MetiTarski Tips from the Field

William Denman
`william.denman@gmail.com`

October 20, 2014

1 Introduction

This is a short document describing some insights into using MetiTarski based on my experiences using it extensively during both my master's and PhD degrees.

2 Tips

2.1 Coding Suggestions

The modified TPTP syntax that MetiTarski uses is definitely not easy on the eyes. My recommendation is to use Emacs along with a custom built mode (metit-mode) for syntax highlighting. It is available at `https://code.google.com/p/metitarski/wiki/MetiTarskiMode`. Emacs also does bracket identification and completion, which you will most surely need for complicated (and large) conjectures.

- Use the `.tptp` extension for your MetiTarski files.
- Whitespace is ignored by the parser, use it to nicely format longer formulas.
- MetiTarski variable names must be uppercase.

2.2 High Variable/Dimensional Problems

Beyond 4 or 5 continuous variables, there is very little hope for MetiTarski in finding a proof. However, there is one critical command line argument when used with the z3 EADM, that has allowed for conjectures of 9 variables to be proved. This is the `--RCFtime` parameter. Experiments show that an RCFtime of 1000 (ms) leads to a nice balance between conjectures proved and conjectures given up on.

2.3 False Implies Anything

Always be wary when a conjecture that has never been proved or usually takes a significant amount of time, suddenly is proved in under a second. This normally indicates the problem has been made trivial in some way. One example is a false statement in the left hand side of an implication. MetiTarski will return Theorem for $![X] : X < 0 \,\&\, X > 0 => X = 0$, since false implies anything.

2.4 autoInclude

One of the most useful command line arguments is `--autoInclude`. I've used it extensively in my verification system QUANTUM that generates thousands of conjectures to be proved. Instead of having to parse the conjectures and manually insert `include` statements. MetiTarski takes care of this automatically. There are cases however where the axioms chosen by autoInclude are not tight enough. In a case like this, you can move on to using `--autoIncludeExtended` and then `--autoIncludeSuperExtended`. Finally, if the conjecture is still not proved, manually putting in the include statements will be required.

Note: `include` statements must end with a period.

2.5 runmetit.pl

The easiest way to use the `runmetit` script is to place it on your $PATH. To run it with `autoInclude` and `RCFtime`, type

```
runmetit.pl --options="--autoInclude --RCFtime 1000"
```

Good luck! And please contact the team with tips of your own..