

TikZ  
L!KΣ

&  
Σ

PGF  
L!KΣ

TikZ 3.0.1a 中文手册

L!KΣ 3.0.1a 中文手册

```
\begin{tikzpicture}
\coordinate (front) at (0,0);
\coordinate (horizon) at (0,.31\paperheight);
\coordinate (bottom) at (0,-.6\paperheight);
\coordinate (sky) at (0,.57\paperheight);
\coordinate (left) at (-.51\paperwidth,0);
\coordinate (right) at (.51\paperwidth,0);

\shade [bottom color=white,
        top color=blue!30!black!50]
        ([yshift=-5mm]horizon -| left)
        rectangle (sky -| right);

\shade [bottom color=black!70!green!25,
        top color=black!70!green!10]
        (front -| left) -- (horizon -| left)
        decorate [decoration=random steps] {
        -- (horizon -| right) }
        -- (front -| right) -- cycle;

\shade [top color=black!70!green!25,
        bottom color=black!25]
        ([yshift=-5mm-1pt]front -| left)
        rectangle ([yshift=1pt]front -| right);

\fill [black!25]
        (bottom -| left)
        rectangle ([yshift=-5mm]front -| right);

\def\nodeshadowed[#1]#2;{
\node[scale=2,above,#1]{
\global\setbox\mybox=\hbox{#2}
\copy\mybox};
\node[scale=2,above,#1,yscale=-1,
scope fading=south,opacity=0.4]{\box\mybox};
}
```

```
\nodeshadowed [at={(-5,8 )},yslant=0.05]
{\Huge Ti\textcolor{orange}{\emph{k}}Z};
\nodeshadowed [at={( 0,8.3)}]
{\huge \textcolor{green!50!black!50}{\&}};
\nodeshadowed [at={( 5,8 )},yslant=-0.05]
{\Huge \textsc{PGF}};
\nodeshadowed [at={( 0,5 )}]
{Manual for Version \pgftypesetversion};

\foreach \where in {-9cm,9cm} {
\nodeshadowed [at={(\where,5cm)}] {\tikz
\draw [green!20!black, rotate=90,
l-system={rule set={F -> FF-[-F+F]+[+F-F]},
axiom=F, order=4,step=2pt,
randomize step percent=50, angle=30,
randomize angle percent=5}] l-system; }}

\foreach \i in {0.5,0.6,...,2}
\fill
[white,opacity=\i/2,
decoration=Koch snowflake,
shift=(horizon),shift={(\rand*11,rnd*7)},
scale=\i,double copy shadow={
opacity=0.2,shadow xshift=0pt,
shadow yshift=3*\i pt,fill=white,draw=none}]
decorate {
decorate {
decorate {
(0,0) - ++(60:1) -- ++(-60:1) -- cycle
} } }

\node (left text) ...
\node (right text) ...

\fill [decorate,decoration={footprints,foot of=gnome},
opacity=.5,brown] (\rand*8,-rnd*10)
to [out=rand*180,in=rand*180] (\rand*8,-rnd*10);
\end{tikzpicture}
```

Für meinen Vater, damit er noch viele schöne T<sub>E</sub>X-Graphiken erschaffen kann.

*Till*

Copyright 2007 to 2013 by Till Tantau

Permission is granted to copy, distribute and/or modify *the documentation* under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

Permission is granted to copy, distribute and/or modify *the code of the package* under the terms of the GNU Public License, Version 2 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled GNU Public License.

Permission is also granted to distribute and/or modify *both the documentation and the code* under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. A copy of the license is included in the section entitled L<sup>A</sup>T<sub>E</sub>X Project Public License.

# TikZ 和 PGF 宏包

3.0.1a 版中文手册<sup>\*†</sup>

<http://sourceforge.net/projects/pgf>

Till Tantau<sup>‡</sup>

Institut für Theoretische Informatik  
Universität zu Lübeck

August 29, 2015 <sup>§</sup>

## Contents

1	引言	5
1.1	TikZ 底下的层 . . . . .	5
1.2	和其他图形宏包的对比 . . . . .	6
1.3	实用宏包 . . . . .	6
1.4	如何阅读本手册 . . . . .	7
1.5	作者与致谢 . . . . .	7
1.6	获取帮助 . . . . .	7
I	教程和指导	8
II	安装和配置	9
III	TikZ ist <i>kein</i> Zeichenprogramm	10
IV	Graph Drawing	11
V	Libraries	12
VI	Data Visualization	13
VII	Utilities	14
VIII	Mathematical and Object-Oriented Engines	15
IX	The Basic Layer	16
X	The System Layer	17

---

<sup>\*</sup>翻译者: Hansimov. 此项目开源在 GitHub, 欢迎提出建议或参与翻译: <https://github.com/Hansimov/pgfmanual-zh>

<sup>†</sup>本手册中译者所加注释均为蓝色, 与本来的注释区分开来。



<sup>‡</sup>该手册的编者。手册的部分内容由另外一些作者写成, 详见 1.5 小节。

<sup>§</sup>本中文手册更新时间: 2018 年 7 月 5 日



# 1 引言

欢迎阅读 TikZ 和底层 PGF 系统的文档。一开始这只是一个小小的 LaTeX 样式，用来在我 (Till Tantau) 的博士论文里画图，如今它已经变成了飞速发展的图形语言，手册有一千多页。TikZ 的大量选项常常吓到新手，不过好消息是，这个文档还有一些慢节奏的教程，你不必看其他部分，就能学到几乎所有你需要知道的关于 TikZ 的内容。

我想从“什么是 TikZ?”这个问题开始。大体上，它只是定义一些 TeX 中的绘图命令。比如说，代码 `\tikz \draw (0pt,0pt) --(20pt,6pt);` 会生成一条线段 ，代码 `\tikz \fill[orange] (1ex,1ex) circle (1ex);` 会生成一个橙色实心圆 。从某种意义上说，当你用 TikZ 的时候，你是在编写图形程序，就像你用 TeX 编写文档程序一样。这也解释了 TikZ 这个名字的由来：它是一个递归缩写，追随了“GNU is not unix”的传统。它的德文含义是“TikZ ist *kein* Zeichenprogramm”，翻译成英文就是“TikZ is not a drawing program”，中文意思是“TikZ 不是一个绘图程序”，它在提醒读者注意这个宏包的真正意图。TikZ 采用“TeX 式的排版”，因此在绘图时沿袭了这样一些优点：迅速创建简单图形，精准定位，可以使用宏，以及一流的排版。同样，它也继承了 TeX 系统的缺点：学习曲线陡峭，不是“所见即所得”(WYSIWYG, What You See Is What You Get)，微小的改变就需要长时间的重新编译，并且代码并不能真正“展示出”事物将有的样子。

现在我们知道什么是 TikZ 了，那么 PGF 呢？我们之前提过，TikZ 一开始是 TeX 中一些绘图的宏，并且能同时用在 pdfLaTeX 和传统的（基于 PostScript 的）LaTeX。换句话说，我当时想的是为 TeX 实现一个“可移植的图形格式”(portable graphics format)，这也是为什么叫 PGF。这些早期的宏仍然在用，并且组成了该手册所述的“基本层”(basic layer)。不过，如今一名写作者几乎都在和 TikZ 打交道——它已经自成一套完整的语言了。

## 1.1 TikZ 底下的层

事实上，在 TikZ 底下有两层 (layer)：

**系统层：** 这一层提供了对驱动的完整抽象。驱动就是一个程序，比如 dvips 或者 dvipdfm，输入 .dvi 文件，输出 .ps 或者 .pdf 文件。（pdfTeX 也可以算作是驱动，尽管它的输入并不是 .dvi 文件。这没有影响。）每个驱动都有一套自己的生成图形的语法，每个想用统一语法绘图的人，都对此感到很头疼。而 PGF 的系统层则将这些差异抽象掉了。比如，系统命令 `\pgfsys@lineto{10pt}{10pt}` 延伸了从当前图片 `{pgfpicture}` 到相对坐标 (10pt, 10pt) 的路径。对于不同的驱动，比如 dvips、dvipdfm 或者是 pdfTeX，系统命令会将其转换成不同的 `\special` 命令。系统层尽可能做到简约，毕竟每加一个额外的命令，都意味着将 PGF 对接到新驱动时，要做更多的工作。

作为一个使用者，你不会直接接触系统层。

**基本层：** 基本层提供了一套基本的命令，在创建复杂图形时，用基本层要比系统层简单的多。比如说，系统层没有画圆的命令，因为圆形几乎可以由更加基础的贝塞尔曲线 (Bézier curves) 组成。然而作为一个使用者，在画圆时，你只想用一条简单的命令（至少我是这样），而不是写上半页纸的贝塞尔曲线的控制点坐标。因此，基本层有一个 `\pgfpathcircle` 的命令，帮你生成必要的曲线坐标。

基本层拥有一个核心 (core)，它包含了几个相互依赖的包，并且只能一起导入，而其它的模块 (modules) 则用于扩展核心，提供更多特定用途的命令，比如操纵节点 (node) 或者作图。例如，BEAMER 宏包只用了基本层的核心，而没有用到 shapes 这样的模块。

理论上，TikZ 本身只是几个可能的“前端”之一，集成了一些命令或者特定的语法，使基本层更容易使用。直接使用基本层有一个问题，就是代码通常都很“冗长”。比如要画一个简单的三角形，用基本层你可能需要多达 5 条命令：第 1 条命令从三角形的一个角开始一条路径，第 2 条命令将这条路径延伸到第二个角，第 3 条命令连接第三个角，第 4 条命令闭合路径，第 5 条命令将三角形“绘画” (paint) 出来（不同于“填充” (fill)）。如果用 TikZ 这个前端，这些可以归为一句 METAFONT 式的简单命令：

```
\draw (0,0) -- (1,0) -- (1,1) -- cycle;
```

实际上, TikZ 是 PGF 唯一“正式”的前端。TikZ 可以访问 PGF 的所有特性, 不过更容易使用。它的语法结合了 MEATFONT 和 PSTricks, 外加一些我自己的想法。除了 TikZ, 还有一些其他的前端, 但它们大多更倾向于“技术研究”, 并且没有 TikZ 正式。特别是 pgfpict2e 这个前端, 使用 PGF 的基本层, 重新实现了标准的 L<sup>A</sup>T<sub>E</sub>X {picture} 环境, 以及 \line 和 \vector 这样的命令。这一层其实并没有必要, 因为 pict2e.sty 宏包在重新实现 {picture} 环境上同样出色。当然, pgfpict2e 宏包背后的出发点是, 简单展示一下如何实现一个前端。

由于大多数使用者只会用到 TikZ, 几乎不会有人直接用到系统层, 因此这个手册的前面部分主要涉及 TikZ, 基本层和系统层将在最后介绍。

## 1.2 和其他图形宏包的对比

TikZ 并不是 T<sub>E</sub>X 中唯一的图形宏包。接下来, 我会将 TikZ 和其他宏包作一个非常合理的对比。

1. 标准的 L<sup>A</sup>T<sub>E</sub>X {picture} 环境可以创建基本图形, 但只是一点点。这并不是 L<sup>A</sup>T<sub>E</sub>X 的设计者缺乏知识或者想象力, 相反, 这是 {picture} 环境的可移植性带来的代价: 它得和所有后端驱动打交道。
2. pstricks 宏包确实很强大, 可以创建任何你能想到的图形, 但是它不够方便。更重要的是, 它既不兼容 pdf<sub>tex</sub>, 也不兼容任何其他驱动, 除了 PostScript 代码, 别的什么也不能产生。  
pstricks 的基础支持和 TikZ 类似, 在过去十年里, 使用者们出于特定用途, 贡献了许多出色的额外宏包。TikZ 的语法比 pstricks 更一致, 因为 TikZ 的开发“更加中心化”, 并且“牢记 pstricks 的缺点”。
3. xypic 是一个更老的绘图宏包, 它更难学习和使用, 因为它的语法和文档有点晦涩。
4. dratex 是一个小的绘图宏包, 相比其他宏包 (包括 TikZ), 它真的很小, 也许是个优点, 也许不是。
5. metapost 是 TikZ 一个强大的替代物, 它一般作为外部程序使用, 不过伴随着一堆问题, 但是 Lua<sub>T<sub>E</sub>X</sub> 现在已经将它集成进去了。metapost 嵌入标签有点麻烦, 而用 PGF 则非常容易。
6. 如果使用者不想编写图形程序 (而这在 TikZ 和其他宏包中是必要的), 那么可以用 xfig 替代 TikZ。有程序可以将 xfig 图形转换成 TikZ。

## 1.3 实用宏包

PGF 宏包伴随着一些实用宏包 (utility package), 它们并不和绘图相关, 并且可以独立于 PGF 使用。不过它们和 PGF 打包在一起, 一部分是为了方便, 另一部分是因为它们的功能和 PGF 紧密相关。这些实用宏包有:

1. pgfkeys 宏包提供了强大的关键字 (key) 管理功能, 可以完全独立于 PGF 使用。
2. pgffor 宏包定义了一个有用的 \foreach 语句。
3. pgfcalendar 宏包定义了创建日历的宏, 通常这些日历可以用 PGF 的图形引擎渲染, 但是你也可以用 pgfcalender 在普通文本中排版日历。该宏包还定义了一些命令用于处理日期。
4. pgfpages 宏包用于将多页合并成一页, 它提供了将多张“虚拟页”合并成单张“物理页”的命令。思路是每当 T<sub>E</sub>X 准备好将一页“输出” (shipout) 时, pgfpages 打断这个输出, 并且将该页存到一个特殊的盒子 (box) 里。等到积累了足够的“虚拟页”, 会将它们按比例缩小, 并且排列到一张“物理页”中, 然后将其真正地输出来。这个机制可以让你把文档的每两页并成一页, 而且直接在 L<sup>A</sup>T<sub>E</sub>X 中操作, 不需要借助任何外部程序。pgfpages 还能做得更多, 你可以在页面中加上徽标 (logo)、水印 (watermark) 和边框 (border), 或者是将 16 页合成 1 页, 等等。

## 1.4 如何阅读本手册

本手册讲述了 TikZ 的设计和使用。文档几乎是遵循“用户友好”的原则组织的，最简单和最常用的命令和子宏包最先介绍，底层的和深奥的特性则在后面介绍。

第一，如果你还没装 TikZ，请先阅读安装步骤。第二，建议你通读教程部分。第三，你可能想浏览一遍 TikZ 的描述。如果你想写自己的前端，或者将 PGF 对接到新的驱动，那么通常情况下，你不必阅读“基本层”那章，只需要阅读“系统层”这章就行了。

系统中“公开”（public）的命令和环境会在正文中介绍，命令、环境和选项用红色，绿色表示该部分是可选的，可以不加。

## 1.5 作者与致谢

大部分 PGF 系统及其文档是 Till Tantau 写的。主要团队的另一位成员是 Mark Wibrow，他负责 PGF 的数学引擎、许多形状、装饰引擎以及矩阵库。第三位成员是 Christian Feuersänger，他贡献了浮点库、图像外化（image externalization）、扩展的关键字处理以及手册中的自动超链接。

还有一些人作出了临时的贡献，包括：Christophe Jorssen、Jin-Hwan Cho、Olivier Binda、Matthias Schulz、Renée Ahrens、Stephan Schuster 以及 Thomas Neumann。

此外，还有许多人也为 PGF 作出了贡献，比如写邮件、发现错误（spot bug）、或者是提交库和补丁。非常感谢他们中的所有人，只是人数太多，无法一一陈列姓名。

## 1.6 获取帮助

如果你在使用 PGF 和 TikZ 时需要帮助，请照下面这样做：

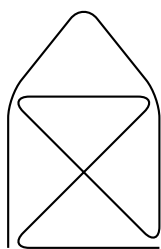
1. 阅读手册，至少是和你的问题相关的那部分。
2. 如果阅读手册没有解决问题，试着看一下 sourceforge 上 PGF 和 TikZ 的开发页面（见本手册标题页）。也许有人已经报告了一个相同的问题，并且有人已经找到了一个解决方法。
3. 在网站上你能找到很多求助的论坛，你可以在论坛发帖，提交错误报告，加入邮件列表，等等。
4. 在你提交错误报告前，尤其是和安装有关的，请确保这真的是个错误。特别是要看一下  $\text{\TeX}$  处理后得到的 `.log` 文件，它应该显示所有正确的文件都从正确的文件夹导入了。几乎所有的安装问题，都能通过查看 `.log` 文件解决。
5. 如果万不得已，你可以给我（Till Tantau）发邮件，如果问题和数学引擎相关，那么找 Mark Wibrow。我不介意收到邮件，我收到太多了。因此，我无法保证及时甚至是能够回复你的邮件。如果你发到 PGF 的邮件列表，也许问题解决的可能性要更高一些（当然了，如果有时间，我会查看这个列表并且回答问题）。

## Part I

# 教程和指导

*by Till Tantau*

为了帮你入门 TikZ，本手册没有立刻给出长长的安装和配置过程，而是直接从教程开始。这些教程解释了该系统所有基本特性和部分高级特性，并不深入所有细节。这部分还指导你在用 TikZ 绘图时，如何继续前进。



```
\tikz \draw[thick,rounded corners=8pt]
(0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0) -- (0,2) -- (2,2) -- (0,0) -- (2,0);
```

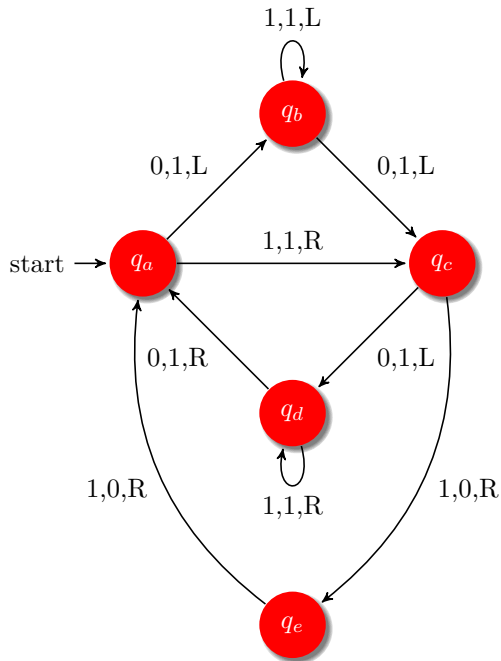


## Part II

# 安装和配置

by Till Tantau

这部分介绍如何安装该系统。通常已经有人帮你装好了，所以你可以跳过这部分；但是如果事与愿违，你是那个不得不自己安装的可怜的家伙，那么请阅读这一部分。



The current candidate for the busy beaver for five states. It is presumed that this Turing machine writes a maximum number of 1's before halting among all Turing machines with five states and the tape alphabet  $\{0, 1\}$ . Proving this conjecture is an open research problem. 中文测试

```

\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,node distance=2.8cm,on grid,semithick,
every state/.style={fill=red,draw=none,circular drop shadow,text=white}]

\node[initial,state] (A) [q_a];
\node[state] (B) [above right=of A] [q_b];
\node[state] (D) [below right=of A] [q_d];
\node[state] (C) [below right=of B] [q_c];
\node[state] (E) [below=of D] [q_e];

\path (A) edge node {0,1,L} (B)
edge node {1,1,R} (C)
(B) edge [loop above] node {1,1,L} (B)
edge node {0,1,L} (C)
(C) edge node {0,1,L} (D)
edge [bend left] node {1,0,R} (E)
(D) edge [loop below] node {1,1,R} (D)
edge node {0,1,R} (A)
(E) edge [bend left] node {1,0,R} (A);

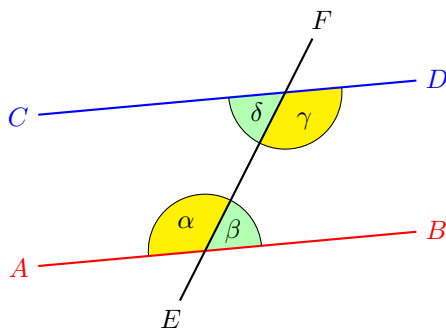
\node [right=1cm,text width=8cm] at (C)
{
The current candidate for the busy beaver for five states. It is
presumed that this Turing machine writes a maximum number of
1's before halting among all Turing machines with five states
and the tape alphabet  $\{0, 1\}$ . Proving this conjecture is an
open research problem. 中文测试
};
\end{tikzpicture}

```

## Part III

# TikZ ist *kein* Zeichenprogramm

by Till Tantau



When we assume that  $AB$  and  $CD$  are parallel, i. e.,  $AB \parallel CD$ , then  $\alpha = \delta$  and  $\beta = \gamma$ .

```
\begin{tikzpicture}[angle radius=.75cm]

\node (A) at (-2,0) [red,left] {$A$};
\node (B) at ( 3,.5) [red,right] {$B$};
\node (C) at (-2,2) [blue,left] {$C$};
\node (D) at ( 3,2.5) [blue,right] {$D$};
\node (E) at (60:-5mm) [below] {$E$};
\node (F) at (60:3.5cm) [above] {$F$};

\coordinate (X) at (intersection cs:first line={(A)--(B)}, second line={(E)--(F)});
\coordinate (Y) at (intersection cs:first line={(C)--(D)}, second line={(E)--(F)});

\path
(A) edge [red, thick] (B)
(C) edge [blue, thick] (D)
(E) edge [thick] (F)
pic ["$\alpha$", draw, fill=yellow] {angle = F--X--A}
pic ["$\beta$", draw, fill=green!30] {angle = B--X--F}
pic ["$\gamma$", draw, fill=yellow] {angle = E--Y--D}
pic ["$\delta$", draw, fill=green!30] {angle = C--Y--E};

\node at ($ (D)!1.5!(B) $) [right=1cm,text width=6cm,rounded corners,fill=red!20,inner sep=1ex]
{
  When we assume that $\color{red}AB$ and $\color{blue}CD$ are
  parallel, i. e., $\color{red}AB \parallel \color{blue}CD$,
  then $\alpha = \delta$ and $\beta = \gamma$.
};
\end{tikzpicture}
```

## Part IV

# Graph Drawing

*by Till Tantau et al.*

*Graph drawing algorithms* do the tough work of computing a layout of a graph for you. *TikZ* comes with powerful such algorithms, but you can also implement new algorithms in the Lua programming language.

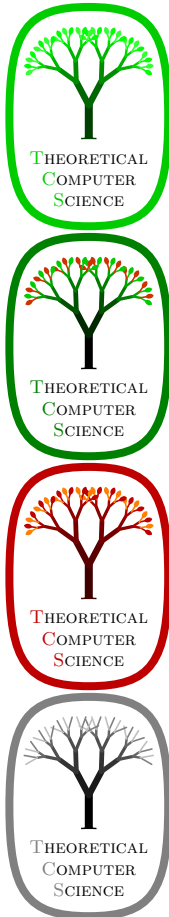
You need to use `LuaTeX` to typeset this part of the manual (and, also, to use algorithmic graph drawing).

# Part V

## Libraries

*by Till Tantau*

In this part the library packages are documented. They provide additional predefined graphic objects like new arrow heads or new plot marks, but sometimes also extensions of the basic PGF or TikZ system. The libraries are not loaded by default since many users will not need them.



```
\tikzset{
  ld/.style={level distance=#1},lw/.style={line width=#1},
  level 1/.style={ld=4.5mm, trunk, lw=1ex, sibling angle=60},
  level 2/.style={ld=3.5mm, trunk!80!leaf a,lw=.8ex,sibling angle=56},
  level 3/.style={ld=2.75mm, trunk!60!leaf a,lw=.6ex,sibling angle=52},
  level 4/.style={ld=2mm, trunk!40!leaf a,lw=.4ex,sibling angle=48},
  level 5/.style={ld=1mm, trunk!20!leaf a,lw=.3ex,sibling angle=44},
  level 6/.style={ld=1.75mm, leaf a, lw=.2ex,sibling angle=40},
}
\pgfarrowsdeclare{leaf}{leaf}
{\pgfarrowslefttextend{-2pt} \pgfarrowsrighttextend{1pt}}
{
  \pgfpathmoveto{\pgfpoint{-2pt}{0pt}}
  \pgfpatharc{150}{30}{1.8pt}
  \pgfpatharc{-30}{-150}{1.8pt}
  \pgfusepathqfill
}

\newcommand{\logo}[5]
{
  \colorlet{border}{#1}
  \colorlet{trunk}{#2}
  \colorlet{leaf a}{#3}
  \colorlet{leaf b}{#4}
  \begin{tikzpicture}
    \scriptsize\scshape
    \draw[border,line width=1ex,yshift=.3cm,
      yscale=1.45,xscale=1.05,looseness=1.42]
      (1,0) to [out=90, in=0] (0,1) to [out=180,in=90] (-1,0)
      to [out=-90,in=-180] (0,-1) to [out=0, in=-90] (1,0) -- cycle;

    \coordinate (root) [grow cyclic,rotate=90]
    child {
      child [line cap=round] foreach \a in {0,1} {
        child foreach \b in {0,1} {
          child foreach \c in {0,1} {
            child foreach \d in {0,1} {
              child foreach \leafcolor in {leaf a,leaf b}
                { edge from parent [color=\leafcolor,-#5] }
            } } }
          } edge from parent [shorten >=-1pt,serif cm-,line cap=butt]
        };

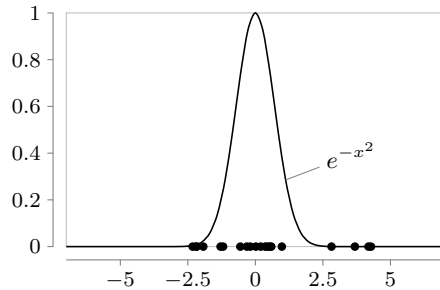
    \node [align=center,below] at (0pt,-.5ex)
    { \textcolor{border}{T}heoretical \ \textcolor{border}{C}omputer \ \
      \textcolor{border}{S}cience };
  \end{tikzpicture}
}

\begin{minipage}[3cm]
  \logo{green!80!black}{green!25!black}{green}{green!80}{leaf}\\
  \logo{green!50!black}{black}{green!80!black}{red!80!green}{leaf}\\
  \logo{red!75!black}{red!25!black}{red!75!black}{orange}{leaf}\\
  \logo{black!50}{black}{black!50}{black!25}{}
\end{minipage}
```

## Part VI

# Data Visualization

by Till Tantau



•  $\sum_{i=1}^{10} x_i$ , where  $x_i \sim U(-1, 1)$

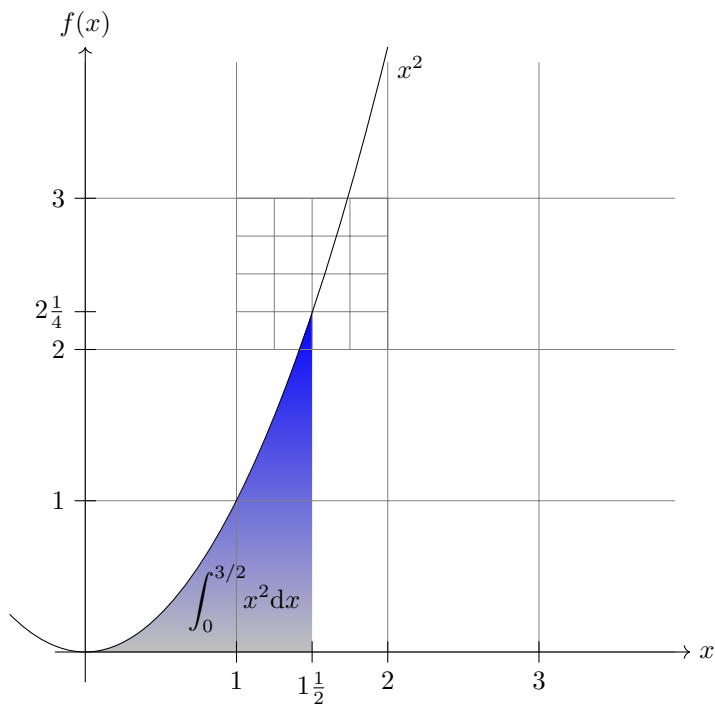
```
\tikz \datavisualization [scientific axes=clean]
[
  visualize as smooth line=Gaussian,
  Gaussian={pin in data={text={\mathit{e}^{-x^2}}},when=x is 1}}
]
data [format=function] {
  var x : interval [-7:7] samples 51;
  func y = exp(-\value x*\value x);
}
[
  visualize as scatter,
  legend={south east outside},
  scatter={
    style={mark=*,mark size=1.4pt},
    label in legend={text={
      \sum_{i=1}^{10} x_i$, where $x_i \sim U(-1,1)$ }}
  }
]
data [format=function] {
  var i : interval [0:1] samples 20;
  func y = 0;
  func x = (rand + rand + rand + rand + rand +
    rand + rand + rand + rand + rand);
};
```

# Part VII

## Utilities

*by Till Tantau*

The utility packages are not directly involved in creating graphics, but you may find them useful nonetheless. All of them either directly depend on PGF or they are designed to work well together with PGF even though they can be used in a stand-alone way.



```
\begin{tikzpicture}[scale=2]
  \shade[top color=blue,bottom color=gray!50] (0,0) parabola (1.5,2.25) |- (0,0);
  \draw (1.05cm,2pt) node[above] {$\displaystyle\int_0^{3/2} \!\! \! x^2 \mathrm{d}x$};

  \draw[help lines] (0,0) grid (3.9,3.9)
    [step=0.25cm] (1,2) grid +(1,1);

  \draw[->] (-0.2,0) -- (4,0) node[right] {$x$};
  \draw[->] (0,-0.2) -- (0,4) node[above] {$f(x)$};

  \foreach \x/\xtext in {1/1, 1.5/1\frac{1}{2}, 2/2, 3/3}
    \draw[shift={(\x,0)}] (0pt,2pt) -- (0pt,-2pt) node[below] {$\xtext$};

  \foreach \y/\ytext in {1/1, 2/2, 2.25/2\frac{1}{4}, 3/3}
    \draw[shift={(0,\y)}] (2pt,0pt) -- (-2pt,0pt) node[left] {$\ytext$};

  \draw (-.5,.25) parabola bend (0,0) (2,4) node[below right] {$x^2$};
\end{tikzpicture}
```

## Part VIII

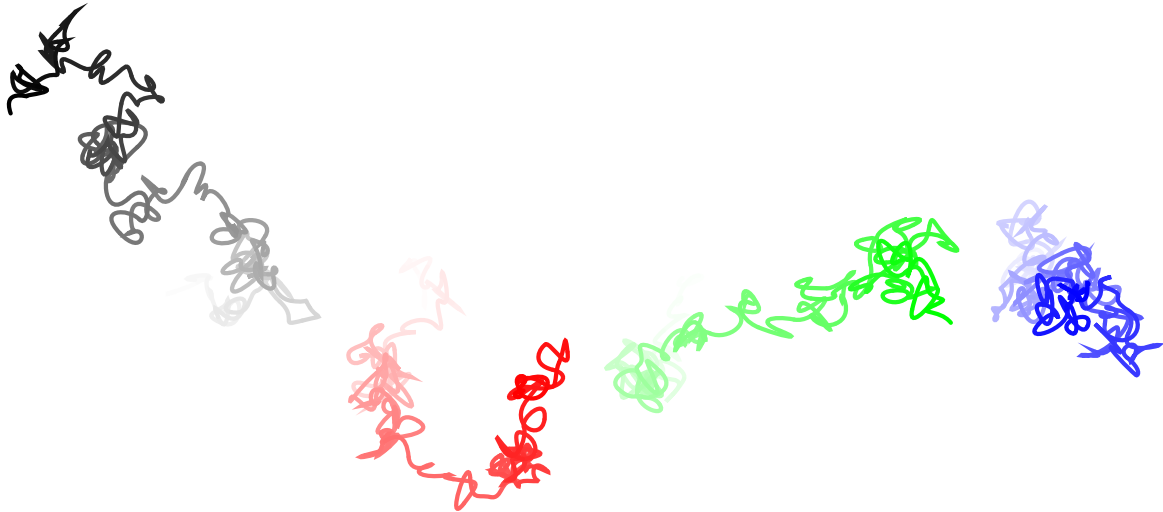
# Mathematical and Object-Oriented Engines

*by Mark Wibrow and Till Tantau*

PGF comes with two useful engines: One for doing mathematics, one for doing object-oriented programming. Both engines can be used independently of the main PGF.

The job of the mathematical engine is to support mathematical operations like addition, subtraction, multiplication and division, using both integers and non-integers, but also functions such as square-roots, sine, cosine, and generate pseudo-random numbers. Mostly, you will use the mathematical facilities of PGF indirectly, namely when you write a coordinate like  $(5\text{cm}*3, 6\text{cm}/4)$ , but the mathematical engine can also be used independently of PGF and TikZ.

The job of the object-oriented engine is to support simple object-oriented programming in T<sub>E</sub>X. It allows the definition of *classes* (without inheritance), *methods*, *attributes* and *objects*.



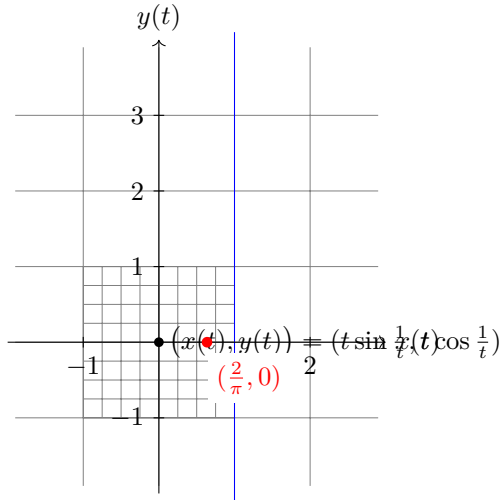
```
\pgfmathsetseed{1}
\foreach \col in {black,red,green,blue}
{
  \begin{tikzpicture}[x=10pt,y=10pt,ultra thick,baseline,line cap=round]
    \coordinate (current point) at (0,0);
    \coordinate (old velocity) at (0,0);
    \coordinate (new velocity) at (rand,rand);

    \foreach \i in {0,1,...,100}
    {
      \draw[\col!\i] (current point)
        .. controls ++([scale=-1]old velocity) and
          ++(new velocity) .. ++(rand,rand)
        coordinate (current point);
      \coordinate (old velocity) at (new velocity);
      \coordinate (new velocity) at (rand,rand);
    }
  \end{tikzpicture}
}
```

## Part IX

# The Basic Layer

by Till Tantau



```
\begin{tikzpicture}
  \draw[gray,very thin] (-1.9,-1.9) grid (2.9,3.9)
    [step=0.25cm] (-1,-1) grid (1,1);
  \draw[blue] (1,-2.1) -- (1,4.1); % asymptote

  \draw[->] (-2,0) -- (3,0) node[right] {$x(t)$};
  \draw[->] (0,-2) -- (0,4) node[above] {$y(t)$};

  \foreach \pos in {-1,2}
    \draw[shift={(\pos,0)}] (0pt,2pt) -- (0pt,-2pt) node[below] {$\pos$};

  \foreach \pos in {-1,1,2,3}
    \draw[shift={(0,\pos)}] (2pt,0pt) -- (-2pt,0pt) node[left] {$\pos$};

  \fill (0,0) circle (0.064cm);
  \draw[thick,parametric,domain=0.4:1.5,samples=200]
    % The plot is reparameterised such that there are more samples
    % near the center.
    plot[id=asymptotic-example] function{((t*t*t)*sin(1/(t*t*t))),(t*t*t)*cos(1/(t*t*t))}
    node[right] {$\bigl(x(t),y(t)\bigr) = (t\sin \frac{1}{t}, t\cos \frac{1}{t})$};

  \fill[red] (0.63662,0) circle (2pt)
    node [below right,fill=white,yshift=-4pt] {$(\frac{2}{\pi},0)$};
\end{tikzpicture}
```



## Part X

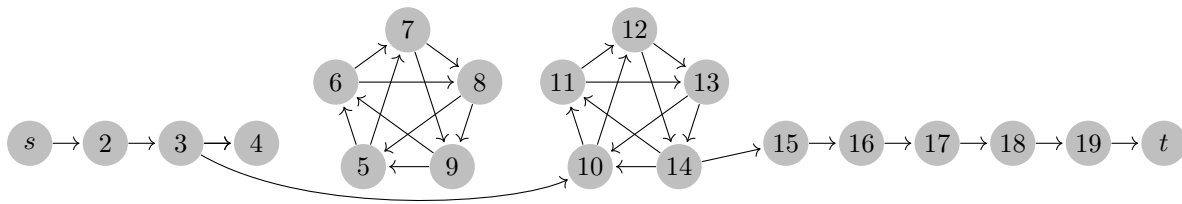
# The System Layer

*by Till Tantau*

This part describes the low-level interface of PGF, called the *system layer*. This interface provides a complete abstraction of the internals of the underlying drivers.

Unless you intend to port PGF to another driver or unless you intend to write your own optimized frontend, you need not read this part.

In the following it is assumed that you are familiar with the basic workings of the `graphics` package and that you know what  $\text{\TeX}$ -drivers are and how they work.



```
\begin{tikzpicture}
  [shorten >=1pt,->,
  vertex/.style={circle,fill=black!25,minimum size=17pt,inner sep=0pt}]

  \foreach \name/\x in {s/1, 2/2, 3/3, 4/4, 15/11, 16/12, 17/13, 18/14, 19/15, t/16}
    \node[vertex] (G-\name) at (\x,0) {$\name$};

  \foreach \name/\angle/\text in {P-1/234/5, P-2/162/6, P-3/90/7, P-4/18/8, P-5/-54/9}
    \node[vertex,xshift=6cm,yshift=.5cm] (\name) at (\angle:1cm) {$\text$};

  \foreach \name/\angle/\text in {Q-1/234/10, Q-2/162/11, Q-3/90/12, Q-4/18/13, Q-5/-54/14}
    \node[vertex,xshift=9cm,yshift=.5cm] (\name) at (\angle:1cm) {$\text$};

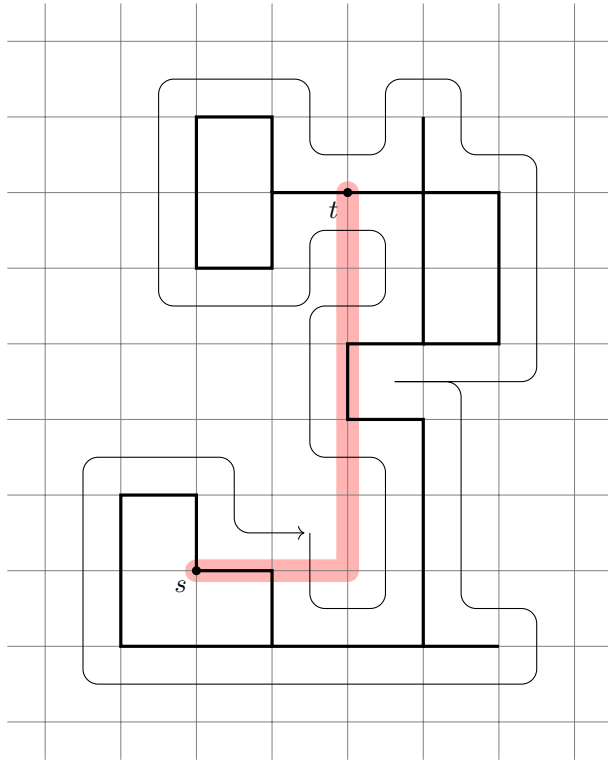
  \foreach \from/\to in {s/2,2/3,3/4,3/4,15/16,16/17,17/18,18/19,19/t}
    \draw (G-\from) -- (G-\to);

  \foreach \from/\to in {1/2,2/3,3/4,4/5,5/1,1/3,2/4,3/5,4/1,5/2}
    { \draw (P-\from) -- (P-\to); \draw (Q-\from) -- (Q-\to); }

  \draw (G-3) .. controls +(-30:2cm) and +(-150:1cm) .. (Q-1);
  \draw (Q-5) -- (G-15);
\end{tikzpicture}
```

## Part XI

# References and Index



```
\begin{tikzpicture}
\draw[line width=0.3cm,color=red!30,line cap=round,line join=round] (0,0)--(2,0)--(2,5);
\draw[help lines] (-2.5,-2.5) grid (5.5,7.5);
\draw[very thick] (1,-1)--(-1,-1)--(-1,1)--(0,1)--(0,0)--
(1,0)--(1,-1)--(3,-1)--(3,2)--(2,2)--(2,3)--(3,3)--
(3,5)--(1,5)--(1,4)--(0,4)--(0,6)--(1,6)--(1,5)
(3,3)--(4,3)--(4,5)--(3,5)--(3,6)
(3,-1)--(4,-1);
\draw[below left] (0,0) node(s){$s$};
\draw[below left] (2,5) node(t){$t$};
\fill (0,0) circle (0.06cm) (2,5) circle (0.06cm);
\draw[->,rounded corners=0.2cm,shorten >=2pt]
(1.5,0.5)-- ++(0,-1)-- ++(1,0)-- ++(0,2)-- ++(-1,0)-- ++(0,2)-- ++(1,0)--
++(0,1)-- ++(-1,0)-- ++(0,-1)-- ++(-2,0)-- ++(0,3)-- ++(2,0)-- ++(0,-1)--
++(1,0)-- ++(0,1)-- ++(1,0)-- ++(0,-1)-- ++(1,0)-- ++(0,-3)-- ++(-2,0)--
++(1,0)-- ++(0,-3)-- ++(1,0)-- ++(0,-1)-- ++(-6,0)-- ++(0,3)-- ++(2,0)--
++(0,-1)-- ++(1,0);
\end{tikzpicture}
```