Regular Chains Development: Quick Start Guide

Paul Vrbik[2]

[2]University of Western Ontario

June 22, 2012

Question.

Installation

.bashrc (Linux) or .bash_profile (OS X)

```
1 #RMAPLE stuff
2 export MAPLE_ROOT=/Applications/rmaple
3 export PATH=$PATH:$MAPLE_ROOT/bin
4 export MAPLELIB=/Users/pvrbik/SVN/DevRegChains/mapleP4/lib
5 export PATH=$PATH:~/depot_tools
6
7 alias mysmaple='smaple -i $HOME/.mapleinit.smaple'
8 alias mysload='sload -I $MAPLELIB $MAPLELIB/RegularChains/
9     src/RegularChains.mpl'
10 alias mysmarch='smarch -c $MAPLELIB/RegularChains.mla'
11 alias mystester='tester -count -maple="smaple -i $HOME/
12     .mapleinit.smaple" -dir  $MAPLELIB/RegularChains/tst'
13 alias rmaple='mysmaple';
14 alias singletester='tester -count -longest=20000'
```

Installation

In `/SVN/DevRegChains/mapleP4/lib/RegularChains`

`.mapleinit` Initialization file for Maple.

`.maplesinit.smaple` Initialization file for sMaple (developer Maple).

The directories you care about

In `/SVN/DevRegChains/mapleP4/lib/RegularChains`

- `src` Not a directory, rather a symbolic link to `dev`.
- `dev` Contains all `.mm` files.
- `tst` Testing files.

The files you care about

In `/SVN/DevRegChains/mapleP4/lib/RegularChains`

`RegularChains.mpl` Giant module defining library.

`user.mm` 11,000 lines (so far). Read by RegularChains.mpl, defines user functions that do argument checking and error reporting etc. (i.e. UI = user level commands)

`you.tst` Create `tst/you.tst` for testing (absolutely necessary). For this I refer you to Maple's advanced programming guide.

`you.mm` The files containing your TRD procedures.

High Level View (dishonest) — RegularChains.mpl

```
1  RegularChains := module()
2      option package;
3      export UI_*;
4      local  TRD*;
5
6      TRDothers := proc() ... end proc;
7      TRDyours  := proc() ... end proc;
8
9      SubPackage := module()
10         option package;
11         export Cmd*;
12         Cmd* := proc()
13             error processing (correct arguments etc);
14             return TRDyours(args);
15         end proc;
16     end module;
17 end module;
```

High Level View (honest) — RegularChains.mpl

```
1 $include <RegularChains/src/macros.mm>
2 RegularChains := module()
3     option package;
4     export UI_*;
5     local  TRD*;
6
7 $include <RegularChains/src/user.mm>
8 $include <RegularChains/src/*.mm>
9
10 # Note: '$' must appear in first column
11 end module;
```

UI Commands — `user.mm`

- Read by the RegularChains module.
- Contains all user level commands.

USER.MM

```
1 ...
2 SubPackage := module()
3     option package;
4     export UI_*;
5
6     UI_* := proc()
7         error processing (correct arguments etc);
8         return TRDyours(args);
9     end proc;
10
11 end module;
12 ...
```

Macros: UI_* explained! — macros.mm

We use macros because:

- they make changing user level names trivial, and
- make our error messages consistent (supposedly).

MACROS.MM

```
1 macro (
2   # Submodules
3     UI_MATRIX    = MatrixTools,
4     UI_CHAIN     = ChainTools,
5     UI_AGT       = AlgebraicGeometryTools,
6       ...
7     UI_YOURS     = YourSubModule
8 );
```

MACROS.MM

```
# 8) AGT : the AlgebraicGeometryTools module
macro(
    AGT_TWM = TriangularizeWithMultiplicity,
    AGT_SCM = SingleChainMultiplicity,
    AGT_TCO = TangentConeAtOrigin,
    AGT_TCC = TangentCone,
    AGT_RCP = RegularChainAtPoint,

    AGT_LTE_LTE = LazyTaylorExpansion,
    AGT_LTE_WRL = NewWorld,
    AGT_LTE_FRC = ExpandLTE,
    AGT_LTE_XPN = ExpansionAtRegularChain
):
```

MACROS.MM

```
1 macro(
2     ERR_VAR_UNDERSCORE   = "variable names should not contain
3     ERR_VAR_UNQIUE       = "%1 parameter should contain
4        variables distinct from those of the polynomial ring.",
5     ERR_VAR_INSUFF       = "%1 parameter contains an inssuff
6        icient number of variables.",
7     INVALID_OPT          = "invalid keyword option."
8 ):
```

Usage:

```
1   if nops(PlaceHolderVar) < nops(var) then
2       error ERR_VAR_INSUFF, "second";
3   end if;
```

"Second parameter should contain variables distinct from those of the polynomial ring."

There is no manual...

Ok fine. I'll make one.