

X. Erdélyi Tudományos Diákköri Konferencia

Kolozsvár, 2007. május 26-27.

A GAP rendszer grafikus kezelőfelülete. Alkalmazások

Şuteu Szöllősi István

Babeş-Bolyai Tudományegyetem, Kolozsvár

Matematika-Informatika kar,

Angol informatika szak, IV. évfolyam

Témavezető: **conf. dr. Septimiu Crivei**

Babeş-Bolyai Tudományegyetem, Kolozsvár

Algebra Tanszék

1 Bevezetés

Napjainkban majdnem minden széles körben elterjedt és használt szoftvertől elvárjuk, hogy könnyen, gyorsan kezelhető legyen, hogy az automatizálható műveletek nagy részét „magára vállalja” - spórolva ezáltal a felhasználó idejét (és hajszárait...). A kereskedelmi szoftverek (closed source and proprietary software) nagy része többé-kevésbé megfelel ezeknek az elvárásoknak, időnként sajnos a minőség rovására menő mértékben. Az ingyenes és nyílt forráskódú (open source) programok világában azonban a figyelem nem esik elsősorban olyan tulajdonságokra, mint: felhasználóbarát kezelőfelület, testreszabhatóság (customizability), vonzó külső design (eye-candy), hanem inkább a szoftver minőségére, stabilitására, megfelelő működésére kerül a hangsúly. Az utóbbi években az open source szélesebb körben való elterjedésének köszönhetően már történtek lépések ebben az irányban. Egyre több ingyenes szoftver legalább annyira felhasználóbarát és széles eszköztárral felszerelt (full-featured), mint kereskedelmi megfelelőik (vagy inkább „vetélytársaik”). Ezek főleg azok, melyek általánosabb használatra hivatottak, nagy a felhasználóbázisuk, és / vagy egy nagy cég, szervezet áll a háttérben (ilyen pl. az OpenOffice, a KDE, a GIMP és sok más grafikai szerkesztőprogram, a Firefox böngésző, stb.). Mások viszont, még mindig a parancssoros (command line) üzemmódnál maradtak. Ennek fő oka a kisebb és sokkal műszakibb beállítottságú felhasználóbázis. Míg e programok tökéletesen megfelelhetnek jelenlegi (műszaki, gyakorlott, guru, stb.) felhasználóiknak, az adott operációs rendszert behatóan nem ismerő, programozással nem foglalkozó átlag felhasználót általában elijesztik a parancssoros üzemmóddal, az eleinte meredek tanulási görbével. Így ezek a programok „elszigetelődnek” és csak a technikai beállítottságú felhasználó számára jelentenek megoldást.

Ebbe az utóbbi táborba tartozik a GAP rendszer is [3]. A GAP egy magas szintű számítógépes algebra rendszer, főleg a matematikai kutatásban és a felsőoktatásban használják – kiváló eszköz a csoportok és csoportábrázolások, tesztek, vektorterek, algebrák stb. tanulmányozására, kombinatorikai számítások végzésére, sejtések ellenőrzésére illetve olyan példák generálására, melyek hosszadalmas és bonyolult számításokat igényelnek. A GAP szoftver ingyenes, nyílt forráskódú és szabadon terjeszthető. Rendeltetésének megfelelően jelenlegi felhasználói főleg matematikusok, fizikusok, és e szakok egyetemistái. Közös bennük az, hogy van idejük, türelmük, szakmai háttérük / tapasztalatuk ahhoz, hogy feltelepítsék (általában több leírás, levelezőlista átböngészése után forráskódból lefordítsák), parancssor

üzem módban használják, és megkeressék egy-egy függvény definícióját, használati módját az összesen több ezer oldalra rúgó dokumentációban.

Az GAP grafikus kezelőfelülete megírásának motivációja ez volt: a GAP-et elérhetővé tenni még több kutató, egyetemi tanár és tanuló számára, azok számára is, akik nem rendelkeznek beható informatikai (vagy éppen programozói háttérrel).

A GAP eme „gyenge pontjának” a szerzői is tudatában vannak; ezt bizonyítja az, hogy a GAP honlapján egy grafikus felület fejlesztése a „legkívántabb projektek listáján” szerepel.

2 A GAP-ről részletesebben

A GAP (Copyright © 1987-2004 GAP Group, incorporating Copyright © 1999, 2000 School of Mathematical and Computational Sciences, University of St. Andrews, North Haugh, St. Andrews, Fife KY16 9SS, Scotland; Copyright © 1992 Lehrstuhl D für Mathematik, RWTH, 52056 Aachen, Germany) ingyenes, nyílt forráskódú szoftver. Szabadon terjeszhető és módosítható, a GNU General Public License értelmében (<http://www.gnu.org/licenses/gpl.html>). A GAP név az angol Groups, Algorithms and Programming szavak kezdőbetűiből tevődik össze. A program fejlesztése Lehrstuhl D für Mathematik, RWTH-Aachen-ben kezdődött el, Joachim Neubuser vezetése alatt (1985). A 2.4-es verzió 1988-ban jelent meg, a 3.1-es pedig 1992-ben. 1997-re a szoftver fejlesztése már nemzetközi erőfeszítéssé vált, és a fejlesztés központját St. Andrews-ba helyezték. 1999-ben a 4.1-es verzió már teljesen újratervezett és felújított belső szerkezettel (maggal) jelenik meg. A szoftver iránt növekvő érdeklődést bizonyítja az is, hogy a 4.4-es verzió fejlesztését a Colorado State University-ből (Fort Collins) irányították.

A GAP rendszer fő komponensei:

1. C nyelvben írt mag (kernel), melynek feladatkörébe a következők tartoznak:
 - a memória automatikus kezelése (a Java-hoz hasonlóan, a felhasználónak nem kell figyelnie arra, hogy az objektumok által elfoglalt memóriát felszabadítsa)
 - olyan gyakran használt eljárások, alapszámítások elvégzése, mint például: alapszámítások tetszőleges hosszúságú egész számokkal, ciklikus formában felírt permutációkkal, véges testek elemeivel és szavakkal. Szintén a kernel kezeli a listákat, record típusú és más adatstruktúrákat.
 - a GAP programozási nyelv interpretálása. A GAP nyelv egy külön algebrai feladatok megoldására kifejlesztett programozási nyelv, olyan beépített adattípusokkal mint a

tetszőleges hosszúságú egész számok, permutációk, véges testek elemei.

- fájlok írása és olvasása
- parancssoros felhasználói felület biztosítása (operációs rendszertől független módon)
 2. több ezer, könyvtárakba (library) rendezett algebrával kapcsolatos algoritmus, GAP nyelven implementálva. A GAP nyelv tehát egyszerre a felhasználó programozási nyelve és az nyelv, amelyben a rendszer nagy része írva volt.
 3. csoportelméleti adatokat tartalmazó adattár (az első pár ezer véges csoport, kristályok szimmetriái, karaktertáblák, stb.).
 4. több ezer oldalas dokumentáció HTML, TEX és PDF formátumban

A GAP rendszer természetesen tovább bővíthető, egyrészt új függvények implementálásával (vagy a létezők javításával), másrészt teljesen új csomagok fejlesztésével. Az új csomagok függvényei használat közben dinamikusan feltölthetők.

Az elég kezdetleges súgó rendszer úgy használható, hogy a parancssorba beírjuk a kérdőjel után a keresett kulcsszót, például:

```
GAP> ? Exponent
```

Jelen példánkban ekkor megjelenik az „Exponent” függvény definíciója, a Unix-os **man** parancshoz hasonló módon.

2.1 Alkalmazások

A GAP egy érdekes alkalmazásaként megemlíthetjük az általunk nemrég kifejlesztett ELISA (Extending Lifting Subgroup Algorithms) csomagot. Ennek segítségével eldönthetjük egy adott kommutatív csoportról, hogy beletartozik-e valamelyikébe a következő osztályoknak (vagy meghatározhatjuk egy adott Ábel-csoport minden olyan alcsoportját, amely): direkt összeg (direct summand), esszenciális alcsoport (essential subgroup), hanyagolható alcsoport (superfluous subgroup), zárt alcsoport (closed subgroup), extending vagy lifting, stb.

Az általunk kifejlesztett módszerek gyakorlatilag gráfelméleti algoritmusok, melyek az alcsoport-hálót irányított gráfként kezelik, és az így „tárolt” bennfoglalási relációt használják fel. Az előbb felsorolt pár fogalom (és ezekhez kapcsolódó más modul-elméletből származó fogalmak) definíciója megtalálható például [1]-ben.

komponensre esett.

3.1 A wxWidgets programozói eszköztár

A wxWidgets [8] egy ingyenes és nyílt forráskódú programozói eszköztár (toolkit) és grafikus widget-készlet. Licence (wxWindows license) a GPL-nél engedékenyebb, olyan értelemben, hogy ingyenes és open source programokon kívül fel lehet használni kereskedelmi szoftverek fejlesztésére, a forráskódhoz való hozzáférés biztosítása nélkül is.

Fő rendeltetése eszközt nyújtani a platformfüggetlen grafikus kezelőfelületek (GUI – Graphical User Interface) fejlesztéséhez. Ezt a Java-tól teljesen eltérő módon valósítja meg. A Java kóddal ellentétben, mely a támogatott operációs rendszertől függetlenül ugyanazon a virtuális gépen fut, az általában C++-ban írt wxWidgets alkalmazások „hagyományos” gépi kódra lefordított programok, melyek semmiben sem különböznek a specifikusan az adott operációs rendszerre írt alkalmazásoktól. A wxWidgets a platformfüggetlenséget úgy oldja meg, hogy meghívja az operációs rendszer vagy a desktop környezet saját grafikus widget-kezelő függvényeit. Más szóval úgy viselkedik, mint egy fordító az alkalmazás és az éppen aktuális desktop környezet között.

A programozó csak a wxWidgets függvénykészletével (API – Application Programming Interface) dolgozik, majd forráskódját lefordítja a kívánt platformra (természetesen a platformnak megfelelő wxWidgets verzióval). Például, ha egy közönséges OK gombot akar létrehozni, így jár el:

```
wxButton* button = new wxButton(this, wxID_OK, wxT("OK"),wxPoint(200, 200));
```

Ez aztán a helyzetnek megfelelően „eredeti” Win32, GTK+, Carbon, vagy egyéb gomb lesz. Kihangsúlyozandó, hogy nem a forráskód automatikus átírásáról van szó, hanem arról, hogy jelen esetben a wxButton konstruktor a desktop környezetnek megfelelő gomb-konstruktor fogja meghívni. Ebből a megközelítésből (megvalósítási módból) legalább a következő két fontos előny is származik: először is a wxWidgets alkalmazások kinézetben nem térnek el az operációs rendszer vagy desktop környezet „honos” alkalmazásaitól (native look-and-feel) – így a felhasználó „otthonosabban” fogja kezelni a programot; másodsor gyorsabban futnak és nem olyan memóriaiigényesek, mint a Java alkalmazások (mivel futtatásukhoz nem szükséges a virtuális gép).

wxWidgets segítségével az összes fő operációs rendszerre vagy desktop környezetre fejleszthetünk alkalmazásokat, a lehetőségeket az következő táblázat foglalja össze:

I. Táblázat: platformfüggetlen szoftverfejlesztés wxWidgets-el

wxWidgets API								
wxWidgets port								
wxMSW	wxGTK	wxX11	wxMotif	wxMac	wxCocoa	wxOS2	wxPalmOS	wxMGL
Platform API								
Win32	GTK+	Xlib	Motif/ Lesstif	Carbon	Cocoa	PM	Palm OS Protein APIs	MGL
Operating System								
Windows/ Windows CE	Unix/Linux			Mac OS 9/ Mac OS X	Mac OS X	OS/2	Palm OS	Unix/ DOS

A wxWidgets valójában több, mint egy platformfüggetlen widget-készlet. A grafikus interfész függvényein kívül, a wxWidgets API biztosítja bizonyos adatstruktúrák használatát, fájlok írását / olvasását, periférikus eszközök használatát (pl. nyomtató), a hálózat elérését, több szálon futó programok írását (multi-threaded programming) - általában mindent, amire egy szokványos alkalmazás írásakor szükség lehet – mindezt oly módon, hogy a programozó elvonatkoztathat a különböző operációs rendszerek sajátosságaitól.

3.2 A Scintilla forráskódszerkesztő komponens

A Scintilla [4] egy ingyenes és open source forráskódszerkesztő komponens, licence nagyjából ugyanaz, mint a wxWidgets-é, vagyis korlátlanul felhasználható ingyenes- és kereskedelmi szoftverfejlesztésben egyaránt. Támogatja a forráskód szintaxis-kiemelő színezését (syntax highlighting) az automatikus kiegészítést (autocompletion), a reguláris kifejezésekkel történő keresést, a szöveg nagyítását / kicsinyítését (zooming), könyvjelzőket (bookmarks), behúzások és zárójelek kiemelését (code folding and brace matching), stb. (Megjegyzés: az előbb említett műveleteket nem végzi automatikusan, hanem olyan függvényeket biztosít, amelyekkel ezek viszonylag könnyebben implementálhatók).

Számos forráskódszerkesztő alkalmazás és integrált programozói környezet (IDE – Integrated Development Environment) Scintillára épül (pl. Code::Blocks, Notepad++, MinGW Developer Studio, Activestate Komodo IDE, a Total Commander forráskódnéző plugin-je, stb.).

A Scintilla jelenleg (2007-05-12) 81 programozási nyelvet „ismer”, (munkánknak köszönhetően) a GAP nyelv a 81-dik. Az utolsó (1.73) verzióban ez még nincs benne, hivatalosan a következő verzió megjelenésétől lehet majd használni (persze a fejlesztés alatt álló új verzió letölthető a CVS-ről). [6]

A Scintillában a forráskód színezését (minden nyelvre külön kifejlesztett) színező modulok, „lexer”-ek végzik. Logikailag egy-egy ilyen lexer nem más, mint egy véges automata. Az automata bemenetét a forráskód egy bizonyos sora adja, ennek beolvasása / elemzése alapján az automata különböző állapotokba kerül, minden állapotnak viszont egy bizonyos szín-konfiguráció felel meg. A lexer színező függvényét a Scintilla hívja meg, annak alapján, hogy mely sorok láthatóak a képernyőn, melyeket változtatott meg a felhasználó, stb.

Az új GAP lexer-t a GAP nyelv formai leírása alapján fejlesztettük ki, mely a következő:

II. Táblázat: A GAP nyelv szintaxisa BNF formában

```

<Ident> := `a'|...|`z'|`A'|...|`Z'|`_ '{`a'|...|`z'|`A'|...|`Z'|`0'|
...|`9'|`_' }

<Var> := <Ident>
| <Var> `.' <Ident>
| <Var> `.' `(' <Expr> `)'
| <Var> `[ ' <Expr> `]'
| <Var> `{ <Expr> }'
| <Var> `(' [ <Expr> { ,<Expr> } ] `)'
| <Var> `!.' <Ident>
| <Var> `!.' `(' <Expr> `)'
| <Var> `![' <Expr> `]'

<List> := `[ ' [ <Expr> ] {`,`' [ <Expr> ] } `]'
| `[ ' <Expr> [, <Expr> ] `..' <Expr> `]'
| <List> `` <List> ``

<Record> := `rec(' [ <Ident> `:= ' <Expr> {`,`' <I
<Permutation> := `(' <Expr> {`,`' <Expr> } `)' { `(' <Expr>
<Function> := `function (' [ <Ident> {`,`' <Ident> } ] `)'
[ `local' <Ident> {`,`' <Ident> } `;' ]
<Statements>
end'
| <Ident> `->' <Expr>

<Char> := '<any character> '
<String> := `"' { <any character> } `"'
<Int> := `0'|`1'|...|`9' {`0'|`1'|...|`9'}
<Atom> := <Int>

```

```

| <Var>
| `(' <Expr> `)'
| <Permutation>
| <Char>
| <String>
| <Function>
| <List>
| <Record>
| { `not' } `true'
| { `not' } `false'
<Factor> := { `+' | `-' } <Atom> [ `^' { `+' | `-' } <Atom> ]
<Term> := <Factor> { `*' | `/' | `mod' <Factor> }
<Arith> := <Term> { `+' | `-' <Term> }
<Rel> := { `not' } <Arith> [ `=' | `<' | `>' | `<=' | `>=' | `in'
<Arith> ]
<And> := <Rel> { `and' <Rel> }
<Logical> := <And> { `or' <And> }
<Expr> := <Logical>
| <Var>
<Statement> := <Expr>
| <Var> `:= ' <Expr>
| `if' <Expr> `then' <Statements>
| { `elif' <Expr> `then' <Statements> }
| [ `else' <Statements> ] `fi
| `for' <Var> `in' <Expr> `do' <Statements> `od'
| `while' <Expr> `do' <Statements> `o
| `repeat' <Statements> `until' <Expr>
| `return' [ <Expr> ]
| `break'
| `quit'
| `QUIT'
<Statements> := { <Statement> `;' }
| `;'

```

Ezenkívül minden sor, ami '#'-val kezdődik, illetve egy adott sorban minden karakter, ami a '#' után következik, magyarázatnak (comment) számít. Természetesen a kód színezése szempontjából nem kell minden a *II. Táblázat*-ban szereplő nyelvi elemet külön állapotnak tekinteni. Nagy részüket össze lehet vonni, így a „színező automatának” a következő érvényes állapotai lesznek:

1. SCE_GAP_DEFAULT – az alapállapot (ide tartoznak a szóközök, üres sorok is)
2. SCE_GAP_IDENTIFIER – változók, konstansok nevei
3. SCE_GAP_KEYWORD – a GAP kulcsszavai (*and, break, continue, do, elif, else, end, fi, for, function, if, in, local, mod, not, od, or, quit, rec, repeat, return, then, until, while, QUIT*)
4. SCE_GAP_KEYWORD2 – a GAP fenntartott szavai (reserved words) (*false, true, IsBound, Unbind, TryNextMethod, Info, Assert, SaveWorkspace, fail*)
5. SCE_GAP_KEYWORD3 – a GAP összes többi beépített függvényének neve
6. SCE_GAP_KEYWORD4 – jelenleg kihasználatlan
7. SCE_GAP_STRING – string konstansok
8. SCE_GAP_CHAR – karakter konstansok
9. SCE_GAP_OPERATOR – az alapl műveletek operátorai és a zárójelek
10. SCE_GAP_COMMENT - magyarázat
11. SCE_GAP_NUMBER – egész számok
12. SCE_GAP_STRINGEOL – egy hibajelző állapot, akkor aktuális, amikor a felhasználó nem tette ki egy string- vagy karakter konstans végére a záró idézőjelet

Mint már említettük, az előbb felsorolt állapotok mindegyikéhez egy-egy színkonfiguráció (és betűtípus) tartozik, ezek természetesen változtathatók. A színezésen kívül a lexer feladata a behúzások (code folds) előkészítése is. Ennek megfelelően minden sorhoz hozzárendel egy behúzási szintet (fold level). A *function, do, if* és *repeat* kulcsszavak eggyel növelik a saját meg a következő sorok behúzási szintjét, az *end, od, fi* és *until* kulcsszavak pedig eggyel csökkentik. A behúzó algoritmus (code folder) azokat a sorokat fogja „összecsukni”, együtt kezelni, melyeknek ugyanannyi a behúzási szintje. Így érhető el, hogy függvénydefiníciók, ciklusok, stb. egy egységként „csukódjanak össze” és „nyíljanak ki”.

Megjegyezzük, hogy projektünkben nem használjuk az eredeti Scintillá-t (ami csak a Win32 és GTK+ desktop környezeteken fut), hanem a Scintilla-nak egy wxWidgets-ben újraimplementált változatát, a wxScintilla komponenst [7]. Ebből semmi lényegi különbség nem adódik, mert a wxScintilla függvénykészlete teljes mértékben kompatibilis a eredeti Scintilla API-val.

4 A GAP rendszer új kezelőfelülete

A kezelőfelület tervezésekor a következő irányelveket vettük figyelembe:

- jól kell illeszkedjen a GAP rendszerbe, de ugyanakkor független is kell legyen (a kód szempontjából), hogy a GAP mag programozása és karbantartása külön és párhuzamosan történhessen a kezelőfelület programozásával / karbantartásával
- legyen minél átláthatóbb és „általánosabb”, hogy később esetleg más open source projekt is felhasználhassa
- használata legyen könnyű, tartalmazza azokat a vizuális elemeket, amikhez a felhasználó hozzászokott (toolbar, menük, stb.)
- a súgó legyen bárhol könnyen elérhető és használható

Ötletet merítettünk a Maple és MatLab jól ismert matematikai programok kezelőfelületeinek design-jából is (a GAP szerkezetében és rendeltetésében a Maple-höz hasonlít a legjobban). Ennek megfelelően a GAP kezelőfelülete valamelyest a Maple-éhez hasonlít, de persze jelentős különbségeket / továbbfejlesztéseket is tartalmaz. Abban, ahogy a forráskódszerkesztőt meg lehet hívni a felületről, a Matlab használata juthat eszünkbe, természetesen itt is vannak eltérések és „GAPreszabások”.

4.1 A forráskódszerkesztő

Az előbb említett irányelveknek megfelelően a forráskódszerkesztő (jelenleg) a következőket támogatja:

- a szerkesztett GAP (nyelvben írt) forráskód szintakszisfüggő színezése (syntax highlighting)
- az ugyanabba a logikai egységbe tartozó kódrészletek behúzása - „nyitása és összecsukása” (code folding)
- automatikus kiegészítés (autocompletion) – nemcsak a GAP beépített több ezer függvényére, hanem az éppen szerkesztett kód változóira és egyéb szavaira is
- a betűméret változtatása (zooming)

Egy kulcsszóra bármikor könnyen és gyorsan rákereshetünk. Ehhez mindössze annyi szükséges, hogy megnyomjuk a megfelelő gombot vagy billentyűkombinációt akkor, amikor a

kurzor a keresendő kulcsszó (függvénynév) fölött van. A sűgőt persze külön is előhívhatjuk a menűből.

A GAP rendszer új sűgójának létrehozásakor arra kellett törekednűnk, hogy a meglévűvel teljes mértékben (tartalmilag) kompatibilis legyen – természetesen könnyebben és gyorsabban használható formában. Ezért a már meglévű terjedelmes dokumentációt újraindexeltűk. Windows-on ez egy CHM fájl (standard Microsoft HTML sűgű), a többi operációs rendszeren pedig egy HTB fájl (a wxWidgets saját platformfüggetlen, szintén HTML-re épűlű sűgű formátuma, melyet a wxHtmlHelpController osztály kezel).

A szerkesztűben egyszerre több dokumentummal is dolgozhatunk, ezek külön lapokon (tab) jelennek meg. Új vagy már létezű fájlokat megnyithatunk a menűből, vagy parancssorból a következű módon:

```
> gapedit algebra.gi 567 10
```

ahol „gapedit” a szerkesztű neve, „algebra.gi” a megnyitandű fájl. Az ezután következű két szám opcionális, ha jelen vannak, akkor a szerkesztű a megnyitás után az adott sorba és oszlopba viszi a kurzort és láthatóvá teszi azt. Ez akkor válik fontossá, amikor a kezelűfelületből hívjuk meg a szerkesztűt egy szintaxishiba jelzésekor. A GAP ugyanis megmondja a hiba helyét (sorát és oszlopát), így a szerkesztűt egyenesen arra a helyre lehet „irányítani”.

Megemlíthető még, hogy ha a szerkesztűt parancssorból hívjuk meg egy új fájlnevvvel, akkor nem nyílik meg új ablak, hanem a már létezű ablakban nyílik meg egy új lap. Ezt egy klienszerver szerű IPC (Inter Process Communication) biztosítja. Amikor a szerkesztű elindul, először megnézi, hogy még működik-e egy másik szerkesztű: ha nem, akkor új ablakot nyit, megnyitja a kívánt fájlt és „szerverré válik”; ha igen, akkor nem nyílik meg új ablak, a szerkesztű kliensként a már futű másik szerkesztű-szerverhez kapcsolódik, átadja neki a parancssorban kapott paramétereket és befejezi a pályafutását. A szerverként futű szerkesztű megnyitja egy új lapon a klienstűl „kapott” fájlt.

```

38 #####
39 ##
40 #M FLMLORByGenerators( <R>, <gens> ) . . . . <R>-FLMLOR generated by <gens>
41 #M FLMLORByGenerators( <R>, <gens>, <zero> )
42 ##
43 InstallMethod( FLMLORByGenerators,
44   "for ring and collection",
45   [ IsRing, IsCollection ],
46   function( R, gens )
47     local A;
48     Abelian
49     AbelianGroup
50     AbelianGroupCons
51     AbelianInvariants
52     AbelianInvariantsMultiplier
53     AbelianInvariantsNormalClosureFpGroup
54     AbelianInvariantsNormalClosureFpGroupRrs
55     AbelianInvariantsOfList
56     AbelianInvariantsSubgroupFpGroup
57   );
58
59 InstallOtherMethod( FLMLORByGenerators,
60   "for ring, homogeneous list, and ring element",
61   [ IsRing, IsHomogeneousList, IsRingElement ],
62   function( R, gens, zero )
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81 #####
82 ##
83 #M FLMLORWithOneByGenerators( <R>, <gens> ) unit. <R>-FLMLOR gen. by <gens>
84 #M FLMLORWithOneByGenerators( <R>, <gens>, <zero> )
85 ##
86 InstallMethod( FLMLORWithOneByGenerators,
87   "for ring and collection",
88
89
90
91
92
93
94
95
96
97
98
99
100

```

A GAP új forráskódszerkesztője használat közben

4.2 A kezelőfelület

A GAP kezelőfelülete egyfajta hibrid a Maple által is képviselt munkalap (worksheet) típusú GUI és a parancssoros üzemmód között – igyekezve megtartani mindkettőnek az előnyös tulajdonságait. Működési módját a következőkben foglaljuk össze:

- támogatja a GAP parancsok színezését, az automatikus kiegészítést, stb. (mivel ugyanazt a wxScintilla komponents használja mint a szerkesztő)
- a GAP „válasza” behúzható, (a lap azon részei, melyek a GAP output-ját tartalmazzák, nem szerkeszthetők, viszont a felhasználó által beírt részek, parancsok igen)
- a forráskódszerkesztőhöz hasonlóan a sűgó könnyen hívható bármilyen kulcsszóra
- a GAP kernel számítása egy gombnyomással leállítható
- munkánkat (a GAP-nek beírt parancsokat a kapott eredményekkel együtt)

elmenthetjük majd újra megnyithatjuk (anélkül hogy a GAP a számításokat újra elvégezné). A munkalapot egy új formátumban (GWS – GAP WorkSheet) menti le, ami nem más, mint egy GAP parancsokat, eredményeket és más beállításokat tartalmazó XML fájl. Amikor egy lementett GWS fájlt újratöltünk, azokat a parancsokat, amiket a felhasználó még nem futtatott le újra, a kezelőfelület a lapon más színnel jelöli, hogy a felhasználó számára egyértelművé tegye, mely változók nincsenek még inicializálva, stb.

A kezelőfelület összekapcsolása a GAP maggal lényegében a GAP I/O átirányításával történik, így elértük azt is, hogy a GAP kernel forráskódja függetlenül legyen a felülettől.

4.3 TODO

Tennivaló még akad bőven:

- még egy pár „feature” implementálása, mint pl. a könyvjelzők kezelése (bookmarks); esetenként a már meglévők továbbfejlesztése (pl. a színek, betűtípusok állítgatása, stb.)
- az interfész és szerkesztő kezelését részletesen elmagyarázó / bemutató súgó megírása
- a programok alapos tesztelése (különösen a Mac OS operációs rendszeren) és javíthatása
- a forráskód feltöltése a SourceForge-ra [6] (és egy dekikált honlap létrehozása valószínűleg szintén a SourceForge-on), hogy bárki aki érdekelt, bekapcsolódhasson a fejlesztésbe / karbantartásba

5 Könyvészet

[1] S. Crivei and S. Suteu-Szollosi, Subgroup Lattice Algorithms Related to Extending and Lifting Abelian Groups, International Electronic Journal of Algebra, Volume 2 (2007) 1-18

[2] S. Crivei, G. Olteanu and S. Suteu Szollosi, GAP algorithms related to extending and lifting abelian groups, Undeposited Contributions section of the GAP webpage.

[3] GAP – Groups, Algorithms, Programming – a System for Computational Discrete Algebra. <http://www.gap-system.org>

[4] Scintilla - A free source code editing component for Win32 and GTK+. <http://www.scintilla.org>

[5] J. Smart, K. Hook, S. Csomor, Cross-Platform GUI Programming with wxWidgets,

Prentice Hall, 2005

[6] SourceForge.net – A collaborative revision control and software development management system. <http://sourceforge.net>

[7] wxScintilla – A wrapper around the Scintilla edit control. <http://sourceforge.net/projects/wxscintilla/>

[8] wxWidgets – A free cross-platform widget toolkit. <http://www.wxwidgets.org>