pi$`

```
<mi> &pi; </mi>

\pi
```

`$i$`

```
<mi> &ImaginaryI; </mi>
```

i

e

`<mi> &ExponentialE; </mi>`

e

### 1.2.3. Számok

Ezzel az elemmel jelöljük az olyan adatokat, melyeket számként szeretnénk megjeleníteni, tehát a számokat leíró szövegeket is jelölhetjük így, ha számértékként akarjuk értelmezni őket (lásd a példákat). Mint látni fogjuk, az **<mn>** elemmel előjel nélküli egészeket és valós számokat jelölhetünk, a többi számfajtát (pl. a negatív és komplex számok, a törtek, stb.) összetett elemformákkal jelöljük.

Az **<mn>** elem *fontstyle* attribútumának alapértelmezett értéke „normal”, tehát nem dőlt és nem vastagított.

A T<sub>E</sub>X matematikai módban automatikusan normál betűs megjelenítésre vált, ha számokat írunk, így erre nincs külön parancs, bár egyes esetekre léteznek parancsok. Érdeemes megjegyezni, hogy ha számot jelentő, de nem számokat tartalmazó szöveget akarunk megjeleníteni, akkor a **\text** parancs segítségével írjuk be a szöveget.

Példák:

2

`<mn> 2 </mn>`

2

0,123

`<mn> 0,123 </mn>`

0,123

1.000.000

`<mn> 1.000.000 </mn>`

1.000.000

2.1e10

`<mn> 2.1e10 </mn>`

$\text{\text{2.1e10}}$

$0xFFEF$

$\langle mn \rangle 0xFFEF \langle /mn \rangle$

$\text{\text{0xFFFF}}$

$MCMLXIX$

$\langle mn \rangle MCMLXIX \langle /mn \rangle$

$\text{\text{MCMLXIX}}$

$huszonegy$

$\langle mn \rangle huszonegy \langle /mn \rangle$

$\text{\text{huszonegy}}$

Példák a többi számfajta jelölésére (negatív számok, komplex számok, törtek, szimbolikus állandók):

$-1$

$\langle mrow \rangle$   
 $\langle mo \rangle - \langle /mo \rangle$   
 $\langle mn \rangle 1 \langle /mn \rangle$   
 $\langle /mrow \rangle$

$-1$

$2+3i$

$\langle mrow \rangle$   
 $\langle mn \rangle 2 \langle /mn \rangle$   
 $\langle mo \rangle + \langle /mo \rangle$   
 $\langle mrow \rangle$   
 $\langle mn \rangle 3 \langle /mn \rangle$   
 $\langle mo \rangle \&InvisibleTimes; \langle /mo \rangle$   
 $\langle mi \rangle \&ImaginaryI; \langle /mi \rangle$   
 $\langle /mrow \rangle$   
 $\langle /mrow \rangle$

$2+3i$

$\frac{1}{2}$

$\langle mfrac \rangle$   
 $\langle mn \rangle 1 \langle /mn \rangle$   
 $\langle mn \rangle 2 \langle /mn \rangle$   
 $\langle /mfrac \rangle$

$\text{\frac{1}{2}}$

Jegyezzük meg a következőket:

$\pi$

$\langle mi \rangle \&pi; \langle /mi \rangle$

$\pi$

$e$

$\langle \text{mi} \rangle \text{ \&ExponentialE; } \langle / \text{mi} \rangle$

$e$

#### 1.2.4. Operátorok

Az **<mo>** elem jeleníti meg a műveleteket, és más, műveleti jelként használt jeleket. Ez utóbbi csoportot két részre oszthatjuk: az elsőbe a különféle záró- és elválasztójelek tartoznak, a másodikba pedig a matematikai felüljelzések (pl. a komplementer jele).

A fent már megismert attribútumokon kívül az **<mo>** a következőket kaphatja meg: *form*, *fence*, *separator*, *lspace*, *rspace*, *stretchy*, *symmetric*, *maxsize*, *minsize*, *largeop*, *movablelimits*, *accent*. Az egyes attribútumokról a későbbiekben részletesebben is szót ejtünk.

A T<sub>E</sub>X-ben ennek az elemnek az egyes műveleti jelek ill. parancsok begépelése felel meg.

Példák a műveleti jelek szokásos jelölésére:

+

$\langle \text{mo} \rangle + \langle / \text{mo} \rangle$

+

<

$\langle \text{mo} \rangle \text{ \&lt; } \langle / \text{mo} \rangle$

<

≤

$\langle \text{mo} \rangle \text{ \&le; } \langle / \text{mo} \rangle$

$\backslash \text{leq}$

++

$\langle \text{mo} \rangle ++ \langle / \text{mo} \rangle$

++

$\Sigma$

```
<mo> &sum; </mo>
```

```
\sum
```

$.NEM.$

```
<mo> .NEM. </mo>
```

```
.NEM.
```

$és$

```
<mo> és </mo>
```

```
\'es
```

„láthatatlan szorzásjel”

```
<mo> &InvisibleTimes; </mo>
```

Ez utóbbira a T<sub>E</sub>X-ben nincs megfelelő parancs.

Példák a záró- és elválasztójelek használatára (Fontos, hogy ezekben a példákban nem a *fence* és *separator* attribútumok használatát mutatjuk be!):

$(a+b)$

```
<mrow>
  <mo> ( </mo>
  <mrow>
    <mi> a </mi>
    <mo> + </mo>
    <mi> b </mi>
  </mrow>
  <mo> ) </mo>
</mrow>
```

$(a+b)$

$[0,1)$

```
<mrow>
  <mo> [ </mo>
  <mrow>
    <mn> 0 </mn>
    <mo> , </mo>
    <mn> 1 </mn>
  </mrow>
  <mo> ) </mo>
</mrow>
```

$[0,1)$

$f(x,y)$

```
<mrow>
  <mi> f </mi>
  <mo> &ApplyFunction; </mo>
  <mrow>
    <mo> ( </mo>
```

```

    <mrow>
      <mi> x </mi>
      <mo> , </mo>
      <mi> y </mi>
    </mrow>
    <mo> ) </mo>
  </mrow>
</mrow>

f(x,y)

```

A matematika jelölésben használunk néhány, úgynevezett „láthatatlan műveletet”, melyek vizuálisan valóban nem látszanak, de logikailag szükséges őket jelölni. Ezek a következők:

*&InvisibleTimes; (röviden: &it;):*  $xy$

```

<mrow>
  <mi> x </mi>
  <mo> &InvisibleTimes; </mo>
  <mi> y </mi>
</mrow>

xy

```

*&ApplyFunction; (röviden: &af;):*  $f(x)$  vagy  $\sin x$

```

<mrow>
  <mi> f </mi>
  <mo> &ApplyFunction; </mo>
  <mrow>
    <mo> ( </mo>
    <mi> x </mi>
    <mo> ) </mo>
  </mrow>
</mrow>

<mrow>
  <mi> sin </mi>
  <mo> &ApplyFunction; </mo>
  <mi> x </mi>
</mrow>

f(x)

\sin x

```

*&InvisibleComma; (röviden: &ic;):*  $m_{12}$

```

<msub>
  <mi> m </mi>
  <mrow>
    <mn> 1 </mn>
    <mo> &InvisibleComma; </mo>
    <mn> 2 </mn>
  </mrow>
</msub>

m_{12}

```

Mint látjuk, T<sub>E</sub>X-ben ezeket nem kell külön jelölni.

A legtöbb matematikai jelnek megvan a tradicionális jele, amelybe beletartozik a körülötte lévő üres hely is. Ezt a MathML megpróbálja szabályosan megjeleníteni, ám vannak olyan helyzetek, mikor ugyanazt a szimbólumot többféle határoló üres hellyel használhatjuk. Erre a legjobb példa a „+” jel: ha összeadást jelölünk vele, akkor mindkét oldalán egyforma üres hely van, de ha a „pozitív” jelzésére használjuk, akkor csak a bal oldalán. Erre használjuk a *form* attribútumot. Három értéke lehet: „*prefix*” (ha csak előtte kell üres hely), „*infix*” (ha mindkét oldalon), „*postfix*” (ha csak utána kell). A fenti okból nem lehet egyértelmű alapértelmezést adni neki, de néhány szabályt azért felállíthatunk:

- Ha a vonatkozó műveleti jel az első az **<mrow>** attribútumai között, akkor „*prefix*” lesz az alapértelmezés.
- Ha a vonatkozó műveleti jel az utolsó az **<mrow>** attribútumai között, akkor „*postfix*” lesz az alapértelmezés.
- Minden más esetben (pl. ha a műveleti jel nem része egyetlen **<mrow>**-nak sem), akkor „*infix*” az alapértelmezett.

A nyitó zárójeleknek „*prefix*”, a záróknak „*postfix*”, az elválasztójeleknek „*infix*” az alapértelmezésük, de néhány esetben ez függhet a környezettől is.

A műveleti jelek köré a megszokottól eltérő méretű üres helyet rendelhetünk az *lspace* és az *rspace* segítségével (a kezdő „l” a bal, az „r” a jobb oldalt jelöli). Az értéket a már előzőekben említett *em* nevű mértékegységben kell megadni, a mértékegység kiírása nélkül.

T<sub>E</sub>X-ben erre nincs szükség, hiszen matematikai módban a matematikai stílusnak megfelelően jeleníti meg az ilyen operátorok körüli üres helyeket. Arra azonban lehetőségünk van, hogy egy adott operátor körül ne jelenjen meg üres hely, ilyenkor tegyük a műveleti jelet **{ }** jelek közé: {+}.

Négy attribútum van, amely segítségével megnyújthatjuk az egyes műveleti jeleket. Ezek: a *stretchy*, a *symmetric*, a *maxsize*, és a *minsize*. Ha egy műveleti jel *stretchy* attribútumának „true” az értéke (ez az alapértelmezés), akkor, ha szükséges, a megjelenítő rendszer meg fogja nyújtani a szükséges elemeket. Ez azt jelenti, hogy az adott jel minden irányban megnő. A növekedés mértékét a *maxsize* és *minsize* attribútumokkal szabályozhatjuk. Ezek értéke egy szám, amely a nyújtás szám-szoros értékét jelenti. Végül a negyedik attribútummal, a *symmetric*-kel az adott karakter tengelyéhez viszonyított függőleges szimmetriáját állíthatjuk be (ha „true”, akkor szimmetrikus lesz a nyújtás). Nem négyzetes mátrixnál vehetjük nagy hasznát, ha „false”-ra állítjuk.

TEX-ben erre parancsok sorozata áll rendelkezésünkre, melyek a következő formájúak (a legkisebbtől a legnagyobbig): **\big $x$** , **\Big $x$** , **\bigg $x$** , **\Bigg $x$** ; ahol az  $x$  a helyzetet mutatja ( $l$  – bal oldal,  $m$  – közép,  $r$  – jobb oldal), erre az üres helyek helyes elrendezése miatt van szükség.

A másik módszer a **\left $x$** , **\right $x$**  parancs pár használata, ahol  $x$  helyén a zárójel alakját kell megadni. Mindig párban és azonos zárójellel kell őket használni. Ha arra van szükségünk, hogy egy adott kifejezés két oldalán különböző zárójeleket adjunk meg, akkor az egyik oldalon használjuk a „.” határoló jelet:

( $a$ ]

`\left(n\left.\right]\right.`

A következő példa a szabályos  $\left(\frac{a}{b}\right)$  forma helyett a  $\left(\frac{a}{b}\right)$  formát adja:

```
<mrow>
  <mo maxsize="1"> ( </mo>
    <mfrac>
      <mi> a </mi>
      <mi> b </mi>
    </mfrac>
  <mo maxsize="1"> ) </mo>
```

```
</mrow>
(\frac ab)
```

Jegyezzük meg, hogy a két zárójel független egymástól, tehát csak egy `maxsize="1"` megadása eltérő méretű zárójelet eredményez.

A függőleges nyújtás szabályai:

- Ha a nyújtani kívánt részkifejezés egy **<mrow>** vagy egy **<mtd>** része, akkor nyújtás a részkifejezés teljes méretére terjed ki, más esetekben pedig a megadott *maxsize*, *minsize* párban megadott értékekre.
- Ha a *symmetric="true"*, akkor ez magasabb prioritású, mint a *maxsize*, *minsize* párban megadott értékek.

A legtöbb zárójel, a  $\sum$ , az  $\int$ , a  $/$ , valamint a függőleges nyílak automatikusan megnyúlnak.

A vízszintes nyújtás szabályai:

- Ha a nyújtani kívánt részkifejezés egy **<munder>**, **<mover>**, **<munderover>**, vagy egy **<mtd>** része, akkor nyújtás a részkifejezés teljes méretére kiterjed.
- Más esetben csak a kifejezés magjában található műveletre terjed ki a hatása.

A legtöbb vízszintes nyíl és felüljelzés automatikusan megnyúlik. Lássunk egy példát a vízszintes nyúlásra is:

```
<mrow>
  <mi> x </mi>
  <mover>
    <mo> &RightArrow; </mo>
    <mtext> leképezés </mtext>
  </mover>
  <mi> y </mi>
</mrow>
x@>lek\ 'epz\ 'es>>y
```

A fenti kód az  $x \xrightarrow{\text{leképezés}} y$  kifejezést eredményezi.

A *largeop* attribútum megnöveli a műveleti jel megjelenítési méretét, ha az **<mstyle>** elem *displaystyle* attribútuma „true”-ra

van állítva. A nagyobb műveleti jelekre legjobb példa az integráljel ( $\int$ ).

Erre a TEX a **\displaystyle** parancsot használja.

Ha a *movablelimits* attribútum „true”-ra van állítva, akkor a felül- és alulírások felső ill. alsó indexként jelennek meg. Erre jó példa a szumma ( $\sum$ ). Hasonló módon működik az *accent* attribútum is a felüljelzésekkel kapcsolatban. TEX-ben a *movablelimits* „true” értékének a **\nolimits**, „false” értékéneke pedig a **\limits** parancs fele meg, melyek után kell megadni a vonatkozó felül- és alulírásokat.

A *separator* attribútum egy automatikus sortörést csinál, a *form* attribútum értékétől függően a művelet előtt, vagy után. TEX-ben erről nekünk kell gondoskodnunk.

A *fence* attribútum vizuálisan nem jelenik meg, akkor van értelme használni, ha valamilyen különleges, nem vizuális megjelenítőn kívánjuk bemutatni kifejezéseinket, ilyenkor ez az attribútum többletinformációt ad a zárójelezésről. Nincs megfelelője TEX-ben.

### 1.2.5. Szövegek

Az **<mtext>** elemmel tetszőleges szöveget helyezhetünk el a kifejezésekben, úgy, hogy az változtatás nélkül megjelenik. Általában a kifejezésekhez nem szervesen kapcsolódó megjegyzéseket hozzáfűzésére használjuk.

Az **<mtext>** elemben használhatunk olyan karaktereket (főként üres helyeket), melyek más elemeknél nem jelennének meg. Ilyenek pl. a `&Thinspace;`, a `&NegativeThinspace;`, vagy a `&zerowidthspace;`. (Lásd az A Függelékét!)

Ez az elem a fejezet elején már említett karakterformázó attribútumokat kaphatja meg.

Néhány példa a használatára:

```
<mtext> 1. tétel: </mtext>
<mtext> &ThinSpace; </mtext>
<mtext> &ThickSpace;&ThickSpace; </mtext>
<mtext> /* egy megjegyzés */ </mtext>
```

Szöveges kifejezések létrehozására használhatjuk az **<ms>** elemet is. Alapértelmezésben idézőjelekkel közrezárt szöveget ír le. Leginkább számítógépes algebrát használó rendszerekkel (pl. programozási nyelvekkel) való kommunikációra használható. Sohasem használhatjuk olyan szövegek jelölésére, amelyet **<mtext>**, **<mo>**, vagy **<mi>** elemmel írunk le.

Az **<ms>** elemben leírt szöveg Unicode karaktereket tartalmazhat, a különleges karakterek közül a legtöbbnek megfelel valamilyen MathML műveleti jel. Például, ha az angol „és” jelet („&” – neve: *ampersand*) kívánjuk leírni, akkor azt a következő módon tehetjük:

```
<ms>&amp;</ms>
```

de ha a jel nevét („&amp;”) akarjuk lejegyezni, akkor a következő formát kell használnunk:

```
<ms>&amp;amp;</ms>
```

Természetesen ebben az elemben is érvényes az üres helyekről már elmondottak, úgyhogy használatakor legyünk erre tekintettel.

A már ismertetett attribútumok mellett az **<ms>** elemnek van két sajátja is: az *lquote* és az *rquote*, melyek segítségével átírhatjuk az alapértelmezett idézőjeleket (&quot;).

Egyes esetekben a szöveget érdekesebb a **<mo>** vagy **<mi>** elemek segítségével megjeleníteni, ha jelentésük inkább kapcsolódik a műveletekhez, vagy azonosítókhoz. A  $\exists \delta > 0 \ni f(x) < 1$  kifejezést „érthetőbben” leírhatjuk úgy is, hogy:

Létezik olyan  $\delta > 0$  amelyikre  $f(x) < 1$ , megjeleníteni pedig a következő kóddal tudjuk:

```
<mrow>
  <mo> Létezik olyan </mo>
  <mrow>
    <mrow>
      <mi> &delta; </mi>
      <mo> &gt; </mo>
      <mn> 0 </mn>
    </mrow>
    <mo> amelyikre </mo>
    <mrow>
      <mrow>
        <mi> f </mi>
        <mo> &ApplyFunction; </mo>
        <mrow>
          <mo> ( </mo>
            <mi> x </mi>
            <mo> ) </mo>
          </mrow>
        </mrow>
      <mo> &lt; </mo>
      <mn> 1 </mn>
    </mrow>
  </mrow>
</mrow>
```

Egy másik példa lehet az  $1 + \dots + n$ , melynek kódja a következő:

```
<mrow>
  <mn> 1 </mn>
  <mo> + </mo>
  <mi> ... </mi>
  <mo> + </mo>
  <mi> n </mi>
</mrow>
```

A nem szervesen kapcsolódó szövegre is álljon itt egy példa: „1.

tétel: ha  $x > 1$ , akkor  $x^2 > x$ ”. Kódja:

```
<mrow>
  <mtext> 1, tétel: ha </mtext>
  <mi> x </mi>
  <mo> &gt; </mo>
  <mn> 1 </mn>
  <mtext> , akkor </mtext>
  <msup>
    <mi> x </mi>
    <mn> 2 </mn>
  </msup>
  <mo> &gt; </mo>
  <mi> x </mi>
</mrow>
```

Ha HTML környezetbe ágyazzuk be a fenti szöveget, akkor célszerűbb a tényleges szöveget HTML-ben megírni, és csak a matematikai kifejezéseket MathML-ben, ezzel is jelezve, hogy a szöveg nem szerves része a kifejezésnek.

TEX matematikai módjában szövegek írására használjuk a **\text** parancsot. Itt **{}** jelek között bármilyen szöveget megadhatunk, az illeszkedni fog a környező kifejezések tulajdonságaihoz (méret, betűtípus, stb.), de normál írással (és nem dőlten) jelenik meg. Szóközöket a normál szövegírásnak megfelelően használhatunk, tehát nincs szükség külön vezérlő parancsokra. Ha lehetőségünk van rá, egyszerűbb, ha csak a matematikai kifejezések miatt kapcsolunk át matematikai módba, egyébként normál szöveges módban írjuk dokumentumunkat.

Ha csak néhány karaktert kell szöveggént kezelni, használhatjuk az egyúttal betűtípusváltó **\roman**, **\bold**, **\slanted**, **\italic** parancsokat. Több karakteres szöveg esetén használjuk a **{}** zárójelpárt!

Példák:

Létezik olyan  $\delta > 0$ , amelyekre  $f(x) < 1$

`\text{L\'etezik olyan} \delta > 0 \text{, amelyekre} f(x) < 1`

vagy

`L\'etezik olyan  $\delta > 0$ , amelyekre  $f(x) < 1$`

$12^{\text{th}}$

`12^{\roman{th}}`

### 1.3. Stílusok

Az **<mstyle>** elemet az egyes kifejezések stílusainak megváltoztatására használjuk. Több vonatkozó argumentum esetén azokat egy **<mrow>** elembe kell zárni.

Egy elem valamely attribútumát kétféle képen tudjuk beállítani, vagy magának az elemnek a nyitótagjában, vagy az **<mstyle>**

elem vonatkozó attribútumának beállításával. A két beállítás között azonban van egy kis különbség: ha egy olyan attribútumot állítunk be az **<mstyle>** elem segítségével, amely nem öröklött, hanem van valamilyen alapértelmezése (pl.: az **<mfrac>** elem *linethickness* attribútuma, melynek alapértelmezése 1), akkor az az összes beágyazott utód elemre vonatkozik, míg az adott elemnél történő beállítás csak az adott elemre. Egyetlen kivétel van, az **<mpadded>** elem, melyre nem vonatkoznak az **<mstyle>** elem beállításai.

Minden argumentum örökli az **<mstyle>** elem beállításait, míg felül nem bíráljuk azt.

A már ismert közös attribútumokon kívül az **<mstyle>**-nak a következő saját attribútumai lehetnek: *displaystyle*, *scriptlevel*, *scriptsizemultiplier*, *scriptminsize*, *color*, *background*.

A *displaystyle* befolyással van egyes más elemek attribútumaira (mint pl. az **<mo>** elem *movablescripts* attribútuma). Általános szokás, hogy ezt az attribútumot „true”-ra állítjuk, kivéve a törteknél, a gyökvonásnál, a táblázatoknál és mátrixoknál, ahol ez az érték „false” lesz, ezzel esetleg kisebb betű és határoló üres hely méretre adva lehetőséget.

A *scriptlevel*-lel a betűtípusok méretét tudjuk beállítani. Minél nagyobb értéket adunk az attribútumnak, annál kisebb lesz a betű mérete (mint az indexekben). Ha előjel nélkül adjuk meg az értéket, akkor a megadott értékre állítjuk be az attribútumot. Ha előjellel adjuk meg, akkor az előjeltől függően az aktuális értéket növeljük vagy csökkentjük. A *fontsize* attribútum módosítása nincs hatással a *scriptlevel*-re.

A *scriptlevel* megváltoztatásával a változás mértékétől függően a *scriptsizemultiplier* értéke automatikusan megsokszorozódik. Például, ha a *scriptlevel*-t növeljük kettővel, akkor a betűk mérete a *scriptsizemultiplier*-szeresére változik kétszer egymás után. Ha a

*scriptlevel* értékét 2-re állítjuk, úgy, hogy előtte 3 volt, akkor a betűméret elosztódik a *scriptsizemultiplier* értékével.

A *scriptsizemultiplier* alapértéke megközelítőleg  $\sqrt{0,5}$ , ezért kapunk kisebb betűméretet a *scriptlevel* növelésével. Hogy megelőzzük a szövegek olvashatatlanságát, a *scriptminsize* értékénél kisebb érték alá nem csökkenhet a betűk mérete. A minimum értéket a már ismert mértékegységekben adhatjuk meg, de itt legtöbbször a *pt*-t használjuk.

Amelyik megjelenítő képes a többszintű stíluslapok megjelenítésére, ott e stíluslapok használatával is beállíthatjuk a fenti attribútumok értékét. Itt viszont figyelembe kell venni azt a tényt, hogyha ugyanarra az elemre beállítunk értékeket a *scriptlevel*-ben és egy stíluslapon is, akkor a megjelenítő először a *scriptlevel* beállításait végzi el, majd a stíluslapét.

Ha egy elem *fontsize* attribútumát egy stíluslap segítségével is és egy *scriptlevel*-el is beállítottuk, akkor annak utódának beállításakor az öröklés után a stíluslap beállításai történnek meg először.

A *color* attribútummal az elem és a hozzá tartozó rajzolatok (törtvonal, gyökjel, stb.) színét adhatjuk meg.

A *background* attribútummal az elem hátterének színét állíthatjuk be. Alapértéke az „átlátszó” („*transparent*”), a beállított értékek nem öröklődnek.

A fenti két attribútum értékét a HTML-ben már megszokott, három vagy hatjegyű színkódokkal - melyeket egy kettős kereszt előz meg -, vagy pedig a HTML-színnevekkel (pl.: „*#0000FF*”, „*#00F*” vagy „*blue*”) állíthatjuk be. A *background* attribútumnál engedélyezett a „*transparent*” szöveg megadása is. Természetesen ezeket az értékeket is definiálhatjuk egy stíluslapon.

TEX-ben ennek az elemnek konkrét megfelelője nincs, attribútumai közül a *displaystyle* attribútumhoz közel áll a TEX

**\displaystyle** parancsa, hasonlóan a *scriptlevellel* megfeleltethető a **\scriptstyle** és a **\scriptscriptstyle**, bár T<sub>E</sub>X-ben ezek fix méretűek (az index és az index indexét jelölhetik) és csak konkrét beállításokat végezhetünk velük, nem lehet örökíteni a tulajdonságokat. Színbeállításra T<sub>E</sub>X nincs lehetőségünk.

Az általam is bemutatni kívánt T<sub>E</sub>X implementáció, az AMS-T<sub>E</sub>X lehetővé teszi stílusállományok használatát, melyeket a **\documentstyle** paranccsal tölthetünk be (a stílusállományok alapértelmezett kiterjesztése .sty). Az AMS-T<sub>E</sub>X esetén azonban ekkor szükség van az amstex.tex dokumentum előzetes megnyitására is:

```
\input amstex
\documentstyle{amsppt}
    saját beállítások
\document
    ide jön a dokumentumunk
\enddocument
```

Ha ez megtörtént, akkor a stílusállományban előre definiált stílusokat mint parancsokat használhatjuk.

Az **<mo>** elemnél látott zárójelnyújtás példája:

```
<mrow>
  <mo maxsize="1"> ( </mo>
    <mfrac>
      <mi> a </mi>
      <mi> b </mi>
    </mfrac>
    <mo maxsize="1"> ) </mo>
</mrow>
```

átírva az **<mstyle>** elemre:

```
<mstyle maxsize="1">
  <mrow>
    <mo> ( </mo>
    <mfrac>
      <mi> a </mi>
      <mi> b </mi>
    </mfrac>
    <mo> ) </mo>
  </mrow>
</mstyle>
```

```
</mrow>  
</mstyle>  
(\frac ab)
```

## 1.4. Üres helyek kihagyása

Üres helyek megadására három féle lehetőségünk van: az **<mspace/>**, az **<mpadded>** és az **<mphantom>** elemek segítségével.

Az **<mspace/>** elem segítségével egy általunk az attribútumaiban meghatározott üres helyet hozhatunk létre, mely csak az utána álló elemekre vonatkozik.

A használható attribútumok: a *width* (oldalirányban), a *height* (fent) és a *depth* (lent). Mindhárom alapértelmezése 0, tehát ha használjuk ezt az elemet, mindenképpen adjunk valamilyen értéket nekik.

Az **<mspace/>** elemen kívül számos más elem rendelkezik azzal a tulajdonsággal, hogy valamilyen üres helyet hozunk létre vele, erre az egyes elemek tárgyalásánál kitérünk.

Az **<mpadded>** elem segítségével úgy jeleníthetjük meg annak tartalmát, hogy az a környező üres helyeket saját szánk íze szerint beállított módon jelenítse meg. Ha több elemet ágyazunk bele, akkor azokat egy **<mrow>** elembe kell zárni. Hatása – az **<mspace/>** elemmel ellentétben – csak a közrezárt elemekre vonatkozik.

Saját attribútumai: *width*, *lspace*, *height*, *depth*. A *width* attribútum a „keret” szélességét határozza meg, növelésével a jobb oldalon növekszik az üres hely. Az *lspace* hasonló a *width*-hez, de ez a bal oldali üres helyért felelős. A *height* az alapvonalról (a normál szöveg legtöbb betűjének alsó szélétől) számított felfelé irányuló magasságot jelöli, a *depth* pedig szintén az alapvonalról számított, de lefelé irányuló magasságot („mélységet”) adja meg. Értékeiket több módon lehet megadni: konkrét számérték

megadásával, növeléssel/csökkentéssel (hasonlóan a *scriptlevel*-hez), vagy százalékos megadással. Számérték megadásakor használhatunk mértékegységet is. Létezik még egy érdekesebb megadási mód is, amikor az ún. „pseudo-mértékkel” adjuk meg az értéket. Ennek módja a következő: megadjuk számmal (vagy százalékkal) a kívánt szorzót és utána mértékegységnek az attribútumok valamelyikét.

Példák az attribútumok megadására: mind a `depth="100% height"` és a `depth="1.0 height"` megadása azt jelenti, hogy az elem mélysége és magassága egyforma legyen. A `depth="105%"` forma a mélység 1,05-szeresét jelenti. A következő két példa ismét ugyanazt jelenti, a mélység legyen kétszeres: `depth="+100%"` vagy `depth="200%"`.

A következő példák ugyanazt az eredményt adják, nem változtatnak a kiírás kiterjedésén:

```
<mpadded width="+0em"> ... </mpadded>
<mpadded width="+0%"> ... </mpadded>
<mpadded width="-0em"> ... </mpadded>
<mpadded width="- 0 height"> ... </mpadded>
<mpadded width="100%"> ... </mpadded>
<mpadded width="100% width"> ... </mpadded>
<mpadded width="1 width"> ... </mpadded>
<mpadded width="1.0 width"> ... </mpadded>
<mpadded> ... </mpadded>
```

Legyünk óvatosak az **<mpadded>** használatával, mivel nincsenek alapértelmezett attribútumok, a különböző megjelenítők más-más formában adhatják vissza az általunk készített kifejezéseket. Ezért ne próbálkozzunk a „negatív üres helyek” segítségével különleges jeleket létrehozni, mert valószínű, hogy azt egy másik megjelenítővel máshogy fogjuk látni. Ráadásul ezek a megoldások értelmezhetetlenek egy hangos megjelenítő, vagy egy számítógépes algebrai rendszer számára. Magyarul: ne üres hellyel próbáljunk jelentést adni a kifejezéseknek! Ha mégis ezt szeretnénk használni inkább a **<semantics>** elemet (lásd ott!).

Lássuk például a következő hibás MathML kódot:

```
<mrow>
  <mpadded width="0">
    <mi> C </mi>
  </mpadded>
  <mstyle width="0.3em"/>
  <mtext> | </mtext>
</mrow>
```

A fenti példa látszólag egy függőlegesen áthúzott nagy C betűt hivatott volna megjeleníteni, ami azonban valójában értelmetlen. Próbálkozzunk inkább a következő formával:

```
<semantics>
  <mrow>
    <mpadded width="0">
      <mi> C </mi>
    </mpadded>
    <mstyle width="0.3em"/>
    <mtext> | </mtext>
  </mrow>
  <annotation-xml encoding="mathml">
    <mi> &DoubleStruckCapitalC; </mi>
  </annotation-xml>
</semantics>
```

Látszólag ugyanaz jelenik meg, de a hozzáfűzött **<annotation-xml>** elem segítségével a nem vizuális felhasználó számára is értelmessé tettük kifejezésünket.

Fontos megjegyezni, hogy az **<mstyle>** elem beállításai nincsenek hatással az **<mpadded>** beállításaira!

Az **<mphantom>** elem vizuálisan nem jelenik meg, de az általa fizikailag elfoglalt hely megmarad, magyarul az így leírt elem helyett a méretének megfelelő üres hely kerül megjelenítésre (hangos megjelenítés esetén az üres helyet szünet jelképezi). Az **<mphantom>** elem teljesen úgy viselkedik, mint egy **<mrow>** elem, azzal a különbséggel, hogy ez utóbbi nem látszik.

Az elemnek nincsen saját attribútuma.

A következő példában a tört számlálójában és nevezőjében található, megegyező változók pontosan egymás alá kerülnek kiírásra:

$$\frac{x+y+z}{x+z}$$

```
<mfrac>
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mrow>
    <mi> x </mi>
    <mphantom>
      <mo form="infix"> + </mo>
      <mi> y </mi>
    </mphantom>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
</mfrac>
```

Az **<mo>** elem *form* attribútumának „infix” beállítása az üres helyek helyes szélessége miatt szükséges (e nélkül a + „prefix” lenne). A fenti példa egy másik megoldása a *form* állítása nélkül:

```
<mfrac>
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mrow>
    <mi> x </mi>
    <mphantom>
      <mo> + </mo>
    </mphantom>
    <mphantom>
      <mi> y </mi>
    </mphantom>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
</mfrac>
```

A T<sub>E</sub>X automatikusan kezeli a kifejezések körüli üres helyeket. Azonban lehetőségünk van vízszintesen plusz hely kihagyására, a következő parancsokkal (a legkisebb helykihagyástól a legnagyobbig): **\,**, **\;**, **\** („\” jel és egy szóköz), **\quad**, **\qquad**. Üres hely csökkentése a **\!** paranccsal lehetséges (akár többször is egymásután. Erre az esetre egy másik megoldás a műveleti

jeleknél, ha  $\{ \}$  zárójelek közé tesszük őket, ekkor a rendszer nem hagy a műveleti jel előtt és után üres helyet.

Függőleges helykihagyásra két lehetőségünk van, azonban ezeket csak kizárólagos matematikai módban (a  $\$$  nyitó és záró karakterek között) használhatjuk. Az egyik a **\spreadlines** (mátrixok esetén **\spreadmatrixlines**) parancs, melyet rögtön a nyitó  $\$$  jelek után kell használni és a matematikai mód zárásáig érvényes. A másik parancs a **\vspace**, amelyet az után a sor után kell beírni, ami után ki akarjuk hagyni a helyet. Ez utóbbi csak olyan parancsoknál használható, amelyek nyitó és záró paranccsal is rendelkeznek (pl. **\align**, lásd később). Mindkét esetben a megismert mértékegységekkel,  $\{ \}$  zárójelek között kell megadni a kihagyni kívánt üres hely nagyságát.

## 1.5. Normál szövegektől eltérő szerkezetek

Ebben a fejezetben az alapjelölések (pl. törtek, gyökök) leírásával foglalkozó elem kerül bemutatásra.

### 1.5.1. Kifejezések csoportba rendezése

Az **<mrow>** elemet több részkifejezés (általában valamilyen műveletei jellel összekapcsolt kifejezés) csoportba rendezésére szolgál.

Ennek az elemnek nincsenek saját argumentumai, csak a már az előző fejezetben megismert általános attribútumokat kaphatja meg.

Az **<mrow>** elem tipikusan egymás után sorban megjelenítendő elemeket kapcsolja össze. Határoló üres helyek tekintetében a felsorolt elemek a meghatározók, az **<mrow>** elem külön nem jelenít meg üres helyet.

Az **<mrow>** elemeket egymásba ágyazhatjuk, ha egy kifejezés logikája azt megkívánja.

TEX-ben a  $\{ \}$  zárójel pár felel meg ennek az elemnek, bár nem mindig azonos helyzetben használjuk. A  $\{ \}$  jeleket olyan parancsok után is használnunk kell, amelyek egyetlen karakterre vonatkoznak, de a zárójelek használatával az egész zárójeles kifejezésre kiterjeszthetők (`\roman{ha}`). Használatos még az üres helyek elnyomására is (`{+}`) Sokszor, a MathML-lel ellentétben, nem használjuk.

$$2x + y - z$$

```
<mrow>
  <mrow>
    <mn> 2 </mn>
    <mo> &InvisibleTimes; </mo>
    <mi> x </mi>
  </mrow>
  <mo> + </mo>
  <mi> y </mi>
  <mo> - </mo>
  <mi> z </mi>
</mrow>
```

$$2x+y-z$$

$$(x,y)$$

```
<mrow>
  <mo> ( </mo>
  <mrow>
    <mi> x </mi>
    <mo> , </mo>
    <mi> y </mi>
  </mrow>
  <mo> ) </mo>
</mrow>
```

$$(x,y)$$

### 1.5.2. Törtek

Az **<mfrac>** elemet használjuk törtek, binomiális együtthatók megjelenítésére.

Egy saját attribútuma van, a *linethickness*, mely segítségével a törtvonal vastagságát állíthatjuk be, alapértelmezése 1 (normál törtvonal). Binomiális együtthatóknál ezt 0-ra kell beállítani, emeletes törteknél alkalmazhatjuk a főtörtvonal megvastagítására a 2-es értéket. Ezeket az értékeket helyettesíthetjük kulcsszavakkal is: „thin” (0), „medium” (1), „thick” (2).

T<sub>E</sub>X-ben a számos parancs felel meg ennek az elemnek. Az egyszerű törteket a **\frac** paranccsal, a binomiális együtthatókat pedig a **\binom** paranccsal jeleníthetjük meg. Létezik parancs olyan törtek előállítására, amelyek egyúttal tartalmazza a zárójeleket is, ez a **\fracwithdelims**, melynél először a zárójelek alakját kell megadni, majd a tört tagjait (ekkor a zárójelek közelebb kerülnek a törthöz, mint a **\left**, **\right** parancsoknál). Létezik parancs a más vastagságú törtvonalra is, a **\thickfrac** mely után a **\thickness** paranccsal kell megadni a vonalvastagságot (0,4 az alapvastagság), majd a tört tagjait. Ennek a zárójeles változata a **\thickfracwithdelims** parancs.

Ezek a parancsok a normál nagyságú írásnál kisebb törteket/binomiális együtthatókat jelenítenek meg. Ha ennél nagyobbra van szükség, akkor használjuk a **\dfrac** és a **\dbinom** parancsokat (ezeknek nincs zárójeles változatuk).

Léteznek még lánctörtek írására alkalmas parancsok, a **\cfrac**, az **\lfrac** és az **\rfrac**, melyeket a **\endcfrac** paranccsal zárunk le. Különbség köztük a számláló elhelyezésében van (a *c* középen, az *l* bal oldalon, az *r* jobb oldalon helyezi el a számlálót).

A fenti lehetőségeken kívül léteznek még speciális törtleírási parancsok is. Megadásuk módja: *számláló* **\parancs** *nevező*. A parancsok: **\over** (normál tört), **\atop** (binomiális együttható), **\above** (vastag törtvonal); ugyanezek zárójelekkel: **\overwithdelims**, **\atopwithdelims**, **\abovewithdelims**; sőt binomiális együttható típusú kifejezésekhez van kerek zárójeles (**\choose**), szögletes zárójeles (**\brack**) és kapcsos zárójeles (**\brace**) parancs is.

Lássunk egy példát a binomiális együtthatóra:  $\binom{a}{b}$

```
<mrow>
  <mo> ( </mo>
  <mfrac linethickness="0">
    <mi> a </mi>
    <mi> b </mi>
```

```

    </mfrac>
    <mo> ) </mo>
  </mrow>

  \binom{a}{b}

```

Egy másik példa egy emeletes törtre:  $\frac{\frac{a}{b}}{\frac{c}{d}}$

```

<mfrac linethickness="2">
  <mfrac>
    <mi> a </mi>
    <mi> b </mi>
  </mfrac>
  <mfrac>
    <mi> c </mi>
    <mi> d </mi>
  </mfrac>
</mfrac>

{a\over c}\above{c\over d}

```

Egy általánosabb példa:  $\frac{1+\sqrt{5}}{2}$

```

<mfrac>
  <mrow>
    <mn> 1 </mn>
    <mo> + </mo>
    <msqrt>
      <mn> 5 </mn>
    </msqrt>
  </mrow>
  <mn> 2 </mn>
</mfrac>

frac{1+\sqrt{5}}{2}

{1+\sqrt{5}}\over 2

```

### 1.5.3. Gyökös kifejezések

A gyökvonás kifejezésére két lehetőségünk van, négyzetgyök esetén az **<msqrt>** elemet kell használni, más kitevő esetén pedig az **<mroot>** elemet. A két elemnek nincsen saját attribútuma.

Hasonlóan a MathML-hez, T<sub>E</sub>X-ben az **\sqrt** és a **\root** (és a hozzá tartozó **\of**) parancsok szolgálnak a gyökvonás leírására.

$\sqrt{5}$

```

<msqrt>
  <mn> 5 </mn>
</msqrt>

```

$\sqrt[3]{5}$

$\sqrt[3]{a}$

```
<mroot>
  <mi> a </mi>
  <mn> 3 </mn>
</mroot>
```

$\sqrt[3]{a}$

#### 1.5.4. Zárójelek

Az **<mfenced>** elem segítségével zárójelezhetjük kifejezéseinket. Ha az elemnek több argumentuma van, akkor azok közé elválasztójelek kerülnek.

Rögtön lássunk két példát, összehasonlítva a már megismert módszerekkel.

Az első egy egyszerű  $(x)$ :

```
<mfenced>
  <mi> x </mi>
</mfenced>
```

```
<mrow>
  <mo> ( </mo>
  <mi> x </mi>
  <mo> ) </mo>
</mrow>
```

A második pedig az  $(x,y)$  kódja:

```
<mfenced>
  <mi> x </mi>
  <mi> y </mi>
</mfenced>
```

```
<mrow>
  <mo> ( </mo>
  <mrow>
    <mi> x </mi>
    <mo> , </mo>
    <mi> y </mi>
  </mrow>
  <mo> ) </mo>
</mrow>
```

Az **<mfenced>** elemnek három saját attribútuma van: az *open*, a *close* és a *separators*; alapértelmezésük rendre „(”, „)”, „, ”.

Egy általános **<mfenced>** elem a következőképpen néz ki:

```
<mfenced open="nyitó zárójel" close="záró zárójel"
  separators="1. elválasztójel ... (n-1). elválasztójel">
  1. argumentum
  ...
  n. argumentum
</mfenced>
```

Alternatív módon az **<mfenced>** elem behelyettesíthető a következő formával:

```
<mrow>
  <mo fence="true"> nyitó zárójel </mo>
  <mrow>
    1. argumentum
    <mo separator="true"> 1. elválasztójel </mo>
    ...
    <mo separator="true"> (n-1). elválasztójel </mo>
    n. argumentum
  </mrow>
  <mo fence="true"> záró zárójel </mo>
</mrow>
```

Az *open* és *close* tetszőleges szövegek, a nyitó és záró zárójeleket írjuk le velük. A *separator* a argumentumokat elválasztó jeleket leíró szöveg, megadása nem kötelező, ebben az esetben nem lesznek elválasztójelek az egyes argumentumok között. Ha túl sok elválasztójelet adunk meg, a feleslegeseket figyelmen kívül hagyja a megjelenítő. Ha viszont túl keveset, akkor az utolsónak megadott jel fog a szükséges számban ismétlődni. A fenti attribútumokban szereplő üres helyeket nem vesszük figyelembe, viszont megadásuknál használhatunk speciális jeleket is.

Ha az **<mfenced>** elemet tartalom nélkül adjuk meg, akkor egy egyszerű zárójelpárt kapunk eredményül.

Ha egy teljes kifejezést akarunk zárójellezni, akkor azt helyezzük egy **<mrow>**-ba, különben a kifejezés összes eleme után elválasztójel fog megjelenni (lásd az első példát).

Egy megjegyzés: ha lehet, ne használjuk összeadandó kifejezéseknél a „+” jelet elválasztójelként!

TEX-ben ilyen lehetőségünk nincs, a zárójeleket és az elválasztójeleket külön-külön meg kell adnunk. Érdemes megjegyezni azt, hogy zárójellezéskor használjuk a **\left** és **\right** parancsokat, mert ezek dinamikusan változnak a kifejezés méretétől függően.

Példák:

$(a+b)$

```
<mfenced>
  <mrow>
    <mi> a </mi>
    <mo> + </mo>
    <mi> b </mi>
  </mrow>
</mfenced>

\left(a+b\right)
```

Jegyezzük meg, hogy **<mrow>** nélkül az  $(a,+,b)$  formát kaptuk volna.

$[0,1)$

```
<mfenced open="[">
  <mn> 0 </mn>
  <mn> 1 </mn>
</mfenced>

\left[0,1\right)
```

$f(x,y)$

```
<mrow>
  <mi> f </mi>
  <mo> &ApplyFunction; </mo>
  <mfenced>
    <mi> x </mi>
    <mi> y </mi>
  </mfenced>
</mrow>

f(x,y)
```

## 1.6. Indexek és ékezetek

Ebben a szakaszban az alapelem körüli indexelemeket mutatjuk be. A matematikai jelölésrendszer fontos részei az alsó- és felsőindexek. Ezeken kívül a MathML tartalmaz még néhány hasonló, különleges helyű leíróelemet. Ezekre legjobb példa az egyes kifejezések határoló értékeinek megadása az alapelem alatt és/vagy felett.

Természetesen ezekre az elemekre is hatással van a már bemutatott **<mstyle>** elem *scriptlevel* és *displaystyle* attribútuma, mely része a környezeti beállításoknak. Az itt következő elemeknél explicit módon nem lehet e tulajdonságokat beállítani, erre egy külön **<mstyle>** elem megadása ad módot.

### 1.6.1. Indexek

Az alapelemhez tartozó alsó indexet az **<msub>**, a felső indexet az **<msup>**, mindkettőt egyszerre az **<msubsup>** elemmel írhatjuk le.

Az **<msub>** saját attribútuma a *subscriptshift*, mely az alapvonaltól való minimális távolságot határozza meg lefelé. Bármilyen vertikális mértékegységben megadható (általában az *ex*-et használjuk). Ugyanígy az **<msup>** saját attribútuma a *subscriptshift*, mely felfelé határozza meg a távolságot. Az **<msubsup>** elemnél mindkét attribútum használható. Ez utóbbi elem és az **<msub>** - **<msup>** elemek együttes használatában az a számottevő különbség, hogy az **<msubsup>** elemnél a két index között vízszintes irányban kisebb a távolság.

Az **<mmultiscripts>** elemnek a segítségével összetettebb indexeket és elő-indexeket tudunk leírni. Az elem általános formája a következő:

```
<mmultiscripts>
  alap
  1. alsóindex
  1. felsőindex
  ...
  n. alsóindex
  n. felsőindex
  [ <mprescripts/> elő-alsóindex elő-felsőindex ]
</mmultiscripts>
```

Jegyezzük meg, hogy az első argumentum mindig az alapelem, ezt követik az alsó- és felsőindexek (ebben a sorrendben), melyek mindig párban állnak. Ha a páros valamelyik tagját nem akarjuk megjeleníteni, akkor az helyett a **<none/>** elemet kell használnunk. Az elő-indexeket is hasonlóan kell megadnunk, de jeleznünk kell a helyüket a **<mprescripts/>** üres elemmel, mely mindig az összes normál index felsorolása *után* található. Használata csak akkor kötelező, ha ténylegesen megadunk elő-indexeket.

Az **<mprescripts/>** csak a **<mmultiscripts>** elembe ágyazva használható.

Ha nem kívánunk elő-indexeket megjeleníteni, akkor az **<mmultiscripts>** elem páratlan argumentumot tartalmaz, ha igen, akkor párosat. A fent már említett két attribútumot kaphatja meg.

A fenti elemek az index argumentumra vonatkozólag eggyel növelik az **<mstyle>** elem *scriptlevel* attribútumát, a *displaystyle* attribútumot pedig „false”-ra állítják.

TEX-ben az alsó index megadási módja a *azonosító\_alsóindex*, a felső indexé a *azonosító^felsőindex*. Több index esetén, valamint az elő-indexek bevezetésére használjuk a **{ }** zárójeleket.

$z_2$

```
<msub>
  <mi> z </mi>
  <mn> 2 </mn>
</msub>
```

$z_2$

$z^2$

```
<msup>
  <mi> z </mi>
  <mn> 2 </mn>
</msup>
```

$z^2$

$\int_0^1 e^x dx$

```
<mrow>
  <msubsup>
    <mo> &int; </mo>
    <mn> 0 </mn>
    <mn> 1 </mn>
  </msubsup>
  <mrow>
    <msup>
      <mi> &ExponentialE; </mi>
      <mi> x </mi>
    </msup>
    <mo> &InvisibleTimes; </mo>
    <mrow>
      <mo> &DifferentialD; </mo>
      <mi> x </mi>
    </mrow>
  </mrow>
</mrow>

\int_0^1 e^x dx
```

$${}_0F_1(;a;z)$$

```

<mrow>
  <mmultiscripts>
    <mi> F </mi>
    <mn> 1 </mn>
    <none/>
    <mprescripts/>
    <mn> 0 </mn>
    <none/>
  </mmultiscripts>
  <mo> &ApplyFunction; </mo>
  <mrow>
    <mo> ( </mo>
      <mrow>
        <mo> ; </mo>
        <mi> a </mi>
        <mo> ; </mo>
        <mi> z </mi>
      </mrow>
    <mo> ) </mo>
  </mrow>
</mrow>

{}_0F_1(;a;z)

```

$$R_i^j{}_{kl} \text{ (ahol } k \text{ és } l \text{ különböző indexek)}$$

```

<mmultiscripts>
  <mi> R </mi>
  <mi> i </mi>
  <none/>
  <none/>
  <mi> j </mi>
  <mi> k </mi>
  <none/>
  <mi> l </mi>
  <none/>
</mmultiscripts>

R_{i}{}^{j}{}_{k}{}_{l}

```

### 1.6.2. Ékezetek a kifejezések körül

Az indexekhez hasonlóan az alá és föléírások megjelenítésére három elem létezik, az **<munder>**, az **<mover>**, és az **<munderover>**.

E három elemmel olyan kifejezések írhatók le, ahol az alapszámhoz valamilyen aláírt (**<munder>**), föléírt (**<mover>**), vagy alá- és föléírt (**<munderover>**) argumentum tartozik.

Az **<munder>** saját attribútuma az *accentunder*, az **<mover>** elemé pedig az *accent*. Az **<munderover>** elemhez mindkettő használható. Az attribútumok azt határozzák meg, hogy az argumentum hogyan viselkedjen. Ha értékük „true”, akkor közelebb kerül az alapértékhez, mint „false” esetében.

Ha az alap egy műveleti jel, és a *movablelimits* és a *displaystyle* beállítása rendre "true" és "false", akkor az *accentunder* értékét nem vesszük figyelembe. Tipikusan olyan szimbólumoknál alkalmazzuk, mint a  $\sum$ .

Mindhárom elem az argumentumra vonatkozólag eggyel növeli az **<mstyle>** elem *scriptlevel* attribútumát, ha az *accentunder* „false”, a *displaystyle* attribútumot pedig „false”-ra állítja.

TEX-ben több módja van a kifejezések alá és fölé történő írásának. A legegyszerűbb módja az **\underset** *mit \to mi alá*, vagy az **\overset** *mit \to mi fölé* parancspáros. Speciális módja a már fent említett kapcsos zárójelezés, melyre külön parancsok léteznek: **\underbrace** vagy **\overbrace** (csak a zárójelet jeleníti meg) és az **\undersetbrace** *mit \to mi alá* vagy az **\oversetbrace** *mit \to mi alá* parancspáros. Ha vízszintes vonalat kell húzni a kifejezés alá vagy fölé, arra az **\underline** vagy az **\overline** parancs szolgál. A MathML *accentunder* és *accent* argumentumnak nincs megfelelője. Ha csak egy karakter fölé akarunk valamilyen ékezetet tenni, akkor annak egyszerűbb módjai az ékezetparancsok.

MathML-ben többszintű alá- és föléírást a **<munder>** vagy **<mover>** egymásba ágyazásával hozhatunk létre. TEX-ben erre külön parancsok vannak, az aláírásra az **\Sb ... \endSb** parancspár; a föléírásra pedig az **\Sp ... \endSp** parancspár, ahol az egyes sorokat a **\** paranccsal választjuk el egymástól. A sorok távolságának megváltoztatására használhatjuk a **\vspace** parancsot, megadva utána az üres hely nagyságát.

Van egy módja a felül és alulírásnak, amelyeket általában az ún. nagyoperátoroknál (pl. integrál, szumma, stb.) használunk, melyet a **\limits** paranccsal adunk meg, mely után az indexeknek megfelelően határozzuk meg az értékeket.

A MathML **<munderover>** elemének a fenti parancsok együttes használata felel meg.

$$\underbrace{x+y+z}$$

```
<munder accentunder="true">
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mo> &UnderBrace; </mo>
</munder>

\underbrace {x+y+z}
```

$$\lim_{\substack{n \rightarrow +\infty \\ m \rightarrow -\infty}} A$$

```
<munder accentunder="true">
  <munder accentunder="true">
    <mrow>
      <mo> lim </mi>
      <mi> A </mi>
    </mrow>
    <mrow>
      <mi> n </mi>
      <mo> &rightarrow; </mo>
      <mn> +&infinity; </mn>
    </mrow>
  </munder>
  <mrow>
    <mi> m </mi>
    <mo> &rightarrow; </mo>
    <mn> -&infinity; </mn>
  </mrow>
</munder>

\lim\sb n\to +\infty \\\ m\to -\infty\endsb A
```

$$\hat{x}$$

```
<mover accent="false">
  <mi> x </mi>
  <mo> &hat; </mo>
</mover>

\^x
```

$$\overline{x+y+z}$$

```
<mover accent="true">
  <mrow>
    <mi> x </mi>
    <mo> + </mo>
    <mi> y </mi>
    <mo> + </mo>
    <mi> z </mi>
  </mrow>
  <mo> &OverBar; </mo>
</mover>

\overline {x+y+z}
```

$$\int_0^{\infty} x$$

```

<mrow>
  <munderover>
    <mo> &int; </mo>
    <mn> 0 </mn>
    <mi &infinity; </mi>
  </munderover>
  <mi> x </mi>
</mrow>

\int\limits_0^{\infty} x

```

## 1.7. Táblák és mátrixok

### 1.7.1. Táblák, mátrixok felépítése

Mátrixok, tömbök és más táblák leírására a **<mtable>**, **<mtr>** és **<mttd>** elemeket használjuk. Ezek az elemek nagyon hasonlítanak a HTML **<table>**, **<tr>** és **<td>** elemeire, azzal a különbséggel, hogy a MathML-ben számos speciális attribútumot használunk a matematikai formák (diagrammok, blokkmátrixok, stb.) megjelenítéséhez.

Az **<mtable>** elemmel tudunk táblákat vagy mátrixokat készíteni. Az **<mtable>** elembe használhatunk **<mtr>** és **<mttd>** elemeket, ezek használata azonban nem kötelező, ekkor egy egysoros-egyoszlopos táblát kapunk. Bányunk körültekintően az **<mtr>** és **<mttd>** elemekkel, mert nagymértékben ezektől függ táblánk megjelenése.

Az **<mtr>** elem segítségével jeleníthetjük meg a tábla ill. mátrix sorait. Csak az **<mtable>** elembe ágyazva használhatjuk. Egy **<mtr>** elem egy sort határoz meg. A benne elhelyezett argumentumok balról jobbra haladva írják le a tábla egyes celláit.

Ha egy **<mtr>** elembe kevesebb értéket adunk meg, mint ugyanannak a táblának a többi **<mtr>** eleménél, akkor a hiányzó hely üres **<mtr>** elemmel töltődik fel.

Az **<mttd>** elemmel jeleníthetjük meg egy tábla vagy mátrix egy bejegyzését. Csak az **<mtr>** elembe ágyazva használhatjuk.

Ha az **<td>** elemben egynél több argumentumot kívánunk megjeleníteni, akkor azt egy **<mrow>** elembe kell beágyazni.

A legtöbb esetben a táblákat valamilyen zárójellel használjuk, ezt azonban az **<mtable>** elem nem tartalmazza, ezért általában egy **<mrow>** vagy **<mfenced>** elem veszi körbe a táblát.

Az **<mtable>** elemnek a következő saját attribútumai lehetnek: *align*, *rowalign*, *columnalign*, *groupalignment*, *alignmentscope*, *rowspacing*, *columnspacing*, *rowlines*, *columnlines*, *frame*, *framespacing*, *equalrows*, *equalcolumns*, *displaystyle*. Az **<mtr>** elemnek három saját attribútuma van: a *rowalign*, a *columnalign* és a *groupalign*.

Az **<td>** a következő attribútumokat kaphatja meg: *rowspan*, *columnspan*, *rowalign*, *columnalign*, *groupalign*.

Az *align* attribútum azt írja le, hogy a tábla hová legyen rendezve a környezet figyelembe vételével. Értékei lehetnek: „*top*”, „*bottom*”, „*center*”, „*baseline*”, „*axis*”; alapértelmezése „*axis*”. Minden esetben a környezet alapvonalához igazít a paraméternek megfelelően. „*Top/bottom*” esetén a felső/alsó sor, „*center*” és „*baseline*” esetén a tábla közepe, „*axis*” esetén a tábla vízszintes tengelye lesz az alapvonalon. A fenti értékek után megadható egy sorszám is (1-től maximum sorszámig), mellyel azt jelezzük, hogy felülről számítva hányadik sorra lesz érvényes a megadott paraméter.

A *rowalign* és a *columnalign* attribútum azt írja le, hogy a sorok/oszlopok hogyan legyenek rendezve. Sorokra vonatkozóan a „*top*”, „*bottom*”, „*center*”, „*baseline*”, „*axis*” értékeket veheti fel, oszlopokra vonatkozóan pedig a „*left*”, „*center*”, „*right*” értékek valamelyikét. Természetesen több sor/oszlop esetén minden egyes sorra/oszlopra megadhatunk egy értéket egymástól szóközzel elválasztva. Például egy négyoszlopos tábla esetén a *columnalign* értéke a következő lehet: „*left right center center*”. Ha több értéket adunk meg, mint amennyi szükséges, a felesleges elemeket

figyelman kívül hagyja. Ha kevesebb értéket adunk meg, mint amennyire szükség lenne, az utolsó értéket használja a hiányzók helyett.

A *groupalign* és *alignmentscope* attribútumokkal az **<maligngroup/>** és **<malignmark/>** elemeknél ismerkedünk meg. Alapértelmezésük „{left}” és „true”.

A *rowspacing* és *columnspacing* attribútummal határozhatjuk meg a sorok/oszlopok közé kerülő üres helyeket (lásd még a *framespacing* attribútumot is!). Alapértelmezésük rendre „1.0ex” és „0.8em”.

A *rowlines* és *columnlines* attribútumok azt határozzák meg, hogy milyen üres helyek kerüljenek a sorok/oszlopok közé. Három értéket vehet fel: „none”, „solid”, „dashed”, alapértelmezés a „none”. „Solid” esetén szélesebb üres helyek, „dashed” esetén keskenyebbek kerülnek a sorok/oszlopok közé. A kezdő és záró sorok/oszlopok elé/mögé kerülő üres helyeket a *frame* attribútumban határozhatjuk meg.

A *frame* és *framespacing* attribútumokat akkor használjuk, ha valamilyen keretet akarunk a táblánk köré készíteni. Ha a *frame* értéke „none”, akkor a *framespacing* értékét nem vesszük figyelembe. Ha az érték nem „none”, akkor a *framespacing*-ben megadott értékű lesz a keret és az első sor/oszlop között az üres hely. Alapértelmezésben a *frame* értéke „none”, ezen kívül a „solid” és a „dashed” értékeket veheti fel. A *framespacing* megadásakor először az oszlopok közti távolságot kell megadni (alapértelmezés: „0.4 em”), majd a sorok közöttit (alapértelmezés: „0.5 ex”).

Az *equalrows* és *equalcolumns* attribútumok „true” beállítása esetén az összes sor/oszlop egyforma hosszúságú/szélességű lesz. Alapértelmezés mindkettőnél „true”.

A *displaystyle* attribútumról már az **<mstyle>** elemnél szót ejtettünk. Alapértelmezése „false”, ez azt jelenti, hogy a tábla

valamilyen kifejezés része, „true” esetén a cellákat külön-külön kifejezéseknek tekintjük. Beállításával a *scriptlevel* nem változik.

A *rowspan* és *columnspan* értékkel a sor/oszlop magasságát/szélességét határozhatjuk meg. Alapértelmezése mindkettőnek 1.

TEX esetén több parancs pár áll rendelkezésünkre mátrixok készítésére, de ezek alapján csak zárójelezésükben különböznek egymástól: **\matrix** - **\endmatrix** (zárójel nélkül), **\pmatrix** - **\endpmatrix** (kerek zárójellel), **\bmatrix** - **\endbmatrix** (szögletes zárójellel), **\vmatrix** - **\endvmatrix** (függőleges vonallal), **\Vmatrix** - **\endVmatrix** (dupla függőleges vonallal). A mátrix sorait **\** parancssal választjuk el.

Ha szöveg közé kívánjuk helyezni a mátrixot, akkor használjuk a **\smallmatrix** - **\endsmallmatrix** parancspárt, mely kisebb méretű megjelenítést tesz lehetővé.

Formázásra csak vízszintes irányban van lehetőség, mégpedig a **\format** parancs segítségével, melyet a nyitó mátrixparancs után kell használnunk, a következőképpen: minden egyes oszlopra vonatkozóan meg kell adnunk az illesztés irányát (**\l** – balra, **\r** – jobbra, **\c** – középre), és esetleg a kihagyandó hely méretét a már megismert módon (pl. **\quad**-dal).

Táblázatok előállítása TEX-ben egy kicsit másként működik. Legegyszerűbb formája, mikor olyan táblázatot kell létrehoznunk, ahol balra illesztve, keret nélkül állnak az oszlopok, ezt a **\settabs** parancssal végezhetjük el.

Ennek is két módja van, az első, ha egyforma szélességű oszlopaink vannak, akkor a parancs formátuma: **\settabs szélesség\columns \+...\cr...** Ekkor a **\columns** parancs után soroljuk fel a sorokat, **\+** parancssal kezdve, a tagokat **&** jellel elválasztva és **\cr** parancssal lezárva.

Másik módja változó szélességű oszlopok esetén használatos, ekkor a **\settabs** parancs után elsőnek, egy formátumsorban kell

megadni az oszlopok szélességét (célszerű felsorolni az oszlopok legszélesebb elemeit). Látszólag ez a formátumsor ugyanúgy néz ki mint a többi sor (`\+...&...\cr`), de ez nem fog látszani.

Ha nem balra igazított táblázatot, vagy vonalakkal elválasztott táblázatra van szükségünk, akkor használjuk a **\halign** parancsot. Felépítése hasonlít a **\settabs** második használatára, de itt nincsenek nyitó **\+** parancsok, és a felsorolást **{}** zárójelek közé kell tenni: `\halign {formátumsor\cr...\cr}`. Ennél a parancsnál nem soroljuk fel a formátumsorban az elemeket, hanem **#** jelekkel adjuk meg az oszlopok számát (az elválasztójel itt is a **&**, mint a mátrixoknál). Az egyes **#** jelek mellett megadhatjuk az illesztés irányát a **\hfil** (gyenge illesztő), **\hfill** (erős illesztő), **\hss** (kiemelő) parancsokkal: ha az oszlopot balra igazítjuk, akkor a **#** jel mögé, ha jobbra, akkor elé, ha középre, akkor mindkét oldalra el kell helyoznünk valamelyiket. Az egyes oszlopjelek előtt meghatározhatjuk azok betűtípusait, méreteit. Elhelyezhetünk a formátumsorban olyan szöveget, amelyek minden egyes sorban ismétlődik, meghatározhatjuk az oszlopelválasztó vonalainkat, valamint itt kell beállítanunk az oszlopok közötti távolságot is. Amennyiben azonos távolságot akarunk hagyni minden oszlop között, használjuk a **\tabskip** elemet: `\tabskip=méret\halign{...}`.

Sorokra vonatkozóan a következő parancsokat kell megjegyeznünk. Teljes táblázat szélességű vízszintes vonalat a **\noalign{\hrule}** kombinációval húzhatunk. Két sor között helyet hagyhatunk ki a **\noalign{\vskip méret}** parancskombinációval. Lehetőségünk van egy sorban a cellák egyesítésére is, ezt a **\multispan** paranccsal tehetjük meg, mely után meg kell adni az egyesítendő cellák számát, majd kapcsos zárójelek között a megjelenítendő szöveget és formázó parancsait, mivel ilyenkor nem érvényesek a formátumsorban megadott paraméterek (pl. `\multispan3{\hfill\bf Adatok\hfill}\cr`). Ha egy adott sorra nem akarjuk alkalmazni a formátumsorban megadott

beállításokat, akkor az adott sor elejére írjuk be az **\omit** parancsot, ami után megadhatjuk az esetleges lokális beállításainkat is, természetesen az egyes celláknak megfelelően. Ekkor erre a sorra az **\omit** után megadott beállítások lesznek érvényben. Ha olyan táblázatot készítünk, amelyben nincs minden sor között elválasztó vonal, akkor a használjuk az **\offinterlineskip** - **\par** parancspárt (`{\offinterlineskip\halign{...}\par}`), ilyenkor a vonalakkal nem elválasztott sorok közelebb kerülnek egymáshoz, szebb képet eredményezve.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```

<mrow>
  <mo> ( </mo>
  <mtable>
    <mtr>
      <mtr><mn>1</mn></mtr>
      <mtr><mn>0</mn></mtr>
      <mtr><mn>0</mn></mtr>
    </mtr>
    <mtr>
      <mtr><mn>0</mn></mtr>
      <mtr><mn>1</mn></mtr>
      <mtr><mn>0</mn></mtr>
    </mtr>
    <mtr>
      <mtr><mn>0</mn></mtr>
      <mtr><mn>0</mn></mtr>
      <mtr><mn>1</mn></mtr>
    </mtr>
  </mtable>
  <mo> ) </mo>
</mrow>

\pmatrix 1&0&0 \\ 0&1&0 \\ 0&0&1 \endpmatrix

```

### 1.7.2. Egyenletek illesztése

A jobb megjelenés érdekében egyenletrendszereket illeszthetünk egymáshoz az **<maligngroup/>** és az **<malignmark/>** elemek segítségével.

Lássuk a következő példát:

$$\begin{aligned} 8.44x + 55y &= 0 \\ 3.1x - 0.7y &= -1.1 \end{aligned}$$

A fenti egyenletrendszert horizontálisan formázva a következő alakot kapjuk:

$$\begin{array}{l} 8.44x + 55y = 0 \\ 3.1x - 0.7y = -1.1 \end{array}$$

Mint látjuk, itt egy egyszerű horizontális formázás történt, de ha nem ismertük volna a vertikális elrendezést, nem hajthattuk volna végre.

Jelen pillanatban az **<mtable>** elem az egyetlen olyan eszköz, mely lehetőséget ad a fenti sorba rendezésre. A MathML későbbi változataiba talán bekerül egy **<malignscope>** elem, mely alkalmasabb lehet az ilyen sorba rendezésekre, de egyelőre csak táblák segítségével tudjuk megoldani problémánkat. Fontos tudnunk, hogy az **<mtable>** elem *alignmentscope* attribútumának értékét „true”-ra kell állítani (ez az alapértelmezés), mert „false” esetén a megjelenítő nem teszi figyelembe a horizontális rendezést.

Hogy a fenti elvárásoknak eleget tudjunk tenni, szükség van ún. *sorbaállítási csoportok* létrehozására, melyeket majd horizontálisan illeszteni tudunk a kívánalmaknak megfelelően.

A sorbaállítási csoportoknak egy láthatatlan **<maligngroup/>** elemmel kell kezdődnie, mely a megfelelő horizontális üres helyet reprezentálja. A fenti egyenletrendszerben minden egyes tényező, azonosító, műveleti jel előtt áll egy **<maligngroup/>** elem, az egyenlőség mindkét oldalán.

Az **<maligngroup/>** elemhez csak a következő fajta elemek kapcsolódhatnak: **<mrow>**, **<mstyle>**, **<mphantom>**, **<mfenced>**, **<maction>**, **<semantics>** és természetesen a token-elemek. Ezeknek megfelelően kell tehát kialakítani az egyes kifejezések (jelen esetben az egyenletrendszer) egyes tagjainak leírását.

Ha vannak olyan soraink, amelyeket nem akarunk csoportosítani, ott egyszerűen ne használjuk a **<maligngroup/>** elemet!

A fent létrehozott csoportokon belül kialakíthatunk egy-egy olyan pontot, amelyhez fogunk majd igazítani. Ezt a pontot két módon jelölhetjük ki: az egyik mód az **<malignmark/>** elem segítségével, a másik az egyes csoportok, vagy a tábla *groupalign* attribútumával. Ha az **<malignmark/>** elemmel tesszük, akkor az *edge* attribútumban beállíthatjuk az igazítás irányát („left” vagy „right”, alapértelmezés a „left”).

A másik módszer szerint a *groupalign* beállításával adhatjuk meg az igazítási pontot. A *groupalign* a következő értékeket veheti fel: „left”, „center”, „right”, „decimalpoint” (ha jól megnézzük, ezek az értékek megfelelnek a Word szövegszerkesztő tabulátorainak igazításához). Nincs alapértelmezés, az érték öröklődik. Ezért például célszerű az egyes oszlopokhoz tartozó értékeket megadni az **<mtable>** elemnél, így az itt megadott értékek majd továbböröklődnek a csoportok számára.

Hogy érthetőbb legyen a fenti „zavaros” megfogalmazás, álljon itt a fenti „igazított” egyenletrendszer MathML kódja:

```
<mtable groupalign="{decimalpoint left left decimalpoint left
  left decimalpoint}">
  <mtd>
    <mrow>
      <mrow>
        <maligngroup/>
        <mn> 8.44 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> x </mi>
      </mrow>
      <maligngroup/>
      <mo> + </mo>
      <mrow>
        <maligngroup/>
        <mn> 55 </mn>
        <mo> &InvisibleTimes; </mo>
        <maligngroup/>
        <mi> y </mi>
      </mrow>
    </mrow>
    <maligngroup/>
    <mo> = </mo>
    <maligngroup/>
  </mtd>
</mtable>
```

```

        <mn> 0 </mn>
      </mtd>
    </mtr>
    <mtd>
      <mrow>
        <mrow>
          <maligngroup/>
          <mn> 3.1 </mn>
          <mo> &InvisibleTimes; </mo>
          <maligngroup/>
          <mi> x </mi>
        </mrow>
        <maligngroup/>
        <mo> - </mo>
        <mrow>
          <maligngroup/>
          <mn> 0.7 </mn>
          <mo> &InvisibleTimes; </mo>
          <maligngroup/>
          <mi> y </mi>
        </mrow>
      </mrow>
      <maligngroup/>
      <mo> = </mo>
      <maligngroup/>
      <mrow>
        <mo> - </mo>
        <mn> 1.1 </mn>
      </mrow>
    </mtd>
  </mtable>

```

A fenti egyenletillesztésre a T<sub>E</sub>X-ben is van parancs, de itt csak egyetlen ponton kapcsolhatjuk össze az egyenleteinket és csak kizárólagos matematikai módban használhatjuk (**\$\$**). Az erre szolgáló parancs az **\eqalign**, melynél kapcsos zárójelben kell megadni az egyenleteket, **\cr** paranccsal elválasztva őket egymástól. Az illesztés helyét a & jel segítségével adjuk meg, az illesztési hely elé kell beszúrni. A fenti egyenletrendszerekben látható üres helyeket a már megismert módszerekkel adhatjuk meg (pl. **\quad**). A második egyenletrendszer T<sub>E</sub>X-ben:

```

$$\eqalign{8.44x+55\quad y&=\quad 0\cr 3.1\quad x-0.7y&=-1.1}$$

```

## 1.8. Egyéb elemek

### 1.8.1. Hibakezelés

Az **<merror>** elem kiírja a tartalmát, mint egy hibaüzenetet, eltérő formában, mint a környezete (pl. piros színnel, villogó

tartalommal, más háttérszínnel, stb. – ez a megjelenítőtől függ). Tartalma bármilyen matematikai kifejezés lehet. Ha több elemet tartalmaz, akkor azokat egy **<mrow>** elembe kell zárni.

Nincsen saját attribútuma.

Főként olyan programok használják, amelyek MathML kódot készítenek, ezzel az elemmel jelezve a szintaktikai hibákat. Általában ilyenkor a programok legenerálják a hiba fajtáját, majd az eredeti kifejezést, olyan pontosan, ahogy csak lehet, hogy minél könnyebb legyen a hiba javítása. Az elem kihasználtsága, tehát, főként a programozón múlik.

T<sub>E</sub>X-ben nincs megfelelője.

Példa egy hibás bemenetre (nincs olyan MathML elem, hogy **<mfraction>**):

```
<mfraction>
  <mrow>
    <mn> 1 </mn>
    <mo> + </mo>
    <msqrt>
      <mn> 5 </mn>
    </msqrt>
  </mrow>
  <mn> 2 </mn>
</mfraction>
```

Egy program ekkor a következő kódot generálhatja:

```
<merror>
  <mtext>
    Ismeretlen elem: &lt;mfraction>; argumentumai:&nbsp;
  </mtext>
  <mrow>
    <mn> 1 </mn>
    <mo> + </mo>
    <msqrt>
      <mn> 5 </mn>
    </msqrt>
  </mrow>
  <mtext>
    &nbsp;és&nbsp;
  </mtext>
  <mn> 2 </mn>
</merror>
```

OpenMath-ban van egy *error* objektum, az **<OME>**, mely hasonlóan használható. Példa a nullával való osztásra:

```

<OMOBJ>
  <OME>
    <OMS cd="arith" name="DivisionByZero"/>
    <OMA>
      <OMS cd="arith" name="divide"/>
      <OMV name="x"/>
      <OMI>0</OMI>
    </OMA>
  </OME>
</OMOBJ>

```

### 1.8.2. Vezérlő szerkezetek

Az **<maction>** elemmel vezérlő szerkezeteket készíthetünk MathML kifejezésekből. Ha egy más alkalmazásból próbálunk kapcsolódni MathML kódú kifejezéseinkhez, például egy standard XML-nyelvű dokumentumból kapcsolódnánk hozzá (*<element-name ... href="...">*), akkor célszerű ezt az elemet használni.

Két attribútuma lehet: az *actiontype* és a *selection*, melyek közül az elsőt kötelező megadni.

Az *actiontype* attribútumban szerepel a kiválasztott esemény típusa. Részletesen később a példánál.

A *selection* attribútumban határozzuk meg a **<maction>** elem argumentumaiként felsorolt részkifejezések közül melyiket kívánjuk választani. Alapértelmezése 1, maximum értéke a **<maction>** elem alatt felsorolt részkifejezések száma lehet, mindig pozitív egész szám. Ha olyan részkifejezésre hivatkozunk, amelyik nem létezik, akkor hibát kapunk.

Lássunk néhány példát (jegyezzük meg, hogy a következő példákban maga a hivatkozás URL-je nem szerepel, tehát ebben a formában a példák nem használhatóak):

```

<maction actiontype="toggle" selection="pozitív egész szám">
  (első kifejezés)
  (második kifejezés)
  ...
</maction>

```

Ebben az esetben a megjelenítő megjeleníti azt az alkalmazást, amelyre a felhasználó kattintott, a *selection*-ben megadott számú kifejezéssel kezdve. Tipikus felhasználása az oktatásban, hosszú számológépes algebrai kimeneteknél illetve illusztrációknál van.

```

<maction actiontype="statusline">

```

```
(kifejezés)
(üzenet)
</maction>
```

Ebben az esetben a kifejezés megjelenítődik, és ha a felhasználó rákattint, vagy az egeret fölé viszi a böngésző státussorába kiíródik az üzenet.

```
<maction actiontype="tooltip">
  (kifejezés)
  (üzenet)
</maction>
```

Hasonló az előzőhöz, azzal a különbséggel, hogyha az egeret egy bizonyos ideig a kifejezés felett tartjuk, akkor megjelenik egy felbukkanó segítő tipp.

```
<maction actiontype="highlight" other="color='#ff0000'">
  kifejezés
</maction>

<maction actiontype="highlight" other="background='#ff0000'">
  kifejezés
</maction>
```

Ebben az esetben, ha az egeret az adott kifejezés fölé visszük, az kiemeltté válik az *other* attribútumban megadott színeket használva.

```
<maction actiontype="menu" selection="pozitív egész szám">
  (1. menüpont)
  (2. menüpont)
  ...
</maction>
```

Ez a típus egy felbukkanó menüt jelenít meg, lehetőséget biztosítva a kifejezések bemutatására. Több ilyen menü is egymásba ágyazható.

Ennek az elemnek nincsen T<sub>E</sub>X megfelelője.

## 2. A tartalmi jelölés

### 2.1. A tartalmi jelölés elemeiről általában

Ebben a fejezetben részletesen megismerkedünk a MathML tartalmi elemeivel.

Tartsuk szem előtt a következőket:

- A MathML tartalmi elemek „nem hajtják végre” magát a matematikai számítást vagy műveletet, a böngészőben nem fogjuk egy számítás eredményét látni, hiszen a tartalmi elemek csak dokumentálási célokra használhatók.
- A tartalmi elemek ugyan helyesen jelenítik meg a matematikai kifejezéseket, de elsősorban azok tartalmát hivatottak bemutatni. Ha nincs szükségünk a valós tartalomra, akkor inkább használjuk a már megismert megjelenítő elemeket.

Ha egy MathML kifejezést fel kívánunk használni valamilyen más alkalmazásban, akkor célszerű a tartalmi jelölést használni, hiszen ekkor a kiválasztott alkalmazás képes lehet magát a műveletet elvégezni is, így kímélve meg minket sok felesleges munkától.

A tartalmi elemek mindegyike megkaphatja a *class*, *style*, *id* és *other* attribútumokat. Az első három megegyezik a HTML hasonló attribútumaival és a stíluslapokkal (CSS) kapcsolatos. Az *other* olyan dolgok leírására alkalmas, melyre nincs külön specifikáció a MathML-ben (például a megjelenítés formájára vonatkozó megjegyzések, stb.). Használatakor vegyük figyelembe, hogy a legtöbb megjelenítő figyelmen kívül hagyja a tartalmát.

A legtöbb (de nem mind!) tartalmi elem megkaphatja a *definitionURL* attribútumot is, melyben egy olyan URL adható meg, ahol az adott elem (bővebb) szemantikai leírása található meg. Formájára nincs szabály, alapértelmezése sincs.

Ezeknek az attribútumoknak sem az OpenMath-ban, sem a T<sub>E</sub>X-ben nincs megfelelőjük, mert egyiknél sem használunk a MathML-nek megfelelő stíluslapokat. A T<sub>E</sub>X-nek van saját stíluslapja, melyet a **\documentstyle** parancs segítségével használhatunk (Lásd 42. oldal: III/1.2.1. A token-elemek attribútumai című fejezetet). Az OpenMath-nál nincsenek külön stíluslapok, a *Content Dictionary*ben (CD) adunk meg minden paramétert. Ezekben a

CD-kben osztályozhatjuk az elemek paramétereit. Az egyes CD-k tartalmát a *C Függelék* (187. oldal) tartalmazza.

## 2.2. Tokenelemek

### 2.2.1. Számok

A **<cn>** elemet használjuk numerikus állandók specifikálására. Alapértelmezésként bármilyen tízes számrendszerbeli valós érték lehet, leírásakor tízes számrendszerbeli számokat, tizedespontot, előjelet, esetleg előre definiált szimbolikus konstansokat ( $\pi$ ) alkalmazhatunk.

Attribútumai a következők lehetnek: *type*, *base*. A *type* segítségével határozhatjuk meg a leírni kívánt szám típusát, értéke „*real*”, „*integer*”, „*rational*”, „*complex-cartesian*”, „*complex-polar*”, „*constant*” lehet. Alapértelmezése a „*real*”. A *base* a szám alapszámrendszerét adja meg, 2 és 36 közötti értéket vehet fel, alapértelmezése „10”.

Racionális és komplex számok esetén az egyes számrészek elválasztására a **<sep/>** elem használandó.

Alapállapotban az így meghatározott számok úgy jelennek meg, mintha azokat az **<mn>** elemmel írtuk volna le.

TEX-ben ugyanazok érvényesek, amelyeket a megjelenítő elemeknél megismertünk.

OpenMath esetén több lehetőségünk van. Egész számok megadására használjuk az **<OMI>** elemet, melynél fontos, hogy a pozitív előjelet nem használjuk, valamint az adott számot megadhatjuk 16-os számrendszerben is, de ekkor „x”-szel kell kezdenünk. Nem egész számok esetén használjuk az **<OMF/>** elemet. Az elem *dec* vagy *hex* paraméterében adhatjuk meg a számot, a számrendszerétől függően. Természetesen előfordulhat, hogy egy, a MathML-ben számként leírt „objektum” az OpenMath-

ban máshogy írandó le (legtöbbször valamilyen szimbólumként, az **<OMS/>** elemmel). Ezen kívül lehetőségünk van egy **<OMATTR>** elemben megadni a számra vonatkozó típist, ilyenkor a `cd="nums"` attribútumot kell beállítani, és megadni a megfelelő számtípust, ami „*based\_integer*”, „*complex\_cartesian*”, „*complex\_polar*”, „*e*”, „*gamma*”, „*i*”, „*infinity*”, „*NaN*”, „*pi*” és „*rational*” lehet.

Lássunk néhány példát az egyes típusokra (első a MathML, második a T<sub>E</sub>X, harmadik az OpenMath forma):

12345.7

```
<cn type="real"> 12345.7 </cn>
```

12345.7

```
<OMF dec="12345.7"/>
```

12345

```
<cn type="integer"> 12345 </cn>
```

12345

```
<OMI>12345</OMI>
```

AB<sub>3</sub><sub>16</sub>

```
<cn type="integer" base=16> AB3 </cn>
```

AB<sub>3</sub><sub>16</sub>

```
<OMI>xAB3</OMI>
```

12342/2342342

```
<cn type="rational"> 12342 <sep/> 2342342 </cn>
```

12342/2342342

```
<OMF dec="5,269e-3"/>
```

12.3+5*i*

```
<cn type="complex-cartesian"> 12.3 <sep/> 5 </cn>
```

12.3+5*i*

```
<OMATTR>
  <OMATP>
    <OMS cd="nums" name="complex_cartesian"/>
  </OMATP>
  <OMF dec="12.3+5i"/>
</OMATTR>
```

Polar(2,3.1415)

```
<cn type="complex-polar"> 2 <sep/> 3.1415 </cn>
```

```
\text{Polar} (2,3.1415)

<OMATTR>
  <OMATP>
    <OMS cd="nums" name="complex_polar"/>
  </OMATP>
  <OMA>
    <OMI>2</OMI>
    <OMF dec="3.1415"/>
  </OMA>
</OMATTR>
```

$\pi$

```
<cn type="constant"> &pi; </cn>

\pi

<OMS cd="nums" name="pi"/>
```

### 2.2.2. Azonosítók

A **<ci>** elem szimbolikus nevek leírására használatosak. Ezek általában valamilyen matematikai objektumok azonosítói. Alapállapotban az így meghatározott azonosítók úgy jelennek meg, mintha azokat az **<mi>** elemmel írtuk volna le.

Egyetlen saját attribútuma van, a *type*, melynek segítségével határozhatjuk meg a leírni kívánt azonosító típusát, értéke „*real*”, „*integer*”, „*rational*”, „*complex-cartesian*”, „*complex-polar*”, „*constant*”, valamint „*interval*”, „*list*”, „*matrix*”, „*matrixrow*”, „*set*”, „*vector*”, „*apply*”, „*reln*”, „*lambda*”, „*fn*” lehet. Alapértelmezése a „”, azaz nincs meghatározva.

TEX-ben ugyanazok érvényesek, amelyeket a megjelenítő elemeknél megismertünk.

OpenMath-ban az **<OMV/>** elemet használjuk az azonosítók leírására. Egyéb attribútumok (pl. indexek) megadására az **<OMATTR>** és **<OMATP>** elem pár használatos, melyeket a változó előtt adunk meg. Az **<OMATTR>** elem jelzi az attribútum megadását, az **<OMATP>** elem pedig leírja magát, az attribútumot. Hosszabb azonosítóknál használhatjuk az **<OMSTR>**, valamint az **<OMB>** elemeket is. Az előbbi egy szöveget, az utóbbi egy bájt sorozatot (betűk, számok, valamint a „+”, „-” „/” „=” jelek) tartalmazhat.

Íme néhány példa:

$x$

```
<ci> x </ci>
```

$x$

```
<OMV name="x"/>
```

$\mathbf{v}$

```
<ci type="vector"> v </ci>
```

```
\bold v
```

```
<OMATTR>
  <OMATP>
    <OMS cd="ecc" name="type"/>
    <OMS cd="ecc" name="array"/>
  </OMATP>
  <OMV name="v"/>
</OMATTR>
```

$x_a$

```
<ci>
  <msub>
    <mi>x</mi>
    <mi>a</mi>
  </msub>
</ci>
```

$x_a$

```
<OMA>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="style"/>
      <OMSTR> subscripted <OMSTR>
    </OMATP>
    <OMV name="x"/>
  </OMATTR>
  <OMV name="a"/>
</OMA>
```

## 2.3. Tartalmi elemek csoportba rendezése

Az **<apply>** elem egy függvény vagy művelet argumentumaival való megjelenítésére szolgál. A MathML tartalmi elemek legnagyobb részét egy ilyen elemmel vezetjük be (funkciójában hasonlít a megjelenítő **<mrow>** elemre). Az **<apply>** elem első attribútuma mindig az a művelet, amelyet meg szeretnénk valósítani, ezután következnek a művelet argumentumai. Használata nagymértékben függ attól az elemtől, melyre vonatkozik. Nincs *definitionURL* attribútuma!

T<sub>E</sub>X-ben ennek az elemnek leginkább a  $\{\}$  felel meg, de az esetek nagy részében nincs szükség a használatára.

OpenMath-ban az **<OMA>** elem a megfelelője. Itt is az első argumentum mindig az a művelet (szimbólum), amelyet meg szeretnénk valósítani, ezután következnek a művelet argumentumai.

Példák:

3!

```
<apply>
  <factorial/>
  <cn>3</cn>
</apply>
```

3!

```
<OMA>
  <OMS cd="arith" name="factorial"/>
  <OMI>3</OMI>
</OMA>
```

3+4

```
<apply>
  <plus/>
  <cn>3</cn>
  <cn>4</cn>
</apply>
```

3+4

```
<OMA>
  <OMS cd="arith" name="plus"/>
  <OMI>3</OMI>
  <OMI>4</OMI>
</OMA>
```

$\sin x$

```
<apply>
  <sin/>
  <ci>x</ci>
</apply>
```

$\sin x$

```
<OMA>
  <OMS cd="trans" name="sin"/>
  <OMV name="x"/>
</OMA>
```

## 2.4. Deklarációk

A **<declare>** elemnek két fontos szerepe van: az egyik, hogy megváltoztassuk vagy megadjuk az alapértelmezését egy adott

kifejezésnek, a másik, hogy kapcsolatot létesítsünk egy név és egy objektum között. Tipikus használata számítógépes algebra esetén lehetséges.

A **<declare>**-nek egy vagy két argumentuma lehet, az első egy **<ci>**-ben leírt azonosító, a második (opcionálisan) egy objektum, mellyel inicializáljuk a változót:

```
<declare type="vector">
  <ci> V </ci>
  <vector>
    <cn> 1 </cn>
    <cn> 2 </cn>
    <cn> 3 </cn>
  </vector>
</declare>
```

Az objektum típusának és a deklarált elem típusának meg kell egyeznie. Például, ha az attribútumban „*fn*”-nek deklaráltuk a változót, akkor az objektumnak is valamilyen függvény-típusúnak kell lennie. Ha deklarációkor nem határozzuk meg típust, akkor az objektum típusát vesszük alapértelmezésnek.

Alapértelmezésben érvényessége „*local*”, azaz helyi, az adott MathML elemre vonatkozó. Ezt a *scope* attribútum „*global*”-ra állításával kiterjeszthetjük az egész dokumentumra (a nyitó **<math>** elemtől kezdődően).

További attribútumok: *type*, mely a deklaráció típusát határozza meg, bármilyen tartalmi azonosító lehet, alapértelmezése „*ci*”. A *nargs* attribútum a függvénydeklaráció argumentumainak számát határozza meg, alapértelmezése 1. Az *occurence* attribútumban adjuk meg a műveletekbeli szerepet, értéke „*prefix*”, „*infix*” vagy „*function-model*” lehet, ez utóbbi az alapértelmezése.

A **<bvar>** elemet használjuk az egyes műveletek alapváltozóinak leírásához. Tartalmazhat egy **<degree>** elemet is, mellyel a változó foka adható meg.

TEX-ben nincs szükség ilyen parancsra, helyette egyszerűen leírjuk az adott kifejezést.

OpenMath-ban ezeket az elemeket **<OMS/>** és **<OMV/>** elemek kombinációjával helyettesíthetjük.

$$f(x, y) = (F + G)(x, y)$$

```

<declare type="fn" nargs="2" scope="local">
  <ci> f </ci>
  <apply>
    <plus/>
    <ci> F </ci>
    <ci> G </ci>
  </apply>
</declare>

f\left(x,y\right)=\left(F+G\right)\left(x,y\right)

<OMA>
  <OMS cd = "relation" name="eq"/>
  <OMA>
    <OMV name="f"/>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>
        <OMS cd="presentation" name="right"/>
        <OMSTR>)</OMSTR>
      </OMATP>
      <OMA>
        <OMS cd = "linalg" name="vector"/>
        <OMV name="x"/>
        <OMV name="y"/>
      </OMA>
    </OMATTR>
  </OMA>
  <OMA>
    <OMS cd = "arith" name="times"/>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>
        <OMS cd="presentation" name="right"/>
        <OMSTR>)</OMSTR>
      </OMATP>
      <OMA>
        <OMS cd = "arith" name="plus"/>
        <OMV name="F"/>
        <OMV name="G"/>
      </OMA>
    </OMATTR>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>
        <OMS cd="presentation" name="right"/>
        <OMSTR>)</OMSTR>
      </OMATP>
      <OMA>
        <OMS cd = "linalg" name="vector"/>
        <OMV name="x"/>
        <OMV name="y"/>
      </OMA>
    </OMATTR>
  </OMA>
</OMA>

```

$$J(x) = \ln x$$

```

<declare type="fn">
  <ci> J </ci>
  <lambda>
    <bvar>
      <ci> x </ci>
    </bvar>
    <apply>
      <ln/>
      <ci> x </ci>
    </apply>
  </lambda>
</declare>

J\left(x\right)=\ln x

<OMA>
  <OMS cd = "relation" name="eq"/>
  <OMA>
    <OMS cd = "arith" name="times"/>
    <OMV name="J"/>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>
        <OMS cd="presentation" name="right"/>
        <OMSTR>)</OMSTR>
      </OMATP>
      <OMV name="x"/>
    </OMATTR>
  </OMA>
  <OMA>
    <OMS cd="transc" name="ln"/>
    <OMV name="x"/>
  </OMA>
</OMA>

```

$$\int_0^a f(x)dx$$

```

<apply>
  <int/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <lowlimit>
    <cn> 0 </cn>
  </lowlimit>
  <uplimit>
    <ci> a </ci>
  </uplimit>
  <apply>
    <fn>
      <ci> f </ci>
    </fn>
    <ci> x </ci>
  </apply>
</apply>

\int\limits_0^a f(x)dx

<OMA>
  <OMS cd="calculus" name="defint"/>
  <OMA>
    <OMS cd="interval" name="interval"/>
    <OMI>0</OMI>
    <OMV name="a"/>
  </OMA>
</OMA>

```

```

<OMS cd = "arith" name="times"/>
<OMA>
  <OMS cd = "arith" name="times"/>
  <OMA>
    <OMV name="f"/>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>
          <OMS cd="presentation" name="right"/>
          <OMSTR>)</OMSTR>
        </OMATP>
        <OMV name="x"/>
      </OMATTR>
    </OMA>
    <OMV name="d"/>
  </OMA>
  <OMV name="x"/>
</OMA>
</OMA>

```

## 2.5. Aritmetika, algebra, logika

### 2.5.1. Alapműveletek

Az összeadás, kivonás, szorzás, osztás leírására MathML-ben rendre a **<plus/>**, a **<minus/>**, a **<divide/>** és a **<power/>** elemeket használjuk.

A **<plus/>** elemnek több argumentuma lehet, a többinek kizárólag kettő. Bár a **<minus/>** esetén egy argumentummal lehetőségünk van leírni a negatív számokat, de ezt kerüljük el, helyette a **<cn>** elemmel jelöljük a negatív számokat.

TEX-ben az összeadás, a kivonás jelölésére a „+” és „-” jelet használjuk. Szorzás jelölésére – ha szükség van rá – a **\cdot** parancsot használjuk, ekkor látható lesz a szorzásjel. Az osztás leírására az **<mfrac>** elemnél (65. oldal) ismertetett parancsokat használjuk.

OpenMath-ban az alapműveleteket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „*minus*”, „*plus*”, „*times*”, „*divide*” paraméterekre állításával) helyettesítjük. A szükséges CD minden esetben az „*arith*”.

$x + y + z$

<apply>

```
<plus/>
<ci> x </ci>
<ci> y </ci>
<ci> z </ci>
</apply>
```

$x+y+z$

```
<OMA>
  <OMS cd="arith" name="plus"/>
  <OMV name="x"/>
  <OMV name="y"/>
  <OMV name="z"/>
</OMA>
```

$x-y$

```
<apply>
  <minus/>
  <ci> x </ci>
  <ci> y </ci>
</apply>
```

$x-y$

```
<OMA>
  <OMS cd="arith" name="minus"/>
  <OMV name="x"/>
  <OMV name="y"/>
</OMA>
```

$ab$

```
<apply>
  <times/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
```

$ab$

$a\cdot b$

```
<OMA>
  <OMS cd="arith" name="times"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a/b$

```
<apply>
  <divide/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
```

$\frac{a}{b}$

```
<OMA>
  <OMS cd="arith" name="divide"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

### 2.5.2. Fontosabb aritmetikai, algebrai műveletek

A hatványozás leírására a **<power/>** elemet használjuk. Argumentumaiban először az alapot, aztán a kitevőt kell megadni. A **<root/>** elemet használjuk a gyökvonás leírásához. Egy vagy két argumentuma lehet, ha kettő van, akkor az első mindig egy **<degree>** elem, mely a gyökkitevőt adja meg. Ha nem adjuk meg a **<degree>** elemet, akkor a gyökkitevő alapértelmezése 2 lesz.

Az osztás maradékának leírásához a **<rem/>** elemet használjuk, az egész rész meghatározására pedig a **<quotient/>** elem szolgál. Mindkettőnek két argumentuma van.

Az abszolút érték leírására az **<abs/>** elemet használjuk.

A **<gcd/>** elem a legnagyobb közös osztó leírására szolgál, több argumentuma lehet. Faktoriálisok leírására a **<factorial/>** elemet használjuk, egyetlen argumentuma lehet.

Az **<interval>** elem egy valós intervallumot ír le. Egyetlen saját attribútuma van, a *closure*, mely az intervallum zártságát írja le és a következő értékeket veheti fel: „open”, „closed”, „open-closed”, „closed-open”. Alapértelmezés a „closed”. Nincs *definitionURL* attribútuma!

A **<conjugate/>** egy argumentumú elemet használjuk a konjugált komplex gyök leírásához.

TEX-ben a hatványozást egyszerűen a „^” jellel írjuk le. A gyökvonásra viszont két parancsunk van. Ha négyzetgyököt kell írunk használjuk az **\sqrt** parancsot, mely után kapcsos zárójelekben kell megadni a kifejezést. Ha nem négyzetgyökre van szükségünk, akkor használjuk a **\root** - **\of** parancspárt, ahol **\root** után kell megadni a kitevőt, az **\of** után pedig a kifejezést.

A maradék jelölésére a **\bmod** parancsot, az egész rész leírására pedig a „[”, „]” zárójelpárt használjuk, Az abszolút értéket pedig egyszerűen a „|” jellel jelöljük.

A konjugált komplex gyök leírásához használjuk az **\overline** paracsot, mely után kapcsos zárójelek között kell megadni a kifejezést.

A többi elemnek nincsen megfelelője, ezeket a kinézetüknek megfelelően kell leírni.

OpenMath-ban a **<power/>**, a **<root/>**, a **<rem/>**, a **<quotient/>**, **<abs/>**, a **<factorial/>** és a **<conjugate/>**, **<gcd/>** elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „power”, „root”, „rem”, „quotient”, „abs”, „factorial”, „conjugate”, „gcd” beállításával) helyettesítjük. A „root” attribútum esetén az **<OMS/>** elem első argumentuma a kitevő. A szükséges CD „factorial”, „gcd”, „quotient”, „rem” esetén „integer”, egyébként „arith”.

Az **<interval/>** elemnek az „interval” CD-ben szereplő „interval\_cc”, „interval\_co”, „interval\_oc”, „interval\_oo” attribútumok felelnek meg („c” – zárt, „o” – nyílt).

Példák:

$x^3$

```
<apply>
  <power/>
    <ci> x </ci>
    <cn> 3 </cn>
</apply>

x^3

<OMA>
  <OMS cd="arith" name="power"/>
  <OMV name="x"/>
  <OMI> 3 </OMI>
</OMA>
```

$\sqrt[n]{a}$

```
<apply>
  <root/>
    <degree>
      <ci> n </ci>
    </degree>
    <ci> a </ci>
</apply>

\root n\of a

<OMA>
  <OMS cd="arith" name="root"/>
```

```

    <OMV name="n"/>
    <OMV name="a"/>
  </OMA>

```

$a \bmod b$

```

  <apply>
    <rem/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>

a\mod b

<OMA>
  <OMS cd="integer" name="rem"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>

```

$[a/b]$

```

  <apply>
    <quotient/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>

\left[ a/b \right]

<OMA>
  <OMS cd="integer" name="quotient"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>

```

$|x|$

```

  <apply>
    <abs/>
    <ci> x </ci>
  </apply>

|x|

<OMA>
  <OMS cd="arith" name="abs"/>
  <OMV name="x"/>
</OMA>

```

$(a;b;c)$

```

  <apply>
    <gcd/>
    <ci> a </ci>
    <ci> b </ci>
    <ci> c </ci>
  </apply>

\gcd\left(a;b;c\right)

<OMA>
  <OMS cd="integer" name="gcd">
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="left"/>
        <OMSTR>(</OMSTR>

```

```

        <OMS cd="presentation" name="right"/>
        <OMSTR>)</OMSTR>
    </OMATP>
    <OMA>
        <OMS cd="linalg" name="vector"/>
        <OMV name="a"/>
        <OMV name="b"/>
        <OMV name="c"/>
    </OMA>
</OMATTR>
</OMA>

```

$n!$

```

<apply>
  <factorial/>
  <ci> n </ci>
</apply>

```

$n!$

```

<OMA>
  <OMS cd="integer" name="factorial"/>
  <OMV name="n"/>
</OMA>

```

$(a,b]$

```

<interval closure="open-closed">
  <ci> a </ci>
  <ci> b </ci>
</interval>

```

$\left[ a,b \right]$

```

<OMA>
  <OMS cd="interval" name="interval_oc"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>

```

$\overline{x+iy}$

```

<apply>
  <conjugate/>
  <apply>
    <plus/>
    <ci> x </ci>
    <apply>
      <times/>
      <cn> &ImaginaryI; </cn>
      <ci> y </ci>
    </apply>
  </apply>
</apply>

```

$\overline{x+iy}$

```

<OMA>
  <OMS cd="arith" name="conjugate"/>
  <OMA>
    <OMS cd="arith" name="plus"/>
    <OMV name="x"/>
    <OMA>
      <OMS cd="arith" name="times"/>
      <OMV name="i"/>
      <OMV name="y"/>
    </OMA>
  </OMA>

```

</OMA>  
</OMA>

### 2.5.3. Fontosabb aritmetikai, algebrai függvények

A **<max/>** és a **<min/>** elemeket használjuk a maximum és minimum függvények leírásához, több argumentuma lehet.

A logaritmusfüggvények leírására alaptól függően az **<ln/>** és a **<log/>** elemeket használhatjuk. Az **<ln/>** elemet használjuk a természetes alapú, a **<log/>** elemet pedig az  $n$  alapú logaritmus-függvény leírásához. Az előbbinek egy, az utóbbinak egy vagy két argumentuma lehet, ha kettő, akkor az elsőnek egy **<logbase>** elemnek kell lennie, mely megadja a logaritmus alapját. Ha ez nincs megadva, akkor alapértelmezése a 10 lesz.

Az **<exp/>** elem az exponenciális függvénynek (az  $\ln$  függvény inverze) felel meg ( $\exp(1)=2.718281828$ ), egyetlen argumentuma van.

TEX-ben a minimum- és maximum-függvényekre a **\max** és a **\min** parancsokat használjuk.

A logaritmus-függvényekre három parancs áll rendelkezésünkre, az **\ln** (a természetes alapúra), az **\lg** (a tízes alapúra) és a **\log** (az  $n$  alapúra), mely utóbbi után alsóindexben ( $\log_n$ ) megadhatjuk az alapot. Az exponenciális-függvénynek pedig az **\exp** parancs felel meg.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „max”, „min”, „ln”, „log”, „exp” beállításával) helyettesítjük. A szükséges CD a „max” és „min” esetében a „minmax”, a többinél a „transc”.

```
max{a,b}
  <apply>
    <max/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>
  \max\left\{a,b\right\}
<OMA>
```

```

    <OMS cd="minmax" name="max"/>
    <OMV name="a"/>
    <OMV name="b"/>
  </OMA>

```

$\min_x \{x^2 \mid x \notin B\}$

```

  <apply>
    <min/>
    <bvar>
      <ci>x</ci>
    </bvar>
    <condition>
      <reln>
        <notin/>
        <ci> x </ci>
        <ci type="set"> B </ci>
      </reln>
    </condition>
    <apply>
      <power/>
      <ci> x </ci>
      <cn> 2 </cn>
    </apply>
  </apply>

  \min_x\left\{x^2\mid x\notin B

  <OMA>
    <OMATTR>
      <OMATP>
        <OMS cd="presentation" name="style"/>
        <OMSTR> subscripted </OMSTR>
      </OMATP>
    </OMATTR>
    <OMA>
      <OMS cd="minmax" name="min"/>
    </OMA>
    <OMATTR>
      <OMATP>
        <OMS cd="set" name="notin"/>
        <OMV name="x"/>
        <OMV name="B"/>
      </OMATP>
    </OMATTR>
    <OMA>
      <OMS cd="arith" name="power"/>
      <OMV name="x"/>
      <OMI> 2</OMI>
    </OMA>
    </OMATTR>
    <OMV name="x"/>
  </OMA>
  </OMA>
  <OMATTR>
    <OMV name="x"/>
  </OMA>

```

$\ln a$

```

  <apply>
    <ln/>
    <ci> a </ci>
  </apply>

  \ln a

  <OMA>
    <OMS cd="transc" name="ln"/>
    <OMV name="a"/>
  </OMA>

```

$\log_3 x$

```
<apply>
  <log/>
  <logbase>
    <cn> 3 </cn>
  </logbase>
  <ci> x </ci>
</apply>

\log_3 x

<OMA>
  <OMS cd="transc" name="log"/>
  <OMI> 3 </OMI>
  <OMV name="x"/>
</OMA>
```

$e^x$

```
<apply>
  <exp/>
  <ci> x </ci>
</apply>

\exp x

<OMA>
  <OMS cd="transc" name="exp"/>
  <OMV name="x"/>
</OMA>
```

#### 2.5.4. Logikai műveletek

A logikai „és”, „vagy”, „kizáró vagy” és „nem” kifejezések MathML megfelelői rendre az **<and/>**, az **<or/>**, az **<xor/>** és a **<not/>** elemek. A **<not/>** elemnek csak egy, a többinek több argumentuma lehet.

Az implikáció leírásához az **<implies/>** elemet használjuk, melynek két argumentuma van.

A „minden ...-ra/re” kifejezés leírásához a **<forall/>** elemet, a „létezik olyan ..., hogy” kifejezés leírásához pedig az **<exists/>** elemet használjuk. Egy **<apply>** vagy **<reln>** segítségével mindkettő összekapcsolható egy **<condition>** elemmel is.

TEX-ben a fenti logikai műveletek megfelelői az **<xor/>** kivételével rendre a következő parancsok: **\land**, **\lor**, **\neg**, **\Rightarrow**, **\forall**, **\exists**. Az **<xor/>** megfelelője az **\=\lor** összetett parancs.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „and”, „or”, „xor”, „not”, „implies”, „forall”, „exists” beállításával) helyettesítjük. A szükséges CD a „forall” és „exist” esetében „quant”, egyébként „logic”.

$a \wedge b$

```
<apply>
  <and/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
a\land b
<OMA>
  <OMS cd = "logic" name="and"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a \vee b$

```
<apply>
  <or/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
a\lor b
<OMA>
  <OMS cd = "logic" name="or"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a \nabla b$

```
<apply>
  <xor/>
  <ci> a </ci>
  <ci> b </ci>
</apply>
a\=\lor b
<OMA>
  <OMS cd = "logic" name="xor"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$\neg a$

```
<apply>
  <not/>
  <ci> a </ci>
</apply>
\neg a
```

```
<OMA>
  <OMS cd = "logic" name="not"/>
  <OMV name="a"/>
</OMA>
```

$A \Rightarrow B$

```
<reln>
  <implies/>
  <ci> A </ci>
  <ci> B </ci>
</reln>
```

$A \rightarrow B$

```
<OMA>
  <OMS cd = "logic" name="implies"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$\forall n | n \in \mathbb{Z} \wedge n > 0 : \exists x, y, z | x, y, z \in \mathbb{Z} \Rightarrow x^n + y^n = z^n$

```
<apply>
  <forall/>
  <bvar>
    <ci> n </ci>
  </bvar>
  <condition>
    <apply>
      <and/>
      <reln>
        <in/>
        <ci> n </ci>
        <ci type="set"> Z </ci>
      </reln>
      <reln>
        <gt/>
        <ci> n </ci>
        <cn> 0 </cn>
      </reln>
    </apply>
  </condition>
  <apply>
    <exists/>
    <bvar>
      <ci> x </ci>
    </bvar>
    <bvar>
      <ci> y </ci>
    </bvar>
    <bvar>
      <ci> z </ci>
    </bvar>
    <condition>
      <apply>
        <reln>
          <in/>
          <ci> x </ci>
          <ci type="set"> Z </ci>
        </reln>
        <reln>
          <in/>
          <ci> y </ci>
          <ci type="set"> Z </ci>
        </reln>
        <reln>
          <in/>
          <ci> z </ci>
        </reln>
      </condition>
    </apply>
  </condition>
  <reln>
    <in/>
    <ci> z </ci>
  </reln>
</condition>
</apply>
```

```

        <ci type="set"> Z </ci>
      </reln>
    </apply>
  </condition>
</reln>
<eq/>
<apply>
  <plus/>
  <apply>
    <power/>
    <ci> x </ci>
    <ci> n </ci>
  </apply>
  <apply>
    <power/>
    <ci> y </ci>
    <ci> n </ci>
  </apply>
</apply>
<apply>
  <power/>
  <ci> z </ci>
  <ci> n </ci>
</apply>
</reln>
</apply>
</apply>

\forall n\mid n\in Z\land n>0:\exists x,y,z\mid x,y,z\in Z
\Rightarrow x^n+y^n=z^n

<OMA>
  <OMATTR>
    <OMATP>
      <OMS cd="logic" name="and"/>
      <OMA>
        <OMS cd="set" name="in"/>
        <OMV name="n"/>
        <OMV name="Z"/>
      </OMA>
      <OMA>
        <OMS cd="relation" name="gt"/>
        <OMV name="n"/>
        <OMI> 0 </OMI>
      </OMA>
    </OMATP>
  </OMATTR>
  <OMA>
    <OMS cd="quant" name="forall"/>
    <OMV name="n"/>
  </OMA>
</OMATTR>
<OMV name="n"/>
<OMATTR>
  <OMATP>
    <OMS cd="logic" name="impies"/>
    <OMA>
      <OMS cd="set" name="in"/>
      <OMA>
        <OMS cd="linalg" name="vector"/>
        <OMV name="x"/>
        <OMV name="y"/>
        <OMV name="z"/>
      </OMA>
      <OMV name="Z"/>
    </OMA>
  </OMA>
  <OMA>
    <OMS cd="relation" name="eq"/>
    <OMA>
      <OMS cd="arith" name="plus"/>
      <OMA>

```

```

        <OMS cd="arith" name="power"/>
        <OMV name="x"/>
        <OMV name="n"/>
    </OMA>
    <OMA>
        <OMS cd="arith" name="power"/>
        <OMV name="y"/>
        <OMV name="n"/>
    </OMA>
</OMA>
<OMA>
    <OMS cd="arith" name="power"/>
    <OMV name="z"/>
    <OMV name="n"/>
</OMA>
</OMATP>
<OMA>
    <OMS cd="quant" name="exists"/>
    <OMA>
        <OMS cd="linalg" name="vector"/>
        <OMV name="x"/>
        <OMV name="y"/>
        <OMV name="z"/>
    </OMA>
</OMA>
</OMATTR>
<OMS cd="linalg" name="vector"/>
<OMV name="x"/>
<OMV name="y"/>
<OMV name="z"/>
</OMA>

```

### 2.5.5. Relációk

Egy reláció leírására a **<reln>** elemet használjuk. Alkalmazása hasonlít az **<apply>**-ra. Az elem első utóda mindig az a reláció, amelyet meg szeretnénk valósítani, ezután következnek a reláció argumentumai. Az esetek egy részében a relációk meghatározásakor valamilyen feltételt is szabunk. Ezt a MathML-ben a **<condition>** elem segítségével tehetjük. Használhatjuk olyan speciális halmazok jelölésére is, ahol a halmaz elemei megszámlálhatatlanok. Egyiknek sincs *definitionURL* attribútuma!

Magukat a relációkat a következő elemek írják le: **<eq/>** - „egyenlő”; **<neq/>** - „nem egyenlő”; **<gt/>** - „nagyobb, mint”; **<lt/>** - „kisebb, mint”; **<geq/>** - „nagyobb vagy egyenlő”; **<leq/>** - „kisebb vagy egyenlő”.

TEX-ben magát a relációt és a feltételt írjuk le. A fenti relációknak rendre a következő jelek, illetve parancsok felelnek meg: „=”, **\ne**, „>”, „<”, **\geq**, **\leq**.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „eq”, „neq”, „gt”, „lt”, „geq”, „leq” beállításával) helyettesítjük. A szükséges CD minden esetben a „relation”. A MathML **<condition>** elem helyettesítése egy **<OMATTR>** szerkezettel lehetséges.

$a = b$

```
<reln>
  <eq/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

a=b

```
<OMA>
  <OMS cd="relation" name="eq"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a \neq b$

```
<reln>
  <neq/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

a\ne b

```
<OMA>
  <OMS cd="relation" name="neq"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a > b$

```
<reln>
  <gt/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

a>b

```
<OMA>
  <OMS cd="relation" name="gt"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a < b$

```
<reln>
  <lt/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

a<b

```
<OMA>
  <OMS cd="relation" name="lt"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a \geq b$

```
<reln>
  <geq/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

$a \backslash \geq b$

```
<OMA>
  <OMS cd="relation" name="geq"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$a \leq b$

```
<reln>
  <leq/>
  <ci> a </ci>
  <ci> b </ci>
</reln>
```

$a \backslash \leq b$

```
<OMA>
  <OMS cd="relation" name="leq"/>
  <OMV name="a"/>
  <OMV name="b"/>
</OMA>
```

$x \in R$

```
<condition>
  <reln>
    <in/>
    <ci> x </ci>
    <ci type="set"> R </ci>
  </reln>
</condition>
```

$x \backslash \text{in } R$

```
<OMA>
  <OMS cd="set" name="in"/>
  <OMV name="x"/>
  <OMV name="R"/>
</OMA>
```

$\max_x \{x - \sin x \mid 0 < x < 1\}$

```
<apply>
  <max/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <condition>
    <apply>
      <and/>
      <reln>
```

```

        <gt/>
        <ci> x </ci>
        <cn> 0 </cn>
    </reln>
    <reln>
        <lt/>
        <ci> x </ci>
        <cn> 1 </cn>
    </reln>
</apply>
</condition>
<apply>
    <minus/>
    <ci> x </ci>
    <apply>
        <sin/>
        <ci> x </ci>
    </apply>
</apply>
</apply>
\max\left\{x-\sin x\mid 0<x<1\right\}

<OMA>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="style"/>
      <OMSTR> subscripted </OMSTR>
    </OMATP>
    <OMA>
      <OMS cd="minmax" name="min"/>
      <OMA>
        <OMATTR>
          <OMATP>
            <OMS cd="relation" name="lt"/>
            <OMA>
              <OMS cd="relation" name="lt"/>
              <OMI> 0 </OMI>
              <OMV name="x"/>
            </OMA>
            <OMI> 1 </OMI>
          </OMATP>
          <OMA>
            <OMS cd="arith" name="minus"/>
            <OMV name="x"/>
          </OMA>
          <OMA>
            <OMS cd="transc" name="sin"/>
            <OMV name="x"/>
          </OMA>
        </OMATTR>
        <OMV name="x"/>
      </OMA>
    </OMATTR>
    <OMV name="x"/>
  </OMA>
</OMA>

```

## 2.6. Függvénytan

### 2.6.1. Felhasználói függvények

Egy felhasználó által készített függvény leírását az **<fn>** elem segítségével végezhetjük el. Ezen kívül lehetőségünk van egy speciális azonosító-függvény létrehozására is, az **<ident/>** elemmel.

TEX-ben ezeknek az elemeknek nincs megfelelője, csak magát a függvényeket írjuk le.

OpenMath-ban egy **<OMA>** és egy **<OMS/>** elem kapcsolatával fejezhetjük ki, ahol az **<OMS/>** elemben írjuk le magát a felhasználói függvényt. Több, egymásba ágyazott függvény esetén az **<OMA>** elem helyett használhatjuk az **<OMBIND>**, **<OMBVAR>** elempárt is, mely magát az egymásba ágyazást jelöli. Az azonosítót az „fns” CD-beli „identity” attribútummal képezhetjük.

$L$

```
<fn>
  <ci> L </ci>
</fn>
```

$L$

```
<OMA>
  <OMS cd="ecc" name="type"/>
  <OMS cd="ecc" name="function"/>
  <OMV name="L"/>
</OMA>
```

$J_0$

```
<fn definitionURL="http://www.defs.org/MyDefinition.txt">
  <ci>
    <msub>
      <mi>J</mi>
      <mn>0</mn>
    </msub>
  </ci>
</fn>
```

$J_0$

```
<OMA>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="style"/>
```

```

        <OMSTR> subscripted </OMSTR>
      </OMATP>
    <OMA>
      <OMS cd="ecc" name="type"/>
      <OMS cd="ecc" name="function"/>
      <OMV name="j"/>
    </OMA>
  </OMATTR>
  <OMI> 0 </OMI>
</OMA>

```

$f + g$

```

<fn>
  <apply>
    <plus/>
    <ci> f </ci>
    <ci> g </ci>
  </apply>
</fn>

```

$f+g$

```

<OMA>
  <OMS cd="arith" name="plus"/>
  <OMA>
    <OMS cd="ecc" name="type"/>
    <OMS cd="ecc" name="function"/>
    <OMV name="f"/>
  </OMA>
  <OMA>
    <OMS cd="ecc" name="type"/>
    <OMS cd="ecc" name="function"/>
    <OMV name="g"/>
  </OMA>
</OMA>

```

$f \circ f^{-1} = id$

```

<reIn>
  <eq/>
  <apply>
    <compose/>
    <fn>
      <ci> f </ci>
    </fn>
    <apply>
      <inverse/>
      <fn>
        <ci> f </ci>
      </fn>
    </apply>
  </apply>
  <ident/>
</reIn>

```

$f \circ f^{-1} = id$

```

<OMA>
  <OMS cd="relation" name="eq"/>
  <OMA>
    <OMS cd="fns" name="left_compose"/>
    <OMA>
      <OMS cd="ecc" name="type"/>
      <OMS cd="ecc" name="function"/>
      <OMV name="f"/>
    </OMA>
  </OMA>

```

```

<OMS cd="fns" name="inverse"/>
<OMA>
  <OMS cd="ecc" name="type"/>
  <OMS cd="ecc" name="function"/>
  <OMV name="f"/>
</OMA>
</OMA>
</OMA>
<OMA>
  <OMS cd="fns" name="identity"/>
  <OMV name="id"/>
</OMA>
</OMA>

```

### 2.6.2. A *lambda* függvény

A **<lambda>** elemet egy olyan felhasználó által meghatározott függvény esetében használjuk, amelyet egy adott kifejezésből és egy vagy több független változóból készítünk. Utódaiban először a változókat kell megadnunk, majd utoljára az aktuális paraméterértékeket.

Ha egy **<declare>** elem argumentumaként használjuk, akkor meghatározhatjuk a függvény nevét is.

T<sub>E</sub>X-ben nincs szükség ilyen parancsra, helyette egyszerűen leírjuk az adott kifejezést (a függvényhez használhatjuk a **\lambda** parancsot).

OpenMath-ban ezt az elemet az **<OMBIND>** - **<OMBVAR/>** elempárossal írhatjuk le. Az **<OMBVAR/>** elem határozza meg a függvény változóját. A szükséges CD az „fns”.

```

λ(x, sin x + 1)
<lambda>
  <bvar>
    <ci> x </ci>
  </bvar>
  <apply>
    <sin/>
    <apply>
      <plus/>
      <ci> x </ci>
      <cn> 1 </cn>
    </apply>
  </apply>
</lambda>

\lambda\left(x,\sin x+1\right)

<OMBIND>
  <OMS cd="fns" name="lambda"/>

```

```

<OMBVAR>
  <OMV name="x"/>
</OMBVAR>
<OMA>
  <OMS cd="transc" name="sin"/>
  <OMA>
    <OMS cd="arith" name="plus"/>
    <OMV name="x"/>
    <OMI> 1 </OMI>
  </OMA>
</OMA>
</OMBIND>

```

### 2.6.3. Függvényekkel végzett műveletek

Függvények inverzét az **<inverse/>** elemmel írhatjuk le.

A **<compose/>** elem segítségével függvények kompozícióját jelölhetjük.

TEX-ben az inverz-függvénynek nincsen külön parancsa, indexek segítségével írhatjuk le. A függvénykompozíció jelölésére használjuk a **\circ** parancsot.

OpenMath-ban a fenti műveleteket egy **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „*inverse*”, „*left\_compose*” beállításával) érhetjük el. A szükséges CD az „*fns*”.

$f^{-1}$

```

<apply>
  <inverse definitionURL="MyDefinition.def"/>
  <ci> f </ci>
</apply>

f^{-1}

<OMA>
  <OMS cd="fns" name="inverse"/>
  <OMV name="f"/>
</OMA>

```

$(f \circ g)(x)$

```

<apply>
  <apply>
    <compose/>
    <fn>
      <ci> f </ci>
    </fn>
    <fn>
      <ci> g </ci>
    </fn>
  </apply>
  <ci> x </ci>
</apply>

\left(f\circ g\right)\left(x\right)

```

```

<OMA>
  <OMS cd="fns" name="compose"/>
  <OMA>
    <OMATTR>
      <OMATP>
        <OMS cd="ecc" name="type"/>
        <OMS cd="ecc" name="function"/>
      </OMATP>
      <OMV name="f"/>
    </OMATTR>
  </OMA>
  <OMA>
    <OMATTR>
      <OMATP>
        <OMS cd="ecc" name="type"/>
        <OMS cd="ecc" name="function"/>
      </OMATP>
      <OMV name="g"/>
    </OMATTR>
  </OMA>
  <OMV name="x"/>
</OMA>

```

## 2.7. Integrál- és differenciálszámítás

### 2.7.1. Integrálszámítás

Az **<int/>** elemet használjuk az integrál leírásához. A változó megadásához a **<bvar>** elem szükséges. Az alsó és felső határ megadására több módszer közül választhatunk: az egyik a **<lowlimit>** és az **<uplimit>** elemek használata; a másik az **<interval>** elem használata, ebben az esetben mindkét határt az **<interval>** elem utódaként kell megadnunk; a harmadik pedig a **<condition>** elem segítségével, halmazként adjuk meg a határokat.

TEX-ben az **\int** parancsot használjuk az integrál leírására. A határokat a **\limits** parancs után indexek segítségével adjuk meg.

OpenMath-ban az integrál és intervallumának leírására az **<OMS/>** elemet használhatjuk, a megfelelő paraméterezéssel (a *name* attribútum „defint”, „interval”; halmazként való megadás esetén „lambda” beállításával). A szükséges CD „defint” esetén „calculus”, „interval”-nál „interval”, „lambda”-nál pedig „fns”.

Példa a **<lowlimit>**, **<uplimit>**, és **<bvar>** használatára:

$$\int_0^a f(x)dx$$

```

<apply>
  <int/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <lowlimit>
    <cn> 0 </cn>
  </lowlimit>
  <uplimit>
    <ci> a </ci>
  </uplimit>
  <apply>
    <fn>
      <ci> f </ci>
    </fn>
    <ci> x </ci>
  </apply>
</apply>

\int\limits_0^a f(x)dx

<OMA>
  <OMS cd="calculus" name="defint"/>
  <OMA>
    <OMS cd="interval" name="interval"/>
    <OMI> 0 </OMI>
    <OMV name="a"/>
  </OMA>
  <OMA>
    <OMATTR>
      <OMATP>
        <OMS cd="ecc" name="type"/>
        <OMS cd="ecc" name="function"/>
      </OMATP>
      <OMV name="f"/>
    </OMATTR>
    <OMV name="x"/>
  </OMA>
</OMA>

```

Példa az **<interval>** használatára:

$$\int_a^b \cos x dx$$

```

<apply>
  <int/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <interval>
    <ci> a </ci>
    <ci> b </ci>
  </interval>
  <apply>
    <cos/>
    <ci> x </ci>
  </apply>
</apply>

\int\limits_a^b \cos x dx

```

```

<OMA>
  <OMS cd="calculus" name="defint"/>
  <OMA>
    <OMS cd="interval" name="interval"/>
    <OMV name="a"/>
    <OMV name="b"/>
  </OMA>
  <OMA>
    <OMS cd="transc" name="cos"/>
    <OMV name="x"/>
  </OMA>
</OMA>

```

Példa a **<condition>** használatára:

$$\int_{x \in D} f(x) dx$$

```

<apply>
  <int/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <condition>
    <reln>
      <in/>
      <ci> x </ci>
      <ci type="set"> D </ci>
    </reln>
  </condition>
  <apply>
    <fn>
      <ci> f </ci>
    </fn>
    <ci> x </ci>
  </apply>
</apply>

\int\limits_{x\in D} f(x)dx

<OMA>
  <OMS cd="calculus" name="defint"/>
  <OMV name="D"/>
  <OMBIND>
    <OMS cd="fns" name="lambda"/>
    <OMBVAR>
      <OMV name="x"/>
    </OMBVAR>
    <OMA>
      <OMATTR>
        <OMATP>
          <OMS cd="ecc" name="type"/>
          <OMS cd="ecc" name="function"/>
        </OMATP>
        <OMV name="f"/>
      </OMATTR>
      <OMV name="x"/>
    </OMA>
  </OMBIND>
</OMA>

```

### 2.7.2. Differenciálszámítás

A **<diff/>** elemet használjuk a differenciálhányados leírásához. A változó megadásához a **<bvar>** elem szükséges.

A parciális differenciál leírásához a **<partialdiff/>** elemet használjuk. A változó megadásához a **<bvar>** elem szükséges. Ez a **<bvar>** tartalmazhat egy **<degree>** elemet is mellyel a parciális derivált foka adható meg.

TEX-ben nincs rá külön parancs, a kifejezést egyszerűen leírjuk. Egyedül a  $\partial$  jelnek van speciális megjelenítése, ez a **\partial** parancs.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „diff” vagy „partialdiff” beállításával) helyettesítjük. A szükséges CD a „calculus”.

$$\frac{d}{dx} f(x)$$

```
<apply>
  <diff/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <apply>
    <fn>
      <ci> f </ci>
    </fn>
    <ci> x </ci>
  </apply>
</apply>
```

```
\frac d{dx} f(x)
```

```
<OMA>
  <OMS cd="calculus" name="diff"/>
  <OMV name="x"/>
</OMA>
<OMATTR>
  <OMATP>
    <OMS cd="ecc" name="type"/>
    <OMS cd="ecc" name="function"/>
  </OMATP>
  <OMV name="f"/>
</OMATTR>
<OMV name="x"/>
</OMA>
```

$$\frac{\partial^2}{\partial x^2} \frac{\partial}{\partial y} f(x, y)$$

```
<apply>
```

```

<partialdiff/>
<bvar>
  <ci> x </ci>
  <degree>
    <cn> 2 </cn>
  </degree>
</bvar>
<bvar>
  <ci> y </ci>
</bvar>
<apply>
  <fn>
    <ci> f </ci>
  </fn>
  <ci> x </ci>
  <ci> y </ci>
</apply>
</apply>

\frac{\partial^2}{\partial x^2}\frac{\partial}{\partial y}
f(x,y)

<OMA>
  <OMA>
    <OMS cd="calculus" name="partialdiff"/>
    <OMBIND>
      <OMS cd="fns" name="lambda"/>
      <OMBVAR>
        <OMV name="x"/>
      </OMBVAR>
      <OMBVAR>
        <OMV name="y"/>
      </OMBVAR>
      <OMA>
        <OMV name="f"/>
        <OMATTR>
          <OMATP>
            <OMS cd="presentation" name="left"/>
            <OMSTR>(</OMSTR>
            <OMS cd="presentation" name="right"/>
            <OMSTR>)</OMSTR>
          </OMATP>
          <OMV name="x"/>
          <OMV name="y"/>
        </OMATTR>
      </OMA>
    </OMBIND>
  </OMA>
  <OMV name="x"/>
  <OMV name="y"/>
</OMA>

```

## 2.8. Halmazelmélet

### 2.8.1. Halmazok megadása

A **<set>** elemet használjuk általában a halmazok leírásához. Ha a halmaz elemeinek sorrendje is fontos, akkor a **<list>** elemet használjuk. A halmazokat vagy felsorolással, vagy a **<bvar>** és a **<condition>** elemek segítségével adhatjuk meg. A **<set>**-nél

megadható a *type* argumentum, értékei „normal” és „multiset”, ez utóbbi megengedi az elemek ismétlődését, alapértelmezés a „normal”. A **<list>** saját attribútuma az *order*, mellyel a sorrend típusát adhatjuk meg, két értéket vehet fel, a sorrend lehet „lexicographic” vagy „numeric”, alapértelmezés az utóbbi.

Egyiknek sincs *definitionURL* attribútuma!

Hogy egy adott elem része-e a halmaznak, azt az **<in/>** és a **<notin/>** elemmel írhatjuk le. Mindkettőnek két argumentuma van.

TEX-ben a halmazok megadására nincs külön parancs, a halmazok elemeit és egyéb paramétereit egyszerűen leírjuk. Az „eleme”, „nem eleme” kifejezéseket az **\in**, **\notin** parancsokkal írhatjuk le.

OpenMath-ban a halmazok megadását az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „set”, „list”, „in”, „notin” beállításával) végezzük el. A szükséges CD „list” esetén „list”, egyébként „set”.

$\{a,b,c\}$

```
<set>
  <ci> b </ci>
  <ci> a </ci>
  <ci> c </ci>
</set>

\left\{a,b,c\right\}

<OMATTR>
  <OMATP>
    <OMS cd="presentation" name="left"/>
    <OMSTR>{</OMSTR>
    <OMS cd="presentation" name="right"/>
    <OMSTR>}</OMSTR>
  </OMATP>
  <OMA>
    <OMS cd = "set" name="set"/>
    <OMV name="a"/>
    <OMV name="b"/>
    <OMV name="c"/>
  </OMA>
</OMATTR>
```

$\{x \mid x < 5\}$

```
<set>
  <bvar>
    <ci> x </ci>
```

```

    </bvar>
    <condition>
      <reln>
        <lt/>
        <ci> x </ci>
        <cn> 5 </cn>
      </reln>
    </condition>
  </set>

\left\{x\mid x<5\right\}

<OMATTR>
  <OMATP>
    <OMS cd="presentation" name="left"/>
    <OMSTR>{</OMSTR>
    <OMS cd="presentation" name="right"/>
    <OMSTR>}</OMSTR>
  </OMATP>
  <OMA>
    <OMS cd="relation" name="lt"/>
    <OMV name="x"/>
    <OMI> 5 </OMI>
  </OMA>
</OMATTR>

```

$[a,b,c]$

```

<list>
  <ci> b </ci>
  <ci> a </ci>
  <ci> c </ci>
</list>

\left\{[a,b,c\right\}

<OMATTR>
  <OMATP>
    <OMS cd="presentation" name="left"/>
    <OMSTR>[</OMSTR>
    <OMS cd="presentation" name="right"/>
    <OMSTR>]</OMSTR>
  </OMATP>
  <OMA>
    <OMS cd = "set" name="set"/>
    <OMV name="a"/>
    <OMV name="b"/>
    <OMV name="c"/>
  </OMA>
</OMATTR>

```

$[x\mid x<5]$

```

<list order="numeric">
  <bvar>
    <ci> x </ci>
  </bvar>
  <condition>
    <reln>
      <lt/>
      <ci> x </ci>
      <cn> 5 </cn>
    </reln>
  </condition>
</list>

\left\{[x\mid x<5\right\}

<OMATTR>
  <OMATP>

```

```

<OMS cd="presentation" name="left"/>
<OMSTR>[</OMSTR>
<OMS cd="presentation" name="right"/>
<OMSTR>]</OMSTR>
</OMATP>
<OMA>
  <OMS cd="relation" name="lt"/>
  <OMV name="x"/>
  <OMI> 5 </OMI>
</OMATTR>

```

$a \in A$

```

<reln>
  <in/>
  <ci> a </ci>
  <ci type="set"> A </ci>
</reln>

a\in A

<OMA>
  <OMS cd="set" name="in"/>
  <OMV name="a"/>
  <OMV name="A"/>
</OMA>

```

$a \notin A$

```

<reln>
  <notin/>
  <ci> a </ci>
  <ci> A </ci>
</reln>

a\notin A

<OMA>
  <OMS cd="set" name="notin"/>
  <OMV name="a"/>
  <OMV name="A"/>
</OMA>

```

### 2.8.2. Halmazműveletek

A MathML-ben a következő halmazműveletek ismertek:  
**<union/>** - „unió”; **<intersect/>** - „metszet”; **<subset/>** - „részhalmaz”;  
**<prsubset/>** - „valódi részhalmaz”; **<notsubset/>** - „nem részhalmaz”; **<notprsubset/>** - „nem valódi részhalmaz”; **<setdiff/>** - „különbség”. Az **<union/>**-nak és az **<intersect/>**-nek több, a maradéknak két argumentuma lehet.

A fenti elemek T<sub>E</sub>X megfelelői: **\cup**, **\cap**, **\subseteq**, **\subset**, **\not\subseteq**, **\not\subset**, **\setminus**.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „union”, „intersect”, „subset”, „notsubset”, „prsubset”, „notprsubset”, „setdiff” beállításával) helyettesítjük. A szükséges CD minden esetben a „set”.

$A \cup B$

```
<apply>
  <union/>
  <ci> A </ci>
  <ci> B </ci>
</apply>
```

$A \setminus \cup B$

```
<OMA>
  <OMS cd="set" name="union"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \cap B$

```
<apply>
  <intersect/>
  <ci type="set"> A </ci>
  <ci type="set"> B </ci>
</apply>
```

$A \setminus \cap B$

```
<OMA>
  <OMS cd="set" name="intersect"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \subseteq B$

```
<reln>
  <subset/>
  <ci> A </ci>
  <ci> B </ci>
</reln>
```

$A \setminus \subseteq B$

```
<OMA>
  <OMS cd="set" name="subset"/>
  7 <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \subset B$

```
<reln>
  <prsubset/>
  <ci> A </ci>
  <ci> B </ci>
</reln>
```

$A \setminus \subset B$

```
<OMA>
  <OMS cd="set" name="prsubset"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \not\subseteq B$

```
<reln>
  <notsubset/>
  <ci> A </ci>
  <ci> B </ci>
</reln>
```

$A \setminus \text{not} \setminus \text{subsest} B$

```
<OMA>
  <OMS cd="set" name="notsubset"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \not\subset B$

```
<reln>
  <notprsubset/>
  <ci> A </ci>
  <ci> B </ci>
</reln>
```

$A \setminus \text{not} \setminus \text{subset} B$

```
<OMA>
  <OMS cd="set" name="notprsubset"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

$A \setminus B$

```
<apply>
  <setdiff/>
  <ci> A </ci>
  <ci> B </ci>
</apply>
```

$A \setminus \text{setminus} B$

```
<OMA>
  <OMS cd="set" name="setdiff"/>
  <OMV name="A"/>
  <OMV name="B"/>
</OMA>
```

## 2.9. Sorozatok

Egy sorozat határértékének leírásához a **<limit/>** elemet használjuk. A határpont megadása a **<lowlimit>** elemmel, vagy a **<condition>** elemmel lehetséges. Az összegzés indexét a **<bvar>** elemmel adhatjuk meg.

Azt, hogy egy adott sorozat mihez tart, azt a **<tendsto/>** elem segítségével adjuk meg, melynek két argumentuma van. Az irányt a *type* attribútummal határozhatjuk meg, melynek értékei a következők lehetnek: „above”, „below”, „two-sided”, alapértelmezése az „above”.

Egy sorozat elemeinek összegzéséhez, a szumma leírásához a **<sum/>** elemet használjuk. A **<product/>** elemet használjuk a sorozat elemeinek szorzatának, a produktum leírásához. Mindkettőnél a felső és alsó határ megadása az **<int/>** elemnél megismert módon vagy **<uplimit>** és **<lowlimit>** elemekkel, vagy a **<condition>** elemmel lehetséges. Az összegzés indexét a **<bvar>** elemmel adhatjuk meg.

TEX-ben a határérték leírására a **\lim** parancsot használjuk, mely után, a **\limits** és a **\to** parancs párral adhatjuk meg hogy a határérték mihez tart. A MathML **<tendsto/>** elemének megfelelői TEX-ben iránytól és nagyságtól függően a következők: **\leftarrow**, **\rightarrow**, **\leftrightarrow**; **\longleftarrow**, **\longrightarrow**, **\longleftrightarrow**.

A szumma és a produktum leírása a **\sum** és **\prod** parancsokkal történik, melyek után kiemelt matematikai módban indexekkel, egyébként a **\limits** parancssal lehet megadni a kívánt határokat.

A végtelen ( $\infty$ ) leírására használjuk az **\infty** parancsot.

OpenMath-ban a fenti elemeket az **<OMS/>** elem megfelelő paraméterezésével (a *name* attribútum „limit”, „sum”, „product” beállításával) helyettesítjük. A **<tendsto/>** megfelelői az **<OMS>**-ban a *name* attribútum „above”, „below”, „both\_sides” beállításai. A „sum” és „product” beállításnál az intervallum leírására szintén egy **<OMS/>** elem megfelelő paraméterezése (a *name* attribútum „interval”; halmazként való megadás esetén „lambda” beállításával) használatos. A szükséges CD a „sum”,

„product” műveleteknél az „arith”, „interval”-nál „interval”, „lambda”-nál „fns”, a többinél a „limit”.

A végtelen megfelelője az „infinity”, mely a „nums” CD-ben található.

$$\lim_{x \rightarrow 0} \sin x$$

```
<apply>
  <limit/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <lowlimit>
    <cn> 0 </cn>
  </lowlimit>
  <apply>
    <sin/>
    <ci> x </ci>
  </apply>
</apply>

\lim\limits_{n \to 0}\sin x

<OMA>
  <OMS cd="limit" name="limit"/>
  <OMA>
    <OMS cd="transc" name="sin"/>
    <OMV name="x"/>
  </OMA>
</OMA>
```

$$x^2 \rightarrow a^2$$

```
<reln>
  <tendsto type="above"/>
  <apply>
    <power/>
    <ci> x </ci>
    <cn> 2 </cn>
  </apply>
  <apply>
    <power/>
    <ci> a </ci>
    <cn> 2 </cn>
  </apply>
</reln>

x^2\rightarrow a^2

<OMA>
  <OMS cd = "limit" name="above"/>
  <OMA>
    <OMS cd="arith" name="power"/>
    <OMV name="x"/>
    <OMI>2</OMI>
  </OMA>
  <OMA>
    <OMS cd="arith" name="power"/>
    <OMV name="a"/>
    <OMI>2</OMI>
  </OMA>
</OMA>
```

$$\sum_{x=a}^b f(x)$$

```

<apply>
  <sum/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <lowlimit>
    <ci> a </ci>
  </lowlimit>
  <uplimit>
    <ci> b </ci>
  </uplimit>
  <apply>
    <fn>
      <ci> f </ci>
    </fn>
    <ci> x </ci>
  </apply>
</apply>

```

```
\sum\limits_{x=a}^b f(x)
```

```

<OMA>
  <OMS cd="arith" name="sum"/>
  <OMA>
    <OMS cd="interval" name="integer-interval"/>
    <OMV name="a"/>
    <OMV name="b"/>
  </OMA>
  <OMBIND>
    <OMS cd="fns" name="lambda"/>
    <OMBVAR>
      <OMV name="x"/>
    </OMBVAR>
    <OMA>
      <OMV name="f"/>
      <OMATTR>
        <OMATP>
          <OMS cd="presentation" name="left"/>
          <OMSTR></OMSTR>
          <OMS cd="presentation" name="right"/>
          <OMSTR></OMSTR>
        </OMATP>
        <OMV name="x"/>
      </OMATTR>
    </OMA>
  </OMBIND>
</OMA>

```

$$\prod_{x \in B} f(x)$$

```

<apply>
  <product/>
  <bvar>
    <ci> x </ci>
  </bvar>
  <condition>
    <reln>
      <in/>
      <ci> x </ci>
      <ci type="set"> B </ci>
    </reln>
  </condition>
  <apply>
    <fn>

```

```

        <ci> f </ci>
      </fn>
    <ci> x </ci>
  </apply>
</apply>

\prod_{x\in B} f(x)

<OMA>
  <OMS cd="arith" name="product"/>
  <OMV name="B"/>
  <OMBIND>
    <OMS cd="fns" name="lambda"/>
    <OMBVAR>
      <OMV name="x"/>
    </OMBVAR>
    <OMA>
      <OMV name="f"/>
      <OMATTR>
        <OMATP>
          <OMS cd="presentation" name="left"/>
          <OMSTR>(</OMSTR>
            <OMS cd="presentation" name="right"/>
            <OMSTR>)</OMSTR>
          </OMATP>
          <OMV name="x"/>
        </OMATTR>
      </OMA>
    </OMBIND>
  </OMA>

```

## 2.10. Trigonometria

A trigonometriai függvények széles körben ismertek, ezért külön-külön történő ismertetésüktől eltekintek, de álljon itt felsorolásuk:

**<sin/>, <cos/>, <tan/>, <cot/>, <sec/>, <csc/>,  
<sinh/>, <cosh/>, <tanh/>, <coth/>, <sech/>, <csch/>,  
<arcsin/>, <arccos/>, <arctan/>>**

Minden trigonometriai elemnek egyetlen argumentuma van.

TEX-ben a fenti elemeknek megfelelő parancsok rendre: **\sin, \cos, \tan, \cot, \sec, \csc, \sinh, \cosh, \tanh, \coth, \arcsin, \arccos, \arctan**. A **<sech/>>** és a **<csch/>>** elemeknek nincsenek megfelelői.

OpenMath-ban a trigonometriai elemeket egy **<OMS/>>** elem *name* attribútumában adhatjuk meg. A szögfüggvények nevei megegyeznek a fenti nevekkel. A trigonometriai függvényeket tartalmazó CD neve „*transc*”.

$\sin x$

```
<apply>
  <sin/>
  <ci> x </ci>
</apply>

\sin x

<OMA>
  <OMS cd="transc" name="sin"/>
  <OMV name="x"/>
</OMA>
```

$\sin(\cos x + x^3)$

```
<apply>
  <sin/>
  <apply>
    <plus/>
    <apply>
      <cos/>
      <ci> x </ci>
    </apply>
    <apply>
      <power/>
      <ci> x </ci>
      <cn> 3 </cn>
    </apply>
  </apply>
</apply>

\sin(\cos x+x^3)

<OMA>
  <OMS cd="transc" name="sin"/>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="left"/>
      <OMSTR>(</OMSTR>
      <OMS cd="presentation" name="right"/>
      <OMSTR>)</OMSTR>
    </OMATP>
    <OMA>
      <OMS cd="arith" name="plus"/>
      <OMA>
        <OMS cd="transc" name="cos"/>
        <OMV name="x"/>
      </OMA>
      <OMA>
        <OMS cd="arith" name="power"/>
        <OMV name="x"/>
        <OMI>3</OMI>
      </OMA>
    </OMA>
  </OMATTR>
</OMA>
```

## 2.11. Statisztika

A statisztikai műveletek közül a következőket találjuk meg MathML-ben: **<mean/>** - átlag; **<sdev/>** - szórás; **<variance/>** - szórásnégyzet; **<median/>** - medián; **<mode/>**

- módusz; **<moment/>** - statisztikai momentum. Ez utóbbinál a momentum fokát a **<degree>** elemmel határozhatjuk meg.

TEX-ben a fenti elemeknek nincsenek megfelelői.

OpenMath-ban a fenti elemek leírására az **<OMS/>** elemet használhatjuk, a megfelelő paraméterezéssel (a *name* attribútum „mean”, „sdev”, „variance”, „median”, „mode”, „moment” beállításával). A szükséges CD a „stats”.

$\bar{X}$

```
<apply>
  <mean/>
  <ci> X </ci>
</apply>

\overline x

<OMA>
  <OMS cd="stats" name="mean"/>
  <OMV name="x"/>
</OMA>
```

$\sigma(X)$

```
<apply>
  <sdev/>
  <ci> X </ci>
</apply>

\sigma(X)

<OMA>
  <OMS cd="stats" name="sdev"/>
  <OMV name="x"/>
</OMA>
```

$\sigma^2(X)$

```
<apply>
  <variance/>
  <ci> X </ci>
</apply>

\sigma^2 (X)

<OMA>
  <OMS cd="stats" name="variance"/>
  <OMV name="x"/>
</OMA>
```

median( $X$ )

```
<apply>
  <median/>
  <ci> X </ci>
</apply>

\rm{median}(X)
```

```
<OMA>
  <OMS cd="stats" name="median"/>
  <OMV name="x"/>
</OMA>
```

mode( $X$ )

```
<apply>
  <mode/>
  <ci> X </ci>
</apply>
```

$\text{mode}(X)$

```
<OMA>
  <OMS cd="stats" name="mode"/>
  <OMV name="x"/>
</OMA>
```

$\langle X^3 \rangle$

```
<apply>
  <moment/>
  <degree>
    <cn> 3 </cn>
  </degree>
  <ci> X </ci>
</apply>
```

$\left\langle X^3 \right\rangle$

```
<OMA>
  <OMS cd="stats" name="moment"/>
  <OMV name="x"/>
</OMA>
```

## 2.12. Lineáris algebra

### 2.12.1. Vektorok

A **<vector>** elemet használjuk a vektorok leírásához. Argumentumai írják le a vektor komponenseit. A vektorok tartalmilag egy egysoros mátrixhoz hasonlíthatók.

TEX-ben nincs megfelelője.

OpenMath-ban a vektorok leírására az **<OMS/>** elemet használhatjuk, a megfelelő paraméterezéssel (a *name* attribútum „vector” beállításával). A szükséges CD a „linalg”.

(1,2,3, $x$ )

```
<vector>
  <cn> 1 </cn>
  <cn> 2 </cn>
  <cn> 3 </cn>
```

```

    <ci> x </ci>
  </vector>

\left(1,2,3,x\right)

<OMATTR>
  <OMATP>
    <OMS cd="presentation" name="left"/>
    <OMSTR>(</OMSTR>
    <OMS cd="presentation" name="right"/>
    <OMSTR>)</OMSTR>
  </OMATP>
  <OMA>
    <OMS cd = "linalg" name="vector"/>
    <OMI>1</OMI>
    <OMI>2</OMI>
    <OMI>3</OMI>
    <OMV name="x"/>
  </OMA>
</OMATTR>

```

### 2.12.2. Mátrixok

A mátrixok leírásához a **<matrix>** elemet használjuk. Argumentumai csak **<matrixrow>** elemek lehetnek, ezek írják le a mátrix sorait.

TEX-ben a megjelenítő elemek táblákkal és mátrixokkal foglalkozó fejezetében (76. oldal) leírtak érvényesek itt is.

OpenMath-ban a mátrixok leírására az **<OMS/>** elemet használhatjuk, a megfelelő paraméterezéssel (a *name* attribútum beállításával: a „*matrix*” írja le magát a mátrixot, a „*matrixrow*” pedig a mátrix sorait). A szükséges CD az „*linalg*”.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

```

<matrix>
  <matrixrow>
    <cn> 0 </cn>
    <cn> 1 </cn>
    <cn> 0 </cn>
  </matrixrow>
  <matrixrow>
    <cn> 0 </cn>
    <cn> 0 </cn>
    <cn> 1 </cn>
  </matrixrow>
  <matrixrow>
    <cn> 1 </cn>
    <cn> 0 </cn>
    <cn> 0 </cn>
  </matrixrow>
</matrix>

```

```

\bmatrix 0&1&0\\0&0&1\\1&0&0\endbmatrix

<OMATTR>
  <OMATP>
    <OMS cd="presentation" name="left"/>
    <OMSTR>[</OMSTR>
    <OMS cd="presentation" name="right"/>
    <OMSTR>]</OMSTR>
  </OMATP>
  <OMA>
    <OMS cd = "linalg" name="matrix"/>

    <OMA>
      <OMS cd = "linalg" name="matrixrow"/>
      <OMI>0</OMI>
      <OMI>1</OMI>
      <OMI>0</OMI>
    </OMA>
    <OMA>
      <OMS cd = "linalg" name="matrixrow"/>
      <OMI>0</OMI>
      <OMI>0</OMI>
      <OMI>1</OMI>
    </OMA>
    <OMA>
      <OMS cd = "linalg" name="matrixrow"/>
      <OMI>1</OMI>
      <OMI>0</OMI>
      <OMI>0</OMI>
    </OMA>
  </OMA>
</OMATTR>

```

### 2.12.3. Mátrixműveletek

A determinánsok leírására a **<determinant/>** elemet használjuk. Egy mátrix transzponáltját a **<transpose/>** elemmel írjuk le.

A **<selector/>** elemet használjuk egy mátrix, vektor vagy lista elemeinek kiválasztásához. Argumentumai közül az első mindig az objektumot határozza meg, a második kiválasztja az objektum elemét, mátrixok esetén a sort, harmadiknak csak mátrix esetén van értelme (egyébként nem vesszük figyelembe), a megadott sor oszlopát jelöli ki. Mátrix esetén, ha a harmadik argumentum nincs megadva, akkor hatása az egész sorra vonatkozik.

TEX-ben csak a determináns leírására van parancs, ez pedig a **\det**. A transzponált leírását egyszerűen indexeléssel írjuk le. A **<selector/>** elemnek nincs megfelelője, mivel ezt csak számítógépes algebrában használjuk.

OpenMath-ban a fenti elemek leírására az **<OMS/>** elemet használhatjuk, a megfelelő paraméterezéssel (a *name* attribútum „determinant”, „transpose”, „selector” beállításával). A szükséges CD az „linalg”.

$\det A$

```
<apply>
  <determinant/>
  <ci type="matrix"> A </ci>
</apply>

\det\rm\bf A

<OMA>
  <OMS cd="linalg" name="determinant"/>
  <OMV name="A"/>
</OMA>
```

$A^T$

```
<apply>
  <transpose/>
  <ci type="matrix"> A </ci>
</apply>

A^T

<OMA>
  <OMS cd="linalg" name="transpose"/>
  <OMV name="x"/>
</OMA>
```

A következő példában

```
<apply>
  <selector/>
  <matrix>
    <matrixrow>
      <cn> 1 </cn>
      <cn> 2 </cn>
    </matrixrow>
    <matrixrow>
      <cn> 3 </cn>
      <cn> 4 </cn>
    </matrixrow>
  </matrix>
  <cn> 1 </cn>
</apply>

<OMA>
  <OMS cd="linalg" name="selector"/>
  <OMI> 1 </OMI>
  <OMATTR>
    <OMATP>
      <OMS cd="presentation" name="left"/>
      <OMSTR>[</OMSTR>
      <OMS cd="presentation" name="right"/>
      <OMSTR>]</OMSTR>
    </OMATP>
  </OMATTR>
</OMA>
```

```

<OMS cd = "linalg" name="matrix"/>
<OMA>
  <OMS cd = "linalg" name="matrixrow"/>
  <OMI>1</OMI>
  <OMI>2</OMI>
</OMA>
<OMA>
  <OMS cd = "linalg" name="matrixrow"/>
  <OMI>3</OMI>
  <OMI>4</OMI>
</OMA>
</OMA>
</OMATTR>
</OMA>

```

mivel csak két argumentumunk van, a mátrix első sora lesz kiválasztva:

```

<matrixrow>
  <cn> 1 </cn>
  <cn> 2 </cn>
</matrixrow>

<OMA>
  <OMS cd = "linalg" name="matrixrow"/>
  <OMI>1</OMI>
  <OMI>2</OMI>
</OMA>

```

## 2.13. Szemantikai elemek

A **<semantics>** elemet többletinformációk hozzáfűzésére használjuk. Első argumentuma mindig egy MathML kódú kifejezés, amihez a megjegyzést fűzzük. A következő argumentumoknak kötelezően **<annotation>** vagy **<annotation-xml>** elemmel kell kezdődniük, ezek írják le magát a megjegyzést. A két elem között az a különbség, hogy az elsőben csak nem XML-alapú leírást adhatunk meg, a másodikban viszont csak XML-alapút. Egy saját attribútumuk van, az *encoding*, melyben meghatározhatjuk, hogy az adott megjegyzés milyen formátumú. Alapértelmezése „”, vagyis nincs meghatározva.

Általában a kifejezés úgy jelenik meg, ahogyan azt az első argumentumban leírtuk. Esetenként a megjelenítő figyelembe veheti a hozzáfűzött információt, ezt elsősorban a tartalmi és megjelenítő elemek keverésekor használhatjuk ki.

A szemantikai elemeknek nincsenek megfelelőik sem T<sub>E</sub>X-ben, sem OpenMath-ban, viszont egy MathML kódhoz ezekkel az elemekkel kapcsolhatunk T<sub>E</sub>X és OpenMath kódokat is.

A következő példa azt mutatja be, hogy a  $\sin(x)+5$  kifejezést és különböző formátumú értelmezéseit hogyan valósíthatjuk meg:

```
<semantics>
  <apply>
    <plus/>
    <apply>
      <sin/>
      <ci> x </ci>
    </apply>
    <cn> 5 </cn>
  </apply>
  <annotation encoding="Tex">
    \sin x + 5
  </annotation>
  <annotation-xml encoding="OpenMath">
    <OMA>
      <OMS cd="arith" name="plus"/>
      <OMA>
        <OMS cd="transc" name="sin"/>
        <OMV name="x"/>
      </OMA>
      <OMI> 5 </OMI>
    </OMA>
  </annotation-xml>
</semantics>
```

### 3. A jelölések ötvözése

Ebben a részben csak MathML-ről lesz szó, hiszen csak ebben a leírónyelvben van értelme erről beszélni.

#### 3.1. Miért használunk kétféle jelölést?

A megjelenítő jelölés elsősorban a minden médiának megfelelő, helyes megjelenítésre helyezi a hangsúlyt. Ezért általában, a tartalom jelentése sokadrangú szempont az egyes kifejezések megjelenítésében, azok megértését, feldolgozását a felhasználóra bízta.

A tartalmi jelölés a matematikai kifejezések jelentését hangsúlyozza, maga a megjelenítés helyes formájával csak a legszükségesebb mértékben foglalkozik. Természetesen ezért a

matematikai alkalmazásokban nagyobb mértékben tudjuk felhasználni az így kódolt jelöléseket.

Így tehát, ha a matematikai leírónyelvek minden hasznos tulajdonságát ki akarjuk használni (magyarul a helyes tartalom mellett a helyes megjelenést is fontosnak tartjuk), akkor a kétféle jelölési módszert ötvöznünk kell.

Természetesen ennek is megvannak a maga szabályai, amiket be kell tartanunk, ha helyes kódokat szándékozunk készíteni.

## **3.2. A két jelölésfajta összekapcsolása**

### **3.2.1. Megjelenítő elemek a tartalmi elemek között**

Meglehetősen behatárolt eszközök állnak rendelkezésünkre az így módon történő összekapcsolásra, ha nem akarunk homályos kódokat készíteni. Három módon használhatjuk korlátlanul a megjelenítő elemeket a tartalmi elemek között:

- **Megjelenítő elemek a tartalmi token-elemekben:** A `<ci>` és `<cn>` token-elemekben engedélyezett a megjelenítő elemek használata. Ezek a kódblokkok a megjelenítő elemeknek megfelelően fognak megjeleníteni (azaz az `<mi>` és `<mn>` elemeknek megfelelően). Ilyenkor ne használjuk a `<sep/>` elemet, mert a megjelenítő úgyis figyelmen kívül hagyja.
- **Megjelenítő elemek az `<fn>` elemekben:** Az `<fn>` elemekben használhatunk megjelenítő elemeket, de vigyáznunk kell arra, hogy nem keverhetjük össze a tartalmi elemekkel, mert ez hibát okozna.
- **Megjelenítő elemek a `<semantics>` elemekben:** A `<semantics>` elem egyik feladata, hogy a megjelenítő és a tartalmi elemek között kapcsolatot teremtsen. A legtöbb esetben a tartalmi jelölést mellett nincs sok szerepe a

megjelenítő jelölésnek, de a megjelenítőtől függően hatással lehet a kifejezés megjelenítésére, erről bővebben a **<semantics>** elemnél beszéltem.

### 3.2.2. Tartalmi elemek a megjelenítő elemek között

A legfontosabb tudnivaló, hogy az így megszerkesztett kifejezésnek nagyon pontos a megjelenítése. Ilyenkor a tartalmi többletjelentés hatással van a megjelenítésre is. Általában csak a következő tartalmi elemek engedélyezettek: **<apply>**, **<ci>**, **<cn>**, **<declare>**, **<fn>**, **<interval>**, **<lambda>**, **<list>**, **<matrix>**, **<matrixrow>**, **<reln>**, **<set>**, **<vector>**; a műveletek, relációk nem. Fontos tudnunk azt is, hogy a megjelenítő token-elemekben tartalmi elemeket nem használhatunk.

Elmondhatjuk, hogy ebben az összekapcsolási módban a tartalmi elemek helyettesíthetők megjelenítő elemekkel. Lássuk erre a következő példát:

```
<mfrac>
  <mi>x</mi>
  <interval closure="open-closed">
    <cn>1</cn>
    <cn>3</cn>
  </interval>
</mfrac>
```

A fenti kevert jelölés tiszta megjelenítő elemekkel a következő módon állítható elő, természetesen a többletinformáció ilyenkor elveszik:

```
<mfrac>
  <mi>x</mi>
  <mfenced close="]">
    <mn>1</mn>
    <mn>3</mn>
  </mfenced>
</mfrac>
```

### 3.3. Makrók készítése

A MathML-ben lehetőségünk van makrókat készíteni. Például készítsünk egy makrót, mely a mátrix rangját adja meg, és egy olyat, amely leegyszerűsíti a transzponált leírását. Legyen az első a `<rank/>`, melynek kódja:

```
<fn>
  <semantics>
    <ci><mo>rank</mo></ci>
    <annotation-xml encoding="OpenMath">
      <OMS cd="BasicLinAlg" name="matrix-rank"/>
    </annotation-xml>
  </semantics>
</fn>
```

Az egyszerűsítés pedig legyen a `<tran>` „elem”:

```
<apply>
  <transpose/>
  <ci>X</ci>
</apply>
```

Lássuk most a  $\text{rank}(u^T v) = 1$  saját függvény kódolását:

```
<reln>
  <eq>
    <apply>
      <rank/>
      <apply>
        <times/>
        <tran>u</tran>
        <ci>v</ci>
      </apply>
    </apply>
    <cn>1</cn>
  </reln>
```

Mint a fenti példákból látjuk, makrók (vagy egész makrókönyvtárak) segítségével mennyire egyszerűen és hatásosan kombinálhatjuk a megjelenítő és tartalmi jelölésformákat.

## 4. Az interfész

### 4.1. A MathML HTML-be ágyazása

Mióta a MathML megszületett, egyre többen ismerik meg, és használják a matematikában. Számptalan felhasználása létezik, az egyszerű szerkesztőktől kezdve a különböző matematikai alkalmazásokig, de mégis használatának legelterjedtebb módja a HTML dokumentumokba való beágyazás.

A MathML HTML-be ágyazásának háromféle interfészét ismerjük.

Az első, mikor szemantikailag integráljuk a két nyelvet. Ez legtöbbször azt használja ki, hogy a MathML az XML „leszármazottja”, így az XML beágyazásának szabályai érvényesek itt is.

A másik módja, mikor az általunk használt böngésző fel van készítve a MathML szövegek felismerésére, így azokat a matematikai jelölés szabályainak megfelelően jeleníti meg.

Harmadik módszer, mikor olyan segédeszközöket használunk, melyek a MathML kódot kezelni tudják. Ezek az eszközök lehetnek speciális szövegszerkesztők, fordítók, kódgenerátorok számítógépes algebrai rendszerek és más tudományos szoftverek.

#### 4.1.1. A **<math>** elem

A MathML-nek egyetlen interfész eleme van, a **<math>**, mely körbezár minden MathML nyelvű szövegrészt egy adott HTML dokumentumban. Mindig a legfelső szinten áll, ezért nem lehet egymásba ágyazni több ilyen elemet. Ez az elem jelzi a megjelenítőnek, hogy innentől ne HTML-ként, hanem MathML-ként kell értelmeznie a további szöveget.

A **<math>** elemnek korlátlan számú utóda lehet, és alapjában véve úgy viselkedik, mint egy **<mrow>** elem.

A **<math>** elemnek a következő attribútumai lehetnek: *class*, *style* (ezeket a stíluslapokkal való későbbi kompatibilitás miatt specifikálták), *macros* (külső makrók helyét határozhatjuk meg itt), *mode* (a megjelenítés módját határozza meg, hasonlít az **<mstyle>** *displaystyle* attribútumára). Ezeket belső interfész-attribútumoknak hívjuk.

Ahhoz, hogy böngészőnkben helyesen értelmezni és megjeleníteni tudjuk MathML kódjainkat, szükséges lehet még néhány attribútum beállítása. Ezeket szintén a **<math>** elem attribútumaiként adjuk meg, feladatuk a böngésző és a MathML megjelenítő közötti információátadás. Ezek az úgynevezett külső interfész-attribútumok: *type* (a tartalom MIME-típusát adhatjuk itt meg), *name* (a szövegrész neve), *height*, *width*, *baseline* (az elemek adott tulajdonságainak ideiglenes beállítására szolgálnak), *overflow* (ha egy kifejezés valamilyen okból – pl. túl hosszú – nem lenne megjeleníthető, itt adhatunk meg egy alternatív megjelenési módot; értékei: „*scroll*” – görgetést tesz lehetővé görgetősávok alkalmazásával egy új ablakban; „*elide*” – ahol tudja, lerövidíti a kifejezéseinket, pl. egy polinomnál a + ... + alkalmazásával; „*truncate*” – a jobboldali és alsó határon túl eső részeket egyszerűen levágja; „*scale*” – átállítja a betűtípusok méretét, vagy ha lehetséges, nagyobbra állítja az ablak méretét), *altimg*, *alttext* (ezeket akkor használjuk, ha a böngésző nem alkalmas beágyazott elemek megjelenítésére, így valamilyen más alternatív megoldást kell választanunk).

A fenti attribútumok beállítása a dokumentum első beolvasásakor lépnek érvénybe. Ekkor történnek meg az olyan beállítások, mint a betűméretek módosítása is, melyek a MathML szövegrészekre is hatással vannak.

Az olyan dokumentumaink fejrészébe, amelyek tartalmazznak MathML elemeket, célszerű elhelyezni a következő HTML **<META>** elemet:

```
<META content-math-type="text/mathml">
```

Ugyanígy a **<math>** elem *type* attribútumában is megadhatjuk a `text/mathml` MIME-típust. Ezeket a dokumentumokat a standard `.mm1` kiterjesztéssel is megjelölhetjük.

Közvetlenül nincsen lehetőség arra, hogy MathML kódokhoz kapcsoljunk HTML dokumentumokat, mert a MathML specifikációja nem engedélyezi a MathML és HTML elemek egy adott szövegben való összevegyítését. Sajnos, ez a MathML hibája, a későbbi változatoknál meg kellene vizsgálni ennek a lehetőségét.

A fentiek miatt így nincs arra lehetőségünk, hogy MathML szövegben közvetlenül kapcsokat hozzunk létre más dokumentumokkal, mert nem használhatjuk a HTML `<A>` elemét. Szerencsénkre, mivel a MathML egy XML változat, azért van lehetőségünk erre, bár egy kicsit bonyolultabban. Az **<malignmark/>**, az **<maligngroup/>**, a **<none/>**, a **<power/>** és a **<sep/>** kivételével az összes többi elemnek lehet még két attribútuma, az *XML-LINK* és a *HREF*, mellyel a kapcsok készítése megoldható, a következő módon:

```
<mrow XML-LINK="simple" HREF="http://www.mypage.com">  
  .  
  .  
  .  
</mrow>
```

Az *XML-LINK* a következő értékeket veheti fel: „*simple*”, „*extended*”, „*locator*”, „*group*”, „*document*”, melyek közül MathML kódunknál a „*simple*”-t érdemes használni (a többiről bővebb információt a W3C XML specifikációja<sup>5</sup> ad). Az, hogy az így létrehozott kapocs hogyan jelenik meg, teljesen a böngészőtől függ.

---

<sup>5</sup> Lásd: [www.w3.org/TR/PR\\_XML](http://www.w3.org/TR/PR_XML)

A kapcsolatokhoz hasonlóan képeket sem tudunk megjeleníteni MathML kódban, erre áthidaló megoldás sincs. Legjobb módszer, ha ilyenkor a matematikai kifejezést és a megjelenítendő képet valamilyen módon egyesítjük, és ezt a képet jelenítjük meg normál HTML kódból.

#### 4.1.2. Egy egyszerű MathML példa

Ebben a példában egy HTML kódot mutatok be, amely a keretben található szöveget jeleníti meg:

##### MathML példa

A másodfokú egyenletek megoldóképlete:  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

A HTML kód:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
      charset=windows-1250">
    <META Content-math-Type="text/mathml">
    <TITLE>MathML példa</TITLE>
  </HEAD>
  <BODY>
    <H1 align="center">MathML példa</H1>
    <P>A másodfokú egyenletek megoldóképlete:&nbsp;</P>
    <math>
      <mrow>
        <mi>x</mi>
        <mo>=</mo>
        <mfrac>
          <mrow>
            <mrow>
              <mo>-</mo>
              <mi>b</mi>
            </mrow>
            <mo>&PlusMinus;</mo>
            <msqrt>
              <mrow>
                <msup>
```

```

        <mi>b</mi>
        <mn>2</mn>
    </msup>
    <mo>-</mo>
    <mrow>
        <mn>4</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>a</mi>
        <mo>&InvisibleTimes;</mo>
        <mi>c</mi>
    </mrow>
</mrow>
</msqrt>
</mrow>
<mrow>
    <mn>2</mn>
    <mo>&InvisibleTimes;</mo>
    <mi>a</mi>
</mrow>
</mfrac>
</mrow>
</math>

</P>
</BODY>

</HTML>

```

## 4.2. Az OpenMath HTML-be ágyazása

### 4.2.1. Az <OMOBJ> elem

Hasonlóan a MathML-hez, az OpenMath-nak is egyetlen interfész eleme van, az **<OMOBJ>** elem, mely körbezár minden OpenMath nyelvű szövegrészt az adott HTML dokumentumban, és jelzi a megjelenítőnek, hogy innentől kezdve OpenMath nyelvű leírás kezdődik. Mindig az OpenMath nyelvű szövegek legfelső szintjén áll, nem lehet egymásba ágyazni.

Az **<OMOBJ>** elemnek korlátlan argumentuma lehet, viszont nincsenek attribútumai, alapjában véve úgy viselkedik, mint egy **<OMA>** elem.

OpenMath nyelvű dokumentumoknál is tanácsos lehet beállítani a HTML dokumentumunk fejrészában egy **<META>** elemet, a következő módon:

```
<META content-math-type="text/OpenMath">
```

OpenMath-ban egyáltalán nincs lehetőségünk arra, hogy kapcsolatokat hozzunk létre más dokumentumokkal, ezt abból a HTML dokumentumból kell elvégeznünk, amibe beágyaztuk OpenMath nyelvű szövegünket.

### 4.2.2. Egy egyszerű OpenMath példa

A fenti, MathML-nél megadott példa OpenMath-os formája:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<HTML>

  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=windows-1250">
    <META Content-Type="text/OpenMath">
    <TITLE>OpenMath példa</TITLE>
  </HEAD>

  <BODY>
    <H1 align="center">OpenMath példa</H1>
    <P>A másodfokú egyenletek megoldóképlete:&nbsp;  

    <OMOBJ>
      <OMA>
        <OMS cd = "relation" name="eq"/>
        <OMV name="x"/>
        <OMA>
          <OMS cd="arith" name="divide"/>
          <OMA>
            <OMS cd="arith" name="plusminus"/>
            <OMA>
              <OMS cd = "arith" name="unary_minus"/>
              <OMV name="b"/>
            </OMA>
          </OMA>
          <OMS cd="arith" name="root"/>
          <OMA>
            <OMS cd = "arith" name="minus"/>
            <OMA>
              <OMS cd="arith" name="power"/>
              <OMV name="b"/>
              <OMI>2</OMI>
            </OMA>
          </OMA>
          <OMS cd = "arith" name="times"/>
          <OMA>
            <OMS cd = "arith" name="times"/>
            <OMI>4</OMI>
            <OMV name="a"/>
          </OMA>
          <OMV name="c"/>
        </OMA>
      </OMA>
    </OMOBJ>
  </BODY>
</HTML>
```

```
<OMA>
  <OMS cd = "arith" name="times"/>
  <OMI>2</OMI>
  <OMV name="a"/>
</OMA>
</OMA>
</OMA>
</OMOBJ>

</P>
</BODY>

</HTML>
```

### **4.3. A T<sub>E</sub>X kapcsolatai**

Napjainkban sajnos nincsen arra mód, hogy T<sub>E</sub>X-ben megírt szövegeinket megjeleníthessük az Interneten; jelenleg lehetőségeink kimerülnek abban, hogy a T<sub>E</sub>X kódot egyszerűen megjelenítsük.

Létezik ugyan néhány program, amely a T<sub>E</sub>X-ben írt szöveget átkonvertálja HTML-be, de a konvertálás során - a HTML hiányosságai miatt - rengeteg olyan információ elveszik, amely miatt az eredmény korántsem egyezik meg a kiindulási szöveggel. Ennek a módszernek egyetlen előnye az, hogy bármilyen standard böngészőben megjeleníthető, ami a MathML és OpenMath kódú dokumentumokról nem mondható el.

## **IV. RÉSZ: JÖVŐKÉP**

### **1. A MathML jövője**

Mint láttuk, a MathML jelenleg még gyerekcipőben jár. Számos bővítésre és egyes elemeknél felülvizsgálatra lesz szükség. Az egyes bővítések könnyen megjósolhatók (pl. az interfész kibővítése, a különböző leírónyelvekbe való beágyazhatóság, és azok beágyazása a MathML-be, stb.), és nagy hatással lesznek a nyelv további fejlődésére.

Szükséges lesz továbbfejleszteni az olyan eszközöket, mint a makrók és a stíluslapokkal való jobb együttműködés, és talán érdemes lenne a jövőben kiterjeszteni a nyelvet a diagramok irányába is.

Természetesen ehhez arra is szükség lesz, hogy a felhasználók körében egyre ismertebbé váljon ez az eszköz, és hogy a szoftverfejlesztők is felismerjék ennek hasznosságát, és ezután a szükséges szoftverekben (szerkesztőkben, böngészőkben, valamint más matematikával kapcsolatos alkalmazásokban) lehetőséget adjanak a MathML felhasználására is.

#### **1.1. Stíluslapok**

A W3C egyik folyamatban lévő tevékenysége a stíluslap-szerkezetek definiálása. Nagy vonalakban már most megfelel a stíluslapok definíciója a MathML számára, de csak a későbbiekben, a mindennapi használat során derülhet ki, hogy mégis mi hiányozhat ebből a definícióból, amely könnyebbé tehetné a MathML szerzők munkáját.

## 1.2. Makrók

A makrók igen fontos és hasznos szerepet játszanak a matematikai tartalom és jelentés kódolásában. Ezen felül igen nehéz egy következetes, jól használható makró-rendszert megalkotni a MathML számára, mert már számos különböző, MathML makrókon alapuló alkalmazás létezik. Emiatt a W3C matematikai munkacsoportja kifejezetten a MathML céljainak megfelelően kívánja tanulmányozni a makró-mechanizmus definícióját, közreműködve a folyamatban lévő általános XML stíluslap- és makró-specifikáció elkészítésében.

A MathML makróinak néhány lehetséges felhasználási területe:

- **Rövidítések:** Az egyik általánosan ismert használata a makróknak a rövidítés. A szerzőknek szükségük lehet néhány bonyolult, de állandó jelölést ismételgetni, melyeket makróként definiálhatnak. A későbbiekben a makrók paraméterek helyettesítésére is alkalmasak lehetnek.
- **A tartalmi jelölés kiszélesítése:** Szemantikai objektumok (pl. binomiális együttható) makróval történő definiálásával az ember képes lenne kiterjeszteni a MathML tartalmi jelölését, ezáltal könnyebbé válhatna az adatcsere egy külső alkalmazással. Lehetővé válna a tartalmi jelölés standardizálása egységes makró-csomagok használatával (mint az *OpenMath* CD-i).
- **Megjelenítés és stíluskontroll:** Egy másik fontos út a makrók használata a stílusok és a megjelenítés ellenőrzésére magas szintű makró-definíciók segítségével. Nagyon fontos a megjelenítés ellenőrzése a HTML-be ágyazott matematikai tagoknál egy tartalomra érzékeny környezetben. Hasonlóan fontos lehet a makrózás a felhasználó által definiált XML kiterjesztések MathML környezethez való kapcsolásakor.

- **Hozzáférhetőség:** Az olvasó által felügyelt stíluslapok fontosak a MathML-hez való hozzáférés biztosításában. Például, hangos megjelenítő esetén egy stíluslap segítségével hozzárendelhetünk egy adott szöveget bármely elemhez, így érthetőbbé téve a matematikai kifejezést, a jelek egymás utáni felolvasása helyett.

### 1.3. Egyenletrendszerek illesztése

A megjelenítő elemeknél már említettem, hogy jelenleg a MathML egyik nagy hiányossága, hogy nincs benne definiálva egy `<malignscope>` elem, mely megkönnyítené az egyenletrendszerek leírását. Fontos volna tehát ezt az elemet a következő verzióba beépíteni, mert a fenti szerkezet `<mtable>`-lél való megjelenítése egy kicsit nehézkes.

### 1.4. Az interfész hiányosságai

Az 1.0-s változatban azonban a legsúlyosabb hiányosság a más leírónyelvekkel való ötvözés igen korlátozott lehetősége. Kicsit nehézkesen lehet kapcsolatokat készíteni MathML elemeinkhez, mert nem használhatjuk a HTML nyelv `<A>` elemét. Egy ehhez hasonló lehetőséget biztosítani kellene a következő verzióban, amire már történtek is lépések, vázlat szintjén elkészült az *XML Linking Language (XLink)*<sup>6</sup>, mellyel lehetőség fog nyílni egyes XML elemekhez kapcsolatokat készíteni.

Képek, ábrák megjelenítésére egyáltalán nincs lehetőségünk, pedig ezt igen jól ki lehetne használni, gondoljunk például a halmazelmélet diagramjaira, vagy az egyes függvények képeire, de más matematikai területen is alkalmazni lehetne (pl. a geometriában). Szükség lenne tehát, vagy egy olyan elemre,

---

<sup>6</sup> Lásd: <http://www.w3.org/1999/07/WD-xlink-19990726>

amely a HTML `<IMG>` eleméhez hasonlít, vagy definiálni kellene olyan saját MathML elemeket, amelyek a fenti hiányosságokat kiküszöbölnék.

## **2. Az OpenMath jövője**

### **2.1. A Content Dictionary előnye**

A MathML-lel szemben az OpenMath kétségtelenül legnagyobb előnye a Content Dictionary megléte. Ezek segítségével könnyen és egyszerűen bővíthetők a nyelv funkciói. Természetesen, ezt a bővítést csak szervezeten, felügyelet mellett – mely az OpenMath Consortium kezében van jelenleg – érdemes tenni, hiszen csak így marad meg széles körben használhatónak a nyelv.

### **2.2. Hátrányok**

Mindemellett – véleményem szerint – a látszólag egyszerű, kevés elemet tartalmazó OpenMath nyelv bonyolulttá válik, amint komolyabb, összetettebb kifejezéseket kívánunk leírni. Egyetlen produktív elem van a rendszerben, az **<OMS>**, ezzel kell megoldani a matematikai műveletek nagy részét.

Számos, a MathML-ben igen egyszerűen megoldható funkciót (pl. indexelés, egyenletrendszerek illesztése, feltételek meghatározása) OpenMath-ban nehezen megvalósíthatók, sok munkát igényelnek.

A másik nagy hiányossága a nyelvnek, hogy nincs lehetőségünk stíluslapok használatára. Igaz, hogy a stílusok főképp a megjelenítésre vannak hatással, de a tartalmi jelölésnél is jól használhatóak lennének, pl. különböző attribútumok megadására – ezzel is könnyítve a nyelv bonyolultságán.

Az OpenMath-nál is igen korlátozottak a lehetőségeink más leírónyelvekkel való kapcsolattartás lehetőségeiben. Az **<OMOBJ>** interfész elemen kívül (ami alapjában véve csak jelzi, hogy más nyelvű szöveg kezdődik), nincs több olyan elem, amellyel más dokumentumokkal kapcsolatba tudnánk lépni. Akárcsak a MathML-ben, itt sincs lehetőségünk képeket, ábrákat beágyazni OpenMath dokumentumainkba.

### 3. Összegzés

A fent leírtakat tudomásul véve elmondhatjuk, hogy 1.0 verzió létre a MathML és az OpenMath igen hasznos segédeszköz lehet a matematikus társadalom mindennapi munkájában. Remélhetőleg minél többen megismerik majd ezeket a leírónyelvet, így több forrásból nyerhető vissza információ a hiányosságok, hibák kiküszöböléséhez.

Nagy valószínűséggel a  $\text{T}_{\text{E}}\text{X}$  területén nagy változások nem fognak történni. A programrendszer igen kiforrott, mind a normál szövegek, mind a matematikai kifejezések szerkesztésének terén. A Windows operációs rendszerek elterjedésével talán érdemes lenne készíteni egy grafikus felületű, teljes értékű  $\text{T}_{\text{E}}\text{X}$  programot, melyben a szövegszerkesztésen kívül lehetséges lenne konvertálni dokumentumainkat, akár  $\text{T}_{\text{E}}\text{X}$ -ről, akár  $\text{T}_{\text{E}}\text{X}$ -re. Bár lehet, hogy az is elégséges lenne, ha a ma ismert és elterjedt szövegszerkesztő programokban lehetőségünk lenne  $\text{T}_{\text{E}}\text{X}$  dokumentumok olvasására, szerkesztésére és mentésére.

Az egyre ismertebbé, hétköznapibbá váló Internet fejlődésével együtt jár egy olyan matematikai leírónyelv megismerése is, mely könnyen kapcsolható a már ismerté vált HTML nyelvhez. Jelenlegi állapotában a  $\text{T}_{\text{E}}\text{X}$  erre nem alkalmas, ezért nagy a valószínűsége, hogy a  $\text{T}_{\text{E}}\text{X}$  megmarad egy belső (egyetemi, kutatóintézeti, stb.)

kommunikációs eszköznek, nagyobb nemzeti, vagy nemzetközi elektronikus publikáció terén – főleg az Interneten – az OpenMath, a MathML, vagy a hozzájuk hasonló XML-alapú leírónyelv ki fogja szorítani.

Természetesen idő kell ahhoz, hogy a két nyelv igazán használhatóvá váljon. A későbbi verziókban remélhetőleg megszűnnek a jelenlegi hiányosságok, így akár mindkét nyelv teljes értékűvé válhat.

Véleményem szerint az OpenMath és a MathML közül az utóbbi könnyebben el fog tudni terjedni, mivel szerkezete egyszerűbb társáénál, bonyolultabb kifejezéseket könnyebb készíteni benne, mint OpenMath-ban. Az Interneten egyre több olyan programot – böngészőt, szerkesztőt, vagy ezekhez tartozó plug-in-eket – lehet találni, melyek szintén elősegítik a MathML terjedését. Mindezek mellett az sem elhanyagolható, hogy létrehozója az a World Wide Web Consortium, amely a már világszerte ismerté vált HTML-t is szabvánnyá tette. Az OpenMath talán Európában tehet szert előnyre, mivel az Európai Unió szabvánnyá nyilvánította, ezért ebben a térségben könnyebben elterjedhet.

Remélhetőleg ez utóbbi két nyelv következő verziója már híven fogja tükrözni a mai matematikus társadalom igényeit, rugalmasabbá válik, hogy minél nagyobb területet fedhessen le a modern matematika szerteágazó világában.

Ezért érdemes elgondolkodni, hogy nincs-e már itt az ideje, hogy komolyabban is megismerkedjünk a fenti két nyelv valamelyikével, esetleg mindkettővel.

## **A FÜGGELÉK: SPECIÁLIS MATEMATIKAI JELEK**

### **1. Általában a speciális jelekről**

A matematika jelölőrendszere számos olyan jelet használ, mely más területen nem ismert, ezért ezeknek a jeleknek a megjelenítése igen körülményes. Másrészt viszont szükség van rájuk, hiszen nélkülük nem tudnánk megjeleníteni olyan jeleket, amelyeket akár minden nap használunk. A MathML-ben az ilyen jelek kódolására találták ki azokat az escape-szekvenciákat, melyekkel a bonyolult jeleket is szövegesen írhatunk le. A speciális jelek alakja minden esetben a következő: `&név;`. Ezzel a módszerrel könnyebbé válik a szerzők munkája, hiszen nem kell azzal foglalkozniuk, hogy milyen módszerekkel jelenítsék meg az egyes matematikai jeleket, ugyanakkor az olvasók helyzetét is megkönnyíti, hiszen egy szöveget könnyebb értelmezni, mint például egy grafikát, nem beszélve arról, hogy a hangos megjelenítésnél sokkal előnyösebb az így tárolt információ.

A matematikai speciális jelek leírásának alapjául az ISO TR 9573-as és TR 8879-es szabványa szolgált alapul, melyek a matematikai jelölésekre vonatkoznak.

OpenMath-ban nincsenek ilyen escape-szekvenciák, ott a CD-kben kell leírni az egyes jeleket.

Összehasonlításképpen – ahol van értelem – megadom a megfelelő  $\text{T}_{\text{E}}\text{X}$  parancsokat is.

## 2. A speciális jelek részletesen

### 2.1. MathML specifikus jelek

Néhány speciális jel nem kapcsolódik közvetlenül a matematikához, de mivel egy leíró nyelvről van szó, ezekre is szükség van a pontos, jó minőségű megjelenítéshez.

<code>&amp;Tab;</code>	tabulátor
<code>&amp;NewLine;</code>	új sor
<code>&amp;IndentingNewLine;</code>	új bekezdés
<code>&amp;NoBreak;</code>	az adott helyen nem lehet sortörés
<code>&amp;GoodBreak;</code>	ha sortörésre van szükség, ez a megfelelő hely
<code>&amp;BadBreak;</code>	ha nem szükséges, ne itt legyen sortörés
<code>&amp;Space;</code>	szóköz (1 em)
<code>&amp;NonBreakingSpace;</code>	olyan szóköz, ahol nem lehet sortörés
<code>&amp;ZeroWidthSpace;</code>	nulla szélességű szóköz
<code>&amp;VeryThinSpace;</code>	nagyon keskeny szóköz (1/18 em)
<code>&amp;ThinSpace;</code>	keskeny szóköz (3/18 em)
<code>&amp;MediumSpace;</code>	közepes szóköz (4/18 em)
<code>&amp;ThickSpace;</code>	vastag szóköz (5/18 em)
<code>&amp;NegativeVeryThinSpace;</code>	nagyon keskeny szóköz (-1/18 em)
<code>&amp;NegativeThinSpace;</code>	keskeny szóköz (-3/18 em)
<code>&amp;NegativeMediumSpace;</code>	közepes szóköz (-4/18 em)
<code>&amp;NegativeThickSpace;</code>	vastag szóköz (-5/18 em)
<code>&amp;InvisibleComma;</code>	láthatatlan vessző (indexeknél használatos)
<code>&amp;ic;</code>	az <code>&amp;InvisibleComma;</code> rövid alakja
<code>&amp;InvisibleTimes;</code>	láthatatlan szorzásjel (változók között)
<code>&amp;it;</code>	az <code>&amp;InvisibleTimes;</code> rövid alakja
<code>&amp;ApplyFunction;</code>	jelzi, hogy függvényről van szó
<code>&amp;af;</code>	az <code>&amp;ApplyFunction;</code> rövid neve
<code>&amp;CapitalDifferentialD;</code>	differenciál D
<code>&amp;DD;</code>	a <code>&amp;CapitalDifferentialD;</code> rövid neve
<code>&amp;DifferentialD;</code>	differenciál d
<code>&amp;dd;</code>	a <code>&amp;DifferentialD;</code> rövid neve
<code>&amp;ExponentialE;</code>	exponenciális e

<code>&amp;ee;</code>	az <code>&amp;ExponentialE;</code> rövid neve
<code>&amp;ImaginaryI;</code>	imaginárius $i$
<code>&amp;ii;</code>	az <code>&amp;ImaginaryI;</code> rövid neve
<code>&amp;NotANumber;</code>	nem szám
<code>&amp;&gt;false;</code>	logikai „hamis”
<code>&amp;&gt;true;</code>	logikai „igaz”

## 2.3. Betűk

A következő oldalakon a matematikában használatos speciális betűsorokat mutatom be. Első oszlopban a betű képe, másodikban a MathML formát, a harmadikban pedig a T<sub>E</sub>X parancs található.

### 2.3.1. Görög betűk

$\alpha$	<code>&amp;alpha;</code>	<code>\alpha</code>	$\phi$	<code>&amp;phi;</code>	<code>\phi</code>
$\beta$	<code>&amp;beta;</code>	<code>\beta</code>	$\chi$	<code>&amp;chi;</code>	<code>\chi</code>
$\gamma$	<code>&amp;gamma;</code>	<code>\gamma</code>	$\psi$	<code>&amp;psi;</code>	<code>\psi</code>
$\delta$	<code>&amp;delta;</code>	<code>\delta</code>	$\omega$	<code>&amp;omega;</code>	<code>\omega</code>
$\varepsilon$	<code>&amp;epsiv;</code>	<code>\varepsilon</code>	$\Gamma$	<code>&amp;Gamma;</code>	<code>\Gamma</code>
$\epsilon$	<code>&amp;epsi;</code>	<code>\epsilon</code>	$\Delta$	<code>&amp;Delta;</code>	<code>\Delta</code>
$\zeta$	<code>&amp;zeta;</code>	<code>\zeta</code>	$\Theta$	<code>&amp;Theta;</code>	<code>\Theta</code>
$\eta$	<code>&amp;eta;</code>	<code>\eta</code>	$\Lambda$	<code>&amp;Lambda;</code>	<code>\Lambda</code>
$\theta$	<code>&amp;theta;</code>	<code>\theta</code>	$\Xi$	<code>&amp;Xi;</code>	<code>\Xi</code>
$\vartheta$	<code>&amp;thetav;</code>	<code>\vartheta</code>	$\Pi$	<code>&amp;Pi;</code>	<code>\Pi</code>
$\iota$	<code>&amp;iota;</code>	<code>\iota</code>	$\Sigma$	<code>&amp;Sigma;</code>	<code>\Sigma</code>
$\kappa$	<code>&amp;kappa;</code>	<code>\kappa</code>	$\Upsilon$	<code>&amp;Upsilon;</code>	<code>\Upsilon</code>
$\lambda$	<code>&amp;lambda;</code>	<code>\lambda</code>	$\Phi$	<code>&amp;Phi;</code>	<code>\Phi</code>
$\mu$	<code>&amp;mu;</code>	<code>\mu</code>	$\Psi$	<code>&amp;Psi;</code>	<code>\Psi</code>
$\nu$	<code>&amp;nu;</code>	<code>\nu</code>	$\Omega$	<code>&amp;Omega;</code>	<code>\Omega</code>
$\xi$	<code>&amp;xi;</code>	<code>\xi</code>	$\Gamma$	-	<code>\varGamma</code>
$\pi$	<code>&amp;pi;</code>	<code>\pi</code>	$\Delta$	-	<code>\varDelta</code>
$\varpi$	<code>&amp;piv;</code>	<code>\varpi</code>	$\Theta$	-	<code>\varTheta</code>
$\rho$	<code>&amp;rho;</code>	<code>\rho</code>	$\Lambda$	-	<code>\varLambda</code>
$\sigma$	<code>&amp;sigma;</code>	<code>\sigma</code>	$\Xi$	-	<code>\varXi</code>
$\varsigma$	<code>&amp;sigmav;</code>	<code>\varsigma</code>	$\Pi$	-	<code>\varPi</code>
$\tau$	<code>&amp;tau;</code>	<code>\tau</code>	$\Sigma$	-	<code>\varSigma</code>
$\upsilon$	<code>&amp;upsilon;</code>	<code>\upsilon</code>	$\Upsilon$	-	<code>\varUpsilon</code>

$\Phi$	-	<code>\varPhi</code>	$\Omega$	-	<code>\varOmega</code>
$\Psi$	-	<code>\varPsi</code>			

### 2.3.2. Gót betűk

$\mathbb{A}$	<code>&amp;Afr;</code>	<code>\frak A</code>	$\mathbb{Z}$	<code>&amp;Zfr;</code>	<code>\frak Z</code>
$\mathbb{B}$	<code>&amp;Bfr;</code>	<code>\frak B</code>	$\mathfrak{a}$	<code>&amp;afr;</code>	<code>\frak a</code>
$\mathbb{C}$	<code>&amp;Cfr;</code>	<code>\frak C</code>	$\mathfrak{b}$	<code>&amp;bfr;</code>	<code>\frak b</code>
$\mathbb{D}$	<code>&amp;Dfr;</code>	<code>\frak D</code>	$\mathfrak{c}$	<code>&amp;cfr;</code>	<code>\frak c</code>
$\mathbb{E}$	<code>&amp;Efr;</code>	<code>\frak E</code>	$\mathfrak{d}$	<code>&amp;dfr;</code>	<code>\frak d</code>
$\mathbb{F}$	<code>&amp;Ffr;</code>	<code>\frak F</code>	$\mathfrak{e}$	<code>&amp;efr;</code>	<code>\frak e</code>
$\mathbb{G}$	<code>&amp;Gfr;</code>	<code>\frak G</code>	$\mathfrak{f}$	<code>&amp;ffr;</code>	<code>\frak f</code>
$\mathbb{H}$	<code>&amp;Hfr;</code>	<code>\frak H</code>	$\mathfrak{g}$	<code>&amp;gfr;</code>	<code>\frak g</code>
$\mathbb{I}$	<code>&amp;Ifr;</code>	<code>\frak I</code>	$\mathfrak{h}$	<code>&amp;hfr;</code>	<code>\frak h</code>
$\mathbb{J}$	<code>&amp;Jfr;</code>	<code>\frak J</code>	$\mathfrak{i}$	<code>&amp;ifr;</code>	<code>\frak i</code>
$\mathbb{K}$	<code>&amp;Kfr;</code>	<code>\frak K</code>	$\mathfrak{j}$	<code>&amp;jfr;</code>	<code>\frak j</code>
$\mathbb{L}$	<code>&amp;Lfr;</code>	<code>\frak L</code>	$\mathfrak{k}$	<code>&amp;kfr;</code>	<code>\frak k</code>
$\mathbb{M}$	<code>&amp;Mfr;</code>	<code>\frak M</code>	$\mathfrak{l}$	<code>&amp;lfr;</code>	<code>\frak l</code>
$\mathbb{N}$	<code>&amp;Nfr;</code>	<code>\frak N</code>	$\mathfrak{m}$	<code>&amp;mfr;</code>	<code>\frak m</code>
$\mathbb{O}$	<code>&amp;Ofr;</code>	<code>\frak O</code>	$\mathfrak{n}$	<code>&amp;nfr;</code>	<code>\frak n</code>
$\mathbb{P}$	<code>&amp;Pfr;</code>	<code>\frak P</code>	$\mathfrak{o}$	<code>&amp;ofr;</code>	<code>\frak o</code>
$\mathbb{Q}$	<code>&amp;Qfr;</code>	<code>\frak Q</code>	$\mathfrak{p}$	<code>&amp;pfr;</code>	<code>\frak p</code>
$\mathbb{R}$	<code>&amp;Rfr;</code>	<code>\frak R</code>	$\mathfrak{q}$	<code>&amp;qfr;</code>	<code>\frak q</code>
$\mathbb{S}$	<code>&amp;Sfr;</code>	<code>\frak S</code>	$\mathfrak{r}$	<code>&amp;rfr;</code>	<code>\frak r</code>
$\mathbb{T}$	<code>&amp;Tfr;</code>	<code>\frak T</code>	$\mathfrak{s}$	<code>&amp;sfr;</code>	<code>\frak s</code>
$\mathbb{U}$	<code>&amp;Ufr;</code>	<code>\frak U</code>	$\mathfrak{t}$	<code>&amp;tfr;</code>	<code>\frak t</code>
$\mathbb{V}$	<code>&amp;Vfr;</code>	<code>\frak V</code>	$\mathfrak{u}$	<code>&amp;ufr;</code>	<code>\frak u</code>
$\mathbb{W}$	<code>&amp;Wfr;</code>	<code>\frak W</code>	$\mathfrak{v}$	<code>&amp;vfr;</code>	<code>\frak v</code>
$\mathbb{X}$	<code>&amp;Xfr;</code>	<code>\frak X</code>	$\mathfrak{w}$	<code>&amp;wfr;</code>	<code>\frak w</code>
$\mathbb{Y}$	<code>&amp;Yfr;</code>	<code>\frak Y</code>	$\mathfrak{x}$	<code>&amp;xfr;</code>	<code>\frak x</code>

$y$	<code>&amp;yfr;</code>	<code>\frak y</code>	$z$	<code>&amp;zfr;</code>	<code>\frak z</code>
-----	------------------------	----------------------	-----	------------------------	----------------------

### 2.3.3. Duplaszárú betűk

$\mathbb{A}$	<code>&amp;Aopf;</code>	<code>\Bbb A</code>	$\mathbb{N}$	<code>&amp;Nopf;</code>	<code>\Bbb N</code>
$\mathbb{B}$	<code>&amp;Bopf;</code>	<code>\Bbb B</code>	$\mathbb{O}$	<code>&amp;Oopf;</code>	<code>\Bbb O</code>
$\mathbb{C}$	<code>&amp;Copf;</code>	<code>\Bbb C</code>	$\mathbb{P}$	<code>&amp;Popf;</code>	<code>\Bbb P</code>
$\mathbb{D}$	<code>&amp;Dopf;</code>	<code>\Bbb D</code>	$\mathbb{Q}$	<code>&amp;Qopf;</code>	<code>\Bbb Q</code>
$\mathbb{E}$	<code>&amp;Eopf;</code>	<code>\Bbb E</code>	$\mathbb{R}$	<code>&amp;Ropf;</code>	<code>\Bbb R</code>
$\mathbb{F}$	<code>&amp;Fopf;</code>	<code>\Bbb F</code>	$\mathbb{S}$	<code>&amp;Sopf;</code>	<code>\Bbb S</code>
$\mathbb{G}$	<code>&amp;Gopf;</code>	<code>\Bbb G</code>	$\mathbb{T}$	<code>&amp;Topf;</code>	<code>\Bbb T</code>
$\mathbb{H}$	<code>&amp;Hopf;</code>	<code>\Bbb H</code>	$\mathbb{U}$	<code>&amp;Uopf;</code>	<code>\Bbb U</code>
$\mathbb{I}$	<code>&amp;Iopf;</code>	<code>\Bbb I</code>	$\mathbb{V}$	<code>&amp;Vopf;</code>	<code>\Bbb V</code>
$\mathbb{J}$	<code>&amp;Jopf;</code>	<code>\Bbb J</code>	$\mathbb{W}$	<code>&amp;Wopf;</code>	<code>\Bbb W</code>
$\mathbb{K}$	<code>&amp;Kopf;</code>	<code>\Bbb K</code>	$\mathbb{X}$	<code>&amp;Xopf;</code>	<code>\Bbb X</code>
$\mathbb{L}$	<code>&amp;Lopf;</code>	<code>\Bbb L</code>	$\mathbb{Y}$	<code>&amp;Yopf;</code>	<code>\Bbb Y</code>
$\mathbb{M}$	<code>&amp;Mopf;</code>	<code>\Bbb M</code>	$\mathbb{Z}$	<code>&amp;Zopf;</code>	<code>\Bbb Z</code>

### 2.3.4. Kalligrafikus betűk

$\mathcal{A}$	<code>&amp;Ascr;</code>	<code>\Cal A</code>	$\mathcal{L}$	<code>&amp;Lscr;</code>	<code>\Cal L</code>
$\mathcal{B}$	<code>&amp;Bscr;</code>	<code>\Cal B</code>	$\mathcal{M}$	<code>&amp;Mscr;</code>	<code>\Cal M</code>
$\mathcal{C}$	<code>&amp;Cscr;</code>	<code>\Cal C</code>	$\mathcal{N}$	<code>&amp;Nscr;</code>	<code>\Cal N</code>
$\mathcal{D}$	<code>&amp;Dscr;</code>	<code>\Cal D</code>	$\mathcal{O}$	<code>&amp;Oscr;</code>	<code>\Cal O</code>
$\mathcal{E}$	<code>&amp;Escr;</code>	<code>\Cal E</code>	$\mathcal{P}$	<code>&amp;Pscr;</code>	<code>\Cal P</code>
$\mathcal{F}$	<code>&amp;Fscr;</code>	<code>\Cal F</code>	$\mathcal{Q}$	<code>&amp;Qscr;</code>	<code>\Cal Q</code>
$\mathcal{G}$	<code>&amp;Gscr;</code>	<code>\Cal G</code>	$\mathcal{R}$	<code>&amp;Rscr;</code>	<code>\Cal R</code>
$\mathcal{H}$	<code>&amp;Hscr;</code>	<code>\Cal H</code>	$\mathcal{S}$	<code>&amp;Sscr;</code>	<code>\Cal S</code>
$\mathcal{I}$	<code>&amp;Iscr;</code>	<code>\Cal I</code>	$\mathcal{T}$	<code>&amp;Tscr;</code>	<code>\Cal T</code>
$\mathcal{J}$	<code>&amp;Jscr;</code>	<code>\Cal J</code>	$\mathcal{U}$	<code>&amp;Uscr;</code>	<code>\Cal U</code>
$\mathcal{K}$	<code>&amp;Kscr;</code>	<code>\Cal K</code>	$\mathcal{V}$	<code>&amp;Vscr;</code>	<code>\Cal V</code>

$\mathcal{W}$	&wscr;	\cal w	$\mathcal{I}$	&lscr;	-
$\mathcal{X}$	&xscr;	\cal x	$\mathcal{M}$	&mscr;	-
$\mathcal{Y}$	&yscr;	\cal y	$\mathcal{N}$	&nscr;	-
$\mathcal{Z}$	&zscr;	\cal z	$\mathcal{O}$	&oscr;	-
$\mathcal{a}$	&ascr;	-	$\mathcal{P}$	&pscr;	-
$\mathcal{b}$	&bscr;	-	$\mathcal{Q}$	&qscr;	-
$\mathcal{c}$	&cscr;	-	$\mathcal{R}$	&rscr;	-
$\mathcal{d}$	&dscr;	-	$\mathcal{S}$	&sscr;	-
$\mathcal{e}$	&escr;	-	$\mathcal{T}$	&tscr;	-
$\mathcal{f}$	&fscr;	-	$\mathcal{U}$	&uscr;	-
$\mathcal{g}$	&gscr;	-	$\mathcal{V}$	&vscr;	-
$\mathcal{h}$	&hscr;	-	$\mathcal{W}$	&wscr;	-
$\mathcal{i}$	&iscr;	-	$\mathcal{X}$	&xscr;	-
$\mathcal{j}$	&jscr;	-	$\mathcal{Y}$	&yscr;	-
$\mathcal{k}$	&kscr;	-	$\mathcal{Z}$	&zscr;	-

## 2.4. ASCII karakterek















!	&excl	!	.	&period	.
"	&quot	"	/	&sol	/
#	&num	#	:	&colon	:
\$	&dollar	\$	;	&semi	;
%	&percnt	%	<	&lt	<
&	&amp	&	=	&equals	=
'	&apos	'	>	&gt	>
(	&lpar	(	?	&quest	?
)	&rpar	)	@	&commat	@
+	&plus	+	[	&lbrack	[
,	&comma	,		&lqb	\lbrack
			\	&bso	\backslash

]	&rbrack	]	°	&deg	\circ
	&rsqb	\rbrack	±	&PlusMinus	\pm
—	&lowbar	—		&plusmn	
{	&lbrace	{		&pm	
	&lcub	\lbrace	²	&sup2	^2
	&verbar		³	&sup3	^3
	&vert	\vert	μ	&micro	\mu
}	&rbrace	}	¶	&para	\P
	&rcub	\rbrace	·	&CenterDot	\cdot
	&nbsp	-		&centerdot	
¡	&iexcl	!		&middot	
¢	&cent	-	¹	&sup1	^1
£	&pound	-	º	&ordm	^o
¤	&curren	-	»	&raquo	-
¥	&yen	\yen	¼	&frac14	-
¦	&brvbar		½	&frac12	-
§	&sect	\S		&half	
©	&copy	\copy	¾	&frac34	-
ª	&ordf	^a	¿	&iquest	-
«	&laquo	-	×	&times	\times
¬	&not	\not	÷	&div	\div
-	&shy	-		&divide	
®	&circledR	\circledR			
	&reg				












## 2.5. Matematikai ékezetek






























A következő táblázatban a nagybetűvel kezdődő T<sub>E</sub>X parancsok nagybetűre vonatkozó ékezetek.













◌̂	&Diacritica	\grave	◌̃	&Diacritica	\acute
	\lgrave;	\Grave		\lacute;	\Acute
	&grave;			&acute;	















































	<code>&amp;Hat;</code>	<code>\hat</code>		<code>&amp;DotDot;</code>	-
		<code>\Hat</code>		<code>&amp;ring;</code>	-
	<code>&amp;Diacritica</code> <code>lTilde;</code>	<code>\tilde</code> <code>\Tilde</code>		<code>&amp;Diacritica</code> <code>lDoubleAcute;</code>	<code>\H</code>
	<code>&amp;tilde;</code>			<code>&amp;dblac;</code>	
	<code>&amp;OverLine</code>	<code>\bar</code>		<code>&amp;Caron;</code>	<code>\check</code>
	<code>&amp;macr;</code>	<code>\Bar</code>		<code>&amp;Hacek;</code>	<code>\Check</code>
	<code>&amp;Breve;</code>	<code>\breve</code>		<code>&amp;Cedil;</code>	<code>\c</code>
		<code>\Breve</code>		<code>&amp;Cedilla;</code>	
	<code>&amp;Diacritica</code> <code>lDot;</code>	<code>\dot</code> <code>\Dot</code>		<code>&amp;ogon;</code>	-
	<code>&amp;Dot;</code>			<code>&amp;UnderLine;</code>	<code>\b</code>
	<code>&amp;Die;</code>	<code>\ddot</code>			
	<code>&amp;DoubleDot;</code>	<code>\Ddot</code>			
	<code>&amp;Uml;</code>				
	<code>&amp;TripleDot;</code>	<code>\ddd</code>			
	<code>&amp;tdot;</code>				

## 2.6. Nyilak

	<code>&amp;larr;</code>	<code>\leftarrow</code>	<code>&amp;updownarrow;</code>	
	<code>&amp;LeftArrow;</code>		<code>&amp;varr;</code>	
	<code>&amp;leftarrow;</code>			<code>&amp;nwarr;</code> <code>\nwarrow</code>
	<code>&amp;uarr;</code>	<code>\uparrow</code>		<code>&amp;nwarrow;</code>
	<code>&amp;UpArrow;</code>			<code>&amp;UpperLeftArrow;</code>
	<code>&amp;rarr;</code>	<code>\rightarrow</code>		<code>&amp;nearr;</code> <code>\nearrow</code>
	<code>&amp;RightArrow;</code>	<code>\to</code>		<code>&amp;nearrow;</code>
	<code>&amp;rightarrow;</code>			<code>&amp;UpperRightArrow;</code>
	<code>&amp;darr;</code>	<code>\downarrow</code>		<code>&amp;LowerRightArrow;</code> <code>\searrow</code>
	<code>&amp;DownArrow;</code>			<code>&amp;searr;</code>
	<code>&amp;downarrow;</code>			<code>&amp;searrow;</code>
	<code>&amp;harr;</code>	<code>\leftrightarrow</code>		<code>&amp;LowerLeftArrow;</code> <code>\swarrow</code>
	<code>&amp;LeftRightArrow;</code>			<code>&amp;swarr;</code>
	<code>&amp;leftrightarrow;</code>			<code>&amp;swarrow;</code>
	<code>&amp;UpDownArrow;</code>	<code>\updownarrow</code>		<code>&amp;nlarr;</code> <code>\nleftarrow</code>
				<code>&amp;nleftarrow;</code>

	<code>&amp;nrarr;</code> <code>&amp;nrigh tarro w;</code>	<code>\nrightarrow</code>		<code>&amp;Lsh;</code> <code>&amp;lsh;</code>	<code>\Lsh</code>
	<code>&amp;rarrw;</code> <code>&amp;rightsquig arrow;</code>	<code>\rightsquiga rrow</code>		<code>&amp;Rsh;</code> <code>&amp;rsh;</code>	<code>\Rsh</code>
	<code>&amp;Larr;</code> <code>&amp;twoheadlef tarrow;</code>	<code>\twoheadleft arrow</code>		<code>&amp;ldsh;</code> <code>&amp;rdsh;</code>	<code>-</code>
	<code>&amp;Uarr;</code>	<code>-</code>		<code>&amp;cularr;</code> <code>&amp;curvearrow left;</code>	<code>\curvearrowl eft</code>
	<code>&amp;Rarr;</code> <code>&amp;twoheadrig htarrow;</code>	<code>\twoheadrigh tarrow</code>		<code>&amp;curarr;</code> <code>&amp;curvearrow right;</code>	<code>\curvearrowr ight</code>
	<code>&amp;Darr;</code>	<code>-</code>		<code>&amp;cudarr;</code>	<code>-</code>
	<code>&amp;larrtl;</code> <code>&amp;leftarrowt ail;</code>	<code>\leftarrowta il</code>		<code>&amp;circlearro wleft;</code> <code>&amp;olarr;</code>	<code>\circlearrow left</code>
	<code>&amp;rarrtl;</code> <code>&amp;ratail;</code> <code>&amp;rightarrow tail;</code>	<code>\rightarrowt ail</code>		<code>&amp;circlearro wright;</code> <code>&amp;orarr;</code>	<code>\circlearrow right</code>
	<code>&amp;map;</code> <code>&amp;mapsto;</code> <code>&amp;RightTeeAr row;</code>	<code>\mapsto</code>		<code>&amp;leftharpo onup;</code> <code>&amp;LeftVector ;</code>	<code>\leftharpoon up</code>
	<code>&amp;hookleftar row;</code> <code>&amp;larrhk;</code>	<code>\hookleftarr ow</code>		<code>&amp;DownLeftVe ctor;</code> <code>&amp;leftharpo on down;</code>	<code>\leftharpoon down</code>
	<code>&amp;hookrightar row;</code> <code>&amp;rarrhk;</code>	<code>\hookrightar row</code>		<code>&amp;lhard;</code> <code>&amp;RightUpVec tor;</code> <code>&amp;uharr;</code>	<code>\upharpoonri ght</code>
	<code>&amp;larrlp;</code> <code>&amp;looparrowl eft;</code>	<code>\looparrowle ft</code>		<code>&amp;upharpoon right;</code>	
	<code>&amp;looparrowr ight;</code> <code>&amp;rarrlp;</code>	<code>\looparrowri ght</code>		<code>&amp;LeftUpVect or;</code> <code>&amp;uharl;</code>	<code>\upharpoonle ft</code>
	<code>&amp;harrw;</code> <code>&amp;leftrights quigarrow;</code>	<code>\leftrightsq uigarrow</code>		<code>&amp;upharpoonl eft;</code> <code>&amp;rharu;</code> <code>&amp;rightharpo onup;</code> <code>&amp;RightVecto r;</code>	<code>\rightharpoon up</code>
	<code>&amp;nharr;</code> <code>&amp;nleftright arrow;</code>	<code>\nleftrighta rrow</code>			

	<code>&amp;DownRightV ector;</code>	<code>\rightharpo ndown</code>		<code>&amp;dharr;</code>	<code>\downharpoon right</code>		<code>&amp;dharl;</code>	<code>\downharpoon left</code>		<code>&amp;RightArrow LeftArrow;</code>	<code>\rightleftar rows</code>		<code>&amp;udarr;</code>	<code>-</code>		<code>&amp;LeftArrowR ightArrow;</code>	<code>\leftrightar rows</code>		<code>&amp;leftleftar rows;</code>	<code>\leftleftarr ows</code>		<code>&amp;upuparrows ;</code>	<code>\upuparrows</code>		<code>&amp;rightright arrows;</code>	<code>\rightarrow</code>		<code>&amp;ddarr;</code>	<code>\downdownarr ows</code>		<code>&amp;leftrighth arpoons;</code>	<code>\leftrightha rpoons</code>		<code>&amp;Equilibriu</code>	<code>\rightleftha</code>																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
	<code>&amp;rhard;</code>			<code>&amp;downharpoon nright;</code>			<code>&amp;downharpoon nleft;</code>			<code>&amp;rightlefta rrows;</code>			<code>&amp;UpArrowDow nArrow;</code>				<code>\leftrightra rows;</code>			<code>&amp;llarr;</code>			<code>&amp;uuarr;</code>			<code>&amp;rrarr;</code>				<code>&amp;ReverseEqu ilibrium;</code>																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						

	<code>&amp;nwArr;</code>	-	<code>&amp;longrightarrow</code>	
	<code>&amp;neArr;</code>	-	<code>&amp;xrarr;</code>	
	<code>&amp;seArr;</code>	-	$\dashleftarrow$	<code>&amp;lBarr;</code> <code>\dashleftarrow</code>
	<code>&amp;swArr;</code>	-	$\dashrightarrow$	<code>&amp;Ac;</code> <code>\dashrightarrow</code>
	<code>&amp;lAarr;</code>	<code>\Lleftarrow</code>	<code>&amp;dbkarow;</code>	
	<code>&amp;rAarr;</code>	<code>\Rrightarrow</code>	<code>&amp;rBarr;</code>	
	<code>&amp;dzigrarr;</code>	-	$\leftarrow$	<code>&amp;lbarr;</code> -
	<code>&amp;origof;</code>	-	$\rightarrow$	<code>&amp;bkarow;</code> -
	<code>&amp;imof;</code>	-	<code>&amp;rbarr;</code>	
	<code>&amp;multimap;</code>	<code>\multimap</code>	$\mapsto$	<code>&amp;longmapsto</code> <code>\longmapsto</code>
	<code>&amp;mumap;</code>	-	<code>&amp;xmap;</code>	
	<code>&amp;DoubleLongLeftArrow;</code>	<code>\Longleftarrow</code>	$\dashrightarrow$	<code>&amp;drbkarow;</code> -
	<code>&amp;Longleftarrow;</code>		<code>&amp;Rbarr;</code>	
	<code>&amp;Longleftarrow;</code>			<code>&amp;hkswarrow;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;swarhk;</code>
	<code>&amp;Longleftarrow;</code>			<code>&amp;hksearrow;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;searhk;</code>
	<code>&amp;Longleftarrow;</code>			<code>&amp;nwarhk;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;nearhk;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;nesear;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;toea;</code>
	<code>&amp;Longleftarrow;</code>			<code>&amp;midcir;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;tosa;</code>
	<code>&amp;Longleftarrow;</code>			<code>&amp;seswar;</code>
	<code>&amp;Longleftarrow;</code>			<code>&amp;swnwar;</code> -
	<code>&amp;Longleftarrow;</code>			<code>&amp;nwnear;</code> -
	<code>&amp;Longleftarrow;</code>		$\mapsto$	<code>&amp;Map;</code> -
	<code>&amp;Longleftarrow;</code>		$\dashleftarrow$	<code>&amp;lfisht;</code> -
	<code>&amp;Longleftarrow;</code>		$\dashrightarrow$	<code>&amp;rfisht;</code> -
	<code>&amp;Longleftarrow;</code>		$\Uparrow$	<code>&amp;DownArrowUpArrow;</code> -
	<code>&amp;Longleftarrow;</code>		$\Downarrow$	<code>&amp;duarr;</code> -
	<code>&amp;Longleftarrow;</code>		$\Updownarrow$	<code>&amp;duhar;</code> <code>\downupharpoons</code>
	<code>&amp;Longleftarrow;</code>		<code>&amp;ReverseUpEquilibrium;</code>	


$\Uparrow$	<code>&amp;udhar;</code>	<code>\updownharpoons</code>	$\rightarrow$	<code>&amp;srarr;</code>	-
	<code>&amp;UpEquilibrium;</code>		$\leftarrow$	<code>&amp;ShortLeftArrow;</code>	-
$\hookrightarrow$	<code>&amp;rdca;</code>	-	$\leftarrow$	<code>&amp;slarr;</code>	-
$\hookleftarrow$	<code>&amp;ldca;</code>	-	$\rightrightarrows$	<code>&amp;simrarr;</code>	-
$\rightsquigarrow$	<code>&amp;nrarrw;</code>	-	$\rightrightarrows$	<code>&amp;rarrap;</code>	-
$\curvearrowright$	<code>&amp;rarrc;</code>	-	$\Rightarrow$	<code>&amp;erarr;</code>	-
$\curvearrowleft$	<code>&amp;nrarrc;</code>	-	$\Uparrow$	<code>&amp;Uarrocir;</code>	-
$\rightarrowtail$	<code>&amp;rarrpl;</code>	-	$\cdots\rightarrow$	<code>&amp;Ddottrahd;</code>	-
$\leftarrowtail$	<code>&amp;larrbfs;</code>	-	$\rightrightarrows$	<code>&amp;Rarrtl;</code>	-
$\rightarrowtail$	<code>&amp;rarrbfs;</code>	-	$\rightarrowtail$	<code>&amp;rAtail;</code>	-
$\leftarrowtail$	<code>&amp;larrfs;</code>	-	$\rightarrowtail$	<code>&amp;latail;</code>	-
$\rightarrowtail$	<code>&amp;rarrfs;</code>	-	$\rightarrowtail$	<code>&amp;lAtail;</code>	-
$\Rightarrow$	<code>&amp;rHar;</code>	-	$\hookrightarrow$	<code>&amp;cudarrl;</code>	-
$\Leftarrow$	<code>&amp;lHar;</code>	-	$\hookrightarrow$	<code>&amp;larrpl;</code>	-
$\Uparrow$	<code>&amp;uHar;</code>	-	$\leftrightarrow$	<code>&amp;harrcir;</code>	-
$\Downarrow$	<code>&amp;dHar;</code>	-	$\rightarrow$	<code>&amp;roarr;</code>	-
$\Jrightarrow$	<code>&amp;ldrushar;</code>	-	$\leftarrow$	<code>&amp;loarr;</code>	-
$\Jleftarrow$	<code>&amp;lurdshar;</code>	-	$\leftrightarrow$	<code>&amp;hoarr;</code>	-
$\Rrightarrow$	<code>&amp;ruluhar;</code>	-	$\rightsquigarrow$	<code>&amp;zigrarr;</code>	-
$\Rleftarrow$	<code>&amp;luruhar;</code>	-	$\Lrightarrow$	<code>&amp;angzarr;</code>	-
$\Rrightarrow$	<code>&amp;ldrdhar;</code>	-	$\curvearrowright$	<code>&amp;curarrm;</code>	-
$\Rleftarrow$	<code>&amp;rdldhar;</code>	-	$\hookrightarrow$	<code>&amp;cularrp;</code>	-
$\Leftarrow$	<code>&amp;lharul;</code>	-	$\Uparrow$	<code>&amp;ufisht;</code>	-
$\Rightarrow$	<code>&amp;lrhard;</code>	-	$\perp$	<code>&amp;dfisht;</code>	-
$\Rightarrow$	<code>&amp;rharul;</code>	-	$\rightrightarrows$	<code>&amp;rarrsim;</code>	-
$\Leftarrow$	<code>&amp;llhard;</code>	-	$\leftarrow$	<code>&amp;larrsim;</code>	-
$\rightarrow$	<code>&amp;ShortRightArrow;</code>	-			

## 2.7. Különféle műveleti és relációs jelek
































$\forall$	<code>&amp;ForAll;</code> <code>\forall</code>	$\dot{+}$	<code>&amp;dotplus;</code> <code>\dotplus</code>
$\complement$	<code>&amp;comp;</code> <code>\complement</code> <code>&amp;complement</code> ;	$*$	<code>&amp;ast;</code> *
$\exists$	<code>&amp;exist;</code> <code>\exists</code>	$\circ$	<code>&amp;lowast;</code> <code>\ast</code> <code>&amp;midast;</code>
$\nexists$	<code>&amp;nexist;</code> <code>\nexists</code> <code>&amp;nexists;</code> <code>&amp;NotExists;</code>	$\circ$	<code>&amp;cir;</code> <code>\circ</code> <code>&amp;circ;</code> <code>&amp;compfn;</code> <code>&amp;SmallCircle;</code>
$\emptyset$	<code>&amp;emptyv;</code> <code>\varnothing</code> <code>&amp;varnothing</code> ;	$\sqrt{\quad}$	<code>&amp;radic;</code> <code>\sqrt{\quad}</code> <code>&amp;Sqrt;</code>
$\nabla$	<code>&amp;Del;</code> <code>\nabla</code> <code>&amp;nabla;</code>	$\infty$	<code>&amp;prop;</code> <code>\propto</code> <code>&amp;Proportional;</code> <code>&amp;propto;</code> <code>&amp;varpropto;</code> <code>&amp;vprop;</code>
$\in$	<code>&amp;Element;</code> <code>\in</code> <code>&amp;isin;</code>	$\infty$	<code>&amp;infin;</code> <code>\infty</code>
$\notin$	<code>&amp;NotElement</code> ; <code>&amp;notin;</code>	$\angle$	<code>&amp;angrt;</code> -
$\epsilon$	<code>&amp;epsi;</code> <code>\epsilon</code> <code>&amp;in;</code> <code>&amp;isin;</code> <code>&amp;straightepsilon;</code>	$\angle$	<code>&amp;ang;</code> <code>\angle</code> <code>&amp;angle;</code>
$\ni$	<code>&amp;niv;</code> <code>\ni</code> <code>&amp;ReverseElement;</code>	$\sphericalangle$	<code>&amp;angmsd;</code> <code>\measuredangle</code> <code>&amp;measuredangle;</code>
$\nmid$	<code>&amp;notni;</code> <code>\nmid</code> <code>&amp;notniva;</code> <code>&amp;NotReverseElement;</code>	$\sphericalangle$	<code>&amp;angsph;</code> <code>\sphericalangle</code>
$\ni$	<code>&amp;ni;</code> <code>\ni</code> <code>&amp;SuchThat;</code>	$\mid$	<code>&amp;mid;</code> <code>\mid</code> <code>&amp;VerticalBar</code> <code>&amp;r;</code>
$\coprod$	<code>&amp;coprod;</code> <code>\coprod</code> <code>&amp;Coproduct;</code>	$\nmid$	<code>&amp;nmid;</code> <code>\nmid</code> <code>&amp;NotVerticalBar;</code>
$\sum$	<code>&amp;Sum;</code> <code>\sum</code> <code>&amp;sum;</code>	$\parallel$	<code>&amp;DoubleVerticalBar;</code> <code>\parallel</code> <code>&amp;par;</code> <code>&amp;parallel;</code>
$\mp$	<code>&amp;MinusPlus;</code> <code>\mp</code> <code>&amp;mnplus;</code> <code>&amp;mp;</code>	$\nparallel$	<code>&amp;NotDoubleVerticalBar;</code> <code>\nparallel</code> <code>&amp;npar;</code> <code>&amp;nparallel;</code>



































$\wedge$	<code>&amp;And;</code>	<code>\land</code>	$\dot{-}$	<code>&amp;dotminus;</code>	-
	<code>&amp;and;</code>	<code>\wedge</code>	$\dot{-}$	<code>&amp;minusd;</code>	
	<code>&amp;wedge;</code>		$\ddot{=}$	<code>&amp;mDDot;</code>	-
$\vee$	<code>&amp;or;</code>	<code>\lor</code>	$\dot{=}$	<code>&amp;homtht;</code>	-
	<code>&amp;vee;</code>	<code>\vee</code>	$\sim$	<code>&amp;sim;</code>	$\sim$
$\cap$	<code>&amp;cap;</code>	<code>\cap</code>	$\sim$	<code>&amp;Tilde;</code>	
$\cup$	<code>&amp;cup;</code>	<code>\cup</code>	$\backsimeq$	<code>&amp;backsim;</code>	<code>\backsim</code>
$\int$	<code>&amp;int;</code>	<code>\int</code>	$\backsimeq$	<code>&amp;bsim;</code>	
	<code>&amp;Integral;</code>		$\approx$	<code>&amp;mstpos;</code>	-
$\iint$	<code>&amp;Int;</code>	<code>\iint</code>	$\Vdash$	<code>&amp;VerticalTilde;</code>	<code>\wr</code>
$\iiint$	<code>&amp;iiint;</code>	<code>\iiint</code>		<code>&amp;wr;</code>	
	<code>&amp;tint;</code>			<code>&amp;wreath;</code>	
$\oint$	<code>&amp;conint;</code>	<code>\oint</code>	$\napprox$	<code>&amp;NotTilde;</code>	<code>\nsim</code>
	<code>&amp;ContourIntegral;</code>			<code>&amp;nsim;</code>	
	<code>&amp;oint;</code>		$\approx$	<code>&amp;eqsim;</code>	-
$\oiint$	<code>&amp;Conint;</code>	-		<code>&amp;EqualTilde;</code>	
	<code>&amp;DoubleContourIntegral;</code>			<code>&amp;esim;</code>	
$\oiiint$	<code>&amp;Cconint;</code>	-	$\approx$	<code>&amp;sime;</code>	<code>\simeq</code>
$\oint$	<code>&amp;cwint;</code>	-		<code>&amp;simeq;</code>	
$\oint$	<code>&amp;ClockwiseContourIntegral;</code>	-		<code>&amp;TildeEqual;</code>	
	<code>&amp;cwconint;</code>		$\neq$	<code>&amp;NotTildeEqual;</code>	<code>\nsimeq</code>
$\oint$	<code>&amp;awconint;</code>	-		<code>&amp;nsime;</code>	
	<code>&amp;CounterClockwiseContourIntegral;</code>		$\equiv$	<code>&amp;cong;</code>	<code>\cong</code>
$\therefore$	<code>&amp;there4;</code>	<code>\therefore</code>		<code>&amp;TildeFullEqual;</code>	
	<code>&amp;Therefore;</code>		$\ncong$	<code>&amp;simne;</code>	-
	<code>&amp;therefore;</code>		$\ncong$	<code>&amp;ncong;</code>	<code>\ncong</code>
$\because$	<code>&amp;becaus;</code>	<code>\because</code>		<code>&amp;NotTildeFullEqual;</code>	
	<code>&amp;Because;</code>		$\approx$	<code>&amp;ap;</code>	<code>\approx</code>
	<code>&amp;because;</code>			<code>&amp;approx;</code>	
$:$	<code>&amp;ratio;</code>	<code>:</code>		<code>&amp;TildeTilde;</code>	
$::$	<code>&amp;Colon;</code>	<code>::</code>		<code>&amp;nap;</code>	<code>\not\approx</code>
	<code>&amp;Proportion;</code>			<code>&amp;napprox;</code>	



















	$\&\text{NotTildeTi}$				$\&\text{trie};$
	$\&\text{lde};$			$\boxed{=}$	$\&\text{quest};$
$\approx$	$\&\text{ape};$	-			-
	$\&\text{approx};$			$\neq$	$\&\text{ne};$
$\approx$	$\&\text{apid};$	-			$\&\text{neq}$
	$\&\text{backcong};$	-			$\&\text{NotEqual};$
	$\&\text{bcong};$			$\equiv$	$\&\text{Congruent};$
$\asymp$	$\&\text{asym};$	$\backslash\text{asym}$			$\backslash\text{equiv}$
	$\&\text{CupCap};$			$\not\equiv$	$\&\text{equiv};$
$\bumpeq$	$\&\text{bump};$	$\backslash\text{Bumpeq}$			$\&\text{NotCongruent};$
	$\&\text{Bumpeq};$				$\backslash\text{equiv}$
	$\&\text{HumpDownHu}$			$\leq$	$\&\text{nequiv};$
	$\&\text{mp};$				$\&\text{le};$
$\bumpeq$	$\&\text{bumpe};$	$\backslash\text{bumpeq}$			$\backslash\text{leq}$
	$\&\text{bumpeq};$			$\geq$	$\&\text{leq};$
	$\&\text{HumpEqual};$				$\&\text{ge};$
$\doteq$	$\&\text{doteq};$	$\backslash\text{doteq}$			$\backslash\text{geq}$
	$\&\text{DotEqual};$				$\&\text{geq};$
	$\&\text{esdot};$				$\&\text{GreaterEqu}$
$\doteqdot$	$\&\text{doteqdot};$	$\backslash\text{doteq}$			$\text{al};$
	$\&\text{eDot};$			$\leq$	$\&\text{le};$
$\fallingdotseq$	$\&\text{efDot};$	$\backslash\text{fallingdots}$			$\backslash\text{leq}$
	$\&\text{fallingdot}$	$\text{eq}$			$\&\text{leqq};$
	$\&\text{seq};$				$\&\text{LessFullEq}$
$\risingdotseq$	$\&\text{erDot};$	$\backslash\text{risingdotse}$			$\text{ual};$
	$\&\text{risingdots}$	$\text{q}$		$\equiv$	$\&\text{leqq};$
	$\&\text{eq};$				$\&\text{ge};$
$\coloneq$	$\&\text{colone};$	$\text{:=}$			$\backslash\text{geqq}$
	$\&\text{coloneq};$				$\&\text{geqq};$
$\eqcolon$	$\&\text{ecolon};$	$\text{=}$			$\&\text{GreaterFul}$
	$\&\text{eqcolon};$				$\text{lEqual};$
$\eqcirc$	$\&\text{ecir};$	$\backslash\text{eqcirc}$			$\&\text{ge};$
	$\&\text{eqcirc};$				$\backslash\text{geqq}$
$\circeq$	$\&\text{circeq};$	$\backslash\text{circeq}$			$\&\text{gne};$
	$\&\text{cire};$				$\&\text{gneq};$
$\hat{=}$	$\&\text{wedgeq};$	$\backslash\text{hat=}$			$\&\text{gneqq};$
$\check{=}$	$\&\text{veeeq};$	$\backslash\text{check=}$			$\&\text{ll};$
$\easter$	$\&\text{easter};$	-			$\backslash\text{ll}$
$\triangleq$	$\&\text{triangleq};$	-			$\&\text{Lt};$
					$\&\text{NestedLess}$
					$\text{Less};$
				$\gg$	$\&\text{gg};$
					$\backslash\text{gg}$
					$\&\text{Gt};$
					$\&\text{NestedGrea}$
					$\text{terGreater};$
				$\emptyset$	$\&\text{between};$
					$\backslash\text{between}$

	<code>&amp;twixt;</code> <code>&amp;nless;</code> <code>\nless</code> <code>&amp;nlt;</code> <code>&amp;NotLess;</code> <code>&amp;nvlt;</code>			<code>de;</code> <code>&amp;ngsim;</code> <code>\not\gtrsim</code> <code>&amp;NotGreater</code> <code>Tilde;</code>
	<code>&amp;ngt;</code> <code>\ngtr</code> <code>&amp;ngtr;</code> <code>&amp;NotGreater</code> <code>;</code> <code>&amp;nvgt;</code>			<code>&amp;LessGreate</code> <code>\lessgtr</code> <code>r;</code> <code>&amp;lessgtr;</code> <code>&amp;lgt;</code> <code>&amp;gl;</code> <code>&amp;GreaterLes</code> <code>s;</code> <code>&amp;gtrless;</code>
	<code>&amp;nleE;</code> <code>\nleqstant</code> <code>&amp;nleqq;</code> <code>&amp;nleqslant;</code> <code>&amp;nles;</code> <code>&amp;NotGreater</code> <code>FullEqual;</code> <code>&amp;NotLessFull</code> <code>Equal;</code> <code>&amp;NotLessSlan</code> <code>teEqual;</code> <code>&amp;nvle;</code>			<code>&amp;NotLessGre</code> <code>\not\lessgtr</code> <code>ater;</code> <code>&amp;ntlg;</code> <code>&amp;ntvlg;</code>
	<code>&amp;ngE;</code> <code>\ngeqstant</code> <code>&amp;ngeqq;</code> <code>&amp;ngeqslant;</code> <code>&amp;nges;</code> <code>&amp;NotGreater</code> <code>SlantEqual;</code> <code>&amp;nvge;</code>			<code>&amp;NotGreater</code> <code>-</code> <code>Less;</code> <code>&amp;ntgl;</code> <code>&amp;ntvgl;</code>
	<code>&amp;lap;</code> <code>\lesssim</code> <code>&amp;lessapprox</code> <code>;</code> <code>&amp;lesssim;</code> <code>&amp;LessTilde;</code> <code>&amp;lsim;</code>			<code>&amp;pr;</code> <code>\prec</code> <code>&amp;prec;</code> <code>&amp;Precedes;</code>
	<code>&amp;gap;</code> <code>\gtrsim</code> <code>&amp;GreaterTil</code> <code>de;</code> <code>&amp;gsim;</code> <code>&amp;gtrapprox;</code> <code>&amp;gtrsim;</code>			<code>&amp;sc;</code> <code>\succ</code> <code>&amp;succ;</code> <code>&amp;Succeeds;</code>
	<code>&amp;nlsim;</code> <code>\not\lesssim</code> <code>&amp;NotLessTil</code>			<code>&amp;prcue;</code> <code>\preccurlyeq</code> <code>&amp;prE;</code> <code>&amp;pre;</code> <code>&amp;preccurlye</code> <code>q;</code> <code>&amp;PrecedesEq</code> <code>ual;</code> <code>&amp;PrecedesSl</code> <code>antEqual;</code> <code>&amp;preceq;</code>
				<code>&amp;sccue;</code> <code>\succcurlyeq</code> <code>&amp;sce;</code> <code>&amp;succcurlye</code> <code>q;</code> <code>&amp;SucceedsEq</code> <code>ual;</code> <code>&amp;SucceedsSl</code> <code>antEqual;</code>

	&succeq;			&SubsetEqual;
	&prap;	\precsim		&supE;
	&precapprox;			&supe;
	&PrecedesTilde;			&SupersetEqual;
	&precsim;			&subsetneq;
	&prsim;			&subneq;
	&sce;			&supneq;
	&scap;	\succsim		&supneq;
	&scsim;			&supneq;
	&succapprox;			&supneq;
	&SucceedsTilde;			&supneq;
	&succsim;			&supneq;
	&NotPrecede;	\not\prec		&supneq;
	&npr;			&supneq;
	&nprec;			&supneq;
	&NotSucceed;	\not\succ		&supneq;
	&nsc;			&supneq;
	&nsucc;			&supneq;
	&sub;	\subset		&supneq;
	&subset;			&supneq;
	&sup;	\supset		&supneq;
	&Superset;			&supneq;
	&supset;			&supneq;
	&NotSubset;	\notsubset		&supneq;
	&nsup;			&supneq;
	&nsupset;			&supneq;
	&NotSuperset;	\not\supset		&supneq;
	&nsup;			&supneq;
	&nsupset;			&supneq;
	&vnsup;			&supneq;
	&subE;	\subseteq		&supneq;
	&sube;			&supneq;
	&subseteq;			&supneq;
	&subseteqq;			&supneq;

	<code>&amp;sqsubseteq;</code> <code>\sqsubseteq</code>	<code>\sqsubseteqeq</code>		<code>&amp;circleddash</code> <code>\circleddash</code>	
	<code>&amp;sqsubseteqeq</code> <code>;</code>			<code>&amp;odash;</code>	
	<code>&amp;SquareSubsetEqual;</code>			<code>&amp;boxplus;</code> <code>\boxplus</code>	
	<code>&amp;sqsupseteq;</code> <code>\sqsupseteq</code>	<code>\sqsupseteqeq</code>		<code>&amp;boxminus;</code> <code>\boxminus</code>	
	<code>&amp;SquareSupersetEqual;</code>			<code>&amp;boxtimes;</code> <code>\boxtimes</code>	
	<code>&amp;sqcap;</code> <code>\sqcap</code>			<code>&amp;dotsquare;</code> <code>\boxdot</code>	
	<code>&amp;SquareIntersection;</code>			<code>&amp;RightTee;</code> <code>\vdash</code>	
	<code>&amp;bigsqcup;</code> <code>\sqcup</code>	<code>\sqcup</code>		<code>&amp;vdash;</code>	
	<code>&amp;sqcup;</code>			<code>&amp;dashv;</code> <code>\dashv</code>	
	<code>&amp;SquareUnion;</code>			<code>&amp;DownTee;</code> <code>\top</code>	
	<code>&amp;bigoplus;</code> <code>\bigoplus</code>	<code>\bigoplus</code>		<code>&amp;top;</code>	
	<code>&amp;CirclePlus;</code> <code>\oplus</code>	<code>\oplus</code>		<code>&amp;bot;</code> <code>\bot</code>	
	<code>&amp;oplus;</code>			<code>&amp;bottom;</code> <code>\perp</code>	
	<code>&amp;xoplus;</code>			<code>&amp;perp;</code>	
	<code>&amp;CircleMinus;</code> <code>\ominus</code>	<code>\ominus</code>		<code>&amp;UpTee;</code>	
	<code>&amp;ominus;</code>			<code>&amp;models;</code> <code>\models</code>	
	<code>&amp;bigotimes;</code> <code>\bigotimes</code>	<code>\bigotimes</code>		<code>&amp;DoubleRightTee;</code> <code>\Vdash</code>	
	<code>&amp;CircleTimes;</code> <code>\otimes</code>	<code>\otimes</code>		<code>&amp;vDash;</code>	
	<code>&amp;xotimes;</code>			<code>&amp;Vdash;</code> <code>\Vdash</code>	
	<code>&amp;oslash;</code> <code>\oslash</code>			<code>&amp;Vdash;</code> <code>\Vdash</code>	
	<code>&amp;osol;</code>			<code>&amp;nvdash;</code> <code>\nvdash</code>	
	<code>&amp;bigodot;</code> <code>\odot</code>	<code>\odot</code>		<code>&amp;nvdash;</code> <code>\nvdash</code>	
	<code>&amp;CircleDot;</code> <code>\odot</code>	<code>\odot</code>		<code>&amp;nvdash;</code> <code>\nvdash</code>	
	<code>&amp;xodot;</code>			<code>&amp;nvdash;</code> <code>\nvdash</code>	
	<code>&amp;circledcirc;</code> <code>\circledcirc</code>	<code>\circledcirc</code>		<code>&amp;nvdash;</code> <code>\nvdash</code>	
	<code>&amp;ocir;</code>			<code>&amp;prurel;</code> <code>-</code>	
	<code>&amp;circledast;</code> <code>\circledast</code>	<code>\circledast</code>		<code>&amp;LeftTriangle;</code> <code>\vartriangleleft</code>	
	<code>&amp;oast;</code>			<code>&amp;vartriangleleft</code> <code>eleft;</code>	

	$\&\text{vltri};$			$\&\text{sdot};$	
	$\&\text{RightTriangle};$	$\backslash\text{vartriangle right}$		$\&\text{sstarf};$	$\backslash\text{star}$
	$\&\text{vartriangle right};$			$\&\text{Star};$	
	$\&\text{vrtri};$			$\&\text{divideontimes};$	$\backslash\text{divideontimes}$
	$\&\text{LeftTriangle};$	$\backslash\text{triangleleft}$		$\&\text{divonx};$	
	$\&\text{ltri};$			$\&\text{bowtie};$	$\backslash\text{bowtie}$
	$\&\text{triangleleft};$			$\&\text{ltimes};$	$\backslash\text{ltimes}$
	$\&\text{RightTriangle};$	$\backslash\text{triangleright}$		$\&\text{rtimes};$	$\backslash\text{rtimes}$
	$\&\text{rtri};$			$\&\text{leftthreetimes};$	$\backslash\text{leftthreetimes}$
	$\&\text{triangleright};$			$\&\text{rightthreetimes};$	$\backslash\text{rightthreetimes}$
	$\&\text{hercon};$	$-$		$\&\text{rthree};$	
	$\&\text{intcal};$	$\backslash\text{intercal}$		$\&\text{backsimeq};$	$\backslash\text{backsimeq}$
	$\&\text{intercal};$			$\&\text{bsime};$	
	$\&\text{veebar};$	$\backslash\text{veebar}$		$\&\text{curlyvee};$	$\backslash\text{curlyvee}$
	$\&\text{barwed};$	$\backslash\text{barwedge}$		$\&\text{cuvee};$	
	$\&\text{barwedge};$			$\&\text{curlywedge};$	$\backslash\text{curlywedge}$
	$\&\text{barvee};$	$\backslash\text{barvee}$		$\&\text{cuwed};$	
	$\&\text{vangrt};$	$-$		$\&\text{Sub};$	$\backslash\text{Subset}$
	$\&\text{bigwedge};$	$\backslash\text{bigwedge}$		$\&\text{Subset};$	
	$\&\text{wedge};$			$\&\text{Sup};$	$\backslash\text{Supset}$
	$\&\text{xwedge};$			$\&\text{Supset};$	
	$\&\text{bigvee};$	$\backslash\text{bigvee}$		$\&\text{Cap};$	$\backslash\text{Cap}$
	$\&\text{Vee};$			$\&\text{fork};$	$\backslash\text{pitchfork}$
	$\&\text{xvee};$			$\&\text{pitchfork};$	
	$\&\text{bigcap};$	$\backslash\text{bigcap}$		$\&\text{epar};$	$-$
	$\&\text{Intersection};$			$\&\text{lessdot};$	$\backslash\text{lessdot}$
	$\&\text{xcap};$			$\&\text{ltdot};$	
	$\&\text{bigcup};$	$\backslash\text{bigcup}$		$\&\text{gtrdot};$	$\backslash\text{gtrdot}$
	$\&\text{Union};$			$\&\text{gtrdot};$	
	$\&\text{xcup};$			$\&\text{Ll};$	$\backslash\text{lll}$
	$\&\text{diam};$	$\backslash\text{diamond}$		$\&\text{Gg};$	$\backslash\text{ggg}$
	$\&\text{diamond};$			$\&\text{ggg};$	
	$\&\text{cdot};$	$\backslash\text{cdot}$			

	<code>&amp;lEg;</code> <code>&amp;leg;</code> <code>&amp;lesseqgtr;</code> <code>&amp;lesseqqgtr;</code> <code>&amp;LessequalGreater;</code>	<code>\lesseqgtr</code>	<code>&amp;precnsim;</code> <code>&amp;prnap;</code> <code>&amp;prnsim;</code>	
	<code>&amp;gEl;</code> <code>&amp;gel;</code> <code>&amp;GreaterEqualLess;</code> <code>&amp;gtreqless;</code> <code>&amp;gtreqqless;</code>	<code>\gtreqless</code>	<code>&amp;scnap;</code> <code>&amp;scnsim;</code> <code>&amp;succnapprox;</code> <code>&amp;succnsim;</code>	<code>\succnsim</code>
	<code>&amp;els;</code> <code>&amp;eqslantless;</code>	<code>\eqslantless</code>	 <code>&amp;nltri;</code> <code>&amp;NotLeftTriangle;</code> <code>&amp;ntriangleleft;</code>	<code>\ntriangleleft</code>
	<code>&amp;egs;</code> <code>&amp;eqslantgtr;</code>	<code>\eqslantgtr</code>	 <code>&amp;NotRightTriangle;</code> <code>&amp;nrttri;</code> <code>&amp;ntriangleright;</code>	<code>\ntriangleright</code>
	<code>&amp;cuepr;</code> <code>&amp;curlyeqprec;</code>	<code>\curlyeqprec</code>	 <code>&amp;nltrie;</code> <code>&amp;NotLeftTriangleEqual;</code> <code>&amp;ntrianglelefteq;</code>	<code>\ntrianglelefteq</code>
	<code>&amp;cuesc;</code> <code>&amp;curlyeqsucc;</code>	<code>\curlyeqsucc</code>	 <code>&amp;NotRightTriangleEqual;</code> <code>&amp;nrttrie;</code> <code>&amp;ntrianglerighteq;</code>	<code>\ntrianglerighteq</code>
	<code>&amp;NotPrecedeSlantEqual;</code> <code>&amp;nprcue;</code>	<code>\not\preccurlyeq</code>	<code>&amp;vellip;</code>	<code>\vdots</code>
	<code>&amp;NotSucceedSlantEqual;</code> <code>&amp;nsccue;</code>	<code>\not\succcurlyeq</code>	<code>&amp;ctdot;</code>	<code>\cdots</code>
	<code>&amp;NotSquaresubsetEqual;</code> <code>&amp;nsqsube;</code>	<code>\not\sqsubseteq</code>	<code>&amp;utdot;</code>	<code>\udots</code>
	<code>&amp;NotSquaresupersetEqual;</code> <code>&amp;nsqsupe;</code>	<code>\not\sqsupseteq</code>	<code>&amp;dtdot;</code>	<code>\ddots</code>
	<code>&amp;lnsim;</code>	<code>\lnsim</code>	 <code>&amp;Barwed;</code> <code>&amp;doublebarwedg;</code>	<code>\doublebarwedg</code>
	<code>&amp;gnsim;</code>	<code>\gnsim</code>	<code>&amp;plusdu;</code>	<code>-</code>
	<code>&amp;precnapprox;</code>	<code>\precnsim</code>	<code>&amp;minusdu;</code>	<code>-</code>
			<code>&amp;loplus;</code>	<code>-</code>
			<code>&amp;roplus;</code>	<code>-</code>

$\otimes$	<code>&amp;lotimes;</code>	-	$\triangleleft$	<code>&amp;triminus;</code>	-
$\oplus$	<code>&amp;ohbar;</code>	-	$\triangleup$	<code>&amp;tritime;</code>	-
$\cap$	<code>&amp;capdot;</code>	-	$\triangle$	<code>&amp;trisb;</code>	-
$\subset$	<code>&amp;subdot;</code>	-	$\boxminus$	<code>&amp;solb;</code>	-
$\supset$	<code>&amp;supdot;</code>	-	$\boxplus$	<code>&amp;bso1b;</code>	-
$\times$	<code>&amp;smashp;</code>	-	$\cap$	<code>&amp;capand;</code>	-
$\triangle$	<code>&amp;wedbar;</code>	-	$\cup$	<code>&amp;cupor;</code>	-
$\pm$	<code>&amp;pluscir;</code>	-	$\bar{\cup}$	<code>&amp;ncup;</code>	-
$\pm$	<code>&amp;pluse;</code>	-	$\bar{\cap}$	<code>&amp;ncap;</code>	-
$\mp$	<code>&amp;eplus;</code>	-	$\oplus$	<code>&amp;odiv;</code>	-
$+_2$	<code>&amp;plustwo;</code>	-	$\otimes$	<code>&amp;odsolid;</code>	-
$\hat{+}$	<code>&amp;plusacir;</code>	-	$\odot$	<code>&amp;ofcir;</code>	-
$\tilde{+}$	<code>&amp;simplus;</code>	-	$\ominus$	<code>&amp;olt;</code>	-
$\pm$	<code>&amp;plussim;</code>	-	$\oslash$	<code>&amp;ogt;</code>	-
$\times$	<code>&amp;timesd;</code>	-	$\oslash$	<code>&amp;opar;</code>	-
$\cup$	<code>&amp;cupcap;</code>	-	$\oslash$	<code>&amp;operp;</code>	-
$\cap$	<code>&amp;capcup;</code>	-	$\otimes$	<code>&amp;Otimes;</code>	-
$\cup$	<code>&amp;cupbrcap;</code>	-	$\otimes$	<code>&amp;otimesas;</code>	-
$\cap$	<code>&amp;capbrcup;</code>	-	$\times$	<code>&amp;timesbar;</code>	-
$\omega$	<code>&amp;cupcup;</code>	-	$\cong$	<code>&amp;acE;</code>	-
$\cap$	<code>&amp;capcap;</code>	-	$\rhd$	<code>&amp;rpargt;</code>	-
$\cup$	<code>&amp;cups;</code>	-	$\lleftarrow$	<code>&amp;lparlt;</code>	-
$\cap$	<code>&amp;caps;</code>	-	$\backslash$	<code>&amp;rmoust;</code>	-
$\sqcup$	<code>&amp;sqcups;</code>	-		<code>&amp;rmoustache</code>	-
$\sqcap$	<code>&amp;sqcaps;</code>	-		<code>&amp;lmoust;</code>	-
$\sqcup$	<code>&amp;ccups;</code>	-		<code>&amp;lmoustache</code>	-
$\sqcap$	<code>&amp;ccaps;</code>	-		<code>&amp;ltrPar;</code>	-
$\boxcup$	<code>&amp;ccupssm;</code>	-		<code>&amp;gtlPar;</code>	-
$\triangle$	<code>&amp;triplus;</code>	-	$\langle$	<code>&amp;langd;</code>	-

$\rangle$	<code>&amp;rangd;</code>	-	$\supsetneq$	<code>&amp;varsupsetneqq;</code>	<code>\varsupsetneqq</code>
$[$	<code>&amp;lbrke;</code>	-		<code>&amp;vsupnE;</code>	
$]$	<code>&amp;rbrke;</code>	-	$\ncong$	<code>&amp;napid;</code>	-
$[$	<code>&amp;lbrkslu;</code>	-	$\ncong$	<code>&amp;ncongdot;</code>	-
$]$	<code>&amp;rbrksld;</code>	-	$\nvap$	<code>&amp;nvap;</code>	-
$[$	<code>&amp;lbrksld;</code>	-	$\napprox$	<code>&amp;nape;</code>	-
$]$	<code>&amp;rbrkslu;</code>	-	$\P$	<code>&amp;parsim;</code>	-
$\gneq$	<code>&amp;gvertneqq;</code>	<code>\gneqq</code>	$\nless$	<code>&amp;nLt;</code>	-
	<code>&amp;gvnE;</code>		$\ngt$	<code>&amp;nGt;</code>	-
$\lnapprox$	<code>&amp;lnap;</code>	<code>\lnapprox-</code>	$\nless$	<code>&amp;nLtv;</code>	-
	<code>&amp;lnapprox;</code>		$\nless$	<code>&amp;NotLessLess;</code>	-
$\lneq$	<code>&amp;lvertneqq;</code>	<code>\lneqq</code>	$\ngtv$	<code>&amp;nGtv;</code>	-
	<code>&amp;lvnE;</code>		$\nless$	<code>&amp;NotGreaterGreater;</code>	-
$\ngeq$	<code>&amp;nge;</code>	<code>\ngeq</code>	$\nless$	<code>&amp;nLl;</code>	-
	<code>&amp;ngeq;</code>		$\ngg$	<code>&amp;nGg;</code>	-
	<code>&amp;NotGreaterEqual;</code>		$\nvrtrie$	<code>&amp;nvrtrie;</code>	-
$\nleq$	<code>&amp;nle;</code>	<code>\nleq</code>	$\nvltrie$	<code>&amp;nvltrie;</code>	-
	<code>&amp;nleq;</code>		$\rnmid$	<code>&amp;rnmid;</code>	-
	<code>&amp;NotLessEqual;</code>		$\emptyset$	<code>&amp;empty;</code>	<code>\emptysetset</code>
$\nshortmid$	<code>&amp;nshortmid;</code>	<code>\nshortmid</code>		<code>&amp;emptyset;</code>	
	<code>&amp;nsmid;</code>		$ $	<code>&amp;shortmid;</code>	<code>\shortmid</code>
$\nshortparallel$	<code>&amp;nshortpara11el;</code>	<code>\nshortparallel</code>		<code>&amp;smid;</code>	
	<code>&amp;nspar;</code>		$\parallel$	<code>&amp;shortparallel;</code>	<code>\shortparallel</code>
$\precneqq$	<code>&amp;precneqq;</code>	<code>\precneqq</code>		<code>&amp;spar;</code>	
	<code>&amp;prnE;</code>		$\smile$	<code>&amp;smallsmile;</code>	<code>\smallsmile</code>
$\succneqq$	<code>&amp;succneqq;</code>	<code>\succneqq</code>		<code>&amp;ssmile;</code>	
	<code>&amp;scnE;</code>		$\thickapprox$	<code>&amp;thickapprox;</code>	<code>\thickapprox</code>
$\varsubsetneqq$	<code>&amp;varsubsetneqq;</code>	<code>\varsubsetneqq</code>		<code>&amp;thkap;</code>	
	<code>&amp;vsubnE;</code>		$\ddot{=}$	<code>&amp;ddotseq;</code>	-
$\varsubsetneq$	<code>&amp;varsubsetneq;</code>	<code>\varsubsetneq</code>		<code>&amp;eDDot;</code>	
	<code>&amp;vsubne;</code>				
$\varsupsetneq$	<code>&amp;varsupsetneq;</code>	<code>\varsupsetneq</code>			
	<code>&amp;vsupne;</code>				

$\pitchfork$	<code>&amp;mlcp;</code>	-	$\equiv$	<code>&amp;eg;</code>	<code>\eqslanygtr</code>
$\sim$	<code>&amp;siml;</code>	-	$\lesssim$	<code>&amp;ltquest;</code>	-
$\gtrsim$	<code>&amp;simg;</code>	-	$\gtrsim$	<code>&amp;gtquest;</code>	-
$\perp$	<code>&amp;vbar;</code>	-	$\lesseqgtr$	<code>&amp;lesg;</code>	-
$\equiv$	<code>&amp;Colone;</code>	-	$\gtrless$	<code>&amp;gesl;</code>	-
$\dashv$	<code>&amp;Dashv;</code>	-	$\lesseqgtrsl$	<code>&amp;lgE;</code>	-
$\vDash$	<code>&amp;vBar;</code>	-	$\gtrlessgtr$	<code>&amp;glE;</code>	-
$\Barv$	<code>&amp;Barv;</code>	-	$\gtrlessgtrsl$	<code>&amp;glj;</code>	-
$\vDash$	<code>&amp;vBarv;</code>	-	$\gtrlessgtrsl$	<code>&amp;gla;</code>	-
$\vdash$	<code>&amp;vdashl;</code>	-	$\lesseqgtrsl$	<code>&amp;lesges;</code>	-
$\cong$	<code>&amp;congdots;</code>	-	$\gtrlessgtrsl$	<code>&amp;gesles;</code>	-
$\approx$	<code>&amp;apE;</code>	-	$\lesseqgtrsl$	<code>&amp;lsime;</code>	-
$\approx$	<code>&amp;Esim;</code>	-	$\gtrlessgtrsl$	<code>&amp;gsime;</code>	-
$\equiv$	<code>&amp;equivDD;</code>	-	$\lesseqgtrsl$	<code>&amp;lsimg;</code>	-
$\doteq$	<code>&amp;sdote;</code>	-	$\gtrlessgtrsl$	<code>&amp;gsiml;</code>	-
$\mcomma$	<code>&amp;mcomma;</code>	-	$\lesseqgtrsl$	<code>&amp;simlE;</code>	-
$\pitchfork$	<code>&amp;forkv;</code>	-	$\gtrlessgtrsl$	<code>&amp;simgE;</code>	-
$\pitchfork$	<code>&amp;topfork;</code>	-	$\lesseqgtrsl$	<code>&amp;smt;</code>	-
$\lesseqgtr$	<code>&amp;lesdot;</code>	-	$\gtrlessgtrsl$	<code>&amp;lat;</code>	-
$\gtrlessgtr$	<code>&amp;gesdot;</code>	-	$\lesseqgtrsl$	<code>&amp;smte;</code>	-
$\lesseqgtr$	<code>&amp;lesdoto;</code>	-	$\gtrlessgtrsl$	<code>&amp;late;</code>	-
$\gtrlessgtr$	<code>&amp;gesdoto;</code>	-	$\lesseqgtrsl$	<code>&amp;smtes;</code>	-
$\lesseqgtr$	<code>&amp;lesdotor;</code>	-	$\gtrlessgtrsl$	<code>&amp;lates;</code>	-
$\gtrlessgtr$	<code>&amp;gesdoto1;</code>	-	$\subarr$	<code>&amp;subrarr;</code>	-
$\lesseqgtr$	<code>&amp;elsdot;</code>	-	$\suparr$	<code>&amp;suplarr;</code>	-
$\egsdot$	<code>&amp;egsdot;</code>	-	$\subplus$	<code>&amp;subplus;</code>	-
$\ltcir$	<code>&amp;ltcir;</code>	-	$\supplus$	<code>&amp;supplus;</code>	-
$\gtcir$	<code>&amp;gtcir;</code>	-	$\submult$	<code>&amp;submult;</code>	-
$\el$	<code>&amp;el;</code>	<code>\eqslantless</code>	$\supmult$	<code>&amp;supmult;</code>	-

$\subseteq$	<code>&amp;subsim;</code>	-	$\infty$	<code>&amp;iinfin;</code>	-
$\supseteq$	<code>&amp;supsim;</code>	-	$\vee$	<code>&amp;Or;</code>	-
$\subseteq$	<code>&amp;subsup;</code>	-	$\oint$	<code>&amp;pointint;</code>	-
$\supseteq$	<code>&amp;supsub;</code>	-	$\oint$	<code>&amp;quatint;</code>	-
$\subseteq$	<code>&amp;subsub;</code>	-	$\iiint$	<code>&amp;iiiint;</code>	<code>\iiiint</code>
$\supseteq$	<code>&amp;supsup;</code>	-		<code>&amp;qint;</code>	
$\supsetsubset$	<code>&amp;suphsub;</code>	-	$\lvert$	<code>&amp;lopar;</code>	-
$\supsetsubset$	<code>&amp;supdsub;</code>	-	$\rangle$	<code>&amp;ropar;</code>	-
$\nsubseteq$	<code>&amp;bsolhsub;</code>	-	$\nsubseteq$	<code>&amp;notinvb;</code>	-
$\nsubseteq$	<code>&amp;suphsol;</code>	-	$\nsubseteq$	<code>&amp;notinvc;</code>	-
$\dot{\subseteq}$	<code>&amp;subedot;</code>	-	$\nsubseteq$	<code>&amp;notnivb;</code>	-
$\dot{\supseteq}$	<code>&amp;supedot;</code>	-	$\nsubseteq$	<code>&amp;notnivc;</code>	-
$\sqsubset$	<code>&amp;csub;</code>	-	$\text{---}$	<code>&amp;strns;</code>	-
$\sqsupset$	<code>&amp;csup;</code>	-	$\nparallel$	<code>&amp;fltns;</code>	-
$\sqsubseteq$	<code>&amp;csube;</code>	-	$//$	<code>&amp;parsl;</code>	-
$\sqsupseteq$	<code>&amp;csupe;</code>	-	$\top$	<code>&amp;topcir;</code>	-
$\triangleleft$	<code>&amp;ltcc;</code>	-	$\#$	<code>&amp;eparsl;</code>	-
$\triangleleft$	<code>&amp;gttcc;</code>	-	$\#$	<code>&amp;smeparsl;</code>	-
$\triangleleft$	<code>&amp;lescc;</code>	-	$\#$	<code>&amp;eqvparsl;</code>	-
$\triangleleft$	<code>&amp;gescc;</code>	-	$\equiv$	<code>&amp;bnequiv;</code>	-
$\triangleleft$	<code>&amp;rtriltri;</code>	-	$\neq$	<code>&amp;bne;</code>	-
$\ntriangleleft$	<code>&amp;Pr;</code>	-	$\nparallel$	<code>&amp;nparsl;</code>	-
$\ntriangleleft$	<code>&amp;Sc;</code>	-	$\ncong$	<code>&amp;nedot;</code>	-
$\ntriangleleft$	<code>&amp;lttlarr;</code>	-	$\sim$	<code>&amp;simdot;</code>	-
$\ntriangleleft$	<code>&amp;gtrarr;</code>	-	$\approx$	<code>&amp;apacir;</code>	-
$\ntriangleleft$	<code>&amp;iff;</code>	-	$\#$	<code>&amp;nhpar;</code>	-
$\ntriangleleft$	<code>&amp;andand;</code>	-	$\nexists$	<code>&amp;nvinfin;</code>	-
$\ntriangleleft$	<code>&amp;notinva;</code>	-	$\nexists$	<code>&amp;npart;</code>	<code>\not\partial</code>
$\ntriangleleft$	<code>&amp;qprime;</code>	-	$\nexists$	<code>&amp;andv;</code>	-










































$\forall$	<code>&amp;orv;</code>	-	$\ominus$	<code>&amp;omid;</code>	-
$\nabla$	<code>&amp;ord;</code>	-	$\oslash$	<code>&amp;ovbar;</code>	-
$\mathbb{A}$	<code>&amp;andd;</code>	-	$\S$	<code>&amp;race;</code>	-
$\oint$	<code>&amp;cirfnint;</code>	-	$\otimes$	<code>&amp;rotimes;</code>	-
$\int$	<code>&amp;fpartint;</code>	-	$\approx$	<code>&amp;gnap;</code>	<code>\gnapprox</code>
$\int$	<code>&amp;rppolint;</code>	-		<code>&amp;gnapprox;</code>	
$\int$	<code>&amp;scpolint;</code>	-	$\nprec$	<code>&amp;NotPrecedeSEqual;</code>	<code>\npreceq</code>
$\int$	<code>&amp;npolint;</code>	-		<code>&amp;npre;</code>	
$\int$	<code>&amp;intlarhk;</code>	-		<code>&amp;npreceq;</code>	
$\int$	<code>&amp;awint;</code>	-	$\nsubseteq$	<code>&amp;NotSucceedSEqual;</code>	<code>\nsucceq</code>
$\dot{\in}$	<code>&amp;isindot;</code>	-		<code>&amp;nsce;</code>	
$\notin$	<code>&amp;notindot;</code>	-		<code>&amp;nsucceq;</code>	
$\in$	<code>&amp;isinE;</code>	-	$\dagger$	<code>&amp;nvsim;</code>	-
$\in$	<code>&amp;disin;</code>	-	$\nmid$	<code>&amp;solbar;</code>	-
$\Rightarrow$	<code>&amp;nisd;</code>	-	$\angle$	<code>&amp;angrtvb;</code>	-
$\oplus$	<code>&amp;isinsv;</code>	-	$\equiv$	<code>&amp;bbrktbrk;</code>	-
$\oplus$	<code>&amp;xnis;</code>	-	$\emptyset$	<code>&amp;bemptyv;</code>	-
$\in$	<code>&amp;isins;</code>	-	$\odot$	<code>&amp;cirE;</code>	-
$\ni$	<code>&amp;nis;</code>	-	$\odot$	<code>&amp;cirscir;</code>	-
$\asymp$	<code>&amp;Acd;</code>	-	$\circledcirc$	<code>&amp;circledS;</code>	-
$\bowtie$	<code>&amp;elinters;</code>	-	$\oslash$	<code>&amp;oS;</code>	-
$\boxtimes$	<code>&amp;olcross;</code>	-	$\Rightarrow$	<code>&amp;backepsilo</code>	-
$\backslash$	<code>&amp;dsol;</code>	-		<code>n;</code>	
$\swarrow$	<code>&amp;dwangle;</code>	-	$\Rightarrow$	<code>&amp;bepsi;</code>	-
$\nearrow$	<code>&amp;uwangle;</code>	-	$\geqslant$	<code>&amp;geqslant;</code>	-
$\Rightarrow$	<code>&amp;Not;</code>	-		<code>&amp;ges;</code>	
$\models$	<code>&amp;bNot;</code>	-		<code>&amp;GreaterSlantEqual;</code>	
$\nabla$	<code>&amp;orslope;</code>	-	$\leqslant$	<code>&amp;leqslant;</code>	-
$\odot$	<code>&amp;olcir;</code>	-		<code>&amp;les;</code>	
				<code>&amp;LessSlantEqual;</code>	
			$\smallfrown$	<code>&amp;sfrown;</code>	<code>\smallfrown</code>
				<code>&amp;smallfrown</code>	













































$\sim$	<code>&amp;thicksim;</code>	<code>\thicksim</code>	$\infty$	<code>&amp;infintie;</code>	-
$\blacklozenge$	<code>&amp;blacklozenge;</code>	<code>\blacklozenge</code>	$\notin$	<code>&amp;notinE;</code>	-
$\wedge$	<code>&amp;andslope;</code>	-	$\mathbb{W}$	<code>&amp;oror;</code>	-
$\mathbb{Z}$	<code>&amp;imped;</code>	-	$\setminus$	<code>&amp;smallsetminus;</code>	<code>\smallsetminus</code>
				<code>&amp;ssetmn;</code>	


















## 2.8. Egyéb jelek

$i$	<code>&amp;imath;</code>	<code>\i</code>	$\dagger$	<code>&amp;dagger;</code>	<code>\dagger</code>
$j$	<code>&amp;jmath;</code>	<code>\j</code>	$\ddagger$	<code>&amp;Dagger;</code>	<code>\ddagger</code>
$\Gamma$	<code>&amp;digamma;</code>	<code>\digamma</code>		<code>&amp;ddagger;</code>	
$\gamma$	<code>&amp;gamma;</code>		$\bullet$	<code>&amp;bull;</code>	<code>\bullet</code>
$\kappa$	<code>&amp;kappav;</code>	<code>\varkappa</code>		<code>&amp;bullet;</code>	
$\varrho$	<code>&amp;rhov;</code>	<code>\varrho</code>	$\ldots$	<code>&amp;nldr;</code>	<code>..</code>
$-$	<code>&amp;dash;</code>	-	$\ldots$	<code>&amp;hellip;</code>	<code>...</code>
$--$	<code>&amp;ndash;</code>	--	$\%$	<code>&amp;permil;</code>	-
$---$	<code>&amp;mdash;</code>	---	$\%$	<code>&amp;pertenk;</code>	-
$----$	<code>&amp;horbar;</code>	----	$\prime$	<code>&amp;prime;</code>	<code>\prime</code>
$\ $	<code>&amp;verbar;</code>	<code>\vert</code>	$\#$	<code>&amp;Prime;</code>	<code>\Prime</code>
$'$	<code>&amp;lsquo;</code>	<code>'</code>	$\#$	<code>&amp;tprime;</code>	-
$\text{‘}$	<code>&amp;OpenCurlyQuote;</code>		$\backprime$	<code>&amp;backprime;</code>	<code>\backprime</code>
$\text{’}$	<code>&amp;rsquo;</code>	<code>'</code>	$\backprime$	<code>&amp;bprime;</code>	
$\text{“}$	<code>&amp;ldquo;</code>	<code>''</code>	$\hat{\cdot}$	<code>&amp;caret;</code>	-
$\text{”}$	<code>&amp;rdquo;</code>	<code>''</code>	$\text{“}$	<code>&amp;hybull;</code>	-
	<code>&amp;OpenCurlyDoubleQuote;</code>		$\%$	<code>&amp;incare;</code>	<code>%</code>
	<code>&amp;CloseCurlyDoubleQuote;</code>		$\mathcal{D}$	<code>&amp;hamilt;</code>	-
			$\hbar$	<code>&amp;hbar;</code>	<code>\hbar</code>
			$\hbar$	<code>&amp;plank;</code>	
			$\hbar$	<code>&amp;hslash;</code>	<code>\hslash</code>
			$\hbar$	<code>&amp;plankv;</code>	
			$\Im$	<code>&amp;Im;</code>	<code>\Im</code>
				<code>&amp;image;</code>	

$\mathcal{L}$	&lagran;	-	$\frac{7}{8}$	&frac78;	-
$\ell$	&ell;	\ell	$\partial$	&part;	\partial
$\textcircled{P}$	&copysr;	-		&PartialD;	
$\wp$	&weierp;	\wp	$\lceil$	&lceil;	\lceil
	&wp;			&LeftCeiling;	
$\Re$	&Re;	\Re	$\lceil$	&rceil;	\rceil
	&real;			&RightCeiling;	
$\mathbb{R}$	&rx;	-	$\lfloor$	&lfloor;	\lfloor
$\text{TM}$	&trade;	\trade		&LeftFloor;	
$\Omega$	&ohm;	\ohm	$\rfloor$	&rfloor;	\rfloor
$\textcircled{U}$	&mho;	\mho		&RightFloor;	
$\imath$	&iota;	-	$\lrcorner$	&drcorner;	-
$\text{\AA}$	&angst;	-	$\llcorner$	&dlcorner;	-
$\mathcal{B}$	&bernou;	-	$\llcorner$	&urcorner;	-
$\mathcal{M}$	&phmmat;	-	$\lrcorner$	&ulcorner;	-
$\mathcal{O}$	&order;	-	$\lrcorner$	&bnot;	-
$\aleph$	&aleph;	\aleph	$\cap$	&proflines;	-
$\beth$	&beth;	\beth	$\cup$	&profsurf;	-
$\gimel$	&gimel;	\gimel	$\textcircled{P}$	&telrec;	-
$\daleth$	&daleth;	\daleth	$\clubsuit$	&target;	-
$\frac{1}{3}$	&frac13;	-	$\lrcorner$	&ulcorner;	\ulcorner
$\frac{1}{5}$	&frac15;	-		&ulcorner;	
$\frac{2}{5}$	&frac25;	-	$\lrcorner$	&urcorner;	\urcorner
$\frac{3}{5}$	&frac35;	-		&urcorner;	
$\frac{4}{5}$	&frac45;	-	$\llcorner$	&dlcorner;	\llcorner
$\frac{1}{6}$	&frac16;	-		&llcorner;	
$\frac{5}{6}$	&frac56;	-	$\lrcorner$	&drcorner;	\lrcorner
$\frac{1}{8}$	&frac18;	-		&llcorner;	
$\frac{3}{8}$	&frac38;	-	$\frown$	&frown;	\frown
$\frac{5}{8}$	&frac58;	-	$\smile$	&smile;	\smile
				&Cup;	
			$\textcircled{P}$	&cylcty;	-

	<code>&amp;profalar;</code>	-		<code>&amp;triangleleft</code>	<code>\triangleleft</code>
	<code>&amp;topbot;</code>	-		<code>&amp;ltri;</code>	
	<code>&amp;uhblk;</code>	-		<code>&amp;loz;</code>	<code>\lozenge</code>
	<code>&amp;lhbblk;</code>	-		<code>&amp;lozenge;</code>	
	<code>&amp;block;</code>	-		<code>&amp;bigcirc;</code>	<code>\bigcirc</code>
	<code>&amp;blk14;</code>	-		<code>&amp;xcirc;</code>	
	<code>&amp;blk12;</code>	-		<code>&amp;tridot;</code>	-
	<code>&amp;blk34;</code>	-		<code>&amp;bigstar;</code>	<code>\bigstar</code>
	<code>&amp;blacksquare;</code>	<code>\blacksquare</code>		<code>&amp;starf;</code>	
	<code>&amp;squarf;</code>			<code>&amp;cross;</code>	-
	<code>&amp;square;</code>	<code>\square</code>		<code>&amp;female;</code>	-
	<code>&amp;squ;</code>			<code>&amp;male;</code>	-
	<code>&amp;bigtriangleup;</code>	<code>\bigtriangleup</code>		<code>&amp;spades;</code>	<code>\spadesuit</code>
	<code>&amp;xutri;</code>			<code>&amp;spadesuit;</code>	
	<code>&amp;blacktriangle;</code>	<code>\blacktriangle</code>		<code>&amp;hearts;</code>	<code>\heartsuit</code>
	<code>&amp;utrif;</code>			<code>&amp;heartsuit;</code>	
	<code>&amp;triangle;</code>	<code>\triangle</code>		<code>&amp;diam;</code>	<code>\diamondsuit</code>
	<code>&amp;utri;</code>			<code>&amp;diamondsuit;</code>	
	<code>&amp;blacktriangleleft;</code>	<code>\blacktriangleleft</code>		<code>&amp;clubs;</code>	<code>\clubsuit</code>
	<code>&amp;utrif;</code>			<code>&amp;clubsuit;</code>	
	<code>&amp;triangleright;</code>	<code>\triangleright</code>		<code>&amp;sung;</code>	<code>\sung</code>
	<code>&amp;rtrif;</code>			<code>&amp;natur;</code>	<code>\natural</code>
	<code>&amp;blacktriangleright;</code>	<code>\blacktriangleright</code>		<code>&amp;natural;</code>	
	<code>&amp;rtri;</code>			<code>&amp;check;</code>	<code>\checkmark</code>
	<code>&amp;triangleright;</code>	<code>\triangleright</code>		<code>&amp;checkmark;</code>	
	<code>&amp;trif;</code>			<code>&amp;malt;</code>	<code>\maltese</code>
	<code>&amp;bigtriangledown;</code>	<code>\bigtriangledown</code>		<code>&amp;maltese;</code>	
	<code>&amp;xdtri;</code>			<code>&amp;lang;</code>	<code>\langle</code>
	<code>&amp;blacktriangledown;</code>	<code>\blacktriangledown</code>		<code>&amp;rangle;</code>	<code>\rangle</code>
	<code>&amp;dtrif;</code>			<code>&amp;LeftAngleBracket;</code>	
	<code>&amp;triangledown;</code>	<code>\triangledown</code>		<code>&amp;rang;</code>	<code>\rangle</code>
	<code>&amp;dtri;</code>			<code>&amp;rangle;</code>	
	<code>&amp;blacktriangleleft;</code>	<code>\blacktriangleleft</code>		<code>&amp;RightAngleBracket;</code>	
	<code>&amp;ltri;</code>			<code>&amp;Lang;</code>	-

	<code>&amp;Rang;</code>	-		<code>&amp;emptyv;</code>	-
	<code>&amp;lbrk;</code>	-		<code>&amp;raemptyv;</code>	-
	<code>&amp;rbrk;</code>	-		<code>&amp;laemptyv;</code>	-
	<code>&amp;loang;</code>	-		<code>&amp;vzigzag;</code>	-
	<code>&amp;roang;</code>	-		<code>&amp;trpezium;</code>	-
	<code>&amp;lbrk;</code>	-		<code>&amp;bsemi;</code>	-
	<code>&amp;robrk;</code>	-		<code>&amp;bbrk;</code>	-
	<code>&amp;cirmid;</code>	-		<code>&amp;tbrk;</code>	-
	<code>&amp;amalg;</code>	<code>\amalg</code>		<code>&amp;benzena;</code>	-
	<code>&amp;intprod;</code>	-		<code>&amp;benzenb;</code>	-
	<code>&amp;ange;</code>	-		<code>&amp;benzenc;</code>	-
	<code>&amp;range;</code>	-		<code>&amp;benzend;</code>	-
	<code>&amp;ang;</code>	-		<code>&amp;benzene;</code>	-
	<code>&amp;angmsdaa;</code>	-		<code>&amp;benzenf;</code>	-
	<code>&amp;angmsdab;</code>	-		<code>&amp;benzeng;</code>	-
	<code>&amp;angmsdac;</code>	-		<code>&amp;benzenh;</code>	-
	<code>&amp;angmsdad;</code>	-		<code>&amp;benzeni;</code>	-
	<code>&amp;angmsdae;</code>	-		<code>&amp;benzenj;</code>	-
	<code>&amp;angmsdaf;</code>	-		<code>&amp;benzenk;</code>	-
	<code>&amp;angmsdag;</code>	-		<code>&amp;benzenl;</code>	-
	<code>&amp;angmsdah;</code>	-		<code>&amp;benzenm;</code>	-
	<code>&amp;angrtvbd</code>	-		<code>&amp;benzenn;</code>	-
	<code>&amp;urtri;</code>	-		<code>&amp;benzeno;</code>	-
	<code>&amp;lrttri;</code>	-		<code>&amp;benzenp;</code>	-
	<code>&amp;ultri;</code>	-		<code>&amp;benzenq;</code>	-
	<code>&amp;lltri;</code>	-		<code>&amp;benzenr;</code>	-
	<code>&amp;boxbox;</code>	-		<code>&amp;benzen;</code>	-
	<code>&amp;demptyv;</code>	-		<code>&amp;hbenzena;</code>	-
				<code>&amp;hbenzenb;</code>	-

	<code>&amp;hbenzenc;</code>	-		<code>&amp;hbenzenl;</code>	-
	<code>&amp;hbenzend;</code>	-		<code>&amp;hbenzenm;</code>	-
	<code>&amp;hbenzene;</code>	-		<code>&amp;hbenzenn;</code>	-
	<code>&amp;hbenzenf;</code>	-		<code>&amp;hbenzeno;</code>	-
	<code>&amp;hbenzeng;</code>	-		<code>&amp;hbenzenp;</code>	-
	<code>&amp;hbenzenh;</code>	-		<code>&amp;hbenzenq;</code>	-
	<code>&amp;hbenzeni;</code>	-		<code>&amp;hbenzenr;</code>	-
	<code>&amp;hbenzenj;</code>	-		<code>&amp;hbenzen;</code>	-
	<code>&amp;hbenzenk;</code>	-			

## B FÜGGELÉK: AZ OPENMATH KÓDFORMÁI

Az OpenMath nyelvnek kétféle kódolási formája van: az egyik az emberi megértésre alkalmas XML formátumú kódolás (ebben a dolgozatban is ezt használtam), a másik pedig az ún. bájtfolym típusú, mely inkább matematikai alkalmazásoknál használatos.

Ez utóbbi kódolásnál minden egyes XML-kódolású OpenMath szimbólumnak van egy bájtokban leírt megfelelője, ezért a terjedősebb XML típusú kódnál jóval rövidebb, tömörebb.

Lássunk egy példát a két kódolás összehasonlítására!

Az  $(x+y)(x+z)$  egyenlet XML kódolással:

```
<OMOBJ>
  <OMA>
    <OMS cd="arith" name="times"/>
    <OMA>
      <OMS cd="arith" name="plus"/>
      <OMV name="x"/>
      <OMV name="y"/>
    </OMA>
    <OMA>
      <OMS cd="arith" name="plus"/>
      <OMV name="x"/>
      <OMV name="z"/>
    </OMA>
  </OMA>
</OMOBJ>
```

Ugyanaz bájtfolymban:

```
18 10 08 05 05 61 72 69 74 68 74 69 6d 65 73 10 08 05 04 61 72
69 74 68 70 6c 75 73 05 01 78 05 01 79 11 10 48 01 45 00 05 01
7a 11 11 19
```

A fenti bájtok jelentése a következő:

18	az objektum kezdete
10	a kifejezés kezdete
08	a művelet kezdete
05	a CD nevének hossza
05	a művelet nevének hossza

innen kezdődik a CD neve

61	$a$
72	$r$
69	$i$
74	$t$
68	$h$

innen kezdődik a művelet neve

74	$t$
69	$i$
6d	$m$
65	$e$
73	$s$

10 a kifejezés kezdete  
 08 a művelet kezdete  
 05 a CD nevének hossza  
 04 a művelet nevének hossza

innen kezdődik a CD neve

61	$a$
72	$r$
69	$i$
74	$t$
68	$h$

innen kezdődik a művelet neve

70	$p$
6c	$l$
75	$u$
73	$s$

05 a változó kezdete  
 01 a változó nevének hossza  
 78 a változó neve („x”)  
 05 a változó kezdete  
 01 a változó nevének hossza  
 79 a változó neve („y”)  
 11 a kifejezés vége  
 10 a kifejezés kezdete  
 48 a művelet kezdete  
 01 utalás az előző műveletre („arith” – „plus”)  
 45 a változó kezdete  
 00 a változó kezdete, utalás az első változóra („x”)  
 05 a változó kezdete  
 01 a változó nevének hossza  
 7a a változó neve („z”)  
 11 kifejezés vége  
 11 kifejezés vége  
 19 objektum vége

## C FÜGGELÉK: AZ OPENMATH CD-I

Az egyes OpenMath *Content Dictionary*khöz tartozó műveletek:

**arith** (aritmetikai műveletek): *abs, conjugate, divide, minus, plus, power, product, root, sum, times, unary\_minus*

**calculus** (differenciál- és integrálszámítás): *defint, diff, int, partialdiff*

**cc** (típusok): *bytearray, float, integer, Lambda, mapsto, omtype, PiType, prop, string, symtype, type, typecoerce*

**ecc** (típusok az ESPRIT Consortiumtól): *array, bytearray, float, integer, Lambda, omtype, Pair, PairProj1, PairProj2, PiType, prop, SigmaType, string, simtype, type*

**fns** (függvények): *identity, inverse, lambda, left\_compose*

**integer** (műveletek egész számokkal): *factorial, gcd, quotient, rem*

**interval** (intervallumok): *integer\_interval, interval, interval\_cc, interval\_co, interval\_oc, interval\_oo*

**limit** (határok): *above, below, both\_sides, limit, null*

**linalg** (lineáris algebra): *matrix, matrixrow, vector, determinant, selector, transpose*

**list** (listák): *list, make\_list*

**logic** (logikai műveletek): *and, false, implies, not, or, true, xor*

**meta** (az OpenMath CD-k szerekezetének leírása): *CD, CDDate, CDName, CDReviewDate, CDStatus, CDURL, CDUses, CDVersion, CMP, Description, Example, FMP, Name*

**minmax** (minimum- és maximumszámítás): *max, min*

**nums** (alap-számhalmazok): *based\_integer, complex\_cartesian, complex\_polar, e, gamma, i, infinity, NaN, pi, rational*

**quant** (feltételek): *exist, forall*

**relation** (relációk): *eq, geq, gt, leq, lt, neq*

**set** (halmazok): *in, intersect, notin, notprsubset, notsubset, prsubset, set, setdiff, subset, union*

**stats** (statisztika): *mean, median, mode, moment, sdev, variance*

**transc** (trigonometria): *arccos, arcsin, arctan, cos, cosh, cot, coth, csc, csch, exp, ln, log, sec, sech, sin, sinh, tan, tanh*

## **D FÜGGELÉK: MATEMATIKAI SZOFTVEREK**

A következőkben felsorolok néhány Internetes címet, ahol információkat lehet szerezni egyes matematikai szoftverekről. A letölthető szoftvereket feltettem a CD mellékletre is, de található néhány nem letölthető program is a Weben.

Egy általános lista a MathML szoftverekről:

<http://indy.cs.concordia.ca/mathml/software/index.html>

Néhány érdekes szerkesztőprogram:

<http://www.mathtype.com>

<http://www.webeq.com>

<http://www.hypernews.org>

A W3C MathML munkacsoportja:

<http://www.w3.org/Math>

Más MathML források:

<http://indy.cs.concordia.ca/mathml>

<http://www2.geom.umn.edu/HyperNews/get.cgi/mathml.html>

<http://www.webeq.com/webeq/mathml/resources.html>

Információk német nyelven:

<http://www.uni-koeln.de/~a0047/mathml-xml-kompass.html>

T<sub>E</sub>X – OpenMath átalakító:

<http://www.maths.tcd.ie/pub/openmath/ml2om-demo.html>

OpenMath - T<sub>E</sub>X átalakító:

<http://www.maths.tcd.ie/~dwmalone/om2la.html>

T<sub>E</sub>X – MathML átalakító:

<http://hutchinson.belmont.ma.us/tth/mm1/tthmm1form.html>

## **E FÜGGELÉK: A CD MELLÉKLET TARTALMA**

A **Dokumentumok** könyvtár tartalma:

**MathML:** A World Wide Web Consortium (W3C) MathML 1.0-s ajánlása, eredeti dokumentáció (**toc.html**);

**OpenMath:** Az OpenMath Consortium OpenMath 1.0 Standard eredeti dokumentációi;

**Szakdolgozat:** Diplomamunkám Word97-es változata (**Szakdolgozat.doc**);

A **Programok** könyvtár tartalma:

**Amaya:** A World Wide Web Consortium (W3C) Web-böngészője és – MathML szerkesztésre is alkalmas – szerkesztője;

**E-lite:** Az ICESoft Java-alapú böngészője MathML implementációkhoz is;

**EzMath:** Egy könnyen használható egyenletszerkesztő MathML-hez;

**Ice Browser 5:** Az ICESoft Java-alapú böngészője MathML implementációkhoz is;

**maje\_dos:** DOS-alapú MathML fordító;

**Publicon:** Professzionális matematikai szerkesztőprogram, későbbi verziókban várható MathML output, valamint TeX konverzió;

**Tex4HT:** Általános TeX – XML fordító;

**webEQ:** Egy Java alapú szerkesztőprogram MathML-hez, TeX konvverterrel.

## **IRODALOMJEGYZÉK**

- [1] Bujdosó Gyöngyi, Fazekas Attila: „TEX kezdőlépések”, Budapest, Tertia Kiadó, 1996
- [2] Stephen Buswell, Stan Devitt, Angel Diaz, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, Stephen Watt: „Mathematical Markup Language (MathML) 1.0 Specification” (REC-MathML-19980407; revised 19990707), The World Wide Web Consortium, <http://www.w3.org/TR/REC-MathML>, 1998-99
- [3] O. Caprotti, A. M. Cohen: „The OpenMath Standard, Version 1.0 (D1.3.2a)” (openmath\_partI), The OpenMath Consortium, [www.nag.co.uk/projects/OpenMath/omstd/partI.pdf](http://www.nag.co.uk/projects/OpenMath/omstd/partI.pdf), 1999
- [4] Obádovics J. Gyula: „Matematika”. Budapest, Sclar Kiadó, 1994
- [5] Obádovics J. Gyula: „Valószínűségszámítás és matematikai statisztika”. Budapest, Sclar Kiadó, 1997
- [6] Országos Pedagógiai Intézet: „Négyjegyű függvénytáblázatok”, szerk.: Hack Frigyes, Kugler Sándor, Balázs Lóránt. Budapest, Tankönyvkiadó, 1989
- [7] The World Wide Web Consortium: „Extensible Markup Language (XML) 1.0” (REC-xml-19980210), The World Wide Web Consortium, <http://www.w3c.org/TR/1998/REC-xml-19980210.html>, 1998
- [8] The OpenMath Consortium: „OpenMath Version1 – Draft”, The OpenMath Consortium, <http://ftp-sop.inria.fr/safir/OM/v1.ps>

## TÁRGYMUTATÓ

- \$, 38
- \$\$, 38, 66, 89
- &, 85
- ;, 66
- \!, 66
- \,, 66
- \\, 78, 84
- \+, 85
- \above, 69
- \abovewithdelims, 69
- \align, 67
- \arccos, 138
- \arctan, 138
- \arcsin, 138
- \atop, 69
- \atopwithdelims, 69
- \bf, 44
- \big, 52
- \Big, 52
- \bigg, 52
- \Bigg, 52
- \binom, 68
- \bmatrix, 84
- \bmod, 106
- \bold, 58
- \brace, 69
- \brack, 69
- \c, 84
- \cap, 132
- \cdot, 104
- \cfrac, 69
- \choose, 69
- \circ, 123
- \column, 84
- \cos, 138
- \cosh, 138
- \cot, 138
- \coth, 138
- \cr, 85, 90
- \cup, 132
- \csc, 138
- \dbinom, 69
- \det, 144
- \dfrac, 69
- \displaystyle, 54, 61
- \documentstyle, 61, 94
- \endbmatrix, 84
- \endcfrac, 69
- \endmatrix, 84
- \endpmatrix, 84
- \endSb, 78
- \endsmallmatrix, 84
- \endSp, 78
- \endvmatrix, 84
- \endVmatrix, 84
- \eqalign, 90
- \exists, 112
- \exp, 110
- \fiverm, 44
- \font, 44
- \forall, 112
- \format, 84
- \frac, 68
- \fracwithdelims, 68
- \gamma, 39
- \Gamma, 39
- \geq, 116
- \halign, 85
- \hfil, 85
- \hfill, 85
- \hrule, 86
- \hss, 85
- \i, 39
- \in, 129
- \infty, 135
- \int, 124
- \it, 43
- \italic, 58
- \j, 39
- \l, 84
- \lambda, 122
- \land, 112
- \left, 53, 69, 73
- \leftarrow, 135
- \leftrightarrow, 135
- \leq, 116
- \lfrac, 69
- \lg, 110
- \lim, 135
- \limits, 55, 78, 124, 135
- \ln, 110
- \log, 110
- \longrightarrow, 135
- \longleftrightarrow, 135

<code>\longrightarrow</code> , 135	<code>\tan</code> , 138
<code>\lor</code> , 112	<code>\tanh</code> , 138
<code>\matrix</code> , 84	<code>\tenit</code> , 44
<code>\max</code> , 110	<code>\tenrm</code> , 44
<code>\min</code> , 110	<code>\text</code> , 46, 57
<code>\multispan</code> , 86	<code>\thickfrac</code> , 69
<code>\ne</code> , 116	<code>\thickfracwithdelims</code> , 69
<code>\neg</code> , 112	<code>\thickness</code> , 69
<code>\noalign</code> , 86	<code>\to</code> , 78, 135
<code>\nolimits</code> , 54	<code>\tt</code> , 44
<code>\not</code> , 132	<code>\underbrace</code> , 78
<code>\notin</code> , 129	<code>\underline</code> , 78
<code>\of</code> , 70, 106	<code>\underset</code> , 78
<code>\offinterlineskip</code> , 86	<code>\undersetbrace</code> , 78
<code>\omit</code> , 86	<code>\vmatrix</code> , 84
<code>\over</code> , 69	<code>\Vmatrix</code> , 84
<code>\overbrace</code> , 78	<code>\vspace</code> , 66, 78
<code>\overline</code> , 78	
<code>\overline</code> , 106	
<code>\overset</code> , 78	<code>\{</code> , 39, 42, 52, 57, 58, 66, 67, 75, 85, 99
<code>\oversetbrace</code> , 78	
<code>\overwithdelims</code> , 69	
<code>\par</code> , 86	<code>&lt;abs/&gt;</code> , 105, 106
<code>\partial</code> , 127	<code>&lt;and/&gt;</code> , 112
<code>\pmatrix</code> , 84	<code>&lt;annotation&gt;</code> , 30, 146
<code>\prod</code> , 135	<code>&lt;annotation-xml&gt;</code> , 30, 65, 146
<code>\qquad</code> , 66	<code>&lt;apply&gt;</code> , 23, 28, 98, 112, 116, 149
<code>\quad</code> , 66, 84, 90	<code>&lt;arccos/&gt;</code> , 138
<code>\r</code> , 84	<code>&lt;arctan/&gt;</code> , 138
<code>\rfac</code> , 69	<code>&lt;arcsin/&gt;</code> , 138
<code>\right</code> , 53, 69, 73	<code>&lt;bvar&gt;</code> , 24, 101, 124, 125, 127, 129, 134
<code>\rightarrow</code> , 135	<code>&lt;ci&gt;</code> , 21, 97, 100, 148, 149
<code>\Rrightarrow</code> , 112	<code>&lt;cn&gt;</code> , 21, 95, 103, 148, 149
<code>\rm</code> , 43	<code>&lt;compose/&gt;</code> , 123
<code>\roman</code> , 58	<code>&lt;condition&gt;</code> , 112, 116, 117, 124, 126, 129, 134
<code>\root</code> , 70, 106	<code>&lt;conjugate/&gt;</code> , 106
<code>\Sb</code> , 78	<code>&lt;cos/&gt;</code> , 138
<code>\scriptscriptstyle</code> , 61	<code>&lt;cosh/&gt;</code> , 138
<code>\scriptstyle</code> , 61	<code>&lt;cot/&gt;</code> , 138
<code>\sec</code> , 138	<code>&lt;coth/&gt;</code> , 138
<code>\setminus</code> , 132	<code>&lt;csc/&gt;</code> , 138
<code>\settabs</code> , 84, 85	<code>&lt;csch/&gt;</code> , 138
<code>\sevenrm</code> , 44	<code>&lt;declare&gt;</code> , 24, 100, 122, 149
<code>\sin</code> , 138	<code>&lt;degree&gt;</code> , 101, 105, 127, 140
<code>\sinh</code> , 138	<code>&lt;determinant/&gt;</code> , 143
<code>\sl</code> , 43	<code>&lt;diff/&gt;</code> , 24, 127
<code>\slanted</code> , 58	<code>&lt;divide/&gt;</code> , 103
<code>\smallmatrix</code> , 84	<code>&lt;eq/&gt;</code> , 116
<code>\Sp</code> , 78	<code>&lt;exists/&gt;</code> , 112
<code>\spreadlines</code> , 66	<code>&lt;exp/&gt;</code> , 110
<code>\spreadmatrixlines</code> , 66	<code>&lt;factorial/&gt;</code> , 105, 106
<code>\sqrt</code> , 70, 106	<code>&lt;fn&gt;</code> , 21, 28, 119, 148, 149
<code>\subset</code> , 132	<code>&lt;forall/&gt;</code> , 112
<code>\subseq</code> , 132	<code>&lt;gcd/&gt;</code> , 105, 106
<code>\sum</code> , 135	
<code>\szóköz</code> , 66	
<code>\tabskip</code> , 85	

<code>&lt;geq/&gt;</code> , 116	<code>&lt;mtd&gt;</code> , 53, 54, 80, 81
<code>&lt;gt/&gt;</code> , 116	<code>&lt;mtext&gt;</code> , 55, 56
<code>&lt;ident/&gt;</code> , 120	<code>&lt;mtr&gt;</code> , 80, 81
<code>&lt;implies/&gt;</code> , 112	<code>&lt;munder&gt;</code> , 22, 54, 77, 78
<code>&lt;in/&gt;</code> , 129	<code>&lt;munderover&gt;</code> , 54, 77, 79
<code>&lt;int/&gt;</code> , 24, 124, 134	<code>&lt;neq/&gt;</code> , 116
<code>&lt;intersect/&gt;</code> , 132	<code>&lt;none/&gt;</code> , 75, 153
<code>&lt;interval/&gt;</code> , 107	<code>&lt;not/&gt;</code> , 112
<code>&lt;interval&gt;</code> , 105, 124, 125, 149	<code>&lt;notin/&gt;</code> , 129
<code>&lt;inverse/&gt;</code> , 123	<code>&lt;notprsubset/&gt;</code> , 132
<code>&lt;lambda&gt;</code> , 122, 149	<code>&lt;notsubset/&gt;</code> , 132
<code>&lt;leq/&gt;</code> , 23, 116	<code>&lt;OMA&gt;</code> , 99, 120, 155
<code>&lt;limit/&gt;</code> , 134	<code>&lt;OMATP&gt;</code> , 97
<code>&lt;list&gt;</code> , 129, 149	<code>&lt;OMATTR&gt;</code> , 96, 97, 117
<code>&lt;ln/&gt;</code> , 109	<code>&lt;OMB&gt;</code> , 98
<code>&lt;log/&gt;</code> , 109	<code>&lt;OMBIND&gt;</code> , 120, 122
<code>&lt;logbase&gt;</code> , 109	<code>&lt;OMBVAR/&gt;</code> , 122
<code>&lt;lowlimit&gt;</code> , 24, 124, 125, 134	<code>&lt;OMBVAR&gt;</code> , 120
<code>&lt;lt/&gt;</code> , 116	<code>&lt;OME&gt;</code> , 91
<code>&lt;maction&gt;</code> , 23, 88, 91, 92	<code>&lt;OMF/&gt;</code> , 36, 95
<code>&lt;maligngroup/&gt;</code> , 82, 87, 88, 153	<code>&lt;OMI&gt;</code> , 95
<code>&lt;malignmark/&gt;</code> , 43, 82, 87, 88, 153	<code>&lt;OMOBJ&gt;</code> , 155, 162
<code>&lt;math&gt;</code> , 20, 100, 151, 152, 153	<code>&lt;OMS/&gt;</code> , 36, 96, 101, 104, 106, 110, 112, 116, 120, 123, 124, 127, 129, 132, 135, 138, 140, 142, 144
<code>&lt;matrix&gt;</code> , 23, 29, 142, 149	<code>&lt;OMS&gt;</code> , 135, 161
<code>&lt;matrixrow&gt;</code> , 142, 149	<code>&lt;OMSTR&gt;</code> , 97
<code>&lt;max/&gt;</code> , 109	<code>&lt;OMV/&gt;</code> , 36, 97, 101
<code>&lt;mean/&gt;</code> , 140	<code>&lt;or/&gt;</code> , 112
<code>&lt;median/&gt;</code> , 140	<code>&lt;partialdiff/&gt;</code> , 23, 127
<code>&lt;merror&gt;</code> , 90	<code>&lt;plus/&gt;</code> , 19, 21, 22, 103
<code>&lt;mfenced&gt;</code> , 27, 71, 72, 81, 88	<code>&lt;power/&gt;</code> , 103, 105, 106, 153
<code>&lt;mfrac&gt;</code> , 23, 59, 68, 104	<code>&lt;product/&gt;</code> , 134
<code>&lt;mi&gt;</code> , 21, 22, 43, 44, 56, 97, 148	<code>&lt;prsubset/&gt;</code> , 132
<code>&lt;min/&gt;</code> , 109	<code>&lt;quotient/&gt;</code> , 105, 106
<code>&lt;minus/&gt;</code> , 103	<code>&lt;reln&gt;</code> , 28, 112, 116, 149
<code>&lt;mmultiscripts&gt;</code> , 23, 74, 75	<code>&lt;rem/&gt;</code> , 105, 106
<code>&lt;mn&gt;</code> , 21, 22, 43, 46, 95, 148	<code>&lt;root/&gt;</code> , 105, 106
<code>&lt;mo&gt;</code> , 21, 22, 48, 56, 59, 61, 65	<code>&lt;sdev/&gt;</code> , 140
<code>&lt;mode/&gt;</code> , 140	<code>&lt;sec/&gt;</code> , 138
<code>&lt;moment/&gt;</code> , 140	<code>&lt;sech/&gt;</code> , 138
<code>&lt;mover&gt;</code> , 54, 77, 78	<code>&lt;selector/&gt;</code> , 144
<code>&lt;mpadded&gt;</code> , 59, 62, 64, 65	<code>&lt;semantics&gt;</code> , 25, 30, 64, 88, 146, 148
<code>&lt;mphantom&gt;</code> , 62, 65, 88	<code>&lt;sep/&gt;</code> , 95, 148, 153
<code>&lt;mprescripts/&gt;</code> , 75	<code>&lt;set&gt;</code> , 23, 129, 149
<code>&lt;mroot&gt;</code> , 70	<code>&lt;setdiff/&gt;</code> , 132
<code>&lt;mrow&gt;</code> , 19, 23, 25, 41, 51, 53, 58, 62, 65, 67, 72, 73, 81, 88, 90, 98, 151	<code>&lt;sin/&gt;</code> , 138
<code>&lt;ms&gt;</code> , 55, 56	<code>&lt;sinh/&gt;</code> , 138
<code>&lt;mSPACE/&gt;</code> , 43, 62, 63	<code>&lt;subset/&gt;</code> , 132
<code>&lt;msqrt&gt;</code> , 70	<code>&lt;sum/&gt;</code> , 134
<code>&lt;mstyle&gt;</code> , 23, 33, 54, 58, 59, 62, 65, 74, 75, 77, 83, 88, 151	<code>&lt;tan/&gt;</code> , 23, 138
<code>&lt;msub&gt;</code> , 22, 74	<code>&lt;tanh/&gt;</code> , 138
<code>&lt;msubsup&gt;</code> , 74	<code>&lt;tendsto/&gt;</code> , 134, 135
<code>&lt;msup&gt;</code> , 19, 74	<code>&lt;transpose/&gt;</code> , 143
<code>&lt;mtable&gt;</code> , 29, 80, 81, 87, 88, 160	<code>&lt;union/&gt;</code> , 132

<b>&lt;uplimit&gt;</b> , 124, 125, 134	<i>height</i> , 62, 63, 152
<b>&lt;variance/&gt;</b> , 140	<i>hex</i> , 36, 95
<b>&lt;vector&gt;</b> , 19, 141, 149	<i>HREF</i> , 153
<b>&lt;xor/&gt;</b> , 112	
<i>accent</i> , 48, 54, 77, 78	<i>id</i> , 43, 94
<i>accentunder</i> , 77, 78	
<i>actiontype</i> , 92	<i>largeop</i> , 48, 54
<i>align</i> , 81, 82	<i>linethickness</i> , 59, 68
<i>alignmentscope</i> , 81, 82, 87	<i>lquote</i> , 56
<i>altimg</i> , 152	<i>lspace</i> , 48, 52, 63
<i>alttext</i> , 152	
<i>background</i> , 59, 60, 61	<i>macros</i> , 151
<i>base</i> , 95	<i>maxsize</i> , 48, 52, 53
<i>baseline</i> , 152	<i>minsize</i> , 48, 52, 53
	<i>mode</i> , 151
<i>cd</i> , 36	<i>movablelimits</i> , 48, 54, 77
<i>class</i> , 43, 94, 151	<i>movablescripts</i> , 59
<i>close</i> , 71, 72	
<i>closure</i> , 105	<i>name</i> , 36, 135, 138, 152
<i>color</i> , 43, 59, 60	<i>nargs</i> , 100
<i>columnalign</i> , 81, 82	
<i>columnlines</i> , 81, 83	<i>occurence</i> , 100
<i>columnspacing</i> , 81, 82	<i>open</i> , 71, 72
<i>columnspan</i> , 81, 84	<i>order</i> , 129
	<i>other</i> , 93, 94
<i>dec</i> , 36, 95	<i>overflow</i> , 152
<i>definitionURL</i> , 94, 99, 106, 116, 129	
<i>depth</i> , 62, 63	<i>rowalign</i> , 81, 82
<i>displaystyle</i> , 54, 59, 74, 75, 77, 78, 81, 83, 151	<i>rowlines</i> , 81, 83
	<i>rowspacing</i> , 81, 82
<i>encoding</i> , 30, 146	<i>rowspan</i> , 81, 84
<i>equalcolumns</i> , 81, 83	<i>rquote</i> , 56
<i>equalrows</i> , 81, 83	<i>rspace</i> , 48, 52
<i>fence</i> , 48, 49, 55	<i>scriptsizemultiplier</i> , 59
<i>fontfamily</i> , 43	<i>scope</i> , 100
<i>fontsize</i> , 43, 59, 60	<i>scriptlevel</i> , 59, 60, 63, 74, 75, 77, 83
<i>fontstyle</i> , 43, 44, 45, 46	<i>scriptminsize</i> , 59, 60
<i>fontweight</i> , 43	<i>scriptsizemultiplier</i> , 59, 60
<i>form</i> , 48, 51, 55, 66	<i>selection</i> , 92
<i>frame</i> , 81, 83	<i>separator</i> , 48, 49, 72
<i>framespacing</i> , 81, 83	<i>separators</i> , 71
	<i>stretchy</i> , 48, 52
<i>groupalign</i> , 81, 82, 88	<i>style</i> , 43, 94, 151
<i>groupalignment</i> , 81	<i>subscriptshift</i> , 74
	<i>symmetric</i> , 48, 52, 53

*szóköz*, 38

**vskip**, 86

*type*, 95, 97, 100, 129, 134, 152,  
153

*width*, 62, 63, 152

*XML-LINK*, 153