

PostScript versions of Dynkin diagrams

This note describes a package of encapsulated PostScript figures of Dynkin diagrams. They were first designed in response to a question of Tonni Springer's in October, 1997. They will be available on the Internet at

www.math.ubc.ca/people/faculty/casselmann/frivolities.

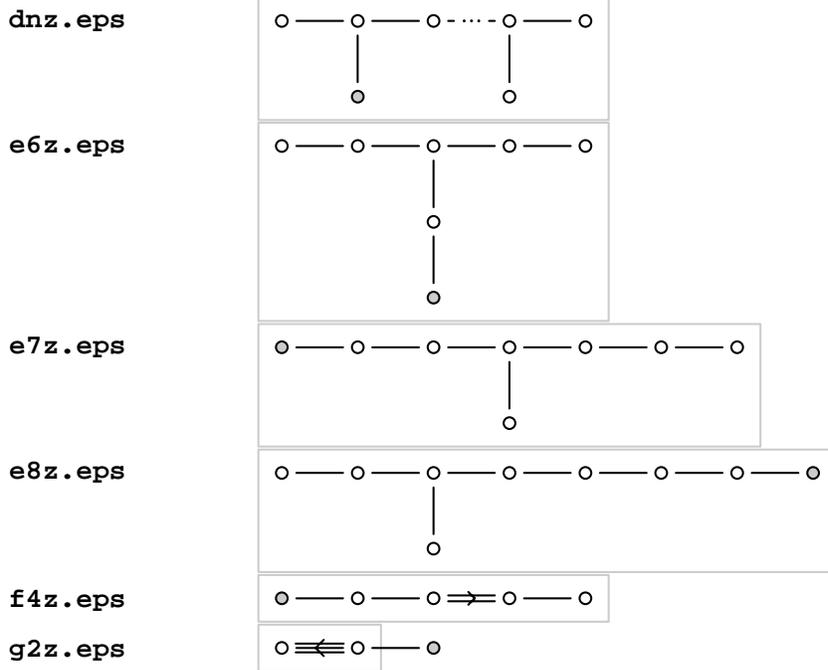
What comes first is a list of all the figures, with their file names at the left. In this list their bounding box envelope is also shown, but of course the figures themselves do not include this.

Here the plain Dynkin diagrams for the finite root systems:

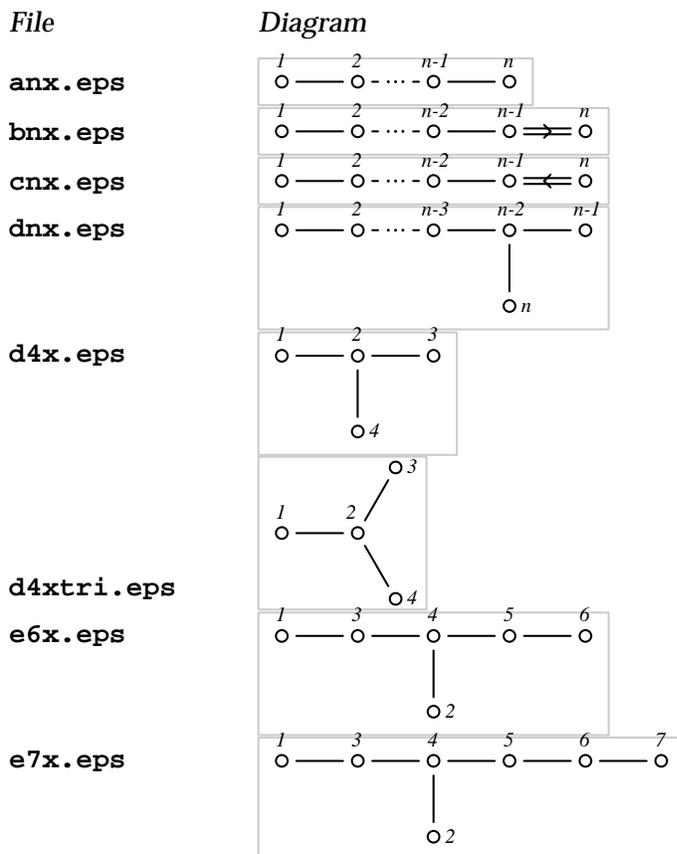
<i>File</i>	<i>Diagram</i>
an.eps	
bn.eps	
cn.eps	
dn.eps	
e6.eps	
e7.eps	
e8.eps	
f4.eps	
g2.eps	

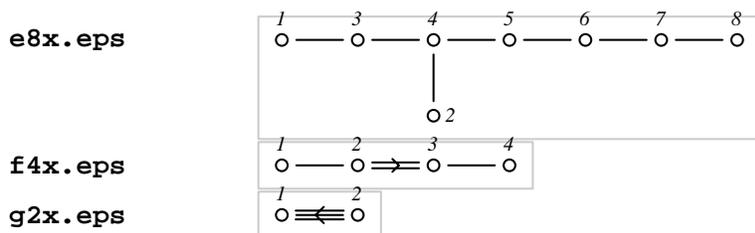
Here are the corresponding affine diagrams:

<i>File</i>	<i>Diagram</i>
anz.eps	
bnz.eps	
cnz.eps	

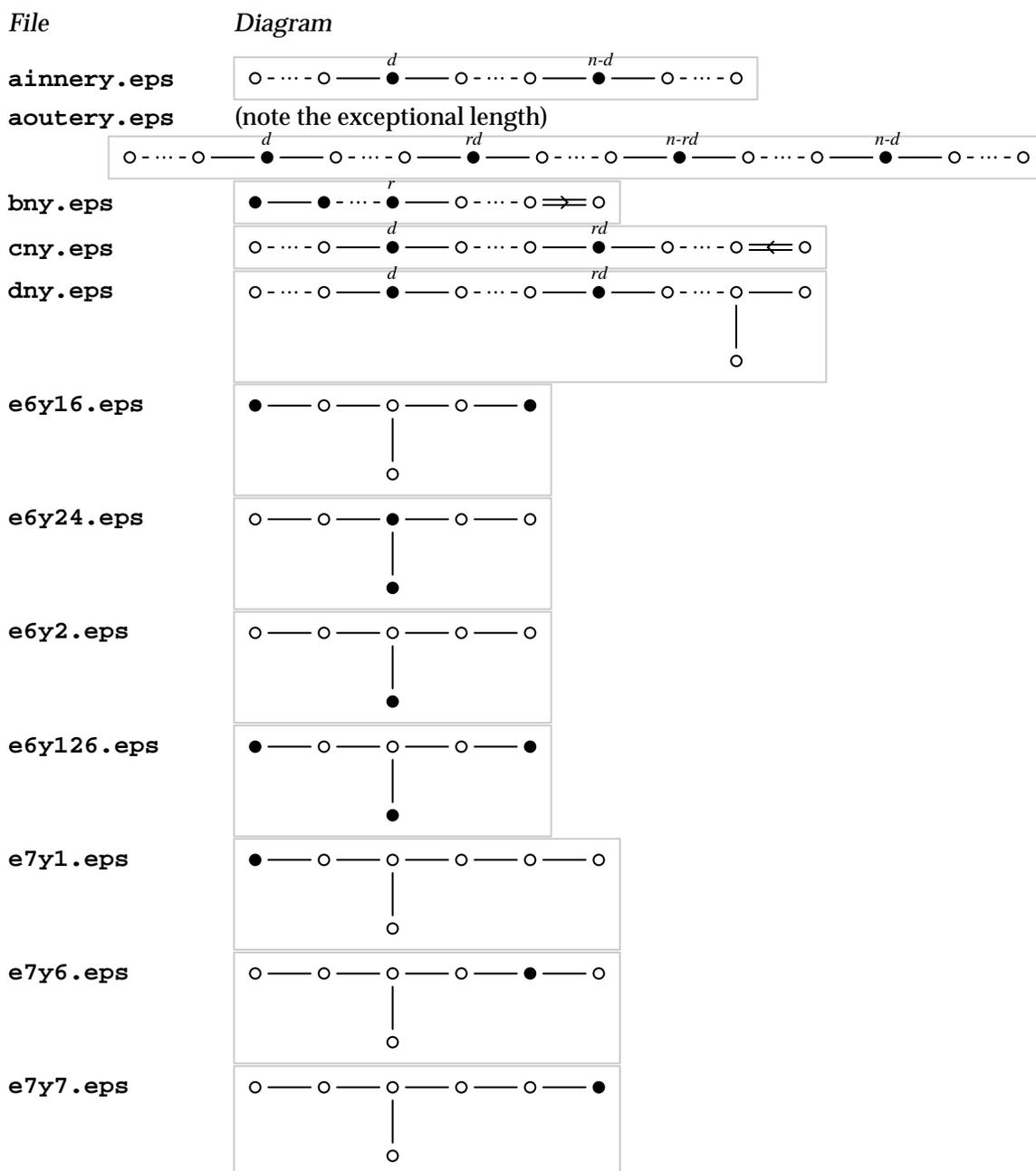


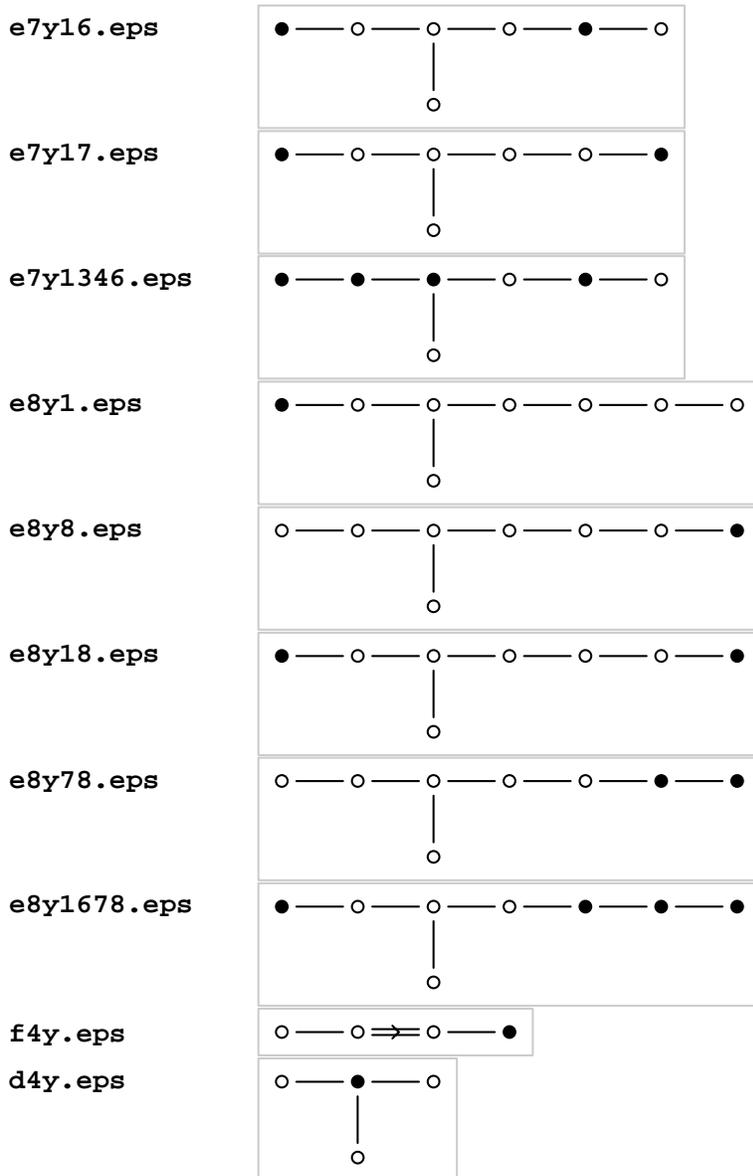
Here with the nodes numbered, following the conventions of Bourbaki's book on Coxeter groups and root systems:





Tits diagrams (in Springer's style):





Each diagram was created originally by hand in a file `*.EPS` which is almost an acceptable encapsulated PostScript file. The figures in this package have the property, however, that several of the parameters are modifiable, but that the bounding box of the final figure depends on the explicit choices made. Therefore these files contain also a line or two giving the dimension of the figures in units of the edge length, one of the variable parameters. The `*.EPS` file is then read and processed by a program I call `sprinc`, which sets the size of the bounding box and includes explicitly all files included implicitly in the `*.EPS` file. Two of these files are `dynkin.inc`, which contains general routines for drawing Dynkin diagrams, and `setup.inc` which fixes certain parameters of the figures such as the width of lines, radii of Dynkin nodes, size of gaps, the fonts used for indexing, etc.

The program `sprinc` is a binary executable, and hence must be recompiled on each type of computer. The source file for `psinc` is in the language of the lexical reader `flex`, and is contained in the directory `src`. It can be recompiled into `C` and thence to a binary executable on any UNIX system which has `flex` loaded. The executable `psinc` should then be put into a directory in the execution path.

The final `*.eps` file produces a `\vbox` with the `tex` command `epsfbox{*.eps}`. But to use the `epsf` macros you must put `\input epsf`. This is part of the standard distribution of `dvips`, available for nearly all computers. There are many ways to input PostScript files in `tex` files, but this `epsf` package supplied with Rokicki's `dvips` is by far the best.

Note that in order to put a figure in a line, you put the `\vbox` produced by `epsf` into an `\hbox`. Note above how `\raise` can put them at the correct level.

The way the figures are named should be apparent, except that there are three series with an arbitrary `x`, `y`, and `z` in their names.

If you want to make a new batch of figures, say with different dimensions, you have several options:

(1) *Most efficient*: You change the line in the file `sprinc.fl` where the edge length and margin are set. Then you remake the binary `sprinc`, and finally rerun it on all the `*.EPS` files.

(2) *Medium efficient*: You add to each `*.EPS` file one or two lines like

```
EDGELENGTH 30
MARGIN 10
```

(units in Adobe points) and rerun `sprinc` them to produce the `*.eps` files.

(3) *Not at all efficient*: Change directly the `*.eps` files.

It should also be apparent from looking at one of the `*.eps` files how to draw completely new figures.