

# The `ot-tableau` package

Adam Baker

April 11, 2010

## 1 Introduction

The `ot-tableau` package makes it easy to create beautiful optimality-theoretic tableaux. The  $\text{\LaTeX}$  source is visually very similar to a formatted tableau, which makes working with the source code painless (well, less painful). A variety of stylistic variants can be modified to suit personal taste.

<code>/stap/</code>	<code>*COMPLEX</code>	<code>ANCHOR-IO</code>	<code>CONTIGUITY-IO</code>
a. <code>stap</code>	<code>*!</code>		
$\mathbb{R}$ b. <code>sap</code>			<code>*</code>
c. <code>tap</code>		<code>*!</code>	

```
\begin{tableau}{c:c|c}
\inp{\ips{stap}}      \const{*Complex} \const{Anchor-IO} \const{Contiguity-IO}
\cand{stap}          \vio{*!}           \vio{}           \vio{}
\cand[\HandRight]{sap} \vio{}           \vio{}           \vio{*}
\cand{tap}           \vio{}           \vio{*!}        \vio{}
\end{tableau}
```

Pertinent features:

- The package introduces the `tableau` environment.
- Indicate solid or dashed lines between constraints with `\begin{tableau}{c:c|c}`. A solid line is indicated by a pipe, a dashed line with a colon.
- The input is specified with the `\inp` command. (Here the `\ips` macro is being used to render the text using TIPA and put it within slashes.)
- Indicate the constraints with the `\const` command.
- Add a candidate with the `\cand` command. An optional argument can be used to annotate the candidate (like with  $\mathbb{R}$ , `\HandRight`)
- Violations are indicated with the `\vio` macro. You need to include these commands even when there are no violations.
- Use of whitespace is optional. Keeping the columns aligned in the source code, though, makes the tableau much easier to edit.

Very large tableaux are not much more difficult—for instance in Table 1.


Input: /RED, ulampoy/	RED= $\sigma$	DEP-IO	MAX-IO	ONSET	No-CODA	ALIGN-RED-L	MAX-BR
a. $\text{u-la-lam.poy}$				*	**	u	mpoy
b. $\text{u-lam-lam.poy}$				*	***!	u	poy
c. $\text{u-u.lam.poy}$				**!	**		lampoy
d. $\text{ul-u.lam.poy}$				**!	***		ampoy
e. $\text{la-lam.poy}$			*!		**		mpoy
f. $\text{tu-tu-lam.poy}$		*!			**		lampoy
g. $\text{u.l-u.lam.poy}$	l!				**		lampoy
h. $\text{u-lam.poy-lam.poy}$	p!oy				****	u	

Table 1: After Kager (1999:229).

## 2 Parameters

### 2.1 Changing formats

The default `\cand` and `\const` commands typeset the argument in IPA (TIPA) and small caps, respectively. Though this is generally appropriate, there are also commands `\cand*` and `\const*` that apply no formatting. The following code and tableau illustrate this.

Illustration	C1	C2
a. Option 1	*!	
 b. Option 2		*


```
\begin{tableau}{c|c}
\inp{Illustration}          \const{C1}  \const*{C2}
\cand*{Option 1}           \vio{*!}   \vio{}
\cand*[\HandRight]{Option 2} \vio{}     \vio{*}
\end{tableau}
```

Usage of `\const*` is illustrated in the next section.

### 2.2 Controlling cell shading

There are two systems in use for shading OT tableaux. One system is to shade cells in a row after the crucial violation. The other system is to shade an entire column, if the associate constraint generates no crucial violations.


`ot-tableau` will do cell-shading automatically, if you invoke the `\ShadingOn` command before creating the tableau.<sup>1</sup> (There is also `\ShadingOff` if you wish to turn this feature off for subsequent tableaux in the document; this is the default setting.) This tableau...

/ba/	*VCD0BS	IDENT-IO-[nas]
a. ba	*!	
 b. pa		*

...is produced by the following code...

```
\ShadingOn
\begin{tableau}{c|c}
\inp{\ips{ba}}          \const{*Vcd0bs}  \const*{\textsc{Ident-IO}-[nas]}
\cand{ba}              \vio{*!}         \vio{}
\cand[\HandRight]{pa}  \vio{}          \vio{*}
\end{tableau}
```

The alternative is to shade an entire column by using ‘s’ instead of ‘c’ in the argument to the `tableau` environment:

/ba/	*VCD0BS	IDENT-IO-[nas]
a. ba	*!	
 b. pa		*

...is produced by...

<sup>1</sup>More specifically, `ot-tableau` will look for the exclamation point. You have to provide the exclamation point.

```

\begin{tableau}{c|s}
\inp{\ips{ba}}      \const{*VcdObs}  \const*{\textsc{Ident-IO}-[nas]}
\cand{ba}           \vio{*!}        \vio{}
\cand[\HandRight]{pa} \vio{}          \vio{*}
\end{tableau}


```

You can also mix the approaches, which is illustrated in the code for Table 1. (That code is not printed in this manual, but it's available in the .tex version of this file.)

You can control the darkness of the shading using a command like `\SetCellShading{0.4}`. A value of 1 corresponds to white, 0 to black. The default is 0.9.

### 2.3 Symbol position

Some people prefer the “finger-of-optimality” to go after the letter. This can be done by using the `\LetterBeforeFinger` command:

/ba/	*VCD OBS	IDENT-IO-[nas]
a. ba	*!	
b.  pa		*

...is produced by the following code...

```




\LeterBeforeFinger
\begin{tableau}{c|c}
\inp{\ips{ba}}      \const{*VcdObs}  \const*{\textsc{Ident-IO}-[nas]}
\cand{ba}           \vio{*!}        \vio{}
\cand[\HandRight]{pa} \vio{}          \vio{*}
\end{tableau}

```

`\FingerBeforeLetter` is also available. This is the default.

### 2.4 Different symbols

Using the optional argument to `\cand`, you can add any annotation to a candidate. The following will be a trip down Memory Lane for some people:

/ba/	*VCD OBS	IDENT-IO-[nas]
(FFC) a. ba	*!	
 b. pa		*
 c. ta		*
 d. sa		*

Generated by...

```

\begin{tableau}{c|c}
\inp{\ips{ba}}      \const{*VcdObs}  \const*{\textsc{Ident-IO}-[nas]}
\cand[(FFC)]{ba}    \vio{*!}        \vio{}
\cand[\HandRight]{pa} \vio{}          \vio{*}
\cand[\ding{96}]{ta} \vio{}          \vio{*}
\cand[\HandLeft]{sa} \vio{}          \vio{*}
\end{tableau}

```

For most needs, `\HandRight` and `\HandLeft` will be sufficient.