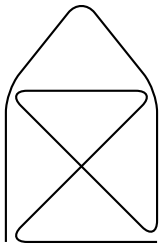
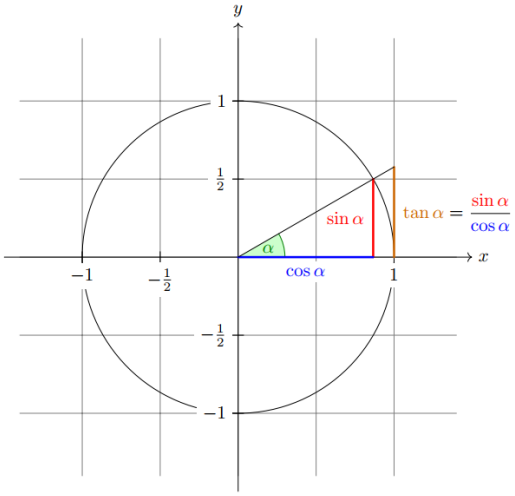
Task 1 Try one's hand at TIKZ



Task 2 A Picture for Karl's Students Step by Step

2.1 Problem Statement

Karl wants to put a graphic on the next worksheet for his students. He is currently teaching his students about sine and cosine. What he would like to have is something that looks like this (ideally):



The angle α is 30° in the example ($\pi/6$ in radians). The sine of α, which is the height of the red line, is

$$\sin\alpha = 1/2.$$

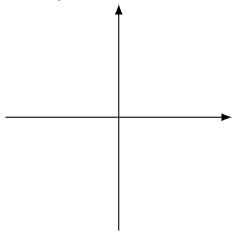
By the Theorem of Pythagoras we have $\cos^2\alpha + \sin^2\alpha = 1$. Thus the length of the blue line, which is the cosine of α, must be

$$\cos\alpha = \sqrt{1 - 1/4} = \tfrac{1}{2}\sqrt{3}.$$

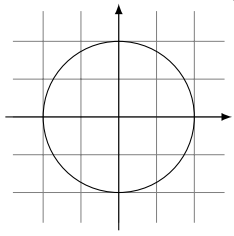
This shows that $\tan\alpha$, which is the height of the orange line, is

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha} = 1/\sqrt{3}.$$

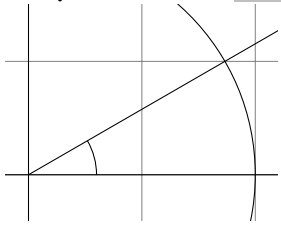
Step 2-1 绘制出带箭头的line axis



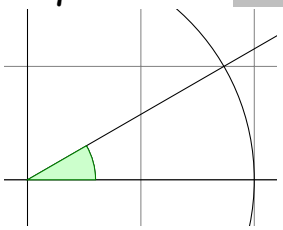
Step 2-2 绘制help lines 与轮廓



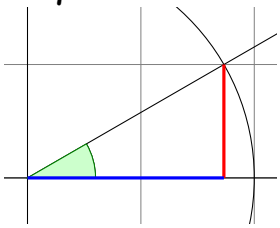
Step 2-3 使用scale 选项并绘制小圆弧和直线



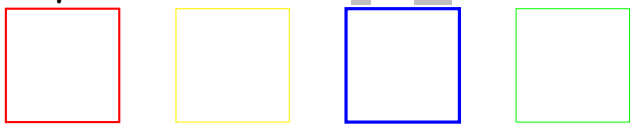
Step 2-4 使用filldraw 命令添加绿色角度标识



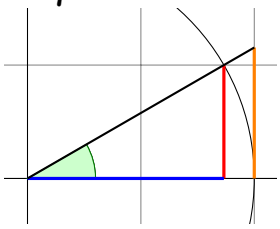
Step 2-5 添加红色与蓝色坐标定位线



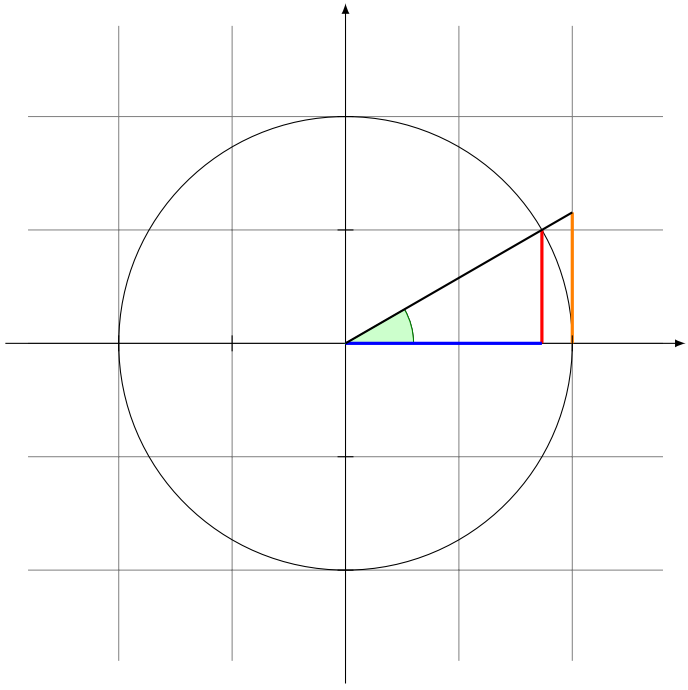
Step 2-6 小彩蛋: 关于+ 与++ 对当前坐标点的影响



Step 2-7 使用路径交点法获取交点并绘制橙色线

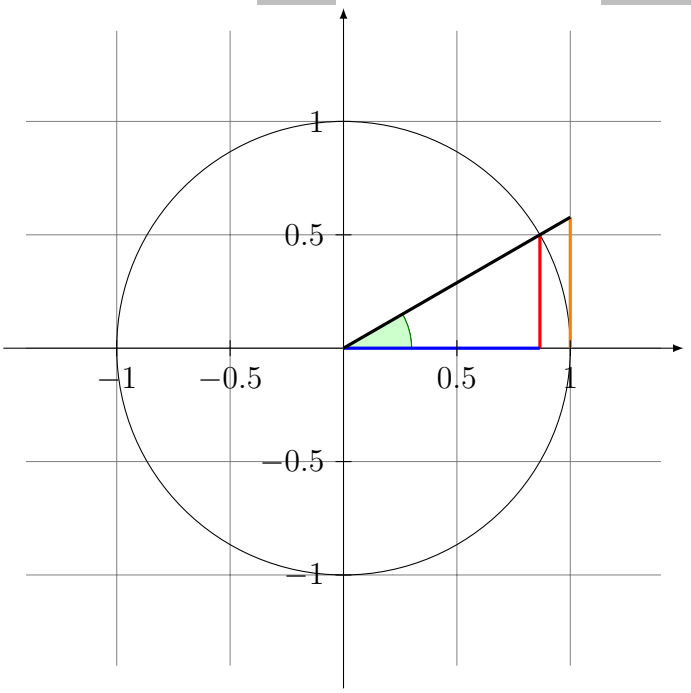


Step 2-8 使用foreach 命令绘制刻度线

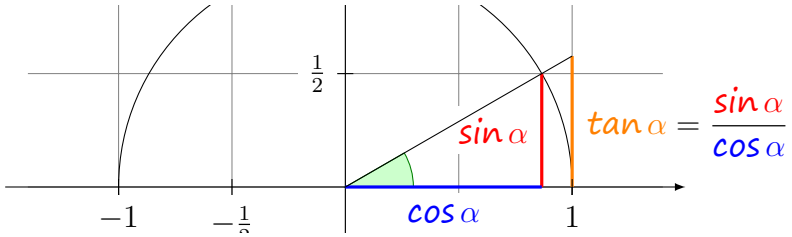


Step 2-9 使用node 命令指定anchor 选项添加文本

实际上这里要注意node 绘制坐标的方向、anchor 并采用数学模式下的文本.

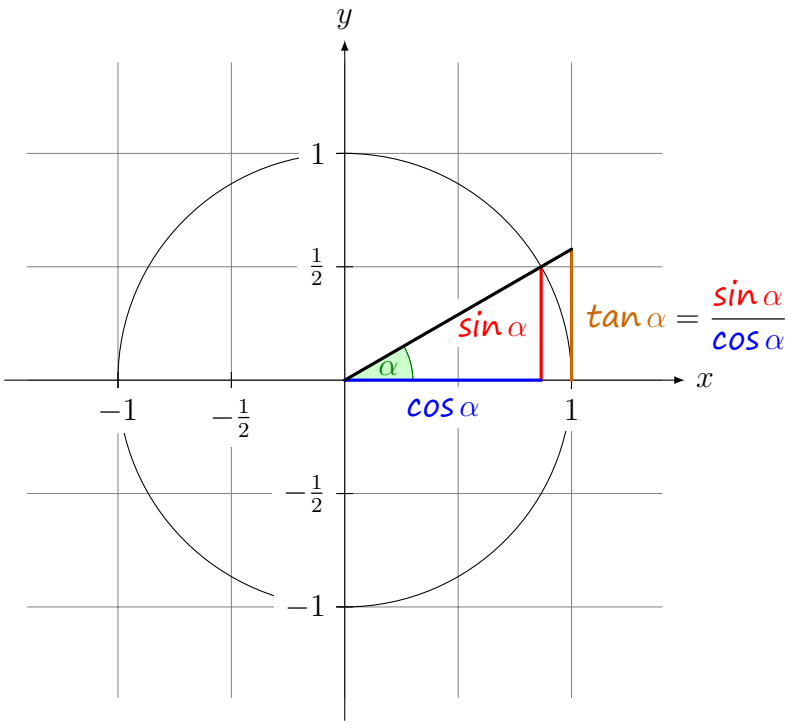


Step 2-10 添加部分颜色信息标注



Step 2-11 更进一步细化修改标注的显示样式 (**Full Code**)

- 设置line cap=rounded 处理线头边缘断线问题
- 在tikzpicture 环境的选项中定义不同线条样式的style
- 使用colorset 预定义颜色
- 使用scope 环境套用axes 样式 (默认样式), 并添加 x、y 轴标识
- 使用颜色anglecolor 补充标注绿色的角度 alpha
- 利用自定义的tikzpicture 选项替换线形与颜色
- 利用xshift 选项补充右侧红色框信息



The angle α is $30°$ in the example ($\pi/6$ in radians). The sine of α, which is the height of the red line, is

$$\sin\alpha = \frac{1}{2}.$$

By the Theorem of Pythagoras we have $\cos^2\alpha + \sin^2\alpha = 1$. Thus the length of the blue line, which is the cosine of α, must be

$$\cos\alpha = \sqrt{1 - \frac{1}{4}} = \frac{\sqrt{3}}{2}$$

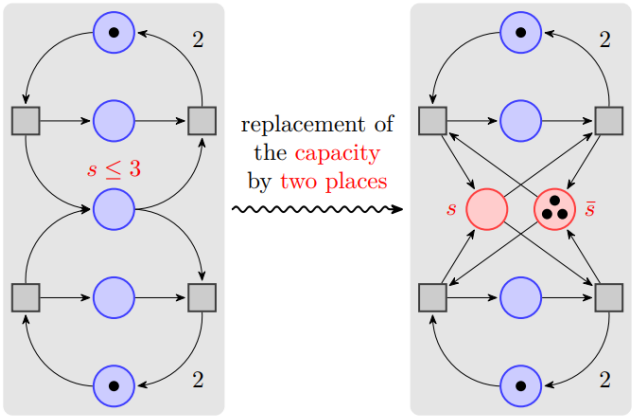
This shows that $\tan\alpha$, which is the height of the orange line, is

$$\tan\alpha = \frac{\sin\alpha}{\cos\alpha} = \frac{1}{\sqrt{3}}$$

Task 3 A Petri-Net for Hagen

3.1 Problem Statement

For his talk, Hagen wishes to create a graphic that demonstrates how a net with place capacities can be simulated by a net without capacities. The graphic should look like this, ideally:



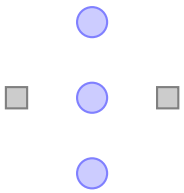
Step 3-1 设置需要调用的TIKZ 库和子图基本框架

○

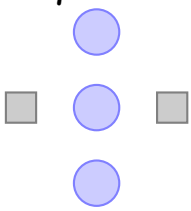
□ ○ □

○

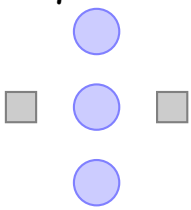
Step 3-2 添加粗细thick 并设置颜色样式draw 与fill 选项



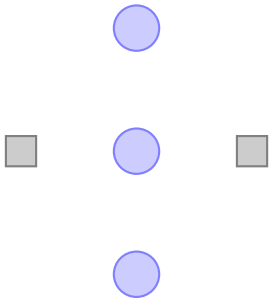
Step 3-3 使用tikz.style 实现代码复用，并设置minimize size 增大 node 内部间距



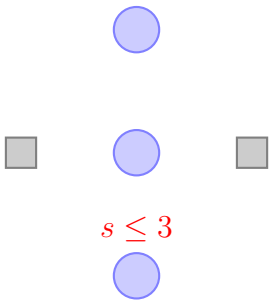
Step 3-4 任意调整node 的syntax 位置并设置坐标点名称



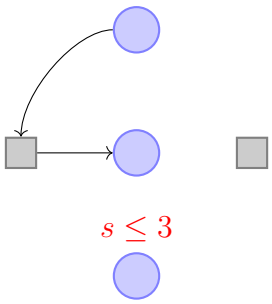
Step 3-5 弃用坐标而改用相对位置参数above、below、left、right 表示各node 间相互关系



Step 3-6 使用label(可以用多个) 添加label 选项进行标注，并在全局 label 选项中使用every 定义标注样式

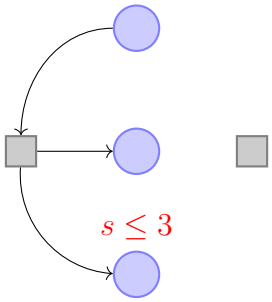


Step 3-7 尝试使用控制点增加node 间的链接箭头，但控制点的距离较难精确把控

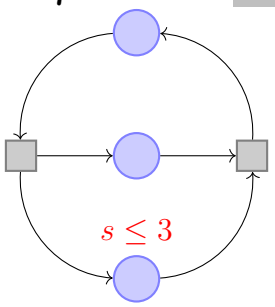


Step 3-8 使用**to** 命令并指定**out** 与**in** 选项控制箭头的输入输出角度，也可以使用**bend left/right** 命令直接指定圆弧状箭头的方向

可以得到更加饱满圆角的弧线箭头 (嘿嘿)

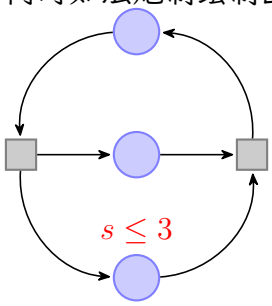


Step 3-9 使用**node** 的**edge** 选项更优雅地一次性指定箭头指向



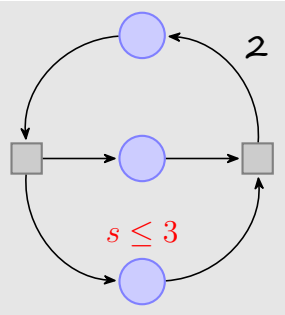
Step 3-10 使用**pre** 与**post** 的箭头样式并绘制整个子图框架

同时如法炮制绘制出整个子图的整体框架



Step 3-11 使用**node** 添加文字标注，并利用**scope** 环境选项**on background layer** 添加阴影

其中**auto** 用于自动确定标注位置,**swap** 用于放置到现有位置的镜像对称位置 (此处使用**swap** 后数字标注将从内侧移动到外侧).



Step 3-12 小彩蛋: 基于*decorations.pathmorphing* 绘制蛇形箭头

- 指定*decorate* 与*decoration=snake* 选项绘制蛇形箭头.

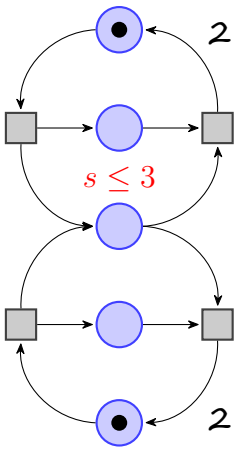
- 指定选项*amplitude*,*amplitude* 以及*amplitude* 指定蛇形路径的弯曲程度、长度等参数.

- 指定*node* 选项*text width* 实现自动换行, 指定*midway* 设置标注位置为当前路径中点.

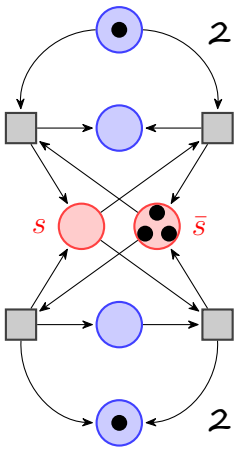
replacement of the capacity by two places

Step 3-13 补充部分细节绘制左半图形

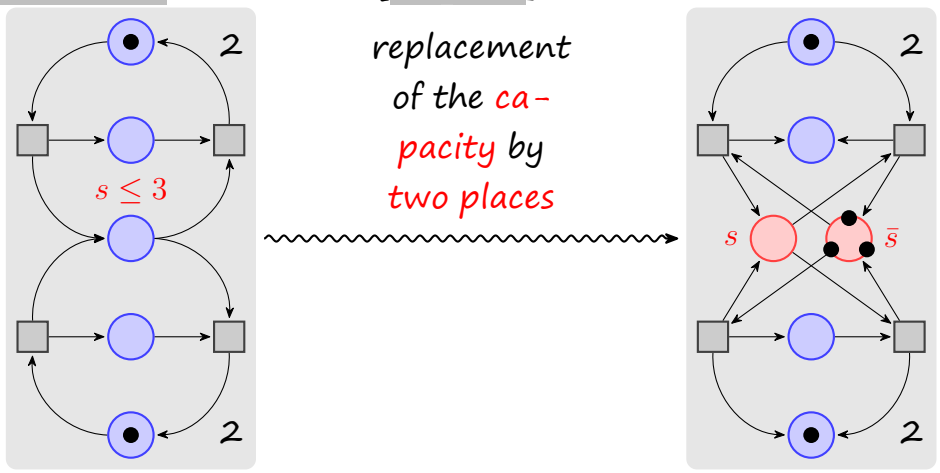
- 设置*node diatance*、*on grid* 选项控制图片样式
- 设置选项*tokens* 选择控制生成点数
- 完善左图的其他交点与连线图形



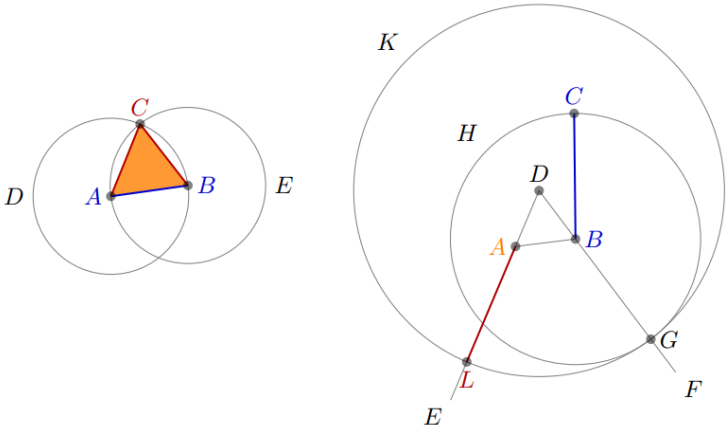
Step 3-14 如法炮制绘制右半部分图



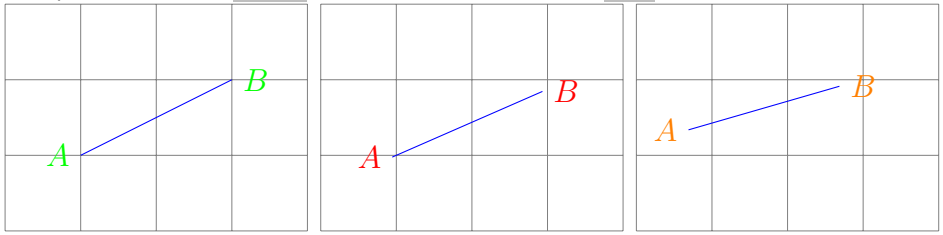
Step 3-15 合并上述所有代码，利用**scope** 环境实现**xshift** 位移并添加阴影，最后利用**decoration=snake** 添加蛇形箭头.[**full code**]



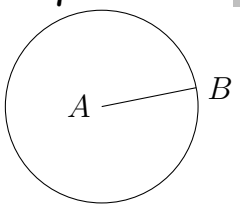
Task 4 Euclid's Amber Version of the Elements



Step 4-1 使用**label** 选项优雅地绘制线段AB

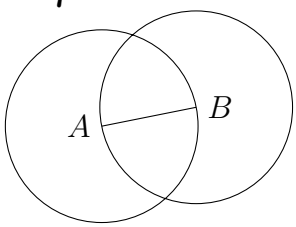


Step 4-2 使用**let<digit>** 关键字获取坐标，使用**veclen** 计算距离

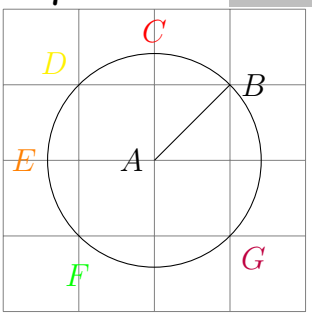


注意在使用坐标计算模式时需要使用 ($ $) 包围，同时指定数字时需要使用大括号 { } 包围.

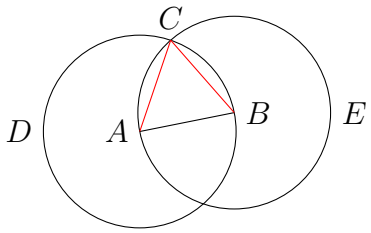
Step 4-3 更进一步利用**let** 指定的坐标同时绘制两个圆



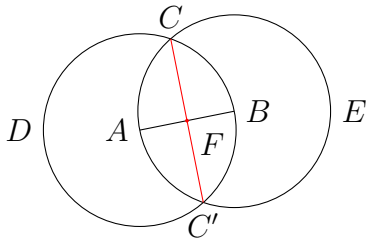
Step 4-4 使用**circle through** 标注圆上的其他点



Step 4-5 使用**name path/intersection** 指定路径与路径的交点



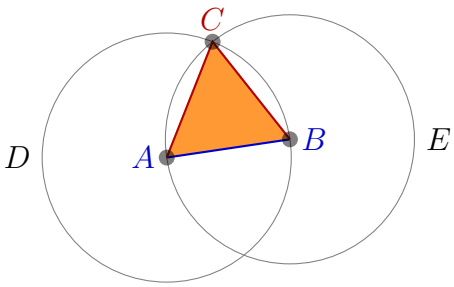
Step 4-6 借助**by** 进一步简化代码并进行交点标注



Step 4-7 更进一步完善左半图的代码

- 在 tikzpicture 环境中添加 help lines 选项
- 使用 colorlet 命令自定义颜色
- 使用 PlainTex 命令 def 定义输入节点文本的命令

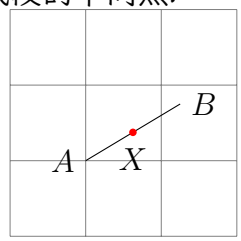
- 使用 foreach 命令遍历目标点序列 A,B,C, 并添加阴影
- 使用 pgfonlayer 命令在背景层绘制橙色三角形



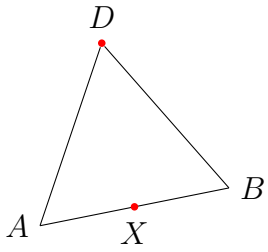
下面让我们进入右半部分图片的绘制嘿嘿.

Step 4-8 使用两种**Partway Calculations** 获取点的变换点

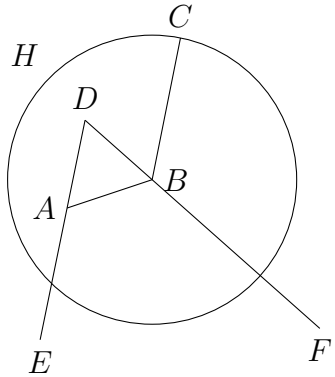
方法一:(坐标比例定点法) 需要在数学计算模式 ($...$) 中使用(Point1)!<scale>!(Point2) 计算线段的中间点.



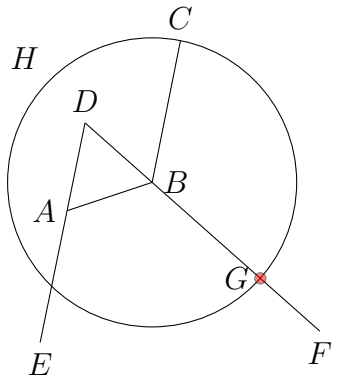
方法二:(旋转拉伸定点法) 在数学计算模式 ($...$) 中使用(Point1)!<proportion>!<angle>:(Point2) 计算点 2 绕点 1 顺时针旋转 <angle> 角度之后拉伸 <scale> 倍的位置.



Step 4-9 精彩再现: 再次使用 **circle through** 标注圆周上的点

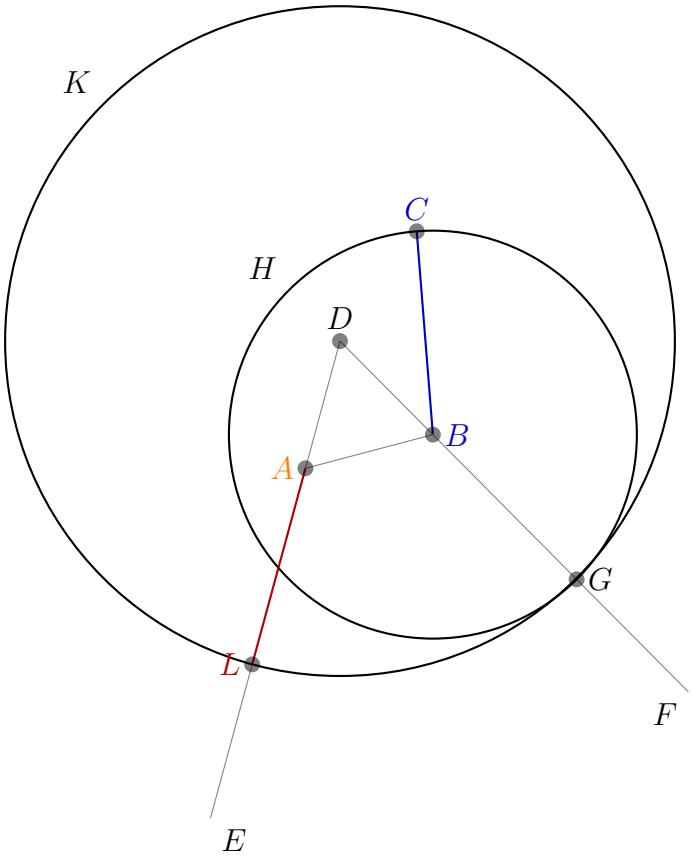


Step 4-10 小彩蛋: 使用**intersection** 定点并添加使用**opacity** 添加半透明阴影点

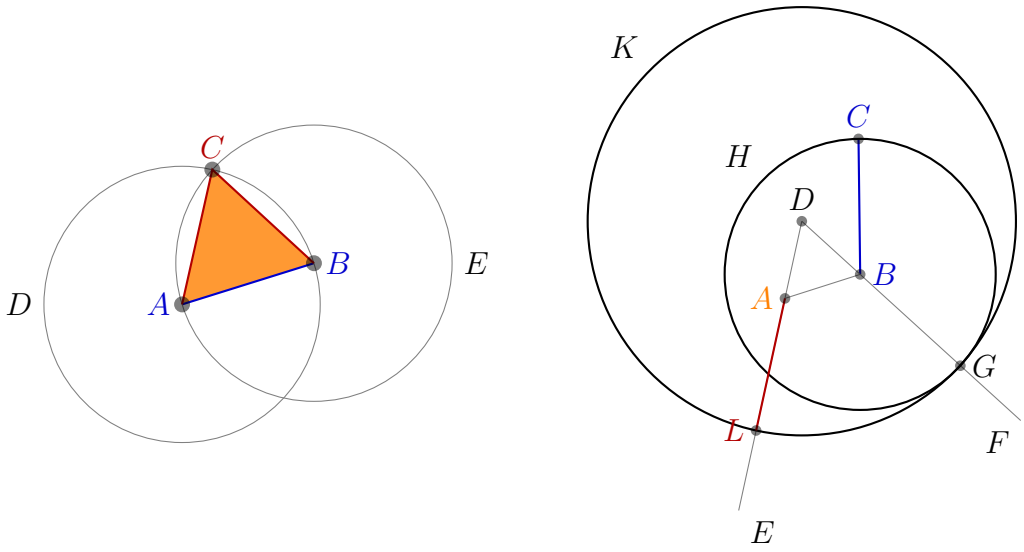


Step 4-11 一鼓作气搞定右半图形

- 使用`def` 结合`colorlet` 命令共同调用颜色
- 使用`name path/intersections` 绘制圆的路径与交点
- 使用`foreach` 结合`opacity` 绘制阴影点



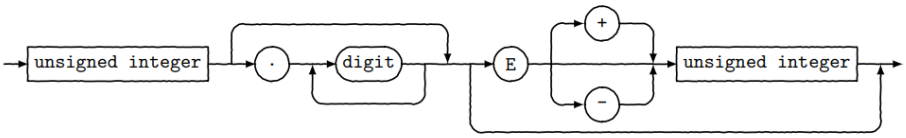
Step 4-12 合并左右子图完善`Euclid` 的工作，完结撒花!(`FullCode`)



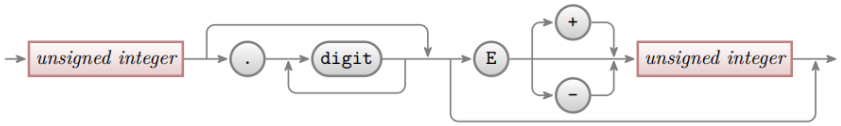
Task 5 Diagrams as Simple Graphs

In this tutorial we have a look at how graphs and matrices can be used to typeset a diagram.

Ilka, who just got tenure for her professorship on Old and Lovable Programming Languages, has recently dug up a technical report entitled *The Programming Language Pascal* in the dusty cellar of the library of her university. Having been created in the good old times using pens and rules, it looks like this[2]:



For her next lecture, Ilka decides to redo this diagram, but this time perhaps a bit cleaner and perhaps also bit "cooler".



Step 5-1 小彩蛋: 设计框线样式nonterminal 与terminal

- 设置形状参数shape 为rectangle
- 设置尺寸参数minimin size 为6mm
- 设置边框的粗细与颜色参数
- 设置边框填充的参数
- 设置填充字体的样式

unsigned integer

. digit long long long long digit E

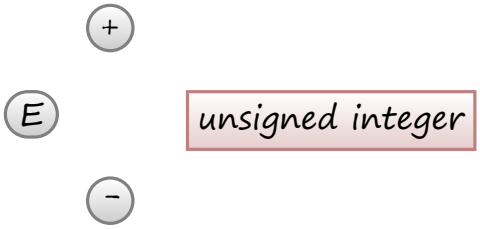
. digit long long long long digit E

Step 5-2 强迫症狂喜: 借助与对齐文本baseline

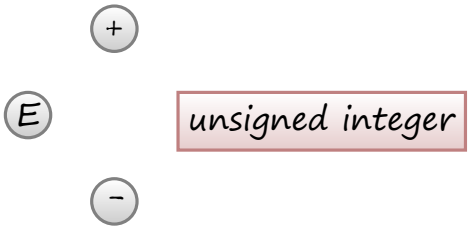
Step 5-3 基本搭建流程图框架，使用**above left** 等选项指定斜向相对位置关系

观察上面的结果我们发现node'+' 和node'-' 的位置由于节点本身的大小限制并不能达到预期，可以尝试采用下面的方式进行修正.

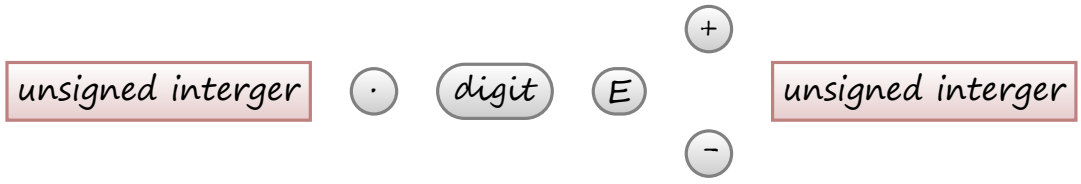
1. 手动将右侧节点利用xshift 参数进行调整.



2. 利用附加选项参数<style name>/.append style 重新换用rounded corners 参数



3. 使用matrix 更优雅地实现各节点的排布



Step 5-4 先来略带痛苦地利用坐标定点绘制出**connections**

但实际上仔细观察发现上面的做法较为麻烦，进一步可以尝试对命令作如下的封装与改进.

Step 5-5 为了代码复用，使用**to path** 选项定义可传入参数的**skip loop** 样式

Step 5-6 继续折腾更精确的路径节点

OS: 其实我更建议加入更多参数来调整起始点的精确位置，避免更为复杂的matrix 规划.