Huhyphn2 – magyar elválasztási mintagyűjtemény

Nagy Bence

2004. szeptember 5.

Kivonat

A Huhyphn2 elválasztási mintagyűjtemény létrehozásának célja az élő magyar nyelv szavainak hibátlan elválasztása szabad szoftveres környezetben. A mintagyűjtemény a TEX-ben és valamennyi a LibHnj programkönyvtárat használó alkalmazásban – a legjelentősebbek az OpenOffice.org és a Scribus – teszi lehetővé az algoritmus által megszabott keretek közötti magyar nyelvű elválasztást.

1. Áttekintés

1.1. Miért kell elválasztani?

A klasszikus könyvművészet a szöveget mindig téglalap alakú szedéstükörbe helyezte el. A nyelv szavai egy általános szövegben azonban sohasem követik egymást olyan sorrendben, hogy egymás mellé helyezve egyenlő hosszúságú sorokat alkossanak. A betűk szerkezeti felépítése, a betűk egymásközti távolsága és a szavak közti távolság nemcsak esztétikai kérdés, hanem az olvashatóság szempontjából is fontos.

Amikor JOHANNES GUTENBERG a híres negyvenkét soros Bibliát készítette, egységesen keskeny szóközöket alkalmazott. A sorkizárást különböző szélességű betűk metszésével, ligatúrák és abbreviatúrák alkalmazásával érte le. Noha a latin nyelv betűi és írásjelei mintegy 60 jelet tesznek ki, Gutenberg 290 különböző jelet metszett. A mai napig esztétikai mintaképül szolgál.

Követője, PETER SCHÖFFER vezette be a manapság is leggyakrabban használt eljárást, a szóközök méretének változtatását. A szöveg szóközeinek növelése vagy csökkentése révén lehet elérni az egyforma hosszúságú sorokat. Esztétikai és olvasáspszichológiai okokból azonban nem szabad eltérni nagymértékben az alapszóköz méretétől. Ez abban az esetben valósítható meg jól, ha egy sorban minél több szóköz van, így az egyes sorhosszkülönbségeket szétosztva, azok kisebb mértékben növelik vagy csökkentik a szóközök méretét. Nemkívánatos az a jelenség, amikor az egyes szóközök mérete olyan nagymértékben nő, hogy azok akár a sorok közti távolságot is meghaladják.

Mivel a magyar nyelv agglutináló, vagyis a szótőhöz hozzáilleszti a toldalékokat, a szövegben a hosszú szavak igen gyakoriak, így az elválasztások alkalmazása nélkülözhetetlen, ha igényes szedés kialakítása a célunk. Ennek jelentősége nyomtatott dokumentumok esetén van, az interneten legelterjedtebb HTML-formátumú weboldalak nem tartalmaznak elválasztást, és alapesetben nem sorkizárt szövegeket jelenítenek meg.

1.2. Az elválasztás jelentősége

Amíg a TEX-rendszer főleg tudományos körökben használt eszköz maradt fejlesztésének évei alatt, a Liang-féle elválasztó algoritmus problémái csak kevés embert érintettek. Mivel a TEX nyílt rendszer, ezért bárki készíthet hozzá kiegészítéseket, javításokat, így korábban a szakértő felhasználók is orvosolni tudták gondjaikat, és elmondhatjuk, hogy a TEX szellemiségébe beleillik ez a fajta felhasználói változtatás.

Az OpenOffice.org általános célú irodai programcsomag, amelybe szintén ezt az elválasztó algoritmust építették be, azonban ez a rendszer már a szélesebb közönséget célozza meg. A program a „középiskolás fokon” oktatott számítástechnikai ismeretekkel is egyszerűen használható, ezért elterjedése nem lehet kétséges. Mivel szabadon terjeszthető, ezért a Linux operációs rendszer elsőszámú irodai programcsomagjává vált, és része a legtöbb disztribúciónak – aki egy modern Linux disztribúciót telepít, az előbb-utóbb találkozik a programmal. Az asztali gépen Linuxot használók száma még csekély a Microsoft Windows felhasználóihoz képest, de valószínűleg ez utóbbi platformon is gyorsan fog nőni a programot használók száma, nem beszélve a többi operációs rendszerről, amelyen elérhető.

Az OpenOffice.org-ba épített LibHnj programkönyvtárat RAPH LEVIEN írta 1998-ban, ennek a magas szintű elválasztás és sorkiegyenlítés a feladata. Forráskódja szintén szabadon felhasználható, ezért várható, hogy újabb programokba is be fogják építeni.

Az egyik ilyen ismert alkalmazás, a Scribus tördelőprogram, melynek 2003. júliusában jelent meg az 1.0-s verziója, és jelenleg úttörőnek mondható a Linuxos DTP területén. Felhasználóinak minden bizonnyal alacsonyabb az OpenOffice.org-énál, viszont ezen a területen nagyobb a jó elválasztás jelentősége.

Magyar nyelvű szövegek szedésének egyik sarkalatos pontja a helyes elválasztás. A helytelen elválasztás helyesírási hiba, az elválasztások kihagyása azonban a sorkizárt szövegben okozhatja a fent említett szóközanomáliát. Nyelvünknek speciális elválasztási kívánalmai is vannak. A nyelvben több egyszerűsítve kettőzött hosszú mássalhangzó van, melyeket nem egyszerűen csak kettéválasztunk, hanem bizonyos betűk a két elválasztott részben megismétlődnek. Hogy érthetőbbé tegyem: az *asszony* szó az *asz-* és a *szony* tagokra választandó szét, erre azonban a TEX beépített algoritmusa nem alkalmas. Mivel az ilyen betűk előtt is és után is egy-egy magánhangzó van, ha nem tudjuk elválasztani a szót, úgy egy legkevesebb öt betűből álló láncot kapunk, mely gondot okozhat az egyenletes szóközök terén, ha a szó pont a sor végére esik.

A TEX egész bekezdést vizsgáló sortörő algoritmusa annyira kifinomult, hogy optimális használatához a forrást a lehető legjobban kell előkészíteni.

1.3. A magyar nyelv elválasztási szabályai

A magyar nyelvnek egyszerűek az elválasztási szabályai: a szavakat szótagokra kell bontani, és az a szótaghatárok mentén elválasztható. A szótagoló elválasztás nem alkalmazható összetett szavak esetén maradéktalanul, itt a szóösszetétel határára kell esnie az elválasztásnak. A szabály idegen szavak esetében a kiejtés szerinti szótagolást részesíti előnyben, ezért sok idegen szó számít csak egy szótagosnak és nem elválaszthatónak, még ha az eredeti alakban vagy annak átvett változatában több magánhangzó is szerepel.

A fenti két szabály ütközik, és ennek következtében a magyar elválasztási szabályok nem adnak egyértelmű utasítást arra az esetre, ha idegen eredetű szavak szerepelnek a szóösszetételekben. Ilyenkor *A magyar helyesírás szabályai*-ban leírtak szerint a meghatározatlan „átlagos magyar nyelvérzéknek” kell döntenie, hogy elfogadja-e vala-

mely összetevőt a magyarban is élő alaknak vagy pedig a szóösszetételtől függetlenül szótagolás szerint kell elválasztani.

A szótagolás szerinti történő elválasztás módszerét a szabályzat 226. paragrafusa tartalmazza. Rövid algoritmusa a következő: minden magánhangzó új szótagot jelöl, és a magánhangzók előtt szereplő mássalhangzók közül mindig az utolsót kell átvinni a következő sorba, és itt a tényleges másalhangzót kell érteni, többjegyű esetén valamennyi betűjének az új sorba kell átkerülnie.

Erre az algoritmusra könnyű programot írni, ez azonban csak a magyar nyelvre lesz használható, a HiOn is eszerint működik és a TEX magyar elválasztási mintáinak 3.12-es verziójáig az ezt megvalósító mintahalmazzal rendelkezett. Az összetett szavak elválasztásának problémáját mindkettő a szó kivételszótárba, illetve a minták közé történő közvetlen felvételével oldotta meg. Ennek hátránya nyilvánvaló: a szóösszetételek száma hatalmas, és ebben az esetben a hibásan elválasztott összetett szavakkal folyamatosan kell bővíteni a szótárakat.

1.4. A TEX módszere

A TEX-rendszerbe FRANK M. LIANG elválasztási algoritmusát építették be, amely előre megadott minták alapján határozza meg az elválasztási helyeket, és így bizonyos megszorításokkal valamennyi latin betűs nyelvre alkalmazható.

Liang 1980 és 1982 között dolgozta ki, és *Word hy-phen-a-tion by computer* című doktori értekezésében 1983-ban publikálta a számítógéppel történő elválasztás egy módszerét, amelyet a TEX-rendszerbe is beépítettek. A Liang-algoritmus gyorsan működik, és valamennyi olyan elválasztási helyet bejelöl, amelyet az elválasztási mintái között eltárolunk. Az algoritmust megvalósító kód igen egyszerű és kevés memóriát igényel, ez az állítás pedig tizenöt évvel kitalálása után hatványozottan igaz.

A TEX a következőképpen választ el. Ha megkap egy szót, először megnézni a kivételszótárat, hogy nincs-e definiálva benne ennek a szónak a különleges elválasztása. Amennyiben nem szerepel benne, úgy a Liang-algoritmussal választja el.

Az angol `hyphenation` szó elválasztásán keresztül mutatom be az algoritmus működését, Knuth maga is ezt a példát hozza fel, és FRICZ CREMER, a *The TEXbook* németre fordítója, is ragaszkodott ehhez.

Az algoritmus a szót először kiegészíti két ponttal: `.hyphenation.`, majd pedig felbontja egyre növekvő hosszúságú szakaszokra. A pontok a szó elejét és végét jelentik, ennek fontos szerepe van, némelyik elválasztási minta konkrétan csak szó eleji vagy végi helyzetben ad helyes elválasztást.

Az így kapott egybetűs szakaszok:

```
. h y p h e n a t i o n .
```

a kétbetűsek:

```
.h hy yp ph he en na at ti io on n.
```

hárombetűsek:

```
.hy hyp ype pen ena nat ati tio ion on.
```

és így tovább.

Az így létrejött `k` hosszúságú szakaszokhoz `k+1` számot társít, melyek az egyes betűk előtt fogják jelölni az elválasztási helyeket. A `hen` szakaszhoz négy szám kerül majd, melyet a következképpen lehet szemléltetni:

```
0h0e2n0
```

A 2-es számjegy az `e` betű utáni lehetséges elválasztás információját tartalmazza.

Az elválasztás folyamata a továbbiakban úgy zajlik, hogy a program az előre eltárolt elválasztási mintákból megkeresi azokat, amelyek megegyeznek a szó szétbontott szakaszaival. A `hyphenation` szóra az angol nyelvi szótárból a következőeket találja meg:

```
0h0y3p0h0
0h0e2n0
0h0e0n0a4
0h0e0n5a0t0
1n0a0
0n2a0t0
1t0i0i0
2i0o0
0o2n0
```

Az elválasztás utolsó lépése az, hogy az egyes betűk közé jutó értékek közül mindig a legnagyobbat választja ki, majd a szavunkba illeszti azokat. A szó végül így fog kinézni:

```
0h0y3p0h0e2n5a4t2i0o2n0
```

Ahol a szóban páratlan szám található, ott elválasztható a szó, ahol páros vagy nulla, ott nem engedélyezett az elválasztás. A hyphenation szó tehát elválasztva: `hy-phen-ation`.

Ha összehasonlítjuk Liang doktori értekezésének címét és a végeredményt, lehet látni, hogy ez utóbbiban eggyel kevesebb elválasztási hely lett bejelölve, maga Knuth azt írja, hogy az elválasztási algoritmus az elválasztási helyek döntő többségét megtalálja. Megfelelően kialakított mintafájl esetén azonban valamennyi elválasztási helyet megkapjuk.

1.4.1. A módszer hiányosságai

Az algoritmus egyik nagy hiányossága, mely minket különösképpen érint, hogy nem képes az egyszerűsítve kettőzött hosszú mássalhangzókat elválasztani.

A TEX elválasztórendszerének másik hiányossága, hogy nem választja el azokat a szavakat, amelyekben kötőjel van. A 6–3-as szabály miatt a hosszú szóösszetételeket kötőjellel kell írni, de földrajzi nevekben is sokszor fordul elő ez az írásmód.

Erre a problémára megoldást jelent BERND RAICHLE `hypht1.tex` nevű fájlja, azonban ehhez mélyebben nyúl bele a TEX elválasztórendszerébe. A `\hyphenation` makróval lehet elválasztásokat megadni, amelynek kötőjellel elválasztott szavakat kell megadni. Kötőjeles alak esetén a kötőjel elé egy egyenlőségejelet kell írni: `\hyphenation{Er-zsé-bet=-híd}`.

A LATEX Babel csomagjához készült stílusfájlban sikerült megoldani az egyszerűsítve kettőzött hosszú mássalhangzók elválasztásának problémáját, azonban az csak jelentős többletmunka árán kelthető életre. A trükk lényege, hogy ezeket a karaktersorozatokat külön meg kell jelölni egy fordított aposztróffal: pl. `loccsan`, `fröccsen`.

2. A magyar elválasztóprogramok története

2.1. A Huhyph-generációk

A TEX-ben és az OpenOffice.org-ban sokáig használt elválasztási mintagyűjtemény a Huhyph 3.12 verziója volt. Ezt a mintakollekciót MIKLÓS DEZSŐ hozta létre 1989-ben, majd MAYER GYULA fejlesztette tovább, és jelenleg is ő ennek a vonalnak a karbantartója. A legújabb változat, amely a 4.0-s verziószámot kapta, kísérleti jellegű, és noha 2002 júniusában megjelent már, hivatalosan soha nem került be egy disztribúcióba sem . A Huhyph 3.12 és a Huhyph 4.0 verzió között lényeges szemléletbeli különbség van.

2.1.1. Huhyph 3.12

A magyar nyelv elválasztásában a fonetikus és az összetétel szerinti szabályok játszanak szerepet. Általánosságban elmondhatjuk, hogy minden szót a fonetikai szabályok szerint kell elválasztani, azonban összetett szavak esetében az elválasztási pontnak az összetétel határára kell esnie. Idegen eredetű szóösszetételeknél e két szabály alkalmazása ingadozhat.

A Huhyph 3.12 ezért azt az elvet követi, hogy az elválasztási minták alapját a kézzel rögzített fonetikai szabályok adják, és majdnem minden egyes összetétel elválasztását külön minta szabályozza. Így minden egyes szó elválasztása a fonetikai szabályok szerint történik, hacsak nem került be a kézzel szerkesztett minták közé az ezt szabályzó kivétel. A módszer hátránya nyilvánvaló, hiszen minden egyes összetett szót, amelynél a fonetikai szabályok szerinti elválasztás nem az összetétel határára esik, külön fel kell venni a minták közé.

Olvasáspszichológiailag indokolható, hogy az összetételek határán lévő egy szótagot képző magánhangzók az elválasztás során az őket tartalmazó összetevőben maradjanak, ezért gyakran az amúgy jól elválasztódó szavakra is külön mintát kell alkalmazni. Nyelvtanilag lehetséges a *rádi-óadó* és a *rádióa-dó* forma, de vitán felül áll a *rádió-adó* változat elsőbbsége.

A kézzel szerkesztett mintagyűjtemény esetén felmerülhet, hogy a kollekció kevésbé optimálisan tárolja az egyes kivételeket lekezelő mintákat. Bővítése során erre tekintettel kell lenni, ezért egy-egy minta hozzáadása alapos utánagondolást és a többi minta felülvizsgálatát igényli.

2.1.2. Huhyph 4.0

A Huhyph 4.0 már a FRANK LIANG és PETER BREITENLOHNER által fejlesztett PatGen programmal készült. A PatGen egy elválasztásokat tartalmazó szótár alapján hozza létre az elválasztási minták kollekcióját, így a Huhyph 4.0 változatának alapja is egy nagyméretű szótár, mely a a TypoTEX Kiadó néhány könyvének szóanyagát tartalmazza. Ezenkívül az Informatikai Kormánybiztosság a Széchenyi-terv keretében pénzzel támogatta a projektet.

A PatGen-nel történő mintagenerálás nagy hiányossága, hogy a létrehozott kollekció teljes bizonyossággal csak a szótárban szereplő szavakon működik: a program optimális eredményt ad a megadott szótárra vonatkoztatva, de ez nem jelenti azt, hogy ez a nyelv szavainak egészére érvényes. Mivel a magyar nyelv agglutináló, ezért sokfajta toldalék, illetve toldalékok kombinációja járulhat egy-egy szóhoz. Nyelvünkben létezik a hangkivetős tövek jelensége is, amikor úgy tűnik, mintha az egyes toldalékok nem

az alapszóhoz, hanem annak módosulatához csatlakoznának. Ezért sok esetben hibás elválasztást kapunk egy ragozott szónál akkor is, ha annak töve szerepel a szótárban.

Mivel összetett szavak elválasztásakor előfordulhat, hogy a szóösszetételi határon önálló szótagot alkotó magánhangzó van, és ennek a másik összetételhez hajtása rontja az értelmezést, Mayer a szóhatárokat +-jellel jelölte. A szótár ezért egy előfeldolgozáson esik át, amely a +-jelet rendes elválasztójellé alakítja át, ugyanakkor az egy betű távolságra lévő elválasztásokat letiltja.

Mivel a magyar elválasztás ingadozik az idegen eredetű szóösszetételek elválasztása tekintetében, az elválasztómodul kétféle összeállításban érkezik: az egyik a tudományos igényű, amely a szavak összetétel voltát tartja előbbre, a másik a fonetikus változat, mely a szótagolás elvét. A dokumentációban leírtak alapján az derül ki, hogy a fejlesztő az elsőt részesíti előnyben, és ezzel az emberek tudatos nyelvhasználatára és az átlagos műveltség emelkedésére törekszik.

2.2. HiOn

Régebben, amíg még nem állt rendelkezésre a TEX-rendszerhez megfelelő magyar elválasztómodul, VERHÁS PÉTER HiOn programja volt a leghasználhatóbb eszköz. Preprocesszorként a szövegben bejelölte az elválasztási helyeket, ezzel a TEX már olyan bemeneti fájlt kapott, ahol nem kellett a szavak elválasztásával foglalkoznia, mivel azokban már szerepeltek a puha elválasztójelek.

A HiOn az elválasztást számos apró beállítási lehetősége révén úgy oldotta meg, hogy semmivel sem csökkentette a TEX tipográfiai képességeit: ugyanis a ligatúrák csak akkor illesztődnek megfelelően a szövegbe, ha nem szerepel az azt alkotó betűk között puha elválasztójel, így ehhez a ligatúrákat diszkrecionális elválasztásként definiálja. Mivel az akkoriban közkedvelt Computer Modern font főleg csak nagybetű-kisbetű egalizálásokat tartalmaz, ezért a kerninget felborító elválasztó makró nem is igen gyakorolt hatást a szedés minőségére. A HiOn viszont ugyanezzel a módszerrel a hosszú dupla mássalhangzók elválasztásának problémáját is megoldotta.

A HiOn gyenge pontja, hogy magánhangzók között nem lehet letiltani az elválasztást, ilyen persze a magyar nyelvben csak idegen eredetű szavak esetén fordulhat elő, ebben az esetben a program működését ideiglenes ki kellene kapcsolni, de ez rengeteg többletmunkával járna. Mivel az igekötőket véges állapotú automata kezeli, egy új szó felvételekor a program forráskódján kell módosítani, és a fordítást meg kell ismételni.

Előnye viszont rendkívüli gyors működése, egy 200 oldalas könyv szövegének elválasztása nem tart tovább 2 másodpercnél egy mai átlagos számítógépen.

Mivel Verhás Péter utoljára a 3.0 verziót adta ki (illetve 4.0 is létezik, de ennek forráskódja megegyezik az előzővel, csak a licence más) még 1994-ben, a fejlesztés megszakadásáról beszélhetünk. Használóinak száma minden bizonnyal csekély lehet, az újabb magyar elválasztómodulok mellett normális esetben nincs szükség a HiOn használatára.

A HiOn C-nyelvű forrását az internetről le lehet tölteni, de lefordításával Windows operációs rendszer alatt komoly problémák voltak. A Borland cég C++ 5.5 fordítójában próbáltam végrehajtani a folyamatot, de az eredmény működésképtelen lett. Ennek oka az volt, hogy a HiOn írásának idejében még a 16-bites Windows-ok korát éltük, és a 32-bites fordító összezavarodott az eltérő egészszám-deklarációk miatt. A fordítás folyamatának egyszerűsítése érdekében az egész rendszer egy fájlból áll, és a forráskód bőven el van látva szöveges megjegyzésekkel, melyek nélkül a kód szinte érthetetlen programozói mű. A mintegy 200 kB kódból a felesleges funkciókat és megjegyzéseket kiszedve mintegy 100 kB olyan kódot kaphatunk, amely ugyanazokat a gyakorlatban

felmerülő feladatokat hajtja végre. Ezt a változatot Linux alá is sikerült változtatások nélkül lefordítani.

A programban súlyos hiba is van: az elválasztási folyamat beindítása előtt a szavakat kisbetűssé alakítja, és a kétjegyű mássalhangzókat az első betűjük nagy változatával kódolja. Mivel a program a `cs` és a `ch` betűkapcsolatokat is mássalhangzónak tekinti, az egyezés miatt a `ch` betűpárt `H`-val helyettesíti. Ha a program a kivételszótárat használja egyes elválasztásokhoz, úgy a kódok feloldásakor nem a kódolt betűhöz tartozó kétjegyű mássalhangzó első betűjét használja, hanem magát a kódot helyettesíti be. Így lehet, hogy a kivételszótár *harminchat* szava elválasztva `harminh-hat` lesz.

2.3. Mspell

Az Mspell a Helyes-e programcsaládról ismert MorphoLogic Unix-alapú rendszerekhez készített terméke. Otthoni használatra ingyenes, ami annak köszönhető, hogy a cég a Széchenyi-terv keretében támogatást kapott az ilyen licencfeltételekkel történő kiadáshoz. Az Mspell a kereskedelmi cégektől talán megszokottnak tekinthető módon így csak egy kiadást élt meg, és nem is nagyon reménykedhetünk újabb build-ban.

Maga a program a Helyes-e konzolon futó változatának tekinthető, ennek megfelelően ugyanazokkal a hiányosságokkal rendelkezik, ezek közül a legbosszantóbb – ami egyébként még indokolható is –, hogy helyesírás-ellenőrző lévén a hibásnak vélt szóalakok elválasztását meg sem próbálja, és így elég sok helyes szó elválasztatlanul marad. Rendelkezik egy súlyos, algoritmikus hibával is: ha összetett szó olyan ragot kap, amely akár önálló jelentéssel rendelkező szó is lehetne, akkor minden esetben újabb összetevőnek feltételezi (pl. *se-géd-mun-kás-oké*), a leggyakrabban hibásan.

Mivel a program csővezetékbe köthető, ezért tesztelési célra alkalmas, amennyiben reguláris kifejezések olyan kombinációját tudjuk meghatározni, amelyekkel az Mspell algoritmikus hibalehetőségeit kiszűrhetjük.

2.4. Huhyphn1

Több mint 450.000 szavas szótárával a Huhyphn előző verziója volt a legnagyobb ilyen irányú projekt a szabad szoftveres elválasztási mintagyűjtemények történetében. Noha a mostani mintagenerálási módszer már rendelkezésre állt, és különféle könyvek (pl. *Egri csillagok*) szóanyagával végzett tesztelés jó eredményeket mutatott, sajnos túlságosan sok hiba jellemezte a Huhyphn1 működését. A fejlesztés során nyilvánvalóvá vált, hogy még ekkora szótárméret sem elegendő.

3. A szótárfejlesztés

3.1. A Huhyph 3.12 és a Huhyph 4.0 által felvetett kérdések

A Huhyph 3.12 és Huhyph 4.0 esetén alkalmazott módszerek akármelyikét vizsgálva jelentős nehézséggel kerülünk szembe.

A kézzel szerkesztett fonetikai bázisra épülő változatnál rengeteg szót kell megvizsgálni és a rájuk alkalmazható mintákat megtalálni. Az új minták felvétele esetén meg kell vizsgálni a régebbieket, hogy azok módosításával elérhető-e a kívánt eredmény, vagy az új minta felvétele esetén találkozunk-e olyan szóval, amelyiket eddig helyesen választott el, de az új minta felvételével már hibásan.

A Huhyph 4.0 módszerével készített kollekció esetén pedig akkor érhetünk el optimális eredményt, ha egy szóhóz valamennyi képzett alakjának elkészítjük az elválasztott formáját, és felvesszük a szótárba. A Huhyph 4.0 szókészlete nem haladja meg a 70.000-es méretet, a mintagenerálás folyamata azonban a dokumentációja szerint egy 1 GHz-es Pentium IV-es számítógépen majdnem nyolc percet vesz igénybe. Amennyiben minden szónak további változatait képezzük, úgy a művelethez szükséges idő nagyságrendekkel nőhet, és akár órákig is tarthat. Nem lehetünk azonban biztosak ekkor sem abban, hogy a szótárból hiányzó szavak megfelelően választódnak el, csak reménykedhetünk, hogy a szótár növekedésével a fonetikai szabályok és a növekvő számú azonos kivételek átlépik azt a kritikus tömeget, hogy érvényesüljenek a nem felvett szavakra is.

3.2. A PatGen

A PatGen kiválóan használható alkalmazás, hogyha meg tudjuk kerülni a használatából származó hátrányait. A fenti problémák alapján két elvárásnak kell megfelelni:

- A generált minták tartalmazzák teljes egészében a fonetikai szabályokat, hogy a szótárban nem szereplő szótagok esetén is érvényesüljenek.
- A generált minták ne befolyásolják a fonetikai szabályok szerint választandó szavak elválasztását.

Az első elvárást egyszerű teljesíteni, a PatGen-t előre elkészített mintákkal kell meghívni, amelyek tartalmazzák a szótagolás szerinti elválasztás szabályait.

A második elvárás teljesítése a nehezebb, ugyanis itt a PatGen működése okozza a problémát. A program ugyanis a bemeneti szótárt alapul véve határozza meg az összetétel szerint elválasztandó szavak esetében a szükséges alkalmazandó mintát. Mivel optimális megoldásra törekszik, ezért ez a minta a lehető legrövidebb lesz, és így könnyen előfordulhat az, hogy egy szótárban nem szereplő, a fonetikai szabályokkal elválasztható szónál hibás elválasztást fogunk kapni.

A PatGen megfelelő paraméterezésével érhetjük el, hogy ne törekedjen optimális eredmény generálására, de ebben az esetben is a bemeneti szótár duzzasztására van szükség. Nem agglutináló nyelvek esetén bőven elég lenne a szó felvétele a szótárba, de a magyarban minél több toldalékolt alakot fel kell venni. Erre egy módszer, ha könyvek szavait a Hunspell-en keresztül átszűrjük. Ez a helyesírás-ellenőrző meg tudja mondani egy szóról, hogy az valamely szótőnek a toldalékolt alakja-e. Ha ismerjük a szótő elválasztását, akkor a toldalékolt alak elválasztását is meghatározhatjuk, mivel a toldalékok minden esetben a fonetikai szabályok szerint választandók el. A szótár növekedése ebben az esetben nem lesz annyira drasztikus, minta egy algoritmussal valamennyi létező toldalékolt alakot felvennénk, ugyanis így csak a ténylegesen használt formák kerülnek a szótárba.

A PatGen alkalmazásával elkerülhető, hogy a TEX kivételszótárát kelljen használni az ismeretlen vagy hibás elválasztású szavak esetén, elég a szót felvenni a program bemeneti szótárába, és az új mintafájl már megfelelően fogja kezelni ezt a szót is.

A szótár gyarapodásával együtt jár a feldolgozási idő növekedése, de ezt az árat meg kell fizetnünk.

3.3. A Szószablya

A Szószablya projekt (`http://www.szoszablya.hu/`) a magyar web feldolgozásával olyan szóanyagot állított össze, amely a ma beszélt magyar nyelv szókészletét nagy mértékben lefedi. A teljes szóanyagból a nagy gyakorisággal előfordulóakat kiválogatva és a Hunspell helyesírás-ellenőrzőn átszűrve olyan szótárat kapunk, amely ideális lehet az elválasztási mintagyűjtemény bemeneteként.

4. Fejlesztőkörnyezet

4.1. Források

4.1.1. Patgen

A minták generálását a teTEX disztribúció patgen alkalmazása végzi. A `magyar.tra`, a `patgen.in` és a `base.pat` fájlok a program működéséhez szükséges paramétereket tartalmazzák. A patgen.patch alkalmazása a nagy mennyiségű adat feldolgozásához szükséges, jelenleg nem ismert, hogy ilyen paraméterek mellett mekkora bemeneti szótár feldolgozására képes, a mintegy 2.500.000 szavas állomány esetén sikerrel használható.

4.1.2. Substrings.pl

Az OpenOffice.org a módosított algoritmusú LibHnj-t használja, ehhez a csomagban lévő `substrings.pl` használatával a kész minták átalakítására van szükség.

4.1.3. A szótár

A bemeneti szótár mintegy 45 MB méretű, és igen gyakran történnek benne hibajavítások, ezért ez egyelőre nem elérhető. Viszont a szótár magját képező szógyűjteményt a `web2.2-mostfrequent-hungarian-words.txt.gz` fájl tartalmazza, ezt a Szószablya projekt weboldaláról lehet letölteni. A majdnem teljes bemeneti szótár előállításához a `test.rb` program módosításával juthatunk, ha azt csővezetékbe kötve futtatjuk a fenti fájlon.

4.2. Működés

A folyamat a `Makefile`-on keresztül a `make` parancs kiadásával történik. A `patgen` formátumfüggetlen mintagyűjteményt hoz létre, melynek felhasználásával az `eruby` program (ezt külön kell telepíteni) a `.tmpl` végű sablonfájlokból elkészíti a mintagyűjtemény különböző változatait.

5. Telepítés

5.1. Telepítés TEX alá

A TEX-rendszerhez rendelkezésre álló mintagyűjtemény a `.tex` nevű fájlban található. A fájlban található karakterek az EC-, T1- vagy más néven Cork-kódolás szerint szerepelnek. Ez a szokásos magyar nyelvű LATEX használat mellett eredményezi a megfelelő

működést. A fájlt a texmf-fa `/tex/generic/hyphen` könyvtárába kell másolni, majd a `mktexlsr` programmal frissíteni a fájlnyilvántartást.

A fájl megfelelő helyre másolása és a konfigurációs fájlok szükséges módosítása után szükség van a formátumfájlok legenerálására, mivel a szótárak feldolgozása nem futásidőben történik. A teTEX-rendszeren a beállítások végrehajthatóak a `texconfig` programmal, amely a különböző makrócsomagoknál teszi lehetővé az eltérő beállítás használatát, és gondoskodik a formátumfájlok elkészítéséről is. Az alábbiakban a kézzel történő beállítások találhatóak.

5.1.1. LaTeX

A LaTeX makrócsomag esetén a betöltendő elválasztási minták a `language.dat` fájlban találhatóak. Ennek szokásos helye a texmf-fa

```
/tex/generic/config
```

könyvtár. Szerepelnie kell benne egy

```
magyar huhyph.tex
```

tartalmú sornak, esetleg százalékjellel az elején. Ezt a sort tegyük megjegyzésbe, és egy másik sorba írjuk a következőt:

```
magyar huhyphn.tex
```

A formátumfájl legenerálásakor már a Huhyphn elválasztási mintái épülnek be.

5.1.2. ConTeXt

A ConTEXt más megközelítést használ. A texmf-fában található a

```
/tex/context/config/cont-usr.tex
```

nevű fájl, melyben a rendszer minden egyes elválasztási mintához definiál egy egységes szinonimát. A magyar nyelv definícióját a következő sor tartalmazza.

```
\definefilesynonym [lang-hu.pat]  [huhyph.tex]
```

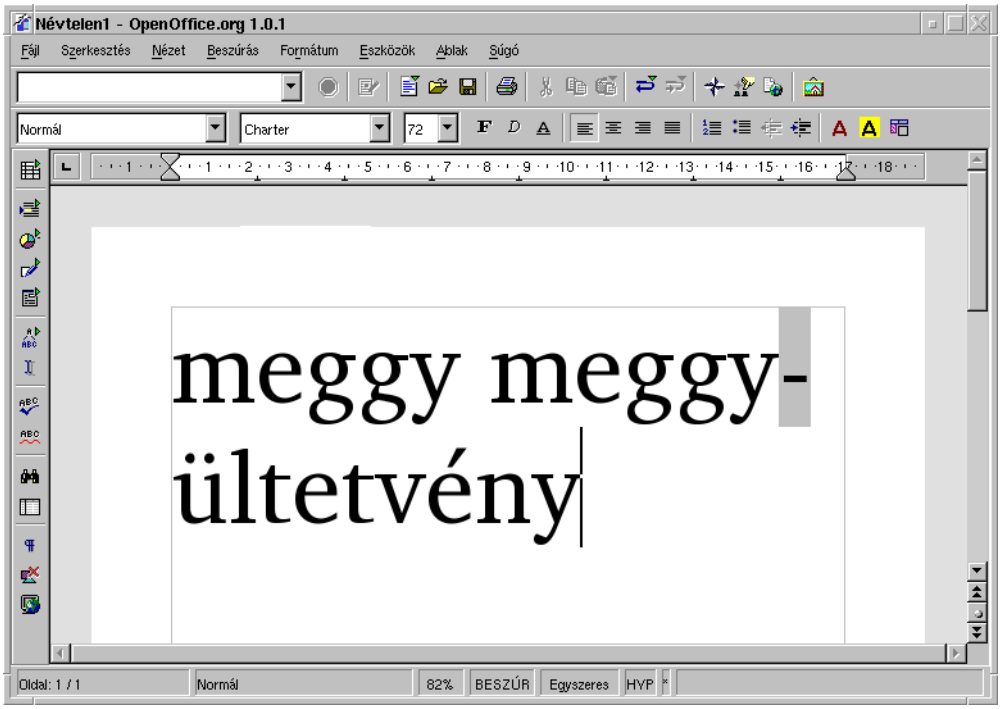
Ez kell módosítani az alábbira:

```
\definefilesynonym [lang-hu.pat]  [huhyphn.tex]
```

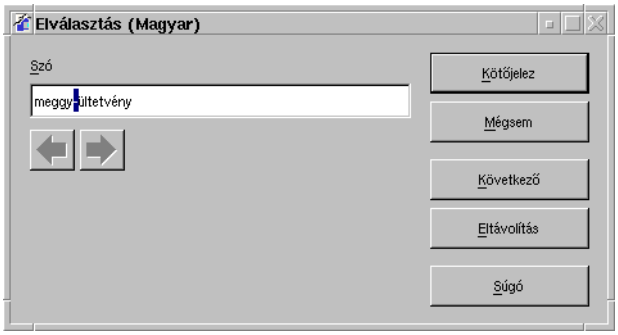
Ha még nem tettük volna meg, töröljük a százalékjelet a következő sor elejéről, ezzel érhetjük el, hogy a magyar elválasztási minták beforduljanak a formátumba.

```
% \installlanguage [\s!hu] [\s!status=\v!start] % hungarian
```

Ezek után generáljuk le a formátumfájlt.



1. ábra. Az *OpenOffice.org* ablakában immár a helyes elválasztás.



2. ábra. Az elválasztás engedélyezésének ablaka.

5.2. Telepítés OpenOffice.org alá

Az *OpenOffice.org* irodai programcsomaghoz használható elválasztási mintafájl a `hyph_hu.dic` nevet viseli. Ezt a következő könyvtárba másolva írjuk felül az eredeti változatot (a könyvtár disztribúciótól függően változhat):

```
/usr/lib/OpenOffice.org/share/dict/ooo/
```

A program ezek után már a *Huhyphn* elválasztási mintáit használja.

5.3. Telepítés Scribus alá

A *Scribus* az elválasztási mintákat a

```
/usr/lib/scribus/dicts/
```

könyvtárban tárolja, ide kell másolni a `hyph_hu.dic` fájlt (az 1.1-es verzió nem engedi szimlink használatát).

6. Az elválasztómodul tesztelése

6.1. Tesztelés

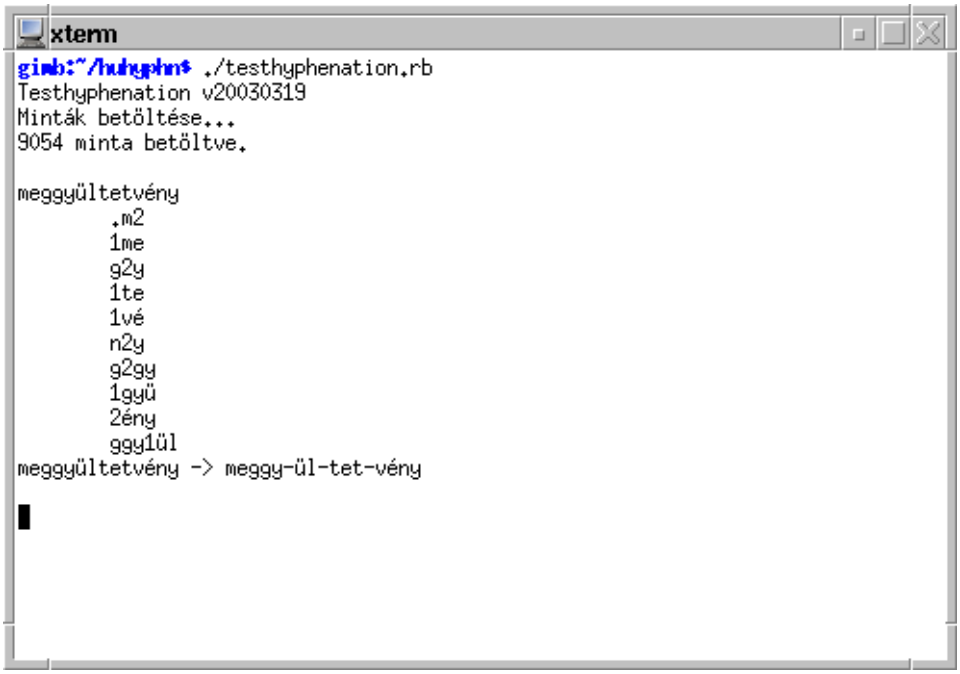
Az elválasztómodul tesztelésére külön alkalmazás szolgál. A *HuhyphnTest* Ruby nyelven írt program, mindössze egy fájlból áll, mely a `test.rb` nevet viseli. A Ruby programnyelvről bővebb információt a `http://www.ruby-lang.org/` oldalon lehet találni, innen tölthető le a rendszer forráskódja is.

A programot elindítva szavakat írhatunk be szöveges terminálon, melyeket kiír elválasztva, valamint az illeszkedő elválasztási mintákat is felsorolja. Így hibás elválasztás esetén egyszerű megállapítani, hogy azt mely helytelen minta okozhatta. Egy üres enter lenyomásával léphetünk ki a programból.

A program működéséhez a TEX-hez készült elválasztási mintafájlt (`huhyphn.tex`) használja. Mivel ebben a fájlban a hungarumlautos ékezetes betűk az EC-kódolás szerintiek, a mintákat beolvasáskor a Latin2-es kódkészletre konvertálja át.

7. Elérhetőség

A `http://www.tipogral.hu/` oldalon érhetőek el a projekt keretében létrehozott elválasztómodulok és programok, melyeket a GPL-licenc feltételei szerint lehet felhasználni.



3. ábra. A *HuhyphnTest* futása szöveges terminálban, az alkalmazott minták egy tabulátorhellyel beljebb íródnak ki az elválasztott szó felett.