Documentation and examples of grapher.py

Manuel Gutierrez Algaba
irmina@ctv.es
http://www.ctv.es/USERS/irmina/texpython.htm

February 1999

1 Copyright issues

2 Beg and cries

If you belong to an important organization:IBM, XeroX, Borland,... (all of them are registered trademarks), then it'd be FINE if you email me saying me how much you like it. Then I can include this CONGRATULATIONS in my curriculum.

3 Introduction

This document explains all the details for the user of grapher.py.

3.1 grapher.py

When I wrote grapher.py there wasn't any automated utility for doing graphs (state machines, data flow diagrams). Of course, you can write them directly in LaTeX or using xfig or something like that. But grapher.py has two major advantages, it's easier to use and it's faster to 'write'. Besides it can be used as an interface by CASE programs. The kinds of available drawings are good for:

- Expressing the flow of control and data.
- Expressing the different parts something can be divided into.

And this is an utility written in python.

`http://www.python.org`

I imagine that it could be written in TEX but It's 3 times easier to use python. And what's more important TEX programmers have a model , if they want to do the translation.
Another point, the python code could be improved ...

And, it generates tex code. So if you want it in postscript, gif or whatever, use the programs dvi, gs or grab directly from a window!

4 How to use it

This piece of python code illustrates the full capabilities of it, basically, you can put several states and transitions. This code:

```
st1 = graphstate("wake up")
st2 = graphstate("breakfast")
st3 = graphstate("homework", "done?")
st4 = graphstate("nice ", "class day")
st5 = graphstate("bad", "class day")
st6 = graphstate("have lunch")
st7 = graphstate("go to","party")
st8 = graphstate("have fun")
st9 = graphstate("next day")
#st1.do_size(2)
st1.rel(n=st2)
st2.rel(n=st3)
st3.rel(n=st4, l='Yes', g=[(st5,'No')])
st4.rel(n=st6,f=st5)
st5.rel(n=st6)
st6.rel(n=st7)
st7.rel(n=st8)
st8.rel(n=st9,g=[(st8, "have fun")])
st9.rel(g=[(st1,"Sleep")])

st1.generate_gpic_code("test1.gpic")
```

generates this :

```
.PS
down
arrowhead=7
arrow
L0:  ellipse "wake up"  ht 0.5 wid 0.84
move 0.2
L1:  ellipse "breakfast"  ht 0.5 wid 1.08
move 0.2
L2:  ellipse "homework" "done?"  ht 0.5 wid 0.96
move 0.5
L3:  ellipse "nice " "class day"  ht 0.5 wid 1.08
move 0.2
L4:  ellipse "bad" "class day"  ht 0.5 wid 1.08
move 0.2
```

```
L5:  ellipse "have lunch"  ht 0.5 wid 1.2
move 0.2
L6:  ellipse "go to" "party"  ht 0.5 wid 0.6
move 0.2
L7:  ellipse "have fun"  ht 0.5 wid 0.96
move 0.2
L8:  ellipse "next day"  ht 0.5 wid 0.96
move 0.2
arrow from L0.s to L1.n
arrow from L1.s to L2.n
arrow "Yes" rjust from L2.s to L3.n
L21 :arc -> from L2.w to L4.w
sprintf("No") ljust at L21.w +(0.2, 0.0)
L31 :arc -> from L3.w to L5.w
arrow from L5.s to L6.n
arrow from L6.s to L7.n
arrow from L7.s to L8.n
L71 : spline from L7.e then up 0.2 right  0.3 then down 0.4 then up 0.2 left 0.3 ->
 box invis  "have fun" at L71 +( 0.6, 0.0)
L81 :arc -> from L8.e to L0.e
sprintf("Sleep") rjust at L81.e +(-0.2, 0.0)
arrow from L4.s to L5.n
.PE
```

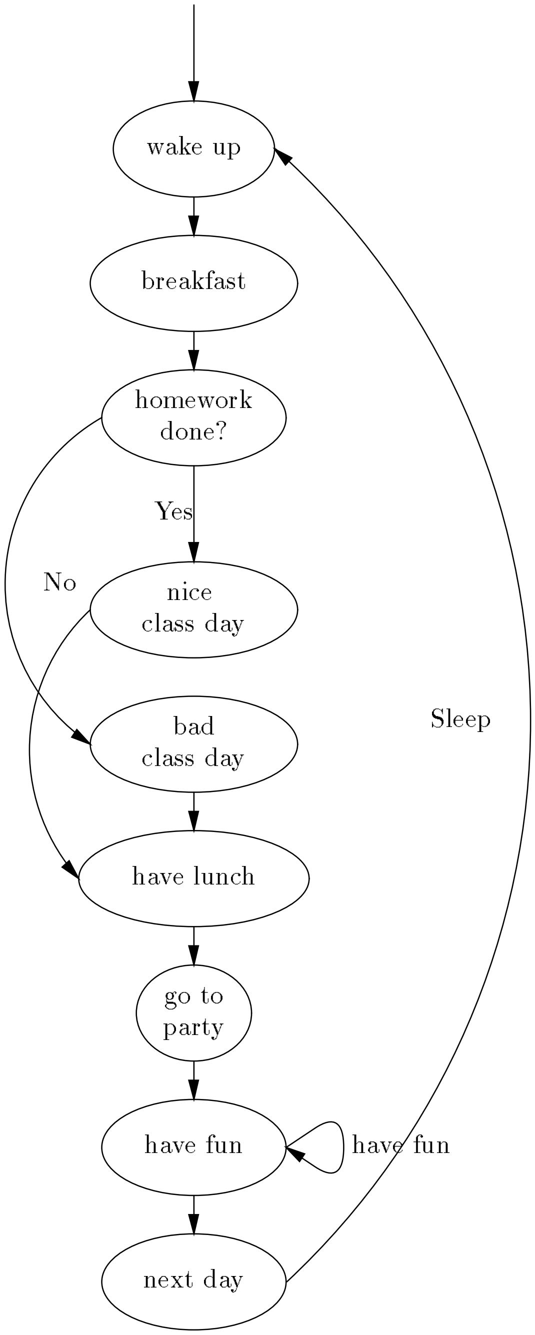
What's this ? ..., uhmmm, ..., many years ago when programmers were programmers , and there was no law to the West of Pecos, ... people used to work in Unix, with small, independent programs , suitable for just one thing, very specialised programs, ..., one of that legendary time is troff , and one of their satellite helpers is gpic, yeah, this forgotten program is what I've recovered from its exile.

What you've seen before it's a gpic specification. Well, to convert it into TEX, just:
gpic -t test1.gpic > test.tex
Then , you use something like this:

```
\documentclass[10pt,a4paper]{article}
\begin{document}
wrekj
\begin{figure}
\input{test} % This is the first line, we include the drawing ,
\centerline{\box\graph} % now, we USE the drawing
\end{figure}
werw
\end{document}
```

Exactly, those two commented lines do the job.

And that's all. Let's take a look at it:



4.0.1 The instructions

definition `st1 = graphstate("wake up")` : This says that state st1 will have inside the words:"wake up"

definition 2 `st2 = graphstate("homework","done?")`: This says that state st1 will have inside the words:"wake up" and in another line "done?"

next relationship `st1.rel(n=st2)`: This says that in the drawing , st2 will be next to st2

goto `st3.rel(n=st4, l='Yes', g=[(st5,'No')])` : This line says , that there's a transition from st3 to st4 and that it's labelled with the word "Yes", and there're another transition to st5, labelled with "No". As you may imagine it may be so many transitions as you like.

forced next `st4.rel(n=st6,f=st5)` : This line says that NOBODY IS PERFECT and that st5 is drawn inmediatly after st4, regardless there's no transition st4-¿st5

4.1 Sources of help

Well, the best you can do is to take a look at the end of grapher.py

Secondly, you should take a look at the source of this document, that is: less grapher.tex

5 Caveats and bugs

grapher has no bugs. Even so, there'll be some bugs.

6 Bye bye

I hope this documentation helps you to use this utility. It's not difficult and greatly profitable. And if you want to get similar drawings or improve some of them just take a look at the code, once you get accostumed to it , you'll find it quite logical.