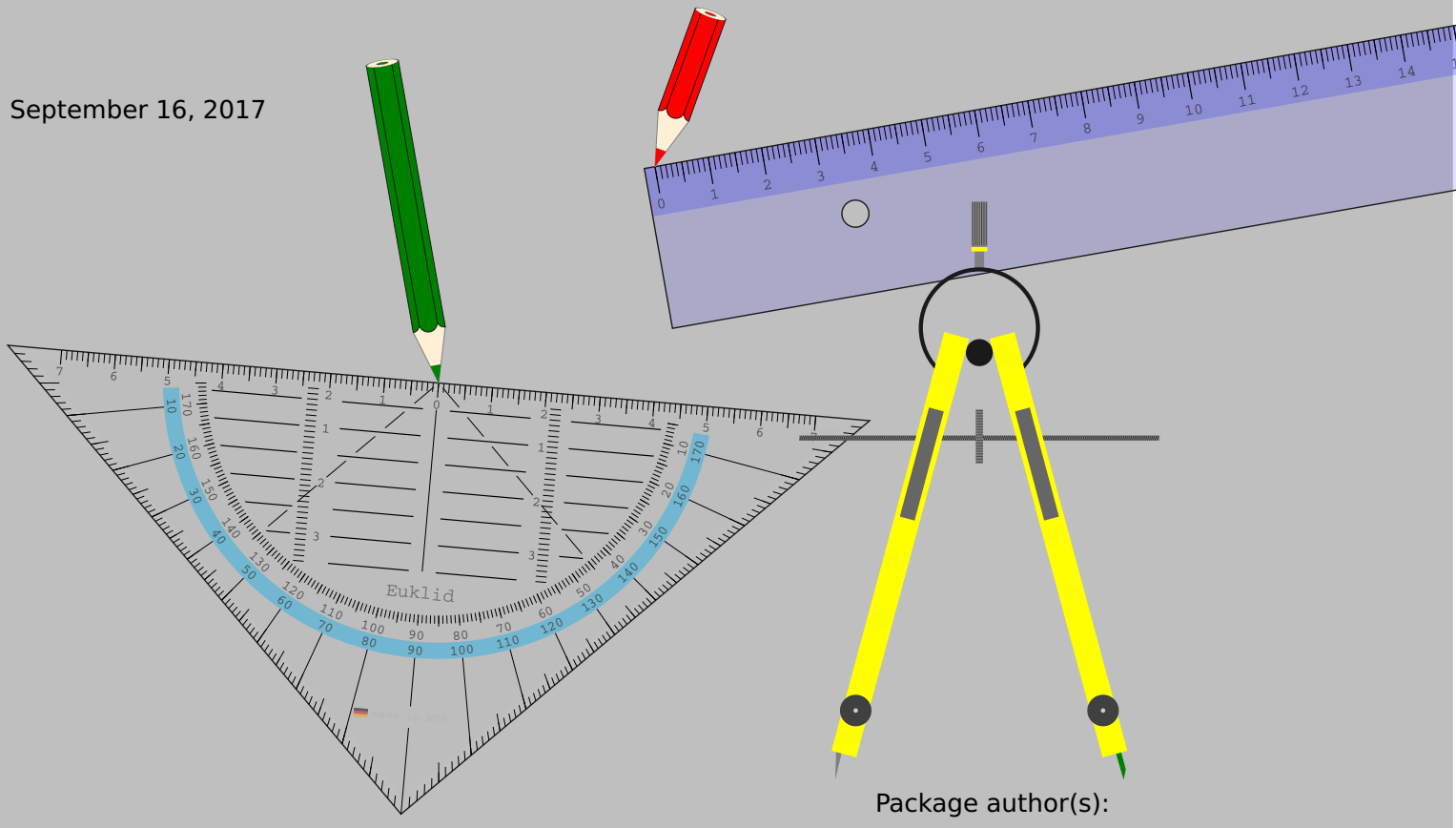
PSTricks

pst-geometrictools v 1.1

A PSTricks package to draw a protractor, a ruler, a compass and pencils

September 16, 2017



Package author(s):
Thomas Söll

Contents

1 How to use the commands 4
1.1 \psProtractor . . . 4
1.2 \psRuler . . . 4
1.3 \psCompass . . . 5
1.4 \psPencil . . . 5
1.5 \psParallels (Idea and realization by *Manuel Luque*) . . . 6

2 Basic Examples 7

3 Advanced examples 10

4 List of all optional arguments for `pst-geometrictools` 13

The package `pst-geometrictools` offers some customizable options to setup a *protractor*, a *ruler*, a *compass* and *pencils* to the users' wishes. Some geometric tools are predefined and ready-to-use for the customer to be able to present some pixel-free graphics showing the handling of some geometric tools.

The geometric tools can be scaled, rotated, positioned, colored as wanted, even labeled—if wanted.

These tools were already available since years within diverse examples—however not yet packaged together in pure PostScript. This was done within this package.

We recommend to use the package `pst-eucl` (by *Dominique Rodriguez*) which makes it easy to position the tools precisely.

Have fun to use it!

1 How to use the commands

1.1 \psProtractor

```
\psProtractor [Options] {angle}(coordinates of the origin)
\psProtractor [Options] {angle}(coordinates of the origin)(coordinates of a second point)
```

The command \psProtractor contains the options ProScale=, ProLineCol=, ProFillCol=, OwnerTxt=, MadeTxt=, PSfontO=, fontsizeO=, PSfontM=, fontsizeM=, country= and Ghost=.

Name	*Default*	*Meaning*
ProScale	1	scale factor
ProFillCol	gray!60	transparent fill color of the protrace
ProLineCol	cyan	linecolor of the protrace
OwnerTxt	T.S.	Name of the owner
MadeTxt	Made in NES	Made in wherever
PSfontO	Symbol	PSfont for the owner
fontsizeO	10pt	fontsize for the owner
PSfontM	Times-Roman	PSfont for Made in wherever
fontsizeM	6pt	fontsize for Made in wherever
country	Germany	Permissible values: Germany, France
Ghost	false	true: not showing the protrace itself, but getting the nodes of its verices

This above introduced command \psProtractor automatically provides three nodes of the vertices of the protractor which are named as follows: GeodrA, GeodrB, GeodrC
These nodes maybe very helpful.
The command \psProtractor offers a positioning

- either by one point and an angle of rotation
- or by two points (and an additional angle of rotation)

1.2 \psRuler

```
\psRuler [Options] {angle}(coordinates of origin)
\psRuler [Options] {angle}(coordinates of origin)(coordinates of a second point)
```

The command \psRuler contains the options RulerScale= and RulerFillCol=.

Name	*Default*	*Meaning*
RulerScale	1	scale factor of the ruler
RulerFillCol	gray	color of the ruler

The command \psRuler offers a positioning

- either by one point and an angle of rotation
- or by two points (and an additional angle of rotation)

1.3 \psCompass

```
\psCompass [Options] {radius}(coordinates of origin)
\psCompass [Options] {radius}(coordinates of origin)(coordinates of a second point)
```

The command \psCompass contains the options PoCAngle=, MCAngle=, PoCLength, PoCFillCol=, PoCMineCol=, RadVS=, AngleVS=, RadMul= and PoCScale=.

Name	*Default*	*Meaning*
PoCAngle	0	angle of rotation
PoCLength	5	length of the compass
PoCFillCol	gray!60	color of the compass
PoCMineCol	gray!60	color of the mine
PoCScale	1	scaling factor
MCAngle	false	suppresses the initial angle, when two points are given
RadVS	RVS	PostScript value for the radius
AngleVS	AVS	PostScript value for the angle
RadMul	1	multiplication factor for the radius

The command \psCompass offers a positioning

- either by one point and the radius—a rotation is set with [PoCAngle=30].
- or by two points (the radius and initial angle then are calculated automatically)—if one sets an additional rotation with [PoCAngle=30], these two angles are automatically added. If one will suppress the initial rotation, then set [MCAngle=false]. Using RadVS=<unique name> and AngleVS=<unique name> you can use the values of the radius respectively the angle within some PostScript calculations. When you like to use a radius other than the distance between the two points that automatically calculates the radius, use RadMul=<decimal number> to multiply the initial radius with this factor.

1.4 \psPencil

```
\psPencil [Options] {angle}(coordinates of the cone end)
```

The command \psPencil contains the options PenScale=, PenLength=, pencilColA= and pencilColB=.

Name	*Default*	*Meaning*
PenScale	1	scale factor
PenLength	5	length of the pencil
pencilColA	red	color of the pencil
pencilColB	HolzCol	color of the wooden cone end

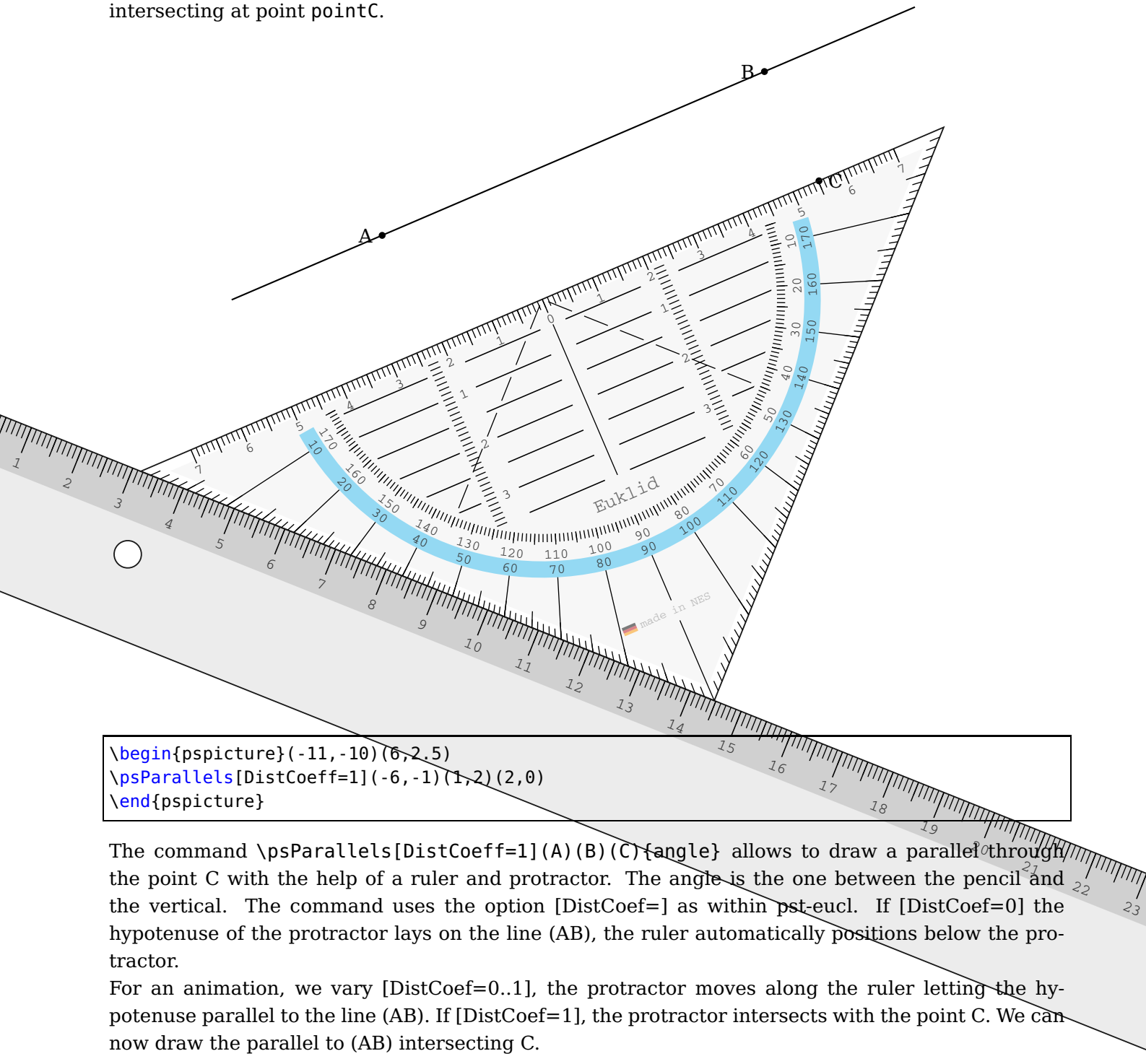
1.5 \psParallels (Idea and realization by *Manuel Luque*)

`\psParallels[Options](pointA)(pointB)(pointC){angle}`

The command `\psParallels` contains the option `DistCoeff=`.

Name	*Default*	*Meaning*
`DistCoeff`	1	relative disctance between (AB) and C

This command is made to directly draw a line parallel to the line containing `pointA` and `pointB` intersecting at point `pointC`.

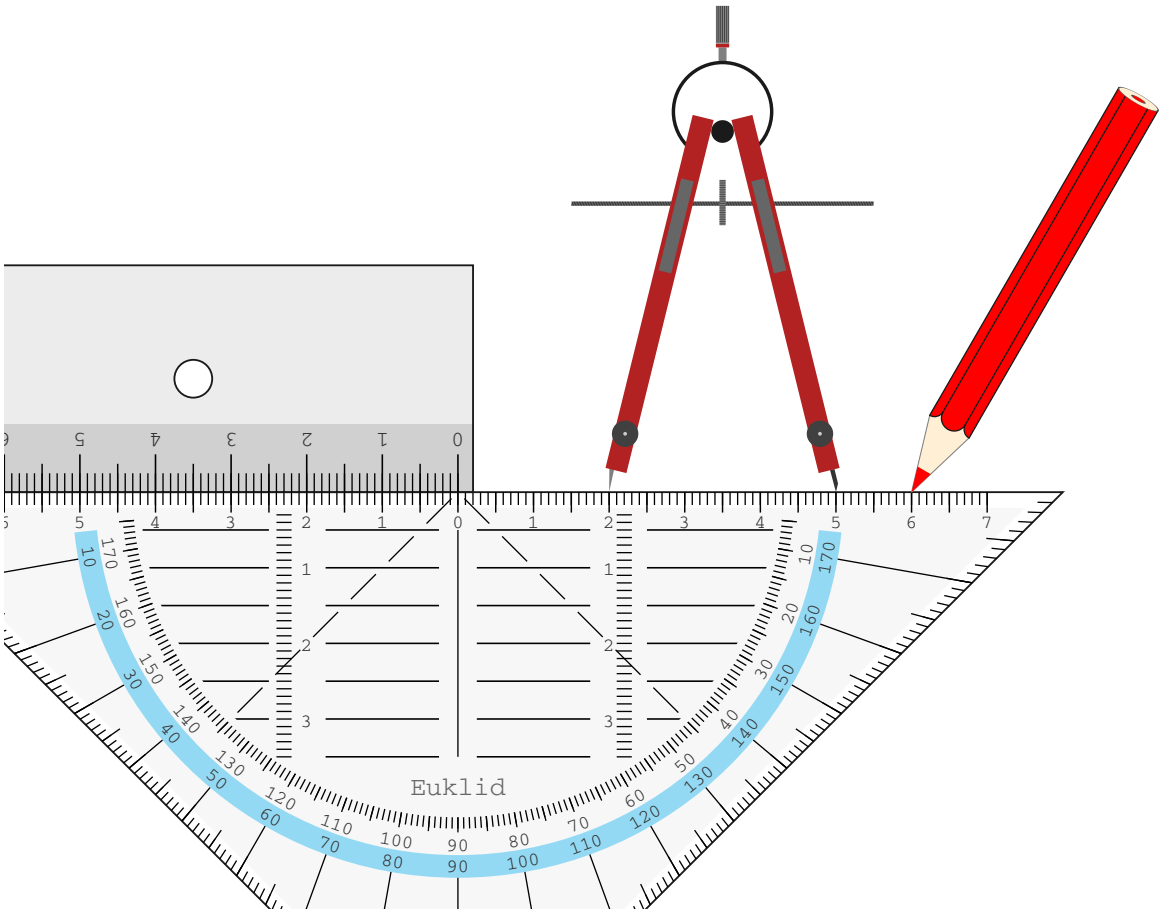



```
\begin{pspicture}(-11,-10)(6,2.5)
\psParallels[DistCoeff=1](-6,-1)(1,2)(2,0)
\end{pspicture}
```

The command `\psParallels[DistCoeff=1](A)(B)(C){angle}` allows to draw a parallel through the point C with the help of a ruler and protractor. The angle is the one between the pencil and the vertical. The command uses the option [DistCoef=] as within pst-eucl. If [DistCoef=0] the hypotenuse of the protractor lays on the line (AB), the ruler automatically positions below the protractor.

For an animation, we vary [DistCoef=0..1], the protractor moves along the ruler letting the hypotenuse parallel to the line (AB). If [DistCoef=1], the protractor intersects with the point C. We can now draw the parallel to (AB) intersecting C.

2 Basic Examples




```
\begin{pspicture}*(-6,-5.5)(10,7)
\psProtractor{0}(0,0)% origin of the protractor
\psRuler{0}(0,0)% origin of the ruler
\psPencil{-30}(6,0)% origin of the pencil
\psCompass{3}(2,0)% origin of the compass
\end{pspicture}
```

We see, that the origins of the *protractor* and *ruler*, *compass* and respectively the *cone end of the pencil* are positioned at $(0|0)$, $(2|0)$, $(6|0)$. Adding an angle rotates the objects around their origins. For the *protractor* and *ruler*, there are two ways to position them:

- one point and an angle, like:

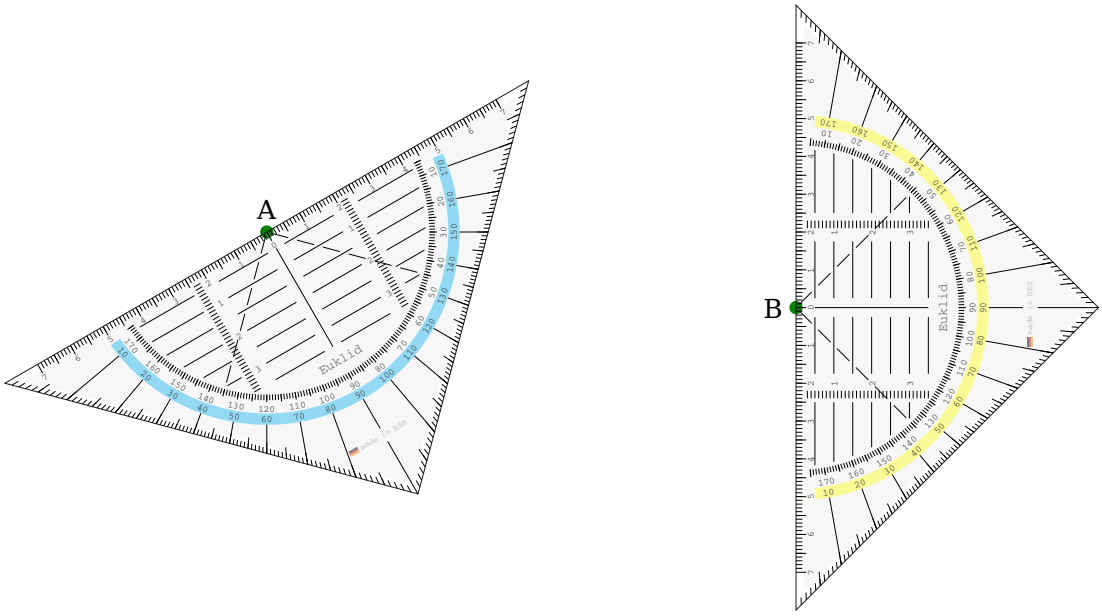
  `\psProtractor [Options] {<angle>}(<point>)`
  `\psRuler [Options] {<angle>}(<point>)`

  This is quite self-explanatory. The origin is positioned at the point and the tool is rotated around this point by the chosen angle.

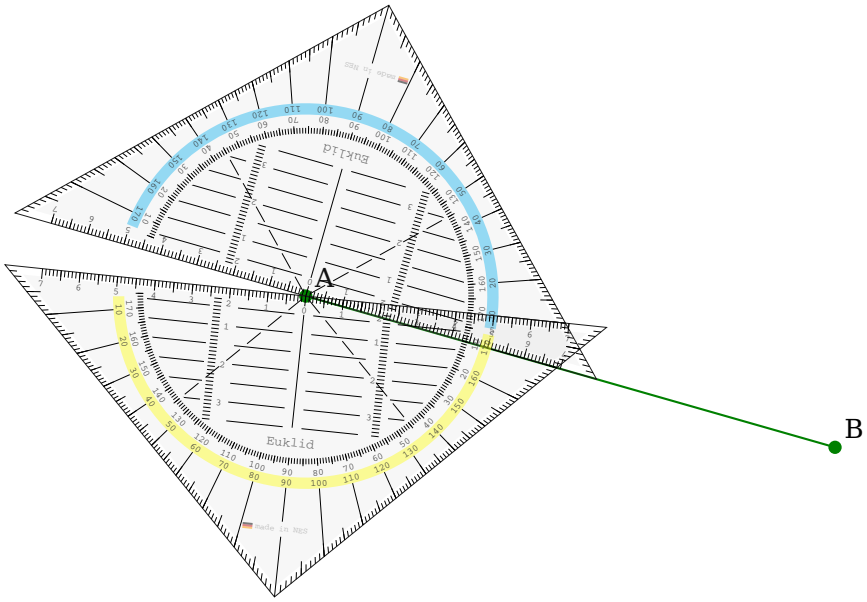
- two points and an additional angle, like:

  `\psProtractor [Options] {<additional angle>}(<pointA>)(<pointB>)`
  `\psRuler [Options] {<additional angle>}(<pointA>)(<pointB>)`

  The origin of the tool is set to pointA. The hypotenuse of the protractor is aligned to the line between pointA and pointB, when the additional angle is chosen to 0.

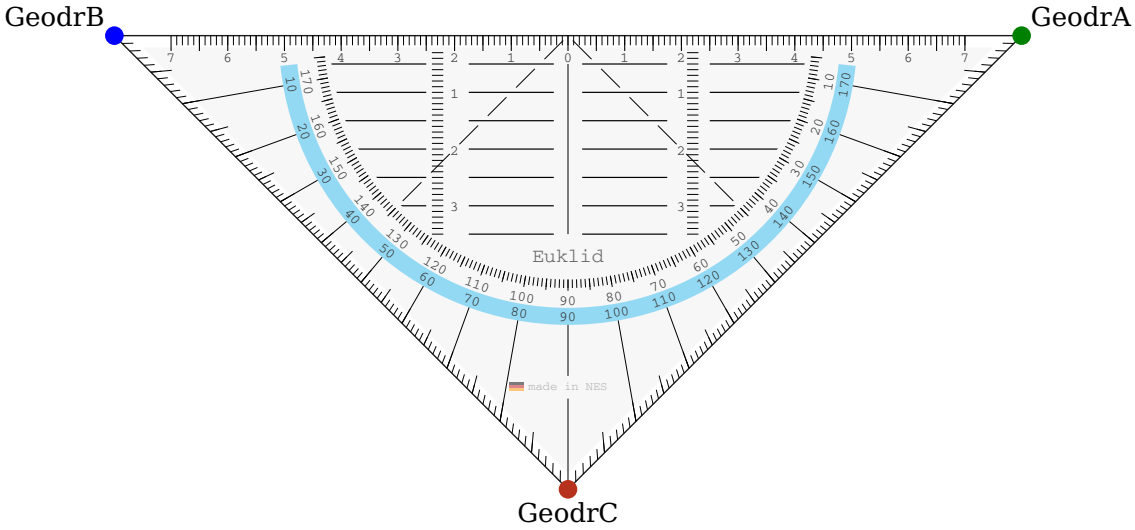



```
\begin{pspicture}(-8,-6.5)(8,2)
\pnode(-2,-1){A}\psdot[linecolor=Green,dotsize=5pt](A)\uput[90](A){A}
\pnode(5,-2){B}\psdot[linecolor=Green,dotsize=5pt](B)\uput[180](B){B}
\psProtractor[ProScale=0.5]{30}(A)
\psProtractor[ProLineCol=Yellow,ProScale=0.5]{90}(B)
\end{pspicture}
```



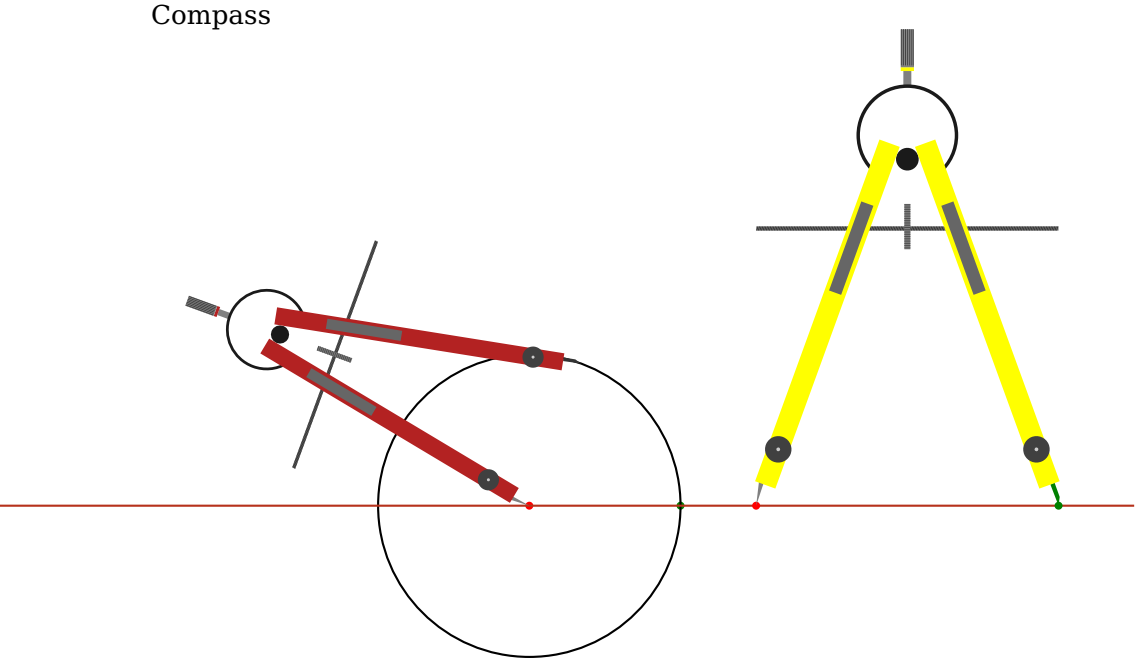

```
\begin{pspicture}(-8,-6.5)(5,2)
\pnode(-2,-2){A}\psdot[linecolor=Green,dotsize=5pt](A)\uput[45](A){A}
\pnode(5,-4){B}\psdot[linecolor=Green,dotsize=5pt](B)\uput[45](B){B}
\pcline[linecolor=Green](A)(B)
\psProtractor[ProScale=0.5]{0}(A)(B)
\psProtractor[ProLineCol=Yellow,ProScale=0.5]{190}(A)(B)
\end{pspicture}
```

The predefined nodes of a protractor.




```
\begin{pspicture}(-8,-6.5)(5,0.3)
\psProtractor[ProScale=0.75]{0}(0,0)
\psdot[linecolor=Green,dotsize=7pt](GeodrA)\uput[45](GeodrA){GeodrA}
\psdot[linecolor=Blue,dotsize=7pt](GeodrB)\uput[135](GeodrB){GeodrB}
\psdot[linecolor=BrickRed,dotsize=7pt](GeodrC)\uput[-90](GeodrC){GeodrC}
\end{pspicture}
```

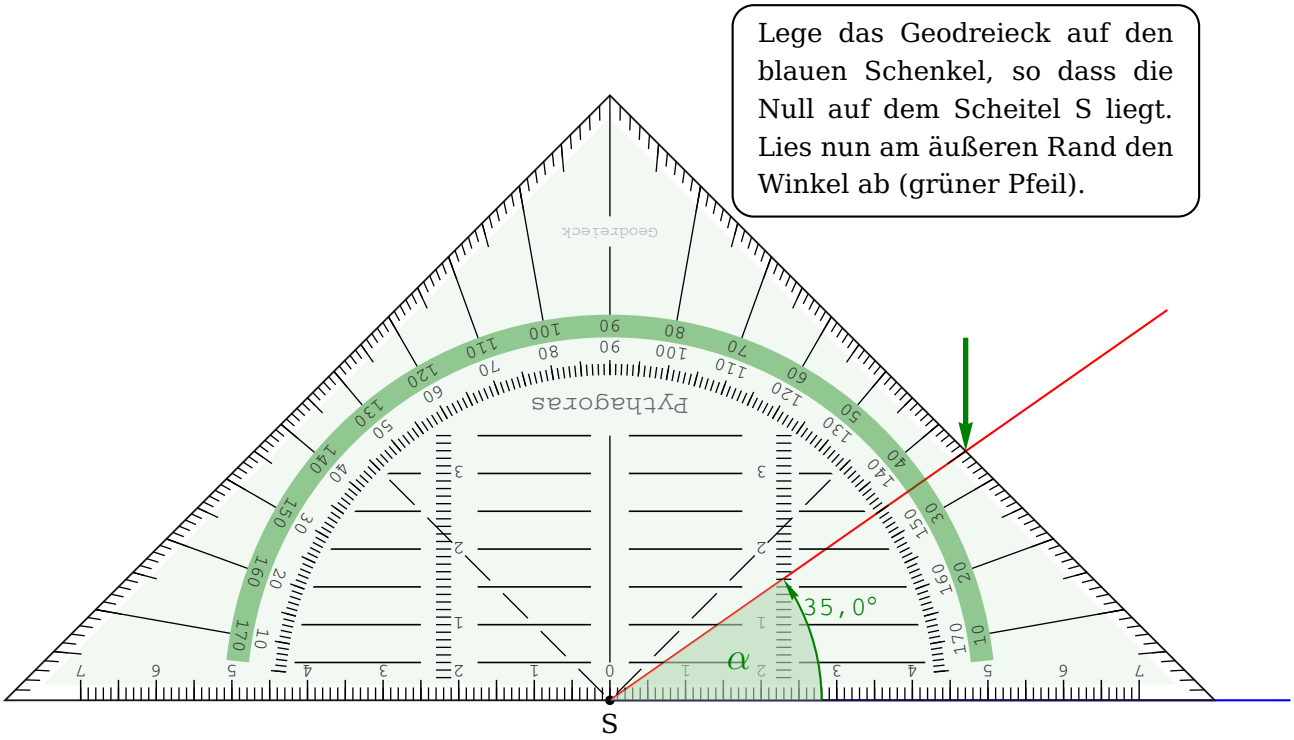
Compass



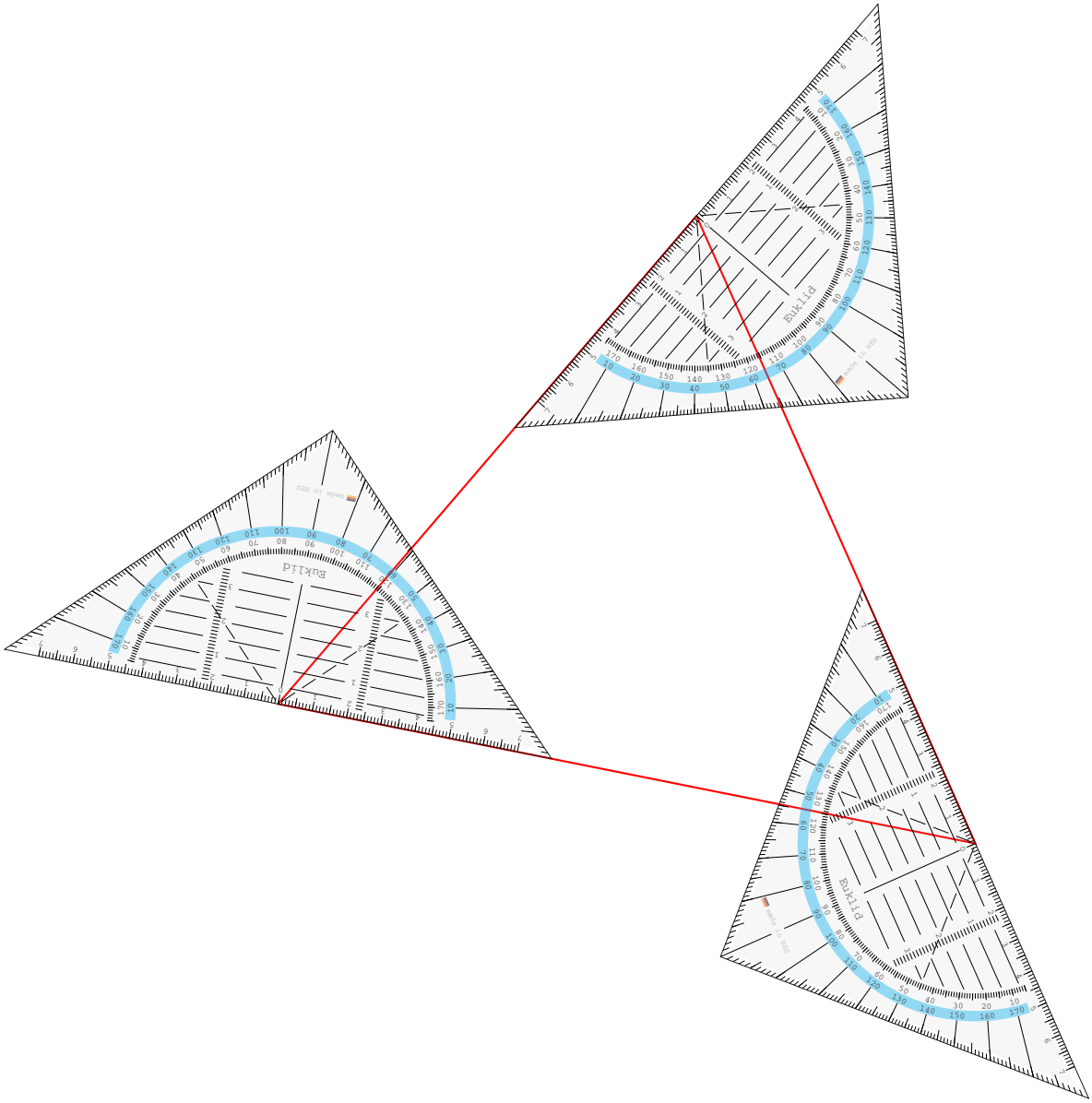
```
\begin{pspicture}(-6,-2)(6,6)%
\pnode(-1,0){A}\psdot[dotsize=3pt,linecolor=red](A)
\pnode(1,0){B}\psdot[dotsize=3pt,linecolor=Green](B)
\pscircle(A){2}
\pcline[linecolor=BrickRed,nodesepA=-9,nodesepB=-6](A)(B)
\psCompass[PoCAngle=70,PoCScale=0.8]{2}(A)
\pnode(2,0){A}\psdot[dotsize=3pt,linecolor=red](A)
\pnode(6,0){B}\psdot[dotsize=3pt,linecolor=Green](B)
\psCompass[PoCScale=1,PoCFillCol=Yellow,PoCAngle=0,PoCMineCol=Green]{5}(A)(B)
\end{pspicture}
```

3 Advanced examples

Measuring the angles of a given triangle.

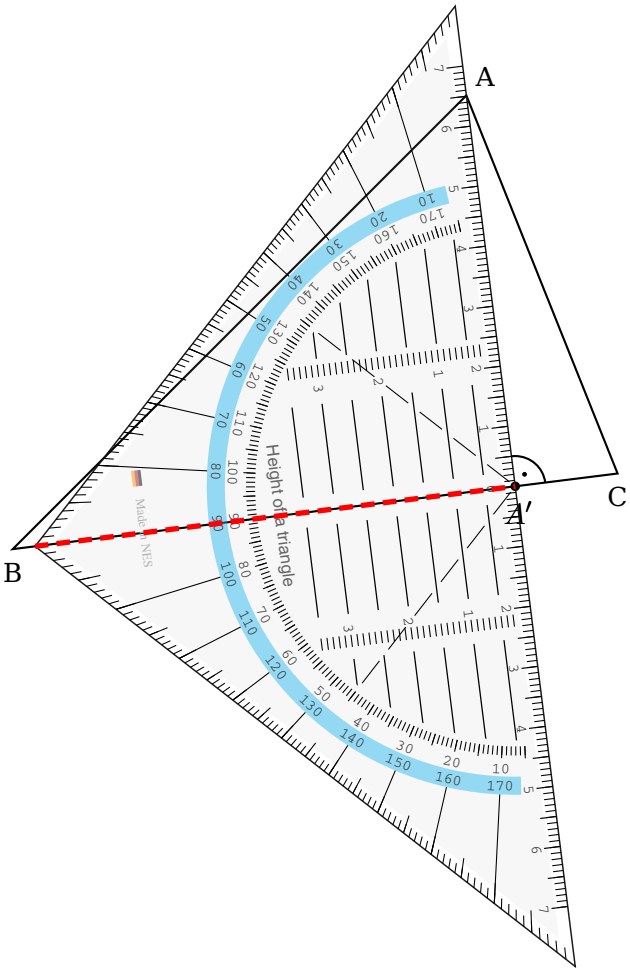



```
\newpsstyle{WkMessung}{linestyle=none,AngleValue=true,ArcColor=Green,arrows=->,WedgeOpacity
    =0.6,WedgeColor=Green!30,LabelSep=1.6,MarkAngleRadius=2.8,linecolor=Green,decimals=1,
    comma,xShift=-6,yShift=9}
\newpsstyle{GeoDrG}{country=G,ProScale=1,ProLineCol=Green,ProFillCol=Green!50,OwnerTxt={
    Pythagoras},MadeTxt={Geodreieck}}
\begin{pspicture}(0,-1)(17,10)
\pnode(8,0){S}\uput[d](S){S}\psdot(S)
\pnode(17,0){B}
\rput(S){\pnode(7;35){C}}
\pcline[linecolor=blue,nodesepB=-0](S)(B)
\pcline[linecolor=red,nodesepB=-2](S)(C)
\psProtractor[style=GeoDrG]{0}(S)(B)
\psIntersectionPoint(S)(C)(GeodrB)(GeodrC){D}% pstricks-add
\pcline[linecolor=Green,arrowinset=0.1,arrowlength=2,linewidth=2pt]{->}([offset=1.5cm]D)(D)
\psGetAngleABC[style=WkMessung](B)(S)(C){\Large\color{Green}$\alpha$}% pst-eucl
\rput([offset=4.5]D){\psframebox[framesep=6pt,framearc=0.2]{
\begin{minipage}[t]{5.5cm}
Lege das Geodreieck auf den blauen Schenkel, so dass die Null auf dem Scheitel S liegt.
    Lies nun am \"au\ss eren Rand den Winkel ab (gr\"uner Pfeil).
\end{minipage}
}}
\end{pspicture}
```



```
\begin{pspicture}(-9,-6)(7,10)
\pnode(1,7){A}
\pnode(-5,0){B}
\pnode(5,-2){C}
\pspolygon[linecolor=red](A)(B)(C)
\psProtractor[ProScale=0.5]{0}(A)(B)
\psProtractor[ProScale=0.5]{0}(B)(C)
\psProtractor[ProScale=0.5]{0}(C)(A)
\end{pspicture}
```

Constructing the height of a given triangle.




```
\begin{pspicture}(-6,-4.5)(5,8)
\pnode(1,7){A}
\pnode(-5,1){B}
\pnode(3,2){C}
\uput[ur](A){A}
\uput[d](B){B}
\uput[d](C){C}
\pspolygon(A)(B)(C)
\pstProjection{B}{C}{A}[A’]% pst-eucl
\psProtractor[ProScale=0.8,OwnerTxt={Height of a triangle},MadeTxt={Made in NES},PSfontO=
    Helvetica,PSfontM=Times-Roman,fontsizeO=8,fontsizeM=6]{0}(A’)(A)
\pcline[linecolor=red,linestyle=dashed,linewidth=2pt](GeodrC)(A’)
\pstRightAngle[RightAngleType=german]{C}{A’}{A}% pst-eucl
\end{pspicture}
```

4 List of all optional arguments for pst-geometrictools

Key	Type	Default
Ghost	boolean	true
ProLineCol	ordinary	cyan
ProFillCol	ordinary	gray!60
ProScale	ordinary	1
OwnerTxt	ordinary	Euklid
MadeTxt	ordinary	made in NES
PSfontO	ordinary	Symbol
fontsizeO	ordinary	10
PSfontM	ordinary	NimbusRomNo9L-Regu
fontsizeM	ordinary	6
country	ordinary	Germany
PenScale	ordinary	1
PenLength	ordinary	5
pencilColA	ordinary	red
pencilColB	ordinary	HolzCol
RulerFillCol	ordinary	cyan!60
RulerScale	ordinary	1
MCAngle	boolean	true
PoCLength	ordinary	5
PoCAngle	ordinary	0
PoCFillCol	ordinary	PoCRed
PoCMineCol	ordinary	black!80
PoCScale	ordinary	1
RadVS	ordinary	RVS
AngleVS	ordinary	AVS
RadMul	ordinary	1
DistCoeff	ordinary	1

Index

AngleVS, 5

country, 4

DistCoeff, 6

fontsizeM, 4
fontsizeO, 4

Ghost, 4

Keyword
    AngleVS, 5
    country, 4
    DistCoeff, 6
    fontsizeM, 4
    fontsizeO, 4
    Ghost, 4
    MadeTxt, 4
    MCAngle, 5
    OwnerTxt, 4
    pencilColA, 5
    pencilColB, 5
    PenLength, 5
    PenScale, 5
    PoCAngle, 5
    PoCFillCol, 5
    PoCLength, 5
    PoCMineCol, 5
    PoCScale, 5
    ProFillCol, 4
    ProLineCol, 4
    ProScale, 4
    PSfontM, 4
    PSfontO, 4
    RadMul, 5
    RadVS, 5
    RulerFillCol, 4
    RulerScale, 4

Macro
    \psCompass, 5
    \psParallels, 6
    \psPencil, 5
    \psProtractor, 4, 7
    \psRuler, 4, 7
MadeTxt, 4

MCAngle, 5

OwnerTxt, 4

Package
    pst-eucl, 3
    pst-geometrictools, 3
pencilColA, 5
pencilColB, 5
PenLength, 5
PenScale, 5
PoCAngle, 5
PoCFillCol, 5
PoCLength, 5
PoCMineCol, 5
PoCScale, 5
ProFillCol, 4
ProLineCol, 4
ProScale, 4
\psCompass, 5
PSfontM, 4
PSfontO, 4
\psParallels, 6
\psPencil, 5
\psProtractor, 4, 7
\psRuler, 4, 7
pst-eucl, 3
pst-geometrictools, 3

RadMul, 5
RadVS, 5
RulerFillCol, 4
RulerScale, 4