OTableau
A LaTeX-friendly editor for Optimality Theory

by Julien Eychenne

October 7, 2007

1   Overview

OTableau is a cross-platform WYSIWYG[1] editor for Optimality Theory (OT) tableaux. Its features include :

- LaTeXcode generation, with optional cell shading

- auto-location of fatal violations

- auto-location of winner(s)

- factorial typology

- import/export of CSV (tab separated files)

- cross-platform (tested under Windows, Mac OS X and Linux)

- dropping of CSV and OTABLEAU files on OTableau's icon (Mac OS X only)

New features in version 1.8 are :

- Display and LaTeX generation of relative tableaux.

- Multiple window interface, similar to that used by Microsoft Word.

- Streamlined LaTeX pasting.

- Recursive constraint demotion.

- Limited import and export of OTSoft tableaux–the first tableau in each table only.

---

[1]What You See Is What You Get

2   Installation

OTableau is written in Python. Python is a programming language which is free (open source) and which runs on many platforms. The drawback is that you need Python to be installed on your computer for OTableau to work.

2.1   Windows

2.1.1   The easy way (without Python)

Download (you should have done it already if you are reading this. . .) the stand alone version for windows. Unzip it: this will produce a directory called `OTableau`. Move this directory wherever you want on you computer (for instance, in `Program files`), and run the program `OTableau(.exe)`. I suggest you add a shortcut pointing to it on your desktop. For the program to work properly, it is crucial that all files remain in the same directory. If for some reason this version didn't work (it has only been tested with Windows XP), try with the instructions below.

2.1.2   The usual way (with Python)

Getting Python working under Windows is easy if you follow these instructions. First, download ActivePython from Activestate[2]. Unless you know what you're doing, you want to download the MSI package for Windows x86. Install ActivePython as any Windows program, then unzip `OTableau.zip`: this will create a folder named OTableau. Move this folder wherever you want (for instance in your Documents). To run OTableau, you only need to double-click the `OTableau.pyw`[3] file (make sure extensions are not hidden). To make things even easier, you can create a shortcut on your desktop.

To read the documentation from within OTableau, you will need to have Acrobat Reader installed. OTableau will check its presence every time you run it. This behaviour can be disabled by editing the `config` file (with blocnote) and switching `check_acrobat` to `no`. Make sure you save the file as a plain text file, with no extension whatsoever.

2.2   Mac OS X

OTableau is provided as a (huge 5 MB) standalone bundle for Mac OS X. It reportedly works on Panther and Tiger. This bundle includes everything

---

[2]`http://www.activestate.com/`
[3]Do not try to run `otableau`, this file is for Unix systems.

you need.

If for some reason it didn't work for you, you may want to try the running the source directly. You can find it at
`http://otableau.googlecode.com/files/otableau-src-1_8.zip`. If you have OS 10.4 or higher, this is all you need. For 10.3 or earlier, you also need to install the newest version of Python. You can find the installer at `http://python.org/download/`.

You should be able to run OTableau by double-clicking `otableau.py`. If you experience any trouble, please contact me (see my address below).

2.3 Linux/Unix

On other Unix-like systems, just check that you have python and Tkinter installed. This can be done by calling `python` from the command line, and then inputting `import Tkinter`. If there is no error, everything is right! Make sure otableau is executable (`chmod +x otableau`) and run it: `/path/to/otableau`. Contact me if you have any problem.

The default PDF viewer is Acrobat reader (`acroread`). If you prefer xpdf or kpdf (or whatever else), you may want to change the `unix_pdf_viewer` in the `config` file.

3 Usage

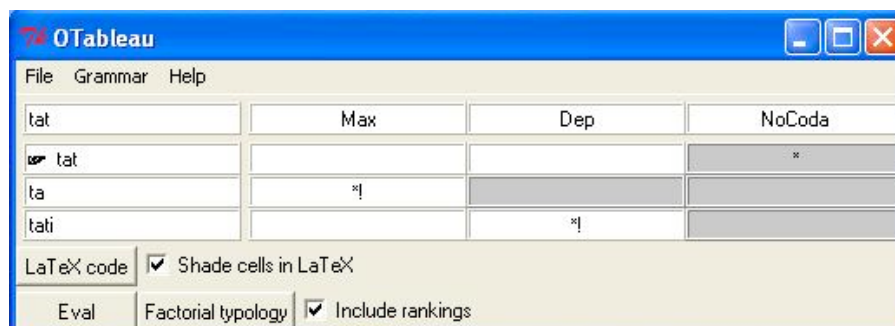Using OTableau is straightforward once you get it installed. The main window looks like the one in figure (1):

Figure 1: Main window under Windows

3.1 Editing your grammar

You can use the example grammar from the `Help` menu. The first row contains constraints, with the input in the leftmost cell. Other rows contain candidates and their constraint violations. Modifying your grammar is made by using the commands in the `Grammar` menu.

You do not need (actually, you must not!) type in fatal violations: they are computed when you click the `Eval` button (or click `Eval` from the `Grammar` menu. Optimal candidate(s) are shown with a yellowish colour.

You can save and open grammars in `.otableau` format. OTableau files are actually zipped XML files. You also have the opportunity to import/export grammars as CSV files (tab separated). You can also import and export OT-Soft CSV files, but unfortunately only the first tableau in each file.

3.2 Generating LaTeXcode

Pressing the `LaTeX code` button will display the LaTeXsource code corresponding to your grammar. It will vary according to your grammar being evaled or not and cell-shading being enabled or not. Here is the source code generated for a simple grammar:

```
\begin{tabular}{|rrl||c|c|c|}\hline
\multicolumn{3}{|c||}{/\ipa{tEt}/} & \textsc{Max} &
\textsc{Dep} & \textsc{NoCoda} \\ \hline\hline
\LCC
&& & &\lightgray &\lightgray \\ \hline
a. & & \ipa{tE} & *! &   &  \\ \hline
\ECC
\LCC
&& & & &\lightgray \\ \hline
b. & \hand & \ipa{tEt} &   &  & * \\ \hline
\ECC
\LCC
&& & & &\lightgray \\ \hline
c. & & \ipa{tEti} &   & *! &  \\ \hline
\ECC
\end{tabular}
```

And here is the result once it has been compiled:

/tɛt/		Max	Dep	NoCoda
a. tɛ		*!		
b. ☞ tɛt				*
c. tɛti			*!	

Candidates are indexed automatically. Because of that, you cannot generate LaTeXcode for a grammar that has more than 26 candidates. This should not be much of a problem, though.

It is also noteworthy that candidates and the input are automatically surrounded with an IPA environment, while constraints are written in small capitals.

3.3   Recursive Constraint Demotion

Recursive constraint demotion support is still rudimentary. There is no way to display strata consisting of multiple constraints even when the algorithm generates one. To use constraint demotion, identify the candidate that will win once constraint demotion has finished. Then move this candidate to the top so that it is the first candidate in the tableau. Finally, choose `Recursive Constraint Demotion` from the `Grammar` menu.

4   Acknowledgements

OTableau has been made possible thanks to the advice found on Doug Arnold's "LaTeX for linguists" website, especially the page dedicated to OT tableaux[4]. I want to express my gratitude to all the contributors.

People from the forum MacBidouille must also be thanked for testing the Mac bundle.

5   Contact

You can contact me at `jeychenne@gmail.com`. I would appreciate if you notified me of any bug or ways to improve the tool. While bugs will be fixed as soon as possible, new features will not be added before October 2006.

---

[4]`http://www.essex.ac.uk/linguistics/clmt/latex4ling/OT/4ling/OT`