

# 学习使用 tikz 宏包

2016 年 2 月 11 日

tikz 的介绍, 在<http://www.tug.org/pracjourn/2007-1/mertz/mertz.pdf>很全面了, 值得仔细阅读, 下面算是读书笔记:-)

通过 tikz 的例子 [1] 学习 tikz 宏包也是个好主意。

## 1 tikz 的基本概念

tikz 是 PGF (Portable Graphics Format) 的前端语言, 使用 tikz 可以在 tex/latex 源文件中直截了当的嵌入漂亮的图形, 最简单的形式如代码清单1所示。

代码清单 1: 包含 tikz 图形的基本 tex 文档

```
1 \documentclass[11pt]{article}
2 ...
3 \usepackage{tikz}
4 % Optional PGF libraries
5 \usepackage{pgflibraryarrows}
6 \usepackage{pgflibrarysnakes}
7 ...
8 \begin{document}
9 ...
10 \begin{tikzpicture}
11 ...
12 \end{tikzpicture}
13 ...
14 \end{document}
```

建议每个 tikz 图形用单独的文件存储, 然后 include 到 tex 或者 lyx 文件中。配合 sublime text 和 latexing(tikz) 插件<sup>1</sup>, 建议将代码清单2保存为 demo.tex。

代码清单 2: demo.tex

```
1 \documentclass{article}
2 \usepackage[utf8]{inputenc}
3 \usepackage{tikz}
```

<sup>1</sup>参见: <http://dz.sdut.edu.cn/blog/subaochen/2016/02/sublime-text-3%E7%9A%84%E4%B8%80%E4%BA%9B%E4%BD%BF%E7%94%A8%E7%BB%8F%E9%AA%8C/>

```
4 % Optional PGF libraries
5 \usepackage{pgflibraryarrows}
6 \usepackage{pgflibrarysnakes}
7
8 \begin{document}
9 \includegraphics{demo.pdf}
10 \end{document}
```

同时将代码清单3作为基本的 tikz 测试文件。

代码清单 3: demo.tikz

```
1 % -*- root: demo.tex -*-
2 \begin{tikzpicture}[scale=.8, z={(-.707,-.3)}]
3 \end{tikzpicture}
```

这样，一个完整的 tikz 测试学习环境就建立起来了，happy sublime + latex + tikz!

## 2 tikz 的基本要素：点 (Point) 和路径 (Path)

先定义 Point 和 Path，然后通过运算符把 Point 用指定的 Path 连接起来，就是一幅图形了。Point 有三种表示形式：

1. 相对于原点的坐标，例如：\path (0,1) coordinate (P) 表示横坐标为 0，总坐标为 1 的一个 Point，以后可以通过 P 来引用这个 Point。
2. 相对于原点的角坐标，例如：\path (30:1cm) coordinate (Q) 表示角度为 30 度，半径为 1cm 的一个 Point
3. 相对坐标，例如：\path (P) ++ (1,1) coordinate (P1) 表示相对于 P 的横坐标和总坐标都增加 1 的 Point，即点 (1,2)，并且画图轨迹的当前点为 P1。如果是 \path (P) + (1.1) coordinate (P1) 则表示相对于 P 的横坐标和纵坐标都增加 1 的 Point，即点 (1,2)，但是画图轨迹的当前点为 P。

所谓路径 (Path) 即通过划线运算符 - 连接起来的轨迹，比如命令：\draw (P) - (Q) - (R)；表示从点 P 画线到 Q，再从 Q 画线到 R，并且当前点为 R。

## 3 tikz 的基本图形

1. 网格 \draw (P) grid (Q)
2. 矩形 \draw (P) rectangle (Q)
3. 园 \draw (P) circle (rim)
4. 圆弧 \draw (P) arc ( : :rim)

5. 椭圆 `\draw (P) ellipse (r1,r2)`

6. 贝叶斯曲线 `\draw (P).. controls (C) and (D).. (Q)`

注意基本图形不能当作 Point 使用，因此在路径中不能作为目标节点，比如代码清单4是正确的，代码清单5是不正确的。

代码清单 4：基本图形可以作为路径的起点

```
1 \begin{tikzpicture}
2 \draw (0:1cm) -- (0:2cm)
3         arc (0:60:2cm) -- (60:1cm)
4         arc (60:0:1cm) -- cycle;
5 \end{tikzpicture}
```

代码清单 5：基本图形不能作为目标节点

```
1 \begin{tikzpicture}
2 \draw (0:1cm) -- (0:2cm) -- arc (0:60:2cm) -- (60:1cm) -- arc (60:0:1cm) -- cycle;
3 \end{tikzpicture}
```

没有三角形？

## 4 tikz 的循环

代码清单 6：bipartite.tikz

```
1 % -*- root: demo.tex -*-
2 \begin{tikzpicture}
3 \foreach \i in {1,...,4}
4 {
5     \path (\i,0) coordinate (X\i);
6     \fill (X\i) circle (1pt);
7 }
8 \foreach \j in {1,...,3}
9 {
10    \path (\j,1) coordinate (Y\j);
11    \fill (Y\j) circle (1pt);
12 }
13 \foreach \i in {1,...,4}
14 {
15     \foreach \j in {1,...,3}
16     {
17         \draw (X\i) -- (Y\j);
18     }
19 }
20 \end{tikzpicture}
```

结果如图1所示：

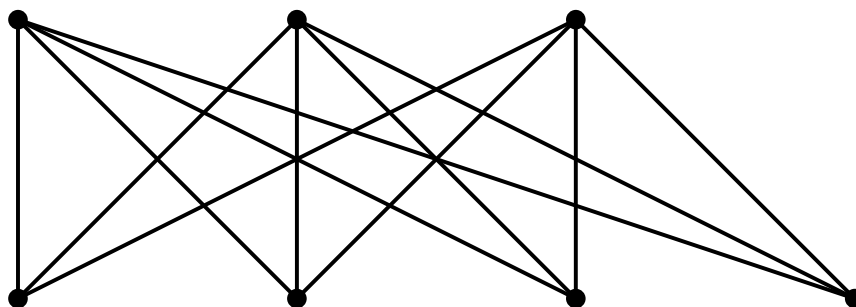


图 1: 有趣的 tikz 循环

## 5 结合 gnuplot

首先要保证安装了 gnuplot, 并且允许 tex 调用 gnuplot 这个外部应用, debian/ubuntu 下是在 `/etc/texmf/texmf.d/` 下面增加一个比如 `100shell_escape.cnf` 文件, 内容为:

```
shell_escape_commands = \
bibtex, bibtex8, bibtexu, pbibtex, upbibtex, biber, \
kpsewhich, \
makeindex, mindex, texindy, \
mpost, pmpost, \
repstopdf, epspdf, extractbb, gnuplot
```

然后执行 `update-texmf` 即可。

比如下面的代码画正弦曲线:

```
1 \begin{tikzpicture}
2 \draw[smooth, domain=0:6.5] plot function {sin(2*x)*exp(-x/4)};
3 \end{tikzpicture}
```

## 参考文献

- [1] “tikz examples.” [Online]. Available: <http://www.texample.net/tikz/examples>