venndiagram v1.2: Drawing Simple Venn Diagrams

Nicola L. C. Talbot
http://www.dickimaw-books.com/

2018-06-07

The venndiagram package is provided to assist generating simple two- and three-set Venn diagrams for lectures or assignment sheets. This package requires the tikz package. As from v1.1, this package no longer requires the intersections library.

The aim of this package is to provide very simple Venn diagrams for assignments or exam questions. If you require more complex diagrams or different layouts it's simpler to directly use the tikz package.

Contents

1 Available Commands and Environments

This package defines two environments:

venndiagram3sets

```
\begin{venndiagram3sets}[⟨options⟩]
```

(for two sets) and

venndiagram2sets

```
\begin{venndiagram2sets}[⟨options⟩]
```

(for three sets).

The optional argument ⟨*options*⟩ is a comma-separated list of ⟨*key*⟩=⟨*value*⟩ settings.

If the ⟨*value*⟩ contains commas or equal signs, make sure you enclose the entire value in braces. For example:

```
\begin{venndiagram3sets}[tikzoptions={scale=2,thick}]
```

The following keys are available:

shade The name of the colour used to shade regions (default: `lightgray`).

labelA The label for the first set (default: `$A$`).

labelB The label for the second set (default: `$B$`).

labelC (Not available for the 2 set version.) The label for the third set (default: `$C$`).

labelOnlyA The label for the region given by $A \setminus (B \cup C)$ (for 3 set version) or $A \setminus B$ (for 2 set version). (Default: empty.)

labelOnlyB The label for the region given by $B \setminus (A \cup C)$ (for 3 set version) or $B \setminus A$ (for 2 set version). (Default: empty.)

labelOnlyC (Not available for 2 set version.) The label for the region given by $C \setminus (A \cup B)$. (Default: empty.)

labelOnlyAB (Not available for 2 set version.) The label for the region given by $(A \cap B) \setminus C$. (Default: empty.)

labelOnlyAC (Not available for 2 set version.) The label for the region given by $(A \cap C) \setminus B$. (Default: empty.)

labelOnlyBC (Not available for 2 set version.) The label for the region given by $(B \cap C) \setminus A$. (Default: empty.)

labelABC (Not available for 2 set version.) The label for the region given by $A \cap B \cap C$. (Default: empty.)

labelNotABC (Not available for 2 set version.) The label for the region given by $(A \cup B \cup C)^c$. (Default: empty.)

labelAB (Not available for 3 set version.) The label for the region given by $A \cap B$. (Default: empty.)

labelNotAB (Not available for 3 set version.) The label for the region given by $(A \cup B)^c$. (Default: empty.)

radius The radius of each set. (Default: 1.2cm.)

hgap The horizontal gap between the outer vertical edge and the nearest set edge. (Default: 0.5cm.)

vgap The vertical gap between the outer horizontal edge and the nearest set edge. (Default: 0.5cm.)

overlap The overlap between the sets. (Default: 0.75cm.)

showframe This is a boolean option (default: `true`). If `true`, the surrounding rectangular frame is drawn. If `false`, the frame isn't drawn but still contributes to the total image size as a hidden path. If the value is omitted `true` is assumed.

tikzoptions Any options to pass to tikzpicture.

Both environments draw the outline of the sets and the rectangular outline of the encompassing universal set. Within the Venn diagram environments commands are provided to shade various regions. (The commands have a cumulative effect, possibly drawing over each other. The set outlines and labels are drawn at the end of the environment.) Available commands are as follows:

`\fillA`

```
\fillA
```

Shades set A.

`\fillB`

```
\fillB
```

Shades set B.

`\fillC`

```
\fillC
```

(Only for 3 set version.) Shades set C.

`\fillAll`

```
\fillAll
```

Shades the entire Venn diagram.

`\fillNotABC`

```
\fillNotABC
```

(Not available for 2 sets version.) Fills $(A \cup B \cup C)^c$.

`\fillOnlyA`

```
\fillOnlyA
```

Shades set $A \setminus (B \cup C)$ (for 3 sets version) or $A \setminus B$ (for 2 sets version).

`\fillOnlyB`

```
\fillOnlyB
```

Shades set $B \setminus (A \cup C)$ (for 3 sets version) or $B \setminus A$ (for 2 sets version).

`\fillOnlyC`

```
\fillOnlyC
```

(Not available for 2 sets version.) Shades $C \setminus (A \cup B)$.

`\fillNotA`

```
\fillNotA
```

Shades everything except A (that is A^c).

`\fillNotB`

```
\fillNotB
```

Shades everything except B (that is B^c).

`\fillNotC`

```
\fillNotC
```

(Not available for 2 set version.) Shades everything except C (that is C^c).

`\fillNotAorB`

```
\fillNotAorB
```

(Not available for 3 set version.) Shades $(A \cup B)^c$

`\fillNotAorNotB`

```
\fillNotAorNotB
```

(Not available for 3 set version.) Shades $(A \cap B)^c$

`\fillANotB`

```
\fillANotB
```

Shades $A \setminus B$.

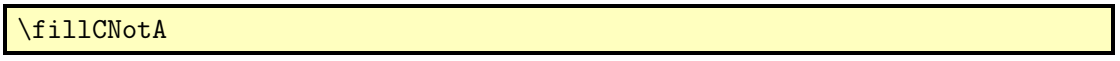
`\fillBNotA`

Shades $B \setminus A$.

`\fillANotC`

(Not available for 2 set version.) Shades $A \setminus C$.

`\fillCNotA`



(Not available for 2 set version.) Shades $C \setminus A$.

`\fillBNotC`

(Not available for 2 set version.) Shades $B \setminus C$.

`\fillCNotB`

(Not available for 2 set version.) Shades $C \setminus B$.

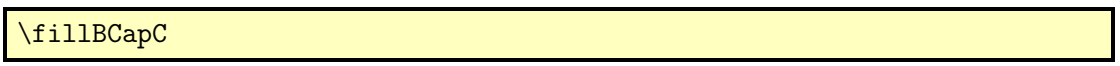
`\fillACapB`

Shades $A \cap B$. (`\fillBCapA` is equivalent to `\fillACapB`.)

`\fillACapC`

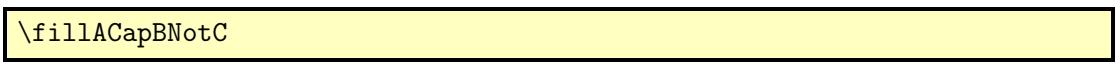
(Not available for 2 set version.) Shades $A \cap C$. (`\fillCCapA` is equivalent to `\fillACapC`.)

`\fillBCapC`



(Not available for 2 set version.) Shades $B \cap C$. (`\fillCCapB` is equivalent to `\fillBCapC`.)

`\fillACapBNotC`



(Not available for 2 set version.) Shades $A \cap B \setminus C$. (`\fillBCapANotC` is equivalent to `\fillACapBNotC`.)

\fillACapCNotB

\fillACapCNotB

(Not available for 2 set version.) Shades $A \cap C \setminus B$. (\fillCCapANotB is equivalent to \fillACapCNotB.)

\fillBCapCNotA

\fillBCapCNotA

(Not available for 2 set version.) Shades $B \cap C \setminus A$. (\fillCCapBNotA is equivalent to \fillBCapCNotA.)

\fillACapBCapC

\fillACapBCapC

(Not available for 2 set version.) Shades $A \cap B \cap C$. (Synonyms: \fillACapCCapB, \fillBCapACapC, \fillBCapCCapA, \fillCCapACapB, \fillCCapBCapA.)

\setpostvennhook

\setpostvennhook{⟨*cmds*⟩}

Sets the hook applied at the very end of the Venn diagram environments (after the outline and labels are drawn but before the end of the tikzpicture environment). The Venn diagram environments create coordinate nodes venn bottom left, venn top left, venn top right and venn bottom right, which may be referenced within the environment or in the hook.

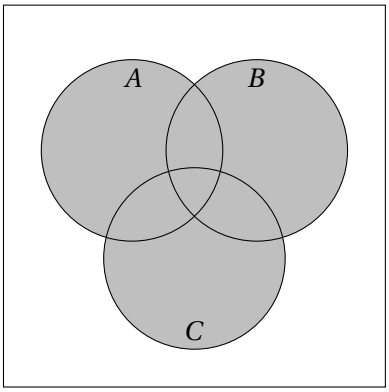
The set labels may also be referenced *but only in* \setpostvennhook: labelOnlyA, labelOnlyB, labelOnlyC (three set version only), labelNotABC (three set version only), labelNotAB (two set version only), labelA, labelB, labelC (three set version only), labelOnlyAB, labelOnlyAC (three set version only), labelOnlyBC (three set version only) and labelAB (two set version only).

2 Examples

1. (Three sets) $A \cup B \cup C$

```
\begin{venndiagram3sets}
 \fillA \fillB \fillC
\end{venndiagram3sets}
```

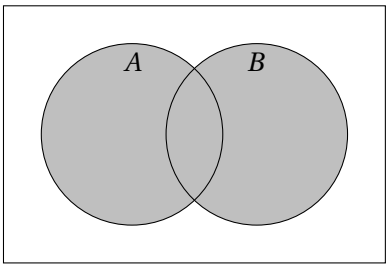
Produces:



2. (Two sets) $A \cup B$

```
\begin{venndiagram2sets}
 \fillA \fillB
\end{venndiagram2sets}
```

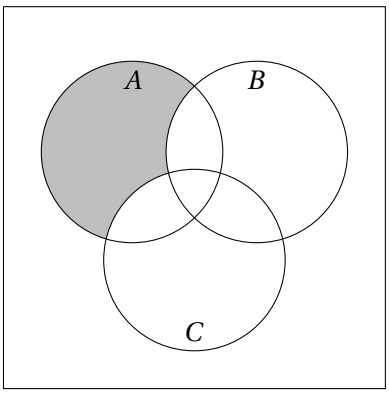
Produces:



3. (Three sets) $A \setminus (B \cup C)$

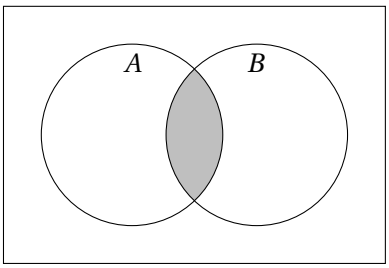
```
\begin{venndiagram3sets}
 \fillOnlyA
\end{venndiagram3sets}
```

Produces:



4. (Two sets) $A \cap B$:

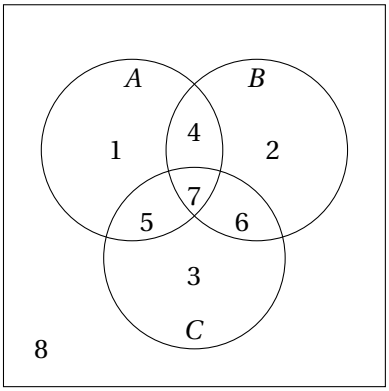
```
\begin{venndiagram2sets}
\fillACapB
\end{venndiagram2sets}
```



5. (Three sets) region labels:

```
\begin{venndiagram3sets}[labelOnlyA={1},labelOnlyB={2},labelOnlyC={3},
 labelOnlyAB={4},labelOnlyAC={5},labelOnlyBC={6},labelABC={7},
 labelNotABC={8}]
\end{venndiagram3sets}
```

Produces:

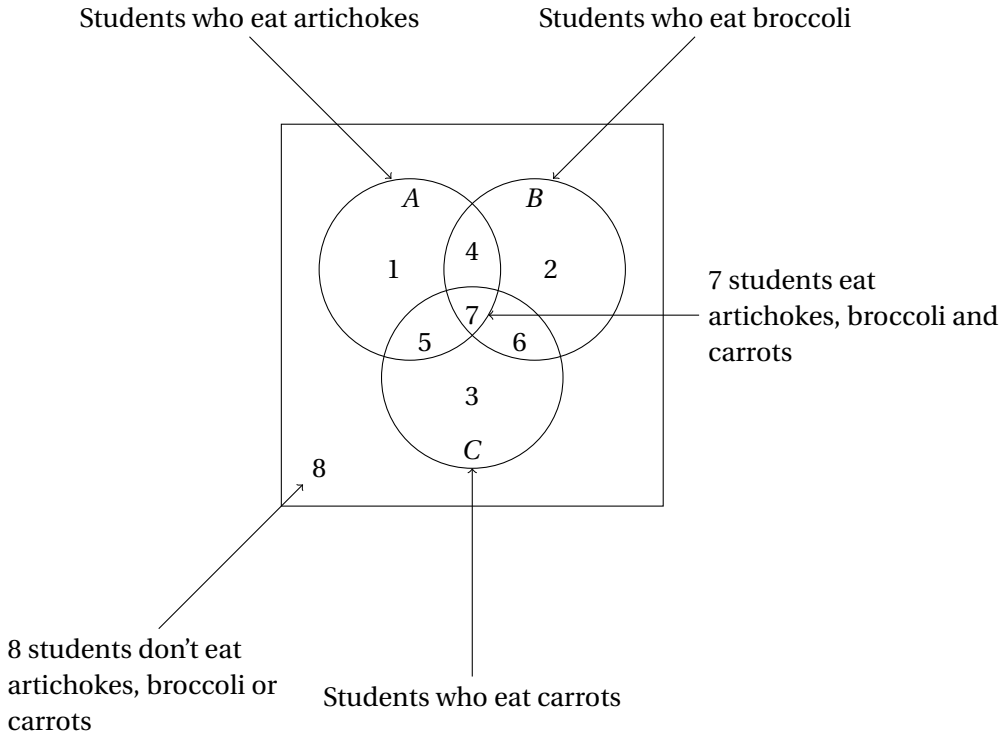


6. Annotating the diagram:

```
\begin{venndiagram3sets}[labelOnlyA={1},labelOnlyB={2},labelOnlyC={3},
 labelOnlyAB={4},labelOnlyAC={5},labelOnlyBC={6},labelABC={7},
 labelNotABC={8}]
\setpostvennhook
{
  \draw[<-] (labelA) -- ++(135:3cm) node[above] {Students who eat
artichokes};
  \draw[<-] (labelB) -- ++(45:3cm) node[above] {Students who eat
broccoli};
```

```
  \draw[<-] (labelC) -- ++(-90:3cm) node[below] {Students who eat
carrots};
  \draw[<-] (labelABC) -- ++(0:3cm)
    node[right,text width=4cm,align=flush left]
   {7 students eat artichokes, broccoli and carrots};
  \draw[<-] (labelNotABC) -- ++(-135:3cm)
    node[below,text width=4cm,align=flush left]
   {8 students don't eat artichokes, broccoli or carrots};
}
\end{venndiagram3sets}
```

Produces:



3 The Code

Package identification:

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{venndiagram}[2018/06/07 v1.2 (NLCT) Venn diagrams]
```

Required packages:

```
3 \RequirePackage{xkeyval}
4 \RequirePackage{tikz}
5 \RequirePackage{etoolbox}
```

TiKZ intersections library no longer needed. (Removed in v1.1)

The pgf package reversed the order of arguments of `atan2` in v3.0, which messes things up a bit. In the event that there are users with older versions of pgf, backward-compatibility is required. Add a switch to determine which syntax to use.

`\ifvennoldpgf` If true use old syntax.

```
\newif\ifvennoldpgf
```

Try to determine this setting.

```
\ifdef\pgfversion
{
  \def\@venn@checkversion#1.#2\@venn@end@checkversion{%
  \ifnum#1<3 \vennoldpgftrue \else \vennoldpgffalse\fi}
  \expandafter\@venn@checkversion\pgfversion.0\@venn@end@checkversion
}
{
  \vennoldpgftrue
}
```

3.1 Initialising the Default Values

Set up macros used by the keys for the Venn diagram options. First the default set labels.

`\@venn@label@A` Set A:

```
\newcommand*{\@venn@label@A}{$A$}
```

`\@venn@label@B` Set B:

```
\newcommand*{\@venn@label@B}{$B$}
```

`\@venn@label@C` Set C:

```
\newcommand*{\@venn@label@C}{$C$}
```

`\@venn@shade` The colour used to shade regions.

```
\newcommand*{\@venn@shade}{lightgray}
```

The default labels for all the other regions are empty.

`enn@label@OnlyA` Only set A:

```
\newcommand*{\@venn@label@OnlyA}{}
```

`enn@label@OnlyB` Only set B:

```
\newcommand*{\@venn@label@OnlyB}{}
```

`enn@label@OnlyC` Only set C:

```
\newcommand*{\@venn@label@OnlyC}{}
```

`nn@label@OnlyAB` Sets A and B but not C:

```
\newcommand*{\@venn@label@OnlyAB}{}
```

nn@label@OnlyAC Sets A and C but not B:

```
\newcommand*{\@venn@label@OnlyAC}{}
```

nn@label@OnlyBC Sets B and C but not A:

```
\newcommand*{\@venn@label@OnlyBC}{}
```

@venn@label@ABC Intersection of sets A, B and C:

```
\newcommand*{\@venn@label@ABC}{}
```

nn@label@NotABC Everything except A, B or C:

```
\newcommand*{\@venn@label@NotABC}{}
```

enn@label@NotAB Everything except A or B (two set version only):

```
\newcommand*{\@venn@label@NotAB}{}
```

\@venn@label@AB Intersection of A and B (two set version only):

```
\newcommand*{\@venn@label@AB}{}
```

Now the default dimensions of the diagrams.

\@venn@radius The radius of the sets.

```
\newcommand*{\@venn@radius}{1.2cm}
```

\@venn@hgap The horizontal distance between the edge of the diagram and the outer edge of the nearest set.

```
\newcommand*{\@venn@hgap}{0.5cm}
```

\@venn@vgap

```
% The vertical distance between the edge of the diagram and the
% outer edge of the nearest set.
\newcommand*{\@venn@vgap}{0.5cm}
```

\@venn@overlap The size of the set overlap.

```
\newcommand*{\@venn@overlap}{0.75cm}
```

enn@tikzoptions Any options to be passed to the tikzpicture environment.

```
\newcommand*{\@venn@tikzoptions}{}
```

Lengths to store the centres of the sets and the overall width and height of the diagram.

\@venn@Ax The x-coordinate of set A:

```
\newlength\@venn@Ax
```

\@venn@Ay The y-coordinate of set A:

```
\newlength\@venn@Ay
```

\@venn@Bx The x-coordinate of set B:

```
\newlength\@venn@Bx
```

`\@venn@By` The y-coordinate of set B:
```
\newlength\@venn@By
```

`\@venn@Cx` The x-coordinate of set C:
```
\newlength\@venn@Cx
```

`\@venn@Cy` The y-coordinate of set C:
```
\newlength\@venn@Cy
```

`\@venn@w` The width of the entire Venn diagram.
```
\newlength\@venn@w
```

`\@venn@h` The height of the entire Venn diagram.
```
\newlength\@venn@h
```

3.2 Defining the key=value Options

Now define the keys for the optional argument of venndiagram2sets and venndiagram3sets. They are all in the family venn.

`shade` Option to set the shading.
```
\define@key{venn}{shade}{\def\@venn@shade{#1}}
```

`labelA` Option to set the label for set A.
```
\define@key{venn}{labelA}{\def\@venn@label@A{#1}}
```

`labelB` Option to set the label for set B.
```
\define@key{venn}{labelB}{\def\@venn@label@B{#1}}
```

`labelC` Option to set the label for set C.
```
\define@key{venn}{labelC}{\def\@venn@label@C{#1}}
```

Now for the region labels.

`labelOnlyA` Option to set the label for only set A.
```
\define@key{venn}{labelOnlyA}{\def\@venn@label@OnlyA{#1}}
```

`labelOnlyB` Option to set the label for only set B.
```
\define@key{venn}{labelOnlyB}{\def\@venn@label@OnlyB{#1}}
```

`labelOnlyC` Option to set the label for only set C.
```
\define@key{venn}{labelOnlyC}{\def\@venn@label@OnlyC{#1}}
```

`labelOnlyAB` Option to set the label for the intersection of A and B.
```
\define@key{venn}{labelOnlyAB}{\def\@venn@label@OnlyAB{#1}}
```

`labelOnlyAC` Option to set the label for the intersection of A and C.

```
\define@key{venn}{labelOnlyAC}{\def\@venn@label@OnlyAC{#1}}
```

`labelOnlyBC` Option to set the label for the intersection of B and C.

```
\define@key{venn}{labelOnlyBC}{\def\@venn@label@OnlyBC{#1}}
```

`labelABC` Option to set the label for the intersection of A, B and C. (Three set version only)

```
\define@key{venn}{labelABC}{\def\@venn@label@ABC{#1}}
```

`labelNotABC` Option to set the label for the region outside the three sets. (Three set version only)

```
\define@key{venn}{labelNotABC}{\def\@venn@label@NotABC{#1}}
```

`labelAB` Option to set the label for the intersection of A and B. (Two set version only)

```
\define@key{venn}{labelAB}{\def\@venn@label@AB{#1}}
```

`labelNotAB` Option to set the label for the region outside the two sets. (Two set version only)

```
\define@key{venn}{labelNotAB}{\def\@venn@label@NotAB{#1}}
```

Now for the dimension options.

`radius` Option to set the radius.

```
\define@key{venn}{radius}{\def\@venn@radius{#1}}
```

`hgap` Option to set the horizontal gap between the outer edge of the diagram and the nearest set edge.

```
\define@key{venn}{hgap}{\def\@venn@hgap{#1}}
```

`vgap` Option to set the vertical gap between the outer edge of the diagram and the nearest set edge.

```
\define@key{venn}{vgap}{\def\@venn@vgap{#1}}
```

`overlap` Option to set the set overlap.

```
\define@key{venn}{overlap}{\def\@venn@overlap{#1}}
```

`showframe` Draw the frame outline.

```
\define@boolkey{venn}[venn]{showframe}[true]{}
\vennshowframetrue
```

Finally the option to set the information to pass to the tikzpicture environment.

`tikzoptions`

```
\define@key{venn}{tikzoptions}{\def\@venn@tikzoptions{#1}}
```

3.3 Circle Intersection

Previously commands like `\fillOnlyA` used pgf path operations to compute the intersection points of the circles, but the code didn't work properly when the co-ordinate system has been scaled. Version 1.1 changes this to calculate the co-ordinates in a more low-level way. We have two circles centred on (a_x, a_y) and (b_x, b_y) both with radius r. This gives the equations:

$$(x-a_x)^2 + (y-a_y)^2 = r^2 \tag{1}$$
$$(x-b_x)^2 + (y-b_y)^2 = r^2 \tag{2}$$

Combining:

$$(x-a_x)^2 - (x-b_x)^2 + (y-a_y)^2 - (y-b_y)^2 = 0 \tag{3}$$

Re-arranging gives the equation of the chord between the two points of intersection:

$$x(b_x-a_x) + y(b_y-a_y) = \frac{b_x^2 - a_x^2 + b_y^2 - a_y^2}{2}$$

Special cases:

1. $b_x = a_x$ (circles vertically stacked):

   $$y(b_y-a_y) = \frac{b_y^2 - a_y^2}{2}$$

   Rearranging gives $y = \frac{1}{2}(b_y + a_y)$. Substituting into (1):

   $$(x-a_x)^2 + \left(\frac{1}{2}(b_y+a_y) - a_y\right)^2 = r^2$$
   $$(x-a_x)^2 + \frac{1}{4}(b_y-a_y)^2 = r^2$$
   $$x^2 - 2xa_x + a_x^2 + \frac{(b_y-a_y)^2}{4} - r^2 = 0$$

   This is a quadratic equation in x with solutions given by

   $$x = a_x \pm \sqrt{a_x^2 - \left(a_x^2 + \frac{(b_y-a_y)^2}{4} - r^2\right)}$$
   $$= a_x \pm \sqrt{r^2 - \frac{1}{4}(b_y-a_y)^2}$$

   If $r^2 < \frac{1}{4}(b_y-a_y)^2$ then no solution exists (circles don't overlap). If $r^2 = \frac{1}{4}(b_y-a_y)^2$ then there's only one point of intersection.

2. $b_y = a_y$ (circles horizontally aligned):

   $$x(b_x-a_x) = \frac{b_x^2 - a_x^2}{2}$$

Rearranging gives $x = \frac{1}{2}(b_x + a_x)$. Substituting into (1):

$$\left(\frac{b_x + a_x}{2} - a_x\right)^2 + (y - a_y)^2 = r^2$$
$$\frac{1}{4}(b_x - a_x)^2 + (y - a_y)^2 = r^2$$
$$y^2 - 2ya_y + a_y^2 + \frac{1}{4}(b_x - a_x)^2 - r^2 = 0$$

This is a quadratic equation in y with solutions given by

$$y = a_y \pm \sqrt{a_y^2 - \left(a_y^2 + \frac{(b_x - a_x)^2}{4} - r^2\right)}$$
$$= a_y \pm \sqrt{r^2 - \frac{1}{4}(b_x - a_x)^2}$$

If $r^2 < \frac{1}{4}(b_x - a_x)^2$ then no solution exists (circles don't overlap). If $r^2 = \frac{1}{4}(b_x - a_x)^2$ then there's only one point of intersection.

The general case has the chord given by equation (3), which can be rewritten in the form $y = mx + c$ where

$$m = \frac{a_x - b_x}{b_y - a_y}$$
$$c = \frac{b_x^2 - a_x^2 + b_y^2 - a_y^2}{2(b_y - a_y)}$$

Substituting into equation (1):

$$(x - a_x)^2 + ((mx + c) - a_y)^2 = r^2$$

Rearranging:

$$(1 + m^2)x^2 + 2x(mc - a_x - a_y m) + a_x^2 + c^2 - 2a_y c + a_y^2 - r^2 = 0$$

This is a quadratic solution in x with solutions given by

$$x = \frac{-(mc - a_x - a_y m) \pm \sqrt{(mc - a_x - a_y m)^2 - (1 + m^2)(a_x^2 + c^2 - 2a_y c + a_y^2 - r^2)}}{1 + m^2}$$

Provide command to compute the intersection of two circles of the same radius (given by `\@venn@radius`.

`ircleintersects` The four arguments are a_x, a_y, b_x and b_y (the centre co-ordinates in lengths). The results are stored in `\@venn@intersect@i@x`, `\@venn@intersect@i@y`, `\@venn@intersect@ii@x` and `\@venn@intersect@ii@y`. The number of intersects (0, 1 or 2) is stored in `\@venn@intersect@n`.

```
\newcommand*{\@venn@computecircleintersects}[4]{%
 \ifdim#1=#3\relax
```

Case 1 ($a_x = b_x$). Compute $y = \frac{1}{2}(b_y + a_y)$.

```
  \pgfmathsetlength{\@venn@intersect@i@y}{0.5*(#4+#2)}%
  \setlength{\@venn@intersect@ii@y}{\@venn@intersect@i@y}%
```

Compute $r^2 - \frac{1}{4}(b_y - a_y)^2$

```
  \pgfmathsetlength{\dimen@i}{#4-#2}%
  \pgfmathsetlength{\dimen@}{\@venn@radius*\@venn@radius-0.25*\dimen@i*\dimen@i}%
  \setlength{\@venn@intersect@i@x}{#1}%
  \setlength{\@venn@intersect@ii@x}{#1}%
  \ifdim\dimen@=0pt\relax
```

One point of intersection.

```
    \def\@venn@intersect@n{1}%
  \else
    \ifdim\dimen@>0pt\relax
```

Two points of intersection.

```
      \def\@venn@intersect@n{2}%
      \pgfmathsetlength{\dimen@i}{sqrt(\dimen@)}%
      \addtolength{\@venn@intersect@i@x}{\dimen@i}%
      \addtolength{\@venn@intersect@ii@x}{-\dimen@i}%
    \else
```

No intersection.

```
      \def\@venn@intersect@n{0}%
    \fi
  \fi
\else
  \ifdim#2=#4\relax
```

Case 2 ($a_y = b_y$). Compute $x = \frac{1}{2}(b_x + a_x)$

```
      \pgfmathsetlength{\@venn@intersect@i@x}{0.5*(#3+#1)}%
      \setlength{\@venn@intersect@ii@x}{\@venn@intersect@i@x}%
```

Compute $r^2 - \frac{1}{4}(b_x - a_x)^2$

```
      \setlength{\dimen@i}{#3}%
      \addtolength{\dimen@i}{-#1}%
      \pgfmathsetlength{\dimen@}{\@venn@radius*\@venn@radius
        -0.25*\dimen@i*\dimen@i}%
      \setlength{\@venn@intersect@i@y}{#2}%
      \setlength{\@venn@intersect@ii@y}{#2}%
      \ifdim\dimen@=0pt\relax
```

One point of intersection.

```
        \def\@venn@intersect@n{1}%
      \else
        \ifdim\dimen@>0pt\relax
```

Two points of intersection.

```
          \def\@venn@intersect@n{2}%
          \pgfmathsetlength{\dimen@i}{sqrt(\dimen@)}%
          \addtolength{\@venn@intersect@i@y}{\dimen@i}%
```

```
        \addtolength{\@venn@intersect@ii@y}{-\dimen@i}%
      \else
```

No intersection.

```
        \def\@venn@intersect@n{0}%
      \fi
    \fi
  \else
```

General case. Convert all lengths to scalar to reduce chances of exceeding max dimension. Using inches to ensure more reasonable values.

```
    \pgfmathparse{#1/72.27}\let\@vnn@ax\pgfmathresult
    \pgfmathparse{#2/72.27}\let\@vnn@ay\pgfmathresult
    \pgfmathparse{#3/72.27}\let\@vnn@bx\pgfmathresult
    \pgfmathparse{#4/72.27}\let\@vnn@by\pgfmathresult
    \pgfmathparse{\@venn@radius/72.27}\let\@vnn@r\pgfmathresult
    \pgfmathparse{\@vnn@ax*\@vnn@ax}\let\@vnn@ax@sq\pgfmathresult
    \pgfmathparse{\@vnn@ay*\@vnn@ay}\let\@vnn@ay@sq\pgfmathresult
    \pgfmathparse{\@vnn@bx*\@vnn@bx}\let\@vnn@bx@sq\pgfmathresult
    \pgfmathparse{\@vnn@by*\@vnn@by}\let\@vnn@by@sq\pgfmathresult
    \pgfmathparse{\@vnn@r*\@vnn@r}\let\@vnn@r@sq\pgfmathresult
```

Set

$$c = \frac{b_x^2 - a_x^2 + b_y^2 - a_y^2}{2(b_y - a_y)}$$

```
    \pgfmathparse{0.5*(\@vnn@bx@sq-\@vnn@ax@sq+\@vnn@by@sq-\@vnn@ay@sq)
       /(\@vnn@by-\@vnn@ay)}%
    \let\@vnn@c\pgfmathresult
```

Set

$$m = \frac{a_x - b_x}{b_y - a_y}$$

```
    \pgfmathparse{(\@vnn@ax-\@vnn@bx)/(\@vnn@by-\@vnn@ay)}%
    \let\@vnn@m\pgfmathresult
```

Compute $(1 + m^2)$.

```
    \pgfmathparse{1+\@vnn@m*\@vnn@m}%
    \let\@vnn@one@plus@m@sq\pgfmathresult
```

Compute

$$mc - a_x - a_y m = m(c - a_y) - a_x$$

```
    \pgfmathparse{\@vnn@m*(\@vnn@c-\@vnn@ay)-\@vnn@ax}%
    \let\@vnn@b\pgfmathresult
```

Denote this value b and now compute

$$b^2 - (1 + m^2)(a_x^2 + a_y^2 + c^2 - 2a_y c - r^2)$$

```
    \pgfmathparse{\@vnn@b*\@vnn@b
     - \@vnn@one@plus@m@sq
      *(\@vnn@ax@sq+\@vnn@ay@sq+\@vnn@c*\@vnn@c
```

```
          -2*\@vnn@ay*\@vnn@c-\@vnn@r@sq)}%
      \let\@vnn@root\pgfmathresult
      \setlength\dimen@{\pgfmathresult in}%
      \ifdim\dimen@=0pt\relax
```

One point of intersection.

```
        \def\@venn@intersect@n{1}%
```

$$x_1 = \frac{-b}{1+m^2}$$

```
        \pgfmathparse{-\@vnn@b/\@vnn@one@plus@m@sq}%
        \setlength{\@venn@intersect@i@x}{\pgfmathresult in}%
        \setlength{\@venn@intersect@ii@x}{\@venn@intersect@i@x}%
```

$$y_1 = mx_1 + c$$

```
        \pgfmathparse{\pgfmathresult*\@vnn@m+c}%
        \setlength{\@venn@intersect@i@y}{\pgfmathresult in}%
        \setlength{\@venn@intersect@ii@y}{\@venn@intesect@i@y}%
      \else
        \ifdim\dimen@>0pt\relax
```

Two points of intersection.

```
          \def\@venn@intersect@n{2}%
          \pgfmathsqrt{\@vnn@root}%
          \let\@vnn@root\pgfmathresult
```

First point.

```
          \pgfmathparse{(-\@vnn@b+\@vnn@root)/\@vnn@one@plus@m@sq}%
          \setlength{\@venn@intersect@i@x}{\pgfmathresult in}%
          \pgfmathparse{\pgfmathresult*\@vnn@m+\@vnn@c}%
          \setlength{\@venn@intersect@i@y}{\pgfmathresult in}%
```

Second point.

```
          \pgfmathparse{(-\@vnn@b-\@vnn@root)/\@vnn@one@plus@m@sq}%
          \setlength{\@venn@intersect@ii@x}{\pgfmathresult in}%
          \pgfmathparse{\pgfmathresult*\@vnn@m+\@vnn@c}%
          \setlength{\@venn@intersect@ii@y}{\pgfmathresult in}%
        \else
```

No intersection.

```
          \def\@venn@intersect@n{0}%
        \fi
      \fi
    \fi
  \fi
}

n@intersect@i@x
\newlength\@venn@intersect@i@x

n@intersect@i@y
\newlength\@venn@intersect@i@y
```

@intersect@ii@x

```
164 \newlength\@venn@intersect@ii@x
```

@intersect@ii@y

```
165 \newlength\@venn@intersect@ii@y
```

3.4 Environment Definitions

enndiagram3sets Environment to draw Venn diagram with three sets.

```
166 \newenvironment{venndiagram3sets}[1][]%
167 {%
```

Disable the keys that aren't applicable.

```
168   \disable@keys{venn}{labelAB,labelNotAB}%
```

Set the key values given in the optional argument.

```
169   \setkeys{venn}{#1}%
```

Calculate centre of set C

```
170   \pgfmathsetlength{\@venn@Cx}{\@venn@hgap + 2*\@venn@radius
171     -0.5*\@venn@overlap}%
172   \pgfmathsetlength{\@venn@Cy}{\@venn@vgap+\@venn@radius}%
```

Calculate centre of set A

```
173   \pgfmathsetlength{\@venn@Ax}{\@venn@hgap+\@venn@radius}%
174   \pgfmathsetlength{\@venn@Ay}{\@venn@Cy
175     + (\@venn@radius - 0.5*\@venn@overlap)*1.73205}%
```

Calculate centre of set B

```
176   \pgfmathsetlength{\@venn@Bx}{\@venn@hgap+3*\@venn@radius
177     -\@venn@overlap}%
178   \setlength{\@venn@By}{\@venn@Ay}%
```

Compute dimensions of entire diagram

```
179   \pgfmathsetlength{\@venn@w}{2*\@venn@hgap+4*\@venn@radius
180       -\@venn@overlap}%
181   \pgfmathsetlength{\@venn@h}{2*\@venn@vgap+4*\@venn@radius
182       -\@venn@overlap}%
```

Define filling commands. Fill all of set A:

```
183   \def\fillA{\path[fill=\@venn@shade] (\@venn@Ax,\@venn@Ay)
184     circle (\@venn@radius);}%
```

Fill all of set B:

```
185   \def\fillB{\path[fill=\@venn@shade] (\@venn@Bx,\@venn@By)
186     circle (\@venn@radius);}%
```

Fill all of set C:

```
187   \def\fillC{\path[fill=\@venn@shade] (\@venn@Cx,\@venn@Cy)
188     circle (\@venn@radius);}%
```

Fill everything:

```
\def\fillAll{\path[fill=\@venn@shade] (0,0)
  rectangle (\@venn@w,\@venn@h);}%
```

Fill everything except set A:

```
\def\fillNotA{\path[fill=\@venn@shade,even odd rule]
  (0,0) rectangle (\@venn@w,\@venn@h)
  (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);}%
```

Fill everything except set B:

```
\def\fillNotB{\path[fill=\@venn@shade,even odd rule]
  (0,0) rectangle (\@venn@w,\@venn@h)
  (\@venn@Bx,\@venn@By) circle (\@venn@radius);}%
```

Fill everything except set C:

```
\def\fillNotC{\path[fill=\@venn@shade,even odd rule]
  (0,0) rectangle (\@venn@w,\@venn@h)
  (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);}%
```

Fill only set A.

```
\let\fillOnlyA\@venn@fillOnlyA@threesets
```

Fill only set B:

```
\let\fillOnlyB\@venn@fillOnlyB@threesets
```

Fill only set C:

```
\let\fillOnlyC\@venn@fillOnlyC@threesets
```

Fill everything except A, B or C.

```
\let\fillNotABC\@venn@fillNotABC@threesets
```

Fill A but not B

```
\def\fillANotB{%
  \begin{scope}
    \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
    \path[fill=\@venn@shade,even odd rule]
      (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
      (\@venn@Bx,\@venn@By) circle (\@venn@radius);
  \end{scope}
}%
```

Fill B but not A

```
\def\fillBNotA{%
  \begin{scope}
    \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \path[fill=\@venn@shade,even odd rule]
      (\@venn@Bx,\@venn@By) circle (\@venn@radius)
      (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
  \end{scope}
}%
```

Fill A but not C

```
\def\fillANotC{%
```

```
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
    \end{scope}
  }%
```

Fill *C* but not *A*

```
  \def\fillCNotA{%
    \begin{scope}
      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
    \end{scope}
  }%
```

Fill *B* but not *C*

```
  \def\fillBNotC{%
    \begin{scope}
      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Bx,\@venn@By) circle (\@venn@radius)
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
    \end{scope}
  }%
```

Fill *C* but not *B*

```
  \def\fillCNotB{%
    \begin{scope}
      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \end{scope}
  }%
```

Fill *A* intersect *B*

```
  \def\fillACapB{%
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \path[fill=\@venn@shade]
        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillBCapA\fillACapB
```

Fill *A* intersect *C*

```
  \def\fillACapC{%
```

```
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \path[fill=\@venn@shade]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillCCapA\fillACapC
```

Fill B intersect C

```
  \def\fillBCapC{%
    \begin{scope}
      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
      \path[fill=\@venn@shade]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillCCapB\fillBCapC
```

Fill A intersect B but not C

```
  \def\fillACapBNotC{%
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Bx,\@venn@By) circle (\@venn@radius)
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillBCapANotC\fillACapBNotC
```

Fill A intersect C but not B

```
  \def\fillACapCNotB{%
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillCCapANotB\fillACapCNotB
```

Fill B intersect C but not A

```
  \def\fillBCapCNotA{%
```

```
    \begin{scope}
      \clip (\@venn@Bx,\@venn@By) circle (\@venn@radius);
      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
      \path[fill=\@venn@shade,even odd rule]
        (\@venn@Cx,\@venn@Cy) circle (\@venn@radius)
        (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
    \end{scope}
  }%
```

Define a synonym:

```
  \let\fillCCapBNotA\fillBCapCNotA
```

Fill the intersection of all three sets

```
  \def\fillACapBCapC{%
    \begin{scope}
      \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
      \clip (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
      \path[fill=\@venn@shade]
        (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \end{scope}
  }%
```

Define synonyms:

```
  \let\fillACapCCapB\fillACapBCapC
  \let\fillBCapACapC\fillACapBCapC
  \let\fillBCapCCapA\fillACapBCapC
  \let\fillCCapACapB\fillACapBCapC
  \let\fillCCapBCapA\fillACapBCapC
```

Start the tikzpicture environment.

```
  \ifdefempty{\@venn@tikzoptions}%
  {%
    \def\@venn@dobegin{\begin{tikzpicture}}%
  }%
  {%
    \edef\@venn@dobegin{\noexpand\begin{tikzpicture}%
       [\expandonce\@venn@tikzoptions]}%
  }%
  \@venn@dobegin
```

coordinates of the Venn diagram corners

```
  \path (0,0) coordinate (venn bottom left)
        (0,\@venn@h) coordinate (venn top left)
        (\@venn@w,\@venn@h) coordinate (venn top right)
        (\@venn@w,0) coordinate (venn bottom right);
}%
```

End environment code:

```
{%
```

Draw outlines

```
    \ifvennshowframe
```

```
      \draw (0,0) rectangle (\@venn@w,\@venn@h);
    \else
      \path (0,0) rectangle (\@venn@w,\@venn@h);
    \fi
    \draw (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
    \draw (\@venn@Bx,\@venn@By) circle (\@venn@radius);
    \draw (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);
```

Draw labels

```
    \draw (\@venn@Ax,\@venn@Ay) node[above,left] (labelOnlyA) {\@venn@label@OnlyA};
    \draw (\@venn@Bx,\@venn@By) node[above,right] (labelOnlyB) {\@venn@label@OnlyB};
    \draw (\@venn@Cx,\@venn@Cy) node[below] (labelOnlyC) {\@venn@label@OnlyC};
```

Region labels

```
    \draw (\@venn@vgap,\@venn@hgap) node (labelNotABC) {\@venn@label@NotABC};
    \draw (\@venn@Ax,\@venn@Ay+\@venn@radius)
      node[below] (labelA) {\@venn@label@A};
    \draw (\@venn@Bx,\@venn@By+\@venn@radius)
      node[below] (labelB) {\@venn@label@B};
    \draw (\@venn@Cx,\@venn@vgap) node[above] (labelC) {\@venn@label@C};
    \draw (\@venn@Cx,0.5*\@venn@h) node (labelABC) {\@venn@label@ABC};
    \draw (\@venn@Cx,\@venn@Ay) node[above] (labelOnlyAB) {\@venn@label@OnlyAB};
    \draw (\@venn@Ax,\@venn@Ay) ++(-60:\@venn@radius-0.5*\@venn@overlap)
      node[below left] (labelOnlyAC) {\@venn@label@OnlyAC};
    \draw (\@venn@Bx,\@venn@By) ++(-120:\@venn@radius-0.5*\@venn@overlap)
      node[below right] (labelOnlyBC) {\@venn@label@OnlyBC};
    \@postvennhook
  \end{tikzpicture}
}
```

`\@postvennhook` Hook called just before the end of the tikzpicture environment.

```
\newcommand*{\@postvennhook}{}
```

`setpostvennhook` User interface to set the post hook.

```
\newcommand*{\setpostvennhook}[1]{\def\@postvennhook{#1}}
```

`OnlyA@threesets` `\fillOnlyA` is set to this for the three set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillOnlyA@threesets{%
```

Get the intersection points between A and B.

```
    \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Bx}{\@venn@By}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
    \ifnum\@venn@intersect@n=2\relax
```

We need the point that's furthest from C.

```
      \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@i@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@i@y}%
      \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@ii@x}%
```

```
      \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@ii@y}%
      \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \ifdim\dimen@>\dimen@i
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
        \edef\@venn@start@pt@x{\the\@venn@intersect@i@x}%
        \edef\@venn@start@pt@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
        \edef\@venn@start@pt@x{\the\@venn@intersect@ii@x}%
        \edef\@venn@start@pt@y{\the\@venn@intersect@ii@y}%
      \fi
```

Compute the initial angle of the first arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@i@angle\pgfmathresult
```

Compute the intersection between A and C.

```
      \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Cx}{\@venn@Cy}%
```

Need the point furthest from B.

```
      \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@i@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@i@y}%
      \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@ii@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@ii@y}%
      \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \ifdim\dimen@>\dimen@i
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
        \edef\@venn@end@pt@i@x{\the\@venn@intersect@i@x}%
        \edef\@venn@end@pt@i@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
        \edef\@venn@end@pt@i@x{\the\@venn@intersect@ii@x}%
        \edef\@venn@end@pt@i@y{\the\@venn@intersect@ii@y}%
      \fi
```

Compute the end angle of the first arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@i@angle\pgfmathresult
```

```
      \pgfmathadd{\@venn@end@i@angle}{360}%
      \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle of the second arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@end@pt@i@x-\@venn@Cx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@i@y-\@venn@Cy}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@ii@angle\pgfmathresult
```

Compute the intersection between B and C.

```
      \@venn@computecircleintersects{\@venn@Bx}{\@venn@By}{\@venn@Cx}{\@venn@Cy}%
```

Need the point closest to A.

```
      \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@i@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@i@y}%
      \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@ii@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@ii@y}%
      \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \ifdim\dimen@<\dimen@i
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Cx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Cy}%
        \edef\@venn@end@pt@ii@x{\the\@venn@intersect@i@x}%
        \edef\@venn@end@pt@ii@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Cx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Cy}%
        \edef\@venn@end@pt@ii@x{\the\@venn@intersect@ii@x}%
        \edef\@venn@end@pt@ii@y{\the\@venn@intersect@ii@y}%
      \fi
```

Compute the end angle of the second arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@ii@angle\pgfmathresult
```

Compute the start angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@end@pt@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@ii@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@iii@angle\pgfmathresult
```

Compute the end angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@start@pt@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@start@pt@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@iii@angle\pgfmathresult
      \pgfmathsubtract{\@venn@end@iii@angle}{360}%
      \let\@venn@end@iii@angle\pgfmathresult
```

Fill path

```
      \path[fill=\@venn@shade] (\@venn@start@pt@x,\@venn@start@pt@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@iii@angle,
      end angle=\@venn@end@iii@angle] -- cycle;
    \else
       \fillOnlyA
    \fi
}%
```

`OnlyB@threesets` `\fillOnlyB` is set to this for the three set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillOnlyB@threesets{%
```

Get the intersection points between B and A.

```
    \@venn@computecircleintersects{\@venn@Bx}{\@venn@By}{\@venn@Ax}{\@venn@Ay}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
    \ifnum\@venn@intersect@n=2\relax
```

We need the point that's furthest from C.

```
      \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@i@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@i@y}%
      \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@ii@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@ii@y}%
      \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \ifdim\dimen@>\dimen@i
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Bx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@By}%
        \edef\@venn@start@pt@x{\the\@venn@intersect@i@x}%
        \edef\@venn@start@pt@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Bx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@By}%
```

```
      \edef\@venn@start@pt@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@start@pt@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the initial angle of the first arc.

```
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@start@i@angle\pgfmathresult
```

Compute the intersection between B and C.

```
    \@venn@computecircleintersects{\@venn@Bx}{\@venn@By}{\@venn@Cx}{\@venn@Cy}%
```

Need the point furthest from A.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@>\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@By}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@i@x}%
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@i@y}%
    \else
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@By}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the end angle of the first arc.

```
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle of the second arc.

```
    \pgfmathsetlength{\dimen@i}{\@venn@end@pt@i@x-\@venn@Cx}%
    \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@i@y-\@venn@Cy}%
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@start@ii@angle\pgfmathresult
```

Compute the intersection between A and C.

```
533       \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Cx}{\@venn@Cy}%
```

Need the point closest to A.

```
534       \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@i@x}%
535       \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@i@y}%
536       \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
537       \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@ii@x}%
538       \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@ii@y}%
539       \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
540       \ifdim\dimen@<\dimen@i
541         \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Cx}%
542         \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Cy}%
543         \edef\@venn@end@pt@ii@x{\the\@venn@intersect@i@x}%
544         \edef\@venn@end@pt@ii@y{\the\@venn@intersect@i@y}%
545       \else
546         \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Cx}%
547         \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Cy}%
548         \edef\@venn@end@pt@ii@x{\the\@venn@intersect@ii@x}%
549         \edef\@venn@end@pt@ii@y{\the\@venn@intersect@ii@y}%
550       \fi
```

Compute the end angle of the second arc.

```
551       \ifvennoldpgf
552         \pgfmathatantwo{\dimen@i}{\dimen@ii}%
553       \else
554         \pgfmathatantwo{\dimen@ii}{\dimen@i}%
555       \fi
556       \let\@venn@end@ii@angle\pgfmathresult
```

Compute the start angle of the third arc.

```
557       \pgfmathsetlength{\dimen@i}{\@venn@end@pt@ii@x-\@venn@Ax}%
558       \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@ii@y-\@venn@Ay}%
559       \ifvennoldpgf
560         \pgfmathatantwo{\dimen@i}{\dimen@ii}%
561       \else
562         \pgfmathatantwo{\dimen@ii}{\dimen@i}%
563       \fi
564       \let\@venn@start@iii@angle\pgfmathresult
```

Compute the end angle of the third arc.

```
565       \pgfmathsetlength{\dimen@i}{\@venn@start@pt@x-\@venn@Ax}%
566       \pgfmathsetlength{\dimen@ii}{\@venn@start@pt@y-\@venn@Ay}%
567       \ifvennoldpgf
568         \pgfmathatantwo{\dimen@i}{\dimen@ii}%
569       \else
570         \pgfmathatantwo{\dimen@ii}{\dimen@i}%
571       \fi
572       \let\@venn@end@iii@angle\pgfmathresult
```

Fill path

```
      \path[fill=\@venn@shade]
      (\@venn@start@pt@x,\@venn@start@pt@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@iii@angle,
      end angle=\@venn@end@iii@angle] -- cycle;
    \else
       \fillOnlyB
    \fi
}%
```

OnlyC@threesets `\fillOnlyC` is set to this for the three set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillOnlyC@threesets{%
```

Get the intersection points between C and B.

```
    \@venn@computecircleintersects{\@venn@Cx}{\@venn@Cy}{\@venn@Bx}{\@venn@By}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
    \ifnum\@venn@intersect@n=2\relax
```

We need the point that's furthest from A.

```
      \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@i@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@i@y}%
      \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@ii@x}%
      \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@ii@y}%
      \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
      \ifdim\dimen@>\dimen@i
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Cx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Cy}%
        \edef\@venn@start@pt@x{\the\@venn@intersect@i@x}%
        \edef\@venn@start@pt@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Cx}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Cy}%
        \edef\@venn@start@pt@x{\the\@venn@intersect@ii@x}%
        \edef\@venn@start@pt@y{\the\@venn@intersect@ii@y}%
      \fi
```

Compute the initial angle of the first arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@i@angle\pgfmathresult
```

Compute the intersection between C and A.

```
      \@venn@computecircleintersects{\@venn@Cx}{\@venn@Cy}{\@venn@Ax}{\@venn@Ay}%
```

Need the point furthest from B.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@>\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Cx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Cy}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@i@x}%
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@i@y}%
    \else
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Cx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Cy}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the end angle of the first arc.

```
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@end@i@angle\pgfmathresult
    \pgfmathsubtract{\@venn@end@i@angle}{360}%
    \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle of the second arc.

```
    \pgfmathsetlength{\dimen@i}{\@venn@end@pt@i@x-\@venn@Ax}%
    \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@i@y-\@venn@Ay}%
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@start@ii@angle\pgfmathresult
```

Compute the intersection between B and A.

```
    \@venn@computecircleintersects{\@venn@Bx}{\@venn@By}{\@venn@Ax}{\@venn@Ay}%
```

Need the point closest to C.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@<\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
```

```
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
        \edef\@venn@end@pt@ii@x{\the\@venn@intersect@i@x}%
        \edef\@venn@end@pt@ii@y{\the\@venn@intersect@i@y}%
      \else
        \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
        \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
        \edef\@venn@end@pt@ii@x{\the\@venn@intersect@ii@x}%
        \edef\@venn@end@pt@ii@y{\the\@venn@intersect@ii@y}%
      \fi
```

Compute the end angle of the second arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@ii@angle\pgfmathresult
```

Compute the start angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@end@pt@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@ii@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@iii@angle\pgfmathresult
```

Compute the end angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@start@pt@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@start@pt@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@iii@angle\pgfmathresult
```

Fill path

```
      \path[fill=\@venn@shade] (\@venn@start@pt@x,\@venn@start@pt@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@iii@angle,
      end angle=\@venn@end@iii@angle] -- cycle;
    \else
       \fillOnlyC
    \fi
}%
```

otABC@threesets `\fillNotABC` is set to this for the three set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillNotABC@threesets{%
```

Get the intersection points between A and B.

```
  \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Bx}{\@venn@By}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
  \ifnum\@venn@intersect@n=2\relax
```

We need the point that's furthest from C.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Cx-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Cy-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@>\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
      \edef\@venn@start@pt@x{\the\@venn@intersect@i@x}%
      \edef\@venn@start@pt@y{\the\@venn@intersect@i@y}%
    \else
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
      \edef\@venn@start@pt@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@start@pt@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the initial angle of the first arc.

```
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@start@i@angle\pgfmathresult
```

Compute the intersection between A and C.

```
    \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Cx}{\@venn@Cy}%
```

We need the point that's furthest from B.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Bx-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@By-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@>\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@i@x}%
```

```
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@i@y}%
    \else
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
      \edef\@venn@end@pt@i@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@end@pt@i@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the end angle of the first arc.

```
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@end@i@angle\pgfmathresult
    \pgfmathadd{\@venn@end@i@angle}{360}%
    \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle of the second arc.

```
    \pgfmathsetlength{\dimen@i}{\@venn@end@pt@i@x-\@venn@Cx}%
    \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@i@y-\@venn@Cy}%
    \ifvennoldpgf
      \pgfmathatantwo{\dimen@i}{\dimen@ii}%
    \else
      \pgfmathatantwo{\dimen@ii}{\dimen@i}%
    \fi
    \let\@venn@start@ii@angle\pgfmathresult
```

Compute the intersection between C and B.

```
    \@venn@computecircleintersects{\@venn@Cx}{\@venn@Cy}{\@venn@Bx}{\@venn@By}%
```

We need the point that's furthest from A.

```
    \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@i@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@i@y}%
    \pgfmathsetlength{\dimen@}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \pgfmathsetlength{\dimen@i}{\@venn@Ax-\@venn@intersect@ii@x}%
    \pgfmathsetlength{\dimen@ii}{\@venn@Ay-\@venn@intersect@ii@y}%
    \pgfmathsetlength{\dimen@i}{\dimen@i*\dimen@i+\dimen@ii*\dimen@ii}%
    \ifdim\dimen@>\dimen@i
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Cx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Cy}%
      \edef\@venn@end@pt@ii@x{\the\@venn@intersect@i@x}%
      \edef\@venn@end@pt@ii@y{\the\@venn@intersect@i@y}%
    \else
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Cx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Cy}%
      \edef\@venn@end@pt@ii@x{\the\@venn@intersect@ii@x}%
      \edef\@venn@end@pt@ii@y{\the\@venn@intersect@ii@y}%
    \fi
```

Compute the end angle of the second arc.

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@ii@angle\pgfmathresult
      \pgfmathadd{\@venn@end@ii@angle}{360}%
      \let\@venn@end@ii@angle\pgfmathresult
```

Compute the start angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@end@pt@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@end@pt@ii@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@iii@angle\pgfmathresult
```

Compute the end angle of the third arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@start@pt@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@start@pt@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@iii@angle\pgfmathresult
```

Fill path

```
      \path[fill=\@venn@shade,even odd rule]
        (0,0) rectangle (\@venn@w,\@venn@h)
       (\@venn@start@pt@x,\@venn@start@pt@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@iii@angle,
      end angle=\@venn@end@iii@angle]
      -- cycle;
    \else
      \path[fill=\@venn@shade,even odd rule]
        (0,0) rectangle (\@venn@w,\@venn@h)
       (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
       (\@venn@Bx,\@venn@By) circle (\@venn@radius)
       (\@venn@Cx,\@venn@Cy) circle (\@venn@radius);%
    \fi
}%
```

enndiagram2sets

```
\newenvironment{venndiagram2sets}[1][]%
{%
```

Disable the keys that aren't applicable.

```
  \disable@keys{venn}{labelABC,labelOnlyC,labelOnlyAC,labelOnlyBC,%
    labelNotABC,labelC,labelOnlyAB}%
```

Set the key values given in the optional argument.

```
  \setkeys{venn}{#1}%
```

Calculate centre of A

```
  \pgfmathsetlength{\@venn@Ax}{\@venn@hgap+\@venn@radius}%
  \pgfmathsetlength{\@venn@Ay}{\@venn@vgap+\@venn@radius}%
```

Calculate centre of B

```
  \pgfmathsetlength{\@venn@Bx}{\@venn@hgap+3*\@venn@radius
    -\@venn@overlap}%
  \setlength{\@venn@By}{\@venn@Ay}%
```

Compute dimensions of entire diagram

```
  \pgfmathsetlength{\@venn@w}{2*\@venn@hgap+4*\@venn@radius
     -\@venn@overlap}%
  \pgfmathsetlength{\@venn@h}{2*\@venn@vgap+2*\@venn@radius}%
```

Define filling commands

```
  \def\fillA{\path[fill=\@venn@shade] (\@venn@Ax,\@venn@Ay)
    circle (\@venn@radius);}%
  \def\fillB{\path[fill=\@venn@shade] (\@venn@Bx,\@venn@By)
    circle (\@venn@radius);}%
  \def\fillAll{\path[fill=\@venn@shade] (0,0)
    rectangle (\@venn@w,\@venn@h);}%
```

Fill only set A

```
  \let\fillOnlyA\@venn@fillOnlyA@twosets
```

Fill only set B

```
  \let\fillOnlyB\@venn@fillOnlyB@twosets
```

Fill everything except A

```
  \def\fillNotA{\path[fill=\@venn@shade,even odd rule]
    (0,0) rectangle (\@venn@w,\@venn@h)
    (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);}%
```

Fill everything except B

```
  \def\fillNotB{\path[fill=\@venn@shade,even odd rule]
    (0,0) rectangle (\@venn@w,\@venn@h)
    (\@venn@Bx,\@venn@By) circle (\@venn@radius);}%
```

Fill everything except A or B ($(A \cup B)^c$)

```
  \def\fillNotAorB{%
    \begin{scope}
    \path[clip]
    (0,0) rectangle (\@venn@w,\@venn@h)
    (\@venn@Bx,\@venn@By) circle (\@venn@radius)
```

```
848      ;
849     \path[fill=\@venn@shade,even odd rule]
850     (0,0) rectangle (\@venn@w,\@venn@h)
851     (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
852     ;
853     \end{scope}
854    }%
```

Fill not A or not B ($(A \cap B)^c$)

```
855   \def\fillNotAorNotB{%
856     \path[fill=\@venn@shade,nonzero rule]
857     (0,0) rectangle (\@venn@w,\@venn@h)
858     (\@venn@Ax,\@venn@Ay) circle (\@venn@radius)
859     (0,0) rectangle (\@venn@w,\@venn@h)
860     (\@venn@Bx,\@venn@By) circle (\@venn@radius)
861     ;
862    }%
```

Fill A but not B (same as only A for two sets).

```
863   \let\fillANotB\fillOnlyA
```

Fill B but not A (same as only B for two sets).

```
864   \let\fillBNotA\fillOnlyB
```

Fill A intersect B

```
865   \def\fillACapB{%
866     \begin{scope}
867       \clip (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
868       \path[fill=\@venn@shade]
869         (\@venn@Bx,\@venn@By) circle (\@venn@radius);
870     \end{scope}
871   }%
```

Define synonym:

```
872   \let\fillBCapA\fillACapB
```

Start the tikzpicture environment.

```
873   \ifdefempty{\@venn@tikzoptions}%
874   {%
875     \def\@venn@dobegin{\begin{tikzpicture}}%
876   }%
877   {%
878     \edef\@venn@dobegin{\noexpand\begin{tikzpicture}%
879       [\expandonce\@venn@tikzoptions]}%
880   }%
881   \@venn@dobegin
```

coordinates of the Venn diagram corners

```
882   \path (0,0) coordinate (venn bottom left)
883         (0,\@venn@h) coordinate (venn top left)
884         (\@venn@w,\@venn@h) coordinate (venn top right)
885         (\@venn@w,0) coordinate (venn bottom right);
886 }%
```

End environment code

```
{%
```

Draw outlines

```
    \ifvennshowframe
      \draw (venn bottom left) rectangle (\@venn@w,\@venn@h);
    \else
      \path (venn bottom left) rectangle (\@venn@w,\@venn@h);
    \fi
    \draw (\@venn@Ax,\@venn@Ay) circle (\@venn@radius);
    \draw (\@venn@Bx,\@venn@By) circle (\@venn@radius);
```

Draw labels

```
    \draw (\@venn@Ax,\@venn@Ay) node[above,left] (labelOnlyA)
       {\@venn@label@OnlyA};
    \draw (\@venn@Bx,\@venn@By) node[above,right] (labelOnlyB)
       {\@venn@label@OnlyB};
```

Region labels

```
    \draw (\@venn@vgap,\@venn@hgap) node (labelNotAB) {\@venn@label@NotAB};
    \draw (\@venn@Ax,\@venn@Ay+\@venn@radius)
      node[below] (labelA) {\@venn@label@A};
    \draw (\@venn@Bx,\@venn@By+\@venn@radius)
      node[below] (labelB) {\@venn@label@B};
    \draw (0.5*\@venn@w,0.5*\@venn@h) node (labelAB) {\@venn@label@AB};
    \@postvennhook
  \end{tikzpicture}
}
```

llOnlyA@twosets `\fillOnlyA` is set to this for the two set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillOnlyA@twosets{%
```

Get the intersection points between A and B.

```
    \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Bx}{\@venn@By}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
    \ifnum\@venn@intersect@n=2\relax
```

Compute the start angle for the first arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@i@angle\pgfmathresult
```

Compute the end angle for the first arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
```

```
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@i@angle\pgfmathresult
      \pgfmathadd{\@venn@end@i@angle}{360}%
      \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle for the second arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@ii@angle\pgfmathresult
```

Compute the end angle for the second arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@ii@angle\pgfmathresult
      \pgfmathadd{\@venn@start@ii@angle}{360}%
      \let\@venn@start@ii@angle\pgfmathresult
```

Fill the path

```
      \path[fill=\@venn@shade]
       (\@venn@intersect@i@x,\@venn@intersect@i@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
       -- cycle;
    \else
       \fillOnlyA
    \fi
}%
```

`llOnlyB@twosets` `\fillOnlyB` is set to this for the two set version. Reimplemented in v1.1 to use new circle intersect code.

```
\newcommand*\@venn@fillOnlyB@twosets{%
```

Get the intersection points between A and B.

```
    \@venn@computecircleintersects{\@venn@Ax}{\@venn@Ay}{\@venn@Bx}{\@venn@By}%
```

If there aren't two points of intersection, then the circles don't overlap.

```
    \ifnum\@venn@intersect@n=2\relax
```

Compute the start angle for the first arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@i@angle\pgfmathresult
```

Compute the end angle for the first arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Bx}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@By}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@i@angle\pgfmathresult
%       \pgfmathadd{\@venn@end@i@angle}{360}%
%       \let\@venn@end@i@angle\pgfmathresult
```

Compute the start angle for the second arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@ii@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@ii@y-\@venn@Ay}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@start@ii@angle\pgfmathresult
```

Compute the end angle for the second arc.

```
      \pgfmathsetlength{\dimen@i}{\@venn@intersect@i@x-\@venn@Ax}%
      \pgfmathsetlength{\dimen@ii}{\@venn@intersect@i@y-\@venn@Ay}%
      \ifvennoldpgf
        \pgfmathatantwo{\dimen@i}{\dimen@ii}%
      \else
        \pgfmathatantwo{\dimen@ii}{\dimen@i}%
      \fi
      \let\@venn@end@ii@angle\pgfmathresult
%       \pgfmathadd{\@venn@start@ii@angle}{360}%
%       \let\@venn@start@ii@angle\pgfmathresult
```

Fill the path

```
      \path[fill=\@venn@shade]
       (\@venn@intersect@i@x,\@venn@intersect@i@y)
      arc[radius=\@venn@radius,start angle=\@venn@start@i@angle,
```

```
      end angle=\@venn@end@i@angle]
      arc[radius=\@venn@radius,start angle=\@venn@start@ii@angle,
      end angle=\@venn@end@ii@angle]
       -- cycle;
   \else
      \fillOnlyB
   \fi
}%
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.