

# DOT语言

维基百科，自由的百科全书

**DOT语言**是一种文本图形描述语言。它提供了一种简单的描述图形的方法，并且可以为人类和计算机程序所理解。DOT语言文件通常是具有`.gv`或是`.dot`的文件扩展名。

很多程序都可以处理DOT文件。其中的一些，例如`dot`，`neato`，`twopi`，`circo`，`fdp`与`sfdp`，会读取DOT文件并将之渲染成为图形格式。其它的一些，比如`gvpr`，`gc`，`accyclic`，`ccomps`，`sccmap`和`tred`，可以读取DOT文件并对它代表的图形进行一些处理。类似于`GVedit`，`lefty`，`dotty`和`grappa`则提供了交互式的界面。以上程序大部分都包括在了Graphviz软件包中。

## 目录

- 1 语法
  - 1.1 图形类别
    - 1.1.1 无向图
    - 1.1.2 有向图
  - 1.2 属性
  - 1.3 注释
- 2 一个简单的例子
- 3 支持的程序
- 4 限制
- 5 参见
- 6 外部链接

## 语法

### 图形类别

#### 无向图

在最简单的应用中，DOT语言可以用来描述一张无向图。无向图显示了对对象间最简单的关系，例如人之间的友谊。使用关键字`graph`开始一张无向图的定义，并用大括号包含要描述的节点，双连字号（`--`）被用来描述节点间的关系。另外，一行的末尾需要加上分号（`;`）。

```
graph graphname {
    a -- b -- c;
    b -- d;
}
```

#### 有向图

类似于无向图，DOT语言也可以用来描述一张有向图，类似于流程图和树状图。其语法与无向图相似，但要在图的最开始使用关键字'digraph'，并用箭头（->）表示节点直接的关系。

```
digraph graphname {  
    a -> b -> c;  
    b -> d;  
}
```

### 属性

DOT语言中，可以对节点和边添加不同的属性。这些属性可以控制节点和边的显示样式，例如颜色，形状和线形。可以在语句和句尾的分号间放置一对方括号，并在其中中放置一个或多个属性-值对。多个属性可以被逗号和空格（,）分开。节点的属性被放置在只包含节点名称的表达式后。

```
graph graphname {  
    // label属性可以改变节点的显示名称  
    a [label="Foo"];  
    // 节点形状被改变了  
    b [shape=box];  
    // a-b边和b-c边有相同的属性  
    a -- b -- c [color=blue];  
    b -- d [style=dotted];  
}
```

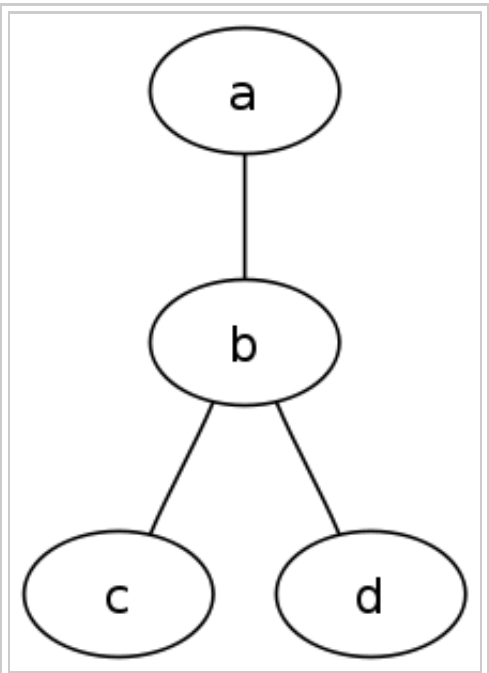
### 注释

DOT语言支持C语言与C++风格的单行与多行注释。另外，也支持Shell脚本风格的以#开头的注释。

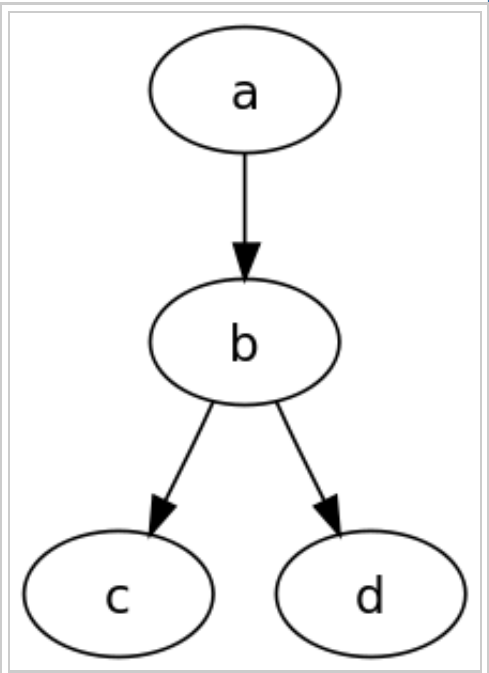
```
// 单行注释  
/* 多行  
   注  
   释 */  
# 如此的行也会被忽略。
```

### 一个简单的例子

以下是一个描述了乙烷化学键结构的示例脚本。这是一个无向图，包括了上述解释的节点属性。

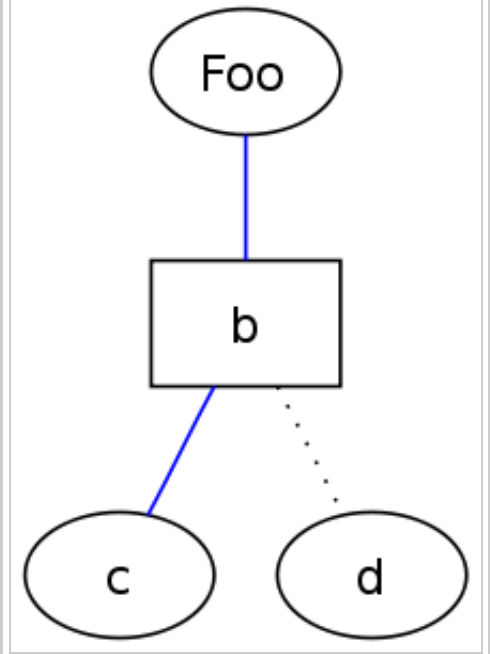


一张无向图



一张有向图

```
graph ethane {
    C_0 -- H_0 [type=s];
    C_0 -- H_1 [type=s];
    C_0 -- H_2 [type=s];
    C_0 -- C_1 [type=s];
    C_1 -- H_3 [type=s];
    C_1 -- H_4 [type=s];
    C_1 -- H_5 [type=s];
}
```



一张有属性的图

## 支持的程序

DOT语言定义了图，但没有提供渲染图的工具。这里列出了一些可以用来渲染，查看与修改DOT图的程序：

- Graphviz - 一组用来修改和渲染图的软件库和程序。
- OmniGraffle 可以导入DOT语言的一个子集，生成一个可编辑的文档，但转换结果不能被导出成为DOT文件。

## 限制

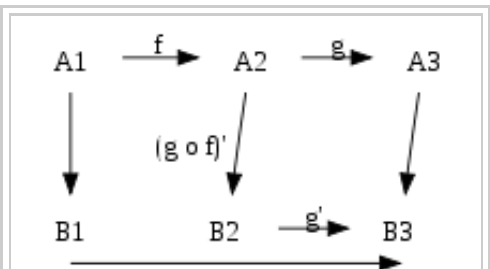
可以用DOT定义图形的位置细节，虽然不是所有实现了DOT语言的工具都可以适当的处理位置属性。因此，取决于所使用的工具，用户必须依赖于自动布局算法（可能造成非预期的输出）或者繁琐的手工布局节点属性。

例如：

```
digraph g {
    node [shape=plaintext]
    A1 -> B1
    A2 -> B2
    A3 -> B3

    A1 -> A2 [label=f]
    A2 -> A3 [label=g]
    B2 -> B3 [label="g'"]
    B1 -> B3 [label="(g o f)" tailport=s]

    { rank=same; A1 A2 A3 }
    { rank=same; B1 B2 B3 }
}
```



一张被不当渲染的图片

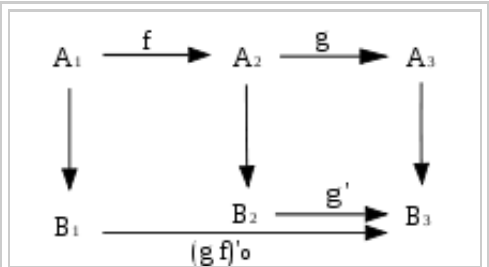
上图有两个问题。右边的方形不是完美的正方形，而且标签 $(g \circ f)$ 的位置有问题。

这些问题可以用Inkscape或其它的SVG编辑器修正。在某些情况下，也可以利用 $pos$ 属性定义位置来修正这

样的问题。

## 参见

- Graphviz
- 图
- `lisp2dot` 将类似Lisp的程序树转换成DOT语言，可以用于遗传编程。



在移动了标签和箭头并改变子脚本的字体大小之后，图形变得正常了

## 外部链接

- DOT语言入门和特性 (<http://www.graphviz.org/Documentation.php>)
- 示例画廊 (<http://www.graphviz.org/Gallery.php>)
- 图形示范 (<http://bloodgate.com/graph-demo>)
- Boost Graph Library (<http://www.boost.org/libs/graph/doc/index.html>)
- Grappa Drawing Package (<http://www.research.att.com/~john/Grappa/>)
- 在线SVG, PNG, GIF, JPG图形发布器 (<http://graph.gafol.net/>)

来自“<http://zh.wikipedia.org/wiki/DOT%E8%AF%AD%E8%A8%80>”

2个分类: 数学软件 | 图形描述语言

- 本页面最后修订于2010年11月29日 (星期一) 04:13。
  - 本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）
- Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。维基媒体基金会是在美国佛罗里达州登记的501(c)(3)免税、非营利、慈善机构。