Tool AddVector(vector)

principle:

The tool returns a list of texts, and each text is a geogebra command.

The list has this caption: "!!! instructions !!!"

A global javascript function captures the name of the last object created. If this object is a list, and if the caption of the list is "!!! instructions !!!", then the list is executed.

This allows sequential instructions activated by a tool, on objects selected by the mouse.

The tool has naturally access to all the objects in the construction. Therefore it is not necessary to pass as parameters all the variables it will use, like conventional tools do. So it is possible to pass as parameters, only those objects that are different for each instance of the construction.

For example, here only the vector to be added is passed to the tool. All the other variables are either taken directly from the construction, or created if they don't exist.

It is also possible to check if a variable exists or not, and to create that variable if it is required.

Some commands require initialising variable names. In that case, 2 successive "execute" are necessary, the first one to replace the names with the variables, the second one to actually execute the command on the variables themselves. The first "execute" creates a variable which is a second list of instructions, where all the variable names have been replaced by the variables, and the second "execute", next in the sequence, executes that list of instructions that was just created.

Compared to the other methods proposed, where all the names are collected from all new texts in a java or GGB script, this construction allows to reduce significantly the number of times a variable is redefined, so the worksheet is more stable.

For an interactive funicular we need a list of objects that has live links to its elements, and that can be modified at will. "SetValue" copies only the values, not the link to the live element. So "SetValue" can't be used and we need to redefine the list each time a vector is added to it or removed from it. This operation is officially not recommended in scripts, because of the declarative nature of Geogebra's engine, and can be done only a limited number of times before Geogebra starts losing tracks and behave strangely. Therefore, from time to time (for example, for every 20 vector added or removed) the file must be saved and reopened, to allow Geogebra to rebuild fresh links.

The script must be edited in word, or in an object's scripting window, and tested line by line in Geogebra. It can then be copied in GGB, when defining a tool, as the object returned by the tool. Comments can be added as "word" comments only. Extra blanks and line feed are removed by GGB, so it becomes almost impossible to understand the script after pasted in Geogebra.

Contents

Global Java Script

```
function ggbOnInit() {
  ggbApplet.registerAddListener("exe")
  ggbApplet.registerRemoveListener("exe")
}

function exe() {
ggbApplet.unregisterAddListener("exe")
ggbApplet.unregisterRemoveListener("exe")

lastname=ggbApplet.getObjectName(ggbApplet.getObjectNumber()-1)
if (ggbApplet.getObjectType(lastname) == "list") {
   if (ggbApplet.getCaption(lastname,false) == "!!! instructions !!!" ) {
       ggbApplet.evalCommand("Execute(Object("+'"' +lastname+ '"'+"))")
       ggbApplet.deleteObject(lastname)
         }
   }

ggbApplet.registerAddListener("exe")
ggbApplet.registerRemoveListener("exe")

}
```

Tool Add Extra Vector (u:vector)

```
{"SetValue(AvInst, {"UnicodeToLetter(34)"
SetValue("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",
KeepIf(IsDefined(Object(AvLi)), AvLi,
     If(Name("Name(u)") ∈ "UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",
        Remove("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",{Name("Name(u)")}),
        Append("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",Name("Name(u)")))))"UnicodeToLetter(34)"})",
"Execute(AvInst)"}
```

tool addvector (u: vector)

```
{"AvInstTemp={}",
"SetValue(AvInstTemp,If(IsDefined(Object("UnicodeToLetter(34)"AvInst"UnicodeToLetter(34)")),{},
          {"UnicodeToLetter(34)"AvInst={}"UnicodeToLetter(34)"}))",
"Execute(AvInstTemp)",
```

Comment [A1]: creates list AvInst={} only once

```
"Delete(AvInstTemp)",

"SetValue(AvInst,If(IsDefined(Object("UnicodeToLetter(34)"tag"UnicodeToLetter(34)")),{},
          {"UnicodeToLetter(34)"tag=1"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,If(IsDefined(Object("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)")),{},
          {"UnicodeToLetter(34)"AvNm="UnicodeToLetter(34)"Name("UnicodeToLetter(34)"$$"UnicodeToLetter(34)")
          "UnicodeToLetter(34)" "UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst,
          {"UnicodeToLetter(34)"SetValue(AvNm,"UnicodeToLetter(34)Name(UnicodeToLetter(34)"Nm"
UnicodeToLetter(34)) "tag" UnicodeToLetter(34) ")"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,
{"UnicodeToLetter(34)"SetVisibleInView(AvNm,1,false) "UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetVisibleInView(AvNm,2,false) "UnicodeToLetter(34)",
"UnicodeToLetter(34)"SetVisibleInView(AvNm,-1,false) "UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,If(IsDefined(Object(AvNm)),{},{ AvNm"UnicodeToLetter(34)"={}"UnicodeToLetter(34)"}))",

"Execute(AvInst)",

"SetValue(AvInst, {"UnicodeToLetter(34)"
SetValue("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",
KeepIf(IsDefined(Object(AvLi)), AvLi,
If(Name("Name(u)") ∈ "UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)",
          Remove("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)", {Name("Name(u)")}),
          Append("UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)", Name("Name(u)")))))"UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A2]: creates number tag=1 if doesn't exist

Comment [A3]: creates text "Nmtag" only once

Comment [A4]: Updates the value of "Nmtag" each time a new list of forces is created

Comment [A5]: creates list namelist={} if doesn't exist

Comment [A6]: updates list namelist

Comment [A7]: creates text addvector_{vectorlist} = "vectors"tag

```
"AvVect="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVect,"UnicodeToLetter(34)"For"UnicodeToLetter(34)"tag)",
```

Comment [A8]:
creates list addvector_{vectorlist}

```
"SetValue(AvInst,If(IsDefined(Object(AvVect)),{},

 {AvVect"UnicodeToLetter(34)"=

          RemoveUndefined(Zip(Object(AvLi), AvLi,

                   "UnicodeToLetter(34)"AvNm"UnicodeToLetter(34)"))

                   "UnicodeToLetter(34)"}))",

"Execute(AvInst)",
```

Comment [A9]:
creates number Length(forces) if doesn't exist

```
"AvLen="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvLen,"UnicodeToLetter(34)"Len"UnicodeToLetter(34)"tag)",

"SetValue(AvInst,If(IsDefined(Object(AvLen)),{},
{AvLen"UnicodeToLetter(34)"=Length("UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")"UnicodeToLetter(34)"}))
","Execute(AvInst)",

"Delete(AvVect)","Delete(AvLen)"}
```

tool resnames (For1:list)

```
{"Ctag="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(Ctag,Take(Name("Name(For1)"),4))",

"AvVect="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVect,"UnicodeToLetter(34)"For"UnicodeToLetter(34)"Ctag)",

"AvPt="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"Ctag)",

"AVxv="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVxv,"UnicodeToLetter(34)"xv"UnicodeToLetter(34)"Ctag)",

"AVyv="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVyv,"UnicodeToLetter(34)"yv"UnicodeToLetter(34)"Ctag)",

"AVxp="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVxp,"UnicodeToLetter(34)"xp"UnicodeToLetter(34)"Ctag)",
"AVyp="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVyp,"UnicodeToLetter(34)"yp"UnicodeToLetter(34)"Ctag)",

"AVxr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVxr,"UnicodeToLetter(34)"xr"UnicodeToLetter(34)"Ctag)",

"AVyr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AVyr,"UnicodeToLetter(34)"yr"UnicodeToLetter(34)"Ctag)",

"AvXr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvXr,"UnicodeToLetter(34)"Xr"UnicodeToLetter(34)"Ctag)",

"AvYr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvYr,"UnicodeToLetter(34)"Yr"UnicodeToLetter(34)"Ctag)",

"AvResA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvResA,"UnicodeToLetter(34)"ResultA"UnicodeToLetter(34)"Ctag)",

"AvResB="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvResB,"UnicodeToLetter(34)"ResultB"UnicodeToLetter(34)"Ctag)",

"AvRes="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvRes,"UnicodeToLetter(34)"result"UnicodeToLetter(34)"Ctag)"}
```

tool delresnames (For1:list)

```
{"SetValue(Ctag,"Name(For1)")",

"Delete(Ctag)",

"Delete(AvVect)",

"Delete(AvPt)",

"Delete(AVxv)",

"Delete(AVyv)",

"Delete(AVxp)",

"Delete(AVyp)",

"Delete(AVxr)",

"Delete(AVyr)",
```

Comment [A10]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

"Delete(AvXr)",

"Delete(AvYr)",

"Delete(AvResA)",

"Delete(AvResB)",

"Delete(AvRes)"}

tool reacnames (For1:list)

{"Ctag="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(Ctag,Take(Name("Name(For1)"),4))",

"AvVect="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVect,"UnicodeToLetter(34)"For"UnicodeToLetter(34)"Ctag)",

"AvPt="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"Ctag)",

"AvResA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvResA,"UnicodeToLetter(34)"ResultA"UnicodeToLetter(34)"Ctag)",

"AvResB="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvResB,"UnicodeToLetter(34)"ResultB"UnicodeToLetter(34)"Ctag)",

"AvRes="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvRes,"UnicodeToLetter(34)"result"UnicodeToLetter(34)"Ctag)",

"AvLin="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvLin,"UnicodeToLetter(34)"Lin"UnicodeToLetter(34)"Ctag)",

"AvA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvB="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvDirA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvDirSeg="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvDirSeg,"UnicodeToLetter(34)"DirSeg"UnicodeToLetter(34)"Ctag)",

"AvPol="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPol,"UnicodeToLetter(34)"Pol"UnicodeToLetter(34)"Ctag)",

"AvF1="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvF1,"UnicodeToLetter(34)"F1"UnicodeToLetter(34)"Ctag)",

"AvRA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvRA,"UnicodeToLetter(34)"rA"UnicodeToLetter(34)"Ctag)",

"AvRB="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvRB,"UnicodeToLetter(34)"rB"UnicodeToLetter(34)"Ctag)"}

tool delreacnames (For1:list)

{"SetValue(Ctag,"Name(For1)")",

"Delete(Ctag)",

"Delete(AvVect)",

"Delete(AvPt)",

"Delete(AvResA)",

"Delete(AvResB)",

"Delete(AvRes)",

"Delete(AvLin)",

"Delete(AvA)",

"Delete(AvB)",

"Delete(AvDirA)",

"Delete(AvDirSeg)",

"Delete(AvPol)",

"Delete(AvF1)",

"Delete(AvRA)",

"Delete(AvRB)"}

Comment [A11]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

tool funinames (For1:list)

{"Ctag="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(Ctag,Take(Name("Name(For1)"),4))",

"AvVect="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVect,"UnicodeToLetter(34)"For"UnicodeToLetter(34)"Ctag)",

"AvLen="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvLen,"UnicodeToLetter(34)"Len"UnicodeToLetter(34)"Ctag)",

"AvPt="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"Ctag)",

"AvLin="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvLin,"UnicodeToLetter(34)"Lin"UnicodeToLetter(34)"Ctag)",

"AvA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvB="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvDirA="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"AvPol="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPol,"UnicodeToLetter(34)"Pol"UnicodeToLetter(34)"Ctag)",

"AvF1="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvF1,"UnicodeToLetter(34)"F1"UnicodeToLetter(34)"Ctag)",

"AvFuni="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvFuni,"UnicodeToLetter(34)"Funi"UnicodeToLetter(34)"Ctag)",

"AvHi="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvHi,"UnicodeToLetter(34)"Hi"UnicodeToLetter(34)"Ctag)",

"AvTr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvTr,"UnicodeToLetter(34)"Tr"UnicodeToLetter(34)"Ctag)"}

tool delfnames (For1:list)

{"SetValue(Ctag,"Name(For1)")",

Comment [A12]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

"Delete(Ctag)",

"Delete(AvVect)",

"Delete(AvLen)",

"Delete(AvPt)",

"Delete(AvLin)",

"Delete(AvA)",

"Delete(AvB)",

"Delete(AvDirA)",

"Delete(AvPol)",

"Delete(AvF1)",

"Delete(AvFuni)",

"Delete(AvHi)",

"Delete(AvTr)"}

tool dynames (For1:list)

{"Ctag="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(Ctag,Take(Name("Name(For1)"),4))",

"AvPol="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPol,"UnicodeToLetter(34)"Pol"UnicodeToLetter(34)"Ctag)",

"AvF1="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvF1,"UnicodeToLetter(34)"F1"UnicodeToLetter(34)"Ctag)",

"AvFuni="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvFuni,"UnicodeToLetter(34)"Funi"UnicodeToLetter(34)"Ctag)",

"AvRe="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvRe,"UnicodeToLetter(34)"Reac"UnicodeToLetter(34)"Ctag)",

"AvVeSum="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVeSum,"UnicodeToLetter(34)"VeSum"UnicodeToLetter(34)"Ctag)",

"AvVeSumTr="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvVeSumTr,"UnicodeToLetter(34)"VeSumTr"UnicodeToLetter(34)"Ctag)"}

tool deldynames (For1:list)

```
{"SetValue(Ctag,"Name(For1)")",
"Delete(Ctag)",
"Delete(AvPol)",
"Delete(AvF1)",
"Delete(AvFuni)",
"Delete(AvRe)",
"Delete(AvVeSum)",
"Delete(AvVeSumTr)"}
```

Comment [A13]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

tool complete set (For1:list, startP, endP, dir: 3 points)

```
{"resultant("Name(For1)")",
"reactions("Name(For1)", "Name(startP)" , "Name(endP)" , "Name(dir)")",
"funicular("Name(For1)", "Name(startP)" , "Name(endP)" , "Name(dir)")",
"dynamic("Name(For1)")",
"segments("Name(For1)")"}
```

tool resultant (For1:list)

```
{"Execute(resnames("Name(For1)"))",

"SetValue(AvInst,{AvPt"UnicodeToLetter(34)"=Zip(Point(AvLi,1),AvLi,"UnicodeToLetter(34)"AvVect"UnicodeToLetter(
34)")"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AVxv"UnicodeToLetter(34)"=Zip(x(AvLi),AvLi,"UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")
"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AVyv"UnicodeToLetter(34)"=Zip(y(AvLi),AvLi,"UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")
"UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A14]: creates Pttag

Comment [A15]: creates xv and yv

```
"SetValue(AvInst,{AVxp"UnicodeToLetter(34)"=Zip(x(AvLi),AvLi,"UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AVyp"UnicodeToLetter(34)"=Zip(y(AvLi),AvLi,"UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A16]: creates xp and yp

```
"SetValue(AvInst,{AVxr"UnicodeToLetter(34)"=Sum("UnicodeToLetter(34)"AVxv"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AVyr"UnicodeToLetter(34)"=Sum("UnicodeToLetter(34)"AVyv"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A17]: creates xr and yr

```
"SetValue(AvInst,

{AvXr"UnicodeToLetter(34)"=

          If("UnicodeToLetter(34)"AVyr"UnicodeToLetter(34)"==0,

                    Sum("UnicodeToLetter(34)"AVxp"UnicodeToLetter(34)")

                    / Length("UnicodeToLetter(34)"AVxp"UnicodeToLetter(34)"),

                    Sum("UnicodeToLetter(34)"AVxp"UnicodeToLetter(34)" "UnicodeToLetter(34)"AVyv
          "UnicodeToLetter(34)")

                    / "UnicodeToLetter(34)"AVyr"UnicodeToLetter(34)")

          "UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A18]: creates Xr

```
"SetValue(AvInst,

{AvYr"UnicodeToLetter(34)"=

          If("UnicodeToLetter(34)"AVxr"UnicodeToLetter(34)"==0,

                    Sum("UnicodeToLetter(34)"AVyp"UnicodeToLetter(34)")

                    / Length("UnicodeToLetter(34)"AVyp"UnicodeToLetter(34)"),

                    Sum("UnicodeToLetter(34)"AVyp"UnicodeToLetter(34)" "UnicodeToLetter(34)"AVxv
          "UnicodeToLetter(34)")
```

Comment [A19]: creates Yr

```
                    / "UnicodeToLetter(34)"AVxr"UnicodeToLetter(34)")

          "UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{AvResA"UnicodeToLetter(34)"=("UnicodeToLetter(34)"AvXr"UnicodeToLetter(34)", "UnicodeToLetter(34)"
AvYr"UnicodeToLetter(34)")"UnicodeToLetter(34)" })",

"Execute(AvInst)",

"SetValue(AvInst,

{AvResB"UnicodeToLetter(34)"=("UnicodeToLetter(34)"AvXr"UnicodeToLetter(34)"+"UnicodeToLetter(34)"AVxr"Uni
codeToLetter(34)" })",

"Execute(AvInst)" ,

"SetValue(AvInst,

{"UnicodeToLetter(34)"AvYr"UnicodeToLetter(34)"+"UnicodeToLetter(34)"AVyr"UnicodeToLetter(34)")"UnicodeToL
etter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AvRes"UnicodeToLetter(34)"=

Vector("UnicodeToLetter(34)"AvResA"UnicodeToLetter(34)",

 "UnicodeToLetter(34)" AvResB"UnicodeToLetter(34)")

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AvRes"UnicodeToLetter(34)"=

Vector("UnicodeToLetter(34)"AvResA"UnicodeToLetter(34)",

 "UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)")

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"Execute(delresnames("Name(For1)"))"}
```

Comment [A20]: creates points ResultA tag and ResultB tag

Comment [A21]: creates vector(ResultA tag, ResultB tag)

tool reactions (For1: list, startP, endP, dir: 3 points)

{"Execute(reacnames("Name(For1)"))",

"SetValue(AvA,"UnicodeToLetter(34) Name(startP) UnicodeToLetter(34)")",

"SetValue(AvB,"UnicodeToLetter(34) Name(endP) UnicodeToLetter(34)")",

"SetValue(AvDirA,"UnicodeToLetter(34) Name(dir) UnicodeToLetter(34)")",

"Execute(reaconly("Name(For1)"))",

"Execute(lines("Name(For1)"))",

"Execute(nicereactions("Name(For1)"))",

"Execute(delreacnames("Name(For1)"))"}

tool reaconly (For1: list)

{"SetValue(Ctag,"Name(For1)")",

Comment [A22]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

"SetValue(AvInst,{AvRA"UnicodeToLetter(34)"=ReactionA("UnicodeToLetter(34)"AvResA"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvA"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvB"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)") "UnicodeToLetter(34)",

AvRB"UnicodeToLetter(34)"=Vector("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",Translate("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)","UnicodeToLetter(34)"AvRes"UnicodeToLetter(34)"-"UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)"))"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,{AvPol"UnicodeToLetter(34)"=(0,0)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetValue("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)",Translate("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)","UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)"))"UnicodeToLetter(34)",

AvDirSeg"UnicodeToLetter(34)"=Segment("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)")"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvF1"UnicodeToLetter(34)"= Translate("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)",-"UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)"}

tool nicereactions (For1:list)

```
{"SetValue(Ctag,"Name(For1)")",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvDirSeg"UnicodeToLetter(34)",black)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineStyle("UnicodeToLetter(34)"AvDirSeg"UnicodeToLetter(34)",1)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvDirSeg"UnicodeToLetter(34)",2)"UnicodeToLetter(
34)",

"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvDirSeg"UnicodeToLetter(34)",false)"UnicodeToLetter(34)
",

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)",9)"UnicodeToLetter(34)"

})","Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",9)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvB"UnicodeToLetter(34)",9)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)",2)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)",9)"UnicodeToLetter(34)"

})","Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)",red)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)",9)"UnicodeToLetter(34)"
,

"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)",true)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvRB"UnicodeToLetter(34)",red)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvRB"UnicodeToLetter(34)",9)"UnicodeToLetter(34)"
,

"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvRB"UnicodeToLetter(34)",true)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvF1"UnicodeToLetter(34)",false)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvF1"UnicodeToLetter(34)",1)"UnicodeToLetter(34)"
```

Comment [A23]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

})","Execute(AvInst)"}

tool lines (For1:list)

------------ names and AvInst must exist before calling this tool ----------------

{"SetValue(AvInst, {"Name(For1)"})",

"SetValue(AvInst,

{AvLin"UnicodeToLetter(34)"=

Zip(Line(AvLi1, AvLi2),

AvLi1, "UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)",

AvLi2, "UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",1,false) "UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",2,false) "UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",-1,false) "UnicodeToLetter(34)"}

)",

"Execute(AvInst)"}

tool funicular (For1: list, startP, endP, dir: 3 points)

{"Execute(funinames("Name(For1)"))",

"SetValue(AvA,"UnicodeToLetter(34) Name(startP) UnicodeToLetter(34)")",

"SetValue(AvB,"UnicodeToLetter(34) Name(endP) UnicodeToLetter(34)")",

"SetValue(AvDirA,"UnicodeToLetter(34) Name(dir) UnicodeToLetter(34)")",

"Execute(funionly("Name(For1)"))",

"Execute(nicefuni("Name(For1)"))",

"Execute(delfnames("Name(For1)"))"}

tool funionly (For1:list)

-- AvInst and names must exist before calling this tool -------

{"SetValue(AvInst, {"Name(For1)"})",

"SetValue(AvInst,{AvFuni "UnicodeToLetter(34)"=IterationList(ilHv4(AvLi,"UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)","UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)","UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)",catch),AvLi,{{1,Sequence("UnicodeToLetter(34)"AvLen"UnicodeToLetter(34)"),"UnicodeToLetter(34)"AvA"UnicodeToLetter(34)",Vector("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)","UnicodeToLetter(34)"AvF1"UnicodeToLetter(34)")}},"UnicodeToLetter(34)" AvLen"UnicodeToLetter(34)") "UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvHi"UnicodeToLetter(34)"=Append(Zip(Element(AvLi,3),AvLi,"UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"),"UnicodeToLetter(34)"AvB"UnicodeToLetter(34)")"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvTr"UnicodeToLetter(34)"=Polyline("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)")"UnicodeToLetter(34)"})","Execute(AvInst)"}

tool nicefuni(For1:list)

{"SetValue(AvInst, {"Name(For1)"})",

Comment [A24]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",5)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",2)"UnicodeToLetter(34)"

})","Execute(AvInst)",

"SetValue(AvInst,{

"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineStyle("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",4)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",7)"UnicodeToLetter(34)",

"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",false)"UnicodeToLetter(34)"

})","Execute(AvInst)"}

tool dynamic (For1:list)

{"Execute(dynames("Name(For1)"))",

"Execute(dynaonly("Name(For1)"))",

"Execute(nicedyn("Name(For1)"))",

"Execute(deldynames("Name(For1)"))"}

tool dynaonly (For1:list)

{"SetValue(AvInst,{"Name(For1)"})",

"SetValue(AvInst,{AvRe"UnicodeToLetter(34)"=Zip(Element(AvLi,4),AvLi,"UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)","SetValue(AvInst,{AvVeSum"UnicodeToLetter(34)"=Zip(Translate("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)",Vector(AvLi)),Vector(AvLi),"UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)")"UnicodeToLetter(34)"})",

"Execute(AvInst)","SetValue(AvInst,{AvVeSumTr"UnicodeToLetter(34)"=Zip(Vector(a,b),a,Append("UnicodeToLetter(34)"AvF1"UnicodeToLetter(34)","UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)"),b,"UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)")"UnicodeToLetter(34)"})","Execute(AvInst)"}

tool nicedyn (For1:list)

{"SetValue(AvInst,{"Name(For1)"})",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)",2)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)",0
)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)",2
)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)",
black)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvVeSumTr"UnicodeToLetter(34)", red)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvVeSumTr"UnicodeToLetter(34)", 2)"UnicodeToLetter(34)"})","Execute(AvInst)"}

tool segments (For1: list)

{"Ctag="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(Ctag,Take(Name("Name(For1)"),4))",

"AvPt="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvPt,"UnicodeToLetter(34)"Pt"UnicodeToLetter(34)"Ctag)",

"AvFuni="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvFuni,"UnicodeToLetter(34)"Funi"UnicodeToLetter(34)"Ctag)",

"AvHi="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvHi,"UnicodeToLetter(34)"Hi"UnicodeToLetter(34)"Ctag)",

"AvOrPt="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvOrPt,"UnicodeToLetter(34)"OrPt"UnicodeToLetter(34)"Ctag)",

"AvSeg="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(AvSeg,"UnicodeToLetter(34)"Seg"UnicodeToLetter(34)"Ctag)",

"SetValue(AvInst,{AvOrPt"UnicodeToLetter(34)"=Sequence("UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)"("UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"(n,1)),n,2,Length("UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"))"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvSeg"UnicodeToLetter(34)"=Zip(Segment(a, b),a,Take("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",2),b,"UnicodeToLetter(34)"AvOrPt"UnicodeToLetter(34)")"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)",black)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)",2)"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineStyle("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)",1)"UnicodeToLetter(34)"})","Execute(AvInst)",

"Delete(Ctag)",

"Delete(AvPt)",

"Delete(AvFuni)",

"Delete(AvHi)",

"Delete(AvOrPt)",

"Delete(AvSeg)"}

++

tool funicular(For1:list)

++

{

"AvInst={}",

Comment [A25]: creates instructions={}

"SetValue(AvInst, {"Name(endP)"})",

Comment [A26]: calls endP for forcing taking endP as a parameter

"SetValue(AvInst, createnames("Name(For1)"))",

Comment [A27]: creates all the names

"Execute(AvInst)",

"SetValue(AvInst,

Comment [A28]: creates list linestag

```
{AvLin"UnicodeToLetter(34)"=
        Zip(Line(AvLi1, AvLi2),
                AvLi1, "UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)",
                AvLi2, "UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")
        "UnicodeToLetter(34)"})",
```

"Execute(AvInst)",

"SetValue(AvInst,

Comment [A29]: makes list linestag invisible

{"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",1,false)
"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",2,false) "UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetVisibleInView("UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)",-1,false) "UnicodeToLetter(34)"}

)",

"Execute(AvInst)",

Comment [A30]: creates poletag

"SetValue(AvInst,

{AvPol"UnicodeToLetter(34)"=Intersect(Line(Intersect(Line("UnicodeToLetter(34)"Name(forcediag) "UnicodeToLetter(34)",Line("UnicodeToLetter(34)"Name(startP) "UnicodeToLetter(34)", "UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)")), Line(Translate("UnicodeToLetter(34)"Name(forcediag) "UnicodeToLetter(34)", Vector(("UnicodeToLetter(34)"AVxr"UnicodeToLetter(34)", "UnicodeToLetter(34)"AVyr"UnicodeToLetter(34)"))), Line("UnicodeToLetter(34)"Name(endP) "UnicodeToLetter(34)", "UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)"))), Line("UnicodeToLetter(34)"Name(startP) "UnicodeToLetter(34)", "UnicodeToLetter(34)" Name(endP) "UnicodeToLetter(34)")), Line("UnicodeToLetter(34)"Name(forcediag) "UnicodeToLetter(34)", Line("UnicodeToLetter(34)"Name(startP) "UnicodeToLetter(34)", "UnicodeToLetter(34)"Name(dir) "UnicodeToLetter(34)"))) "UnicodeToLetter(34)"}

)",

"Execute(AvInst)",

Comment [A31]: creates list funitag

"SetValue(AvInst,

{

AvFuni "UnicodeToLetter(34)" =

IterationList(

    ilHv4(AvLi, "UnicodeToLetter(34)"AvLin"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)", "UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)", catching_{radius}),

    AvLi,

    {{1,

    Sequence(Length("UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")),

    "UnicodeToLetter(34)"Name(startP) "UnicodeToLetter(34)",

    Vector("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)", "UnicodeToLetter(34)"Name(forcediag) "UnicodeToLetter(34)")}},

    Length("UnicodeToLetter(34)"AvVect"UnicodeToLetter(34)")) "UnicodeToLetter(34)"

}

```
)",

"Execute(AvInst)",

"SetValue(AvInst,

{

AvHi"UnicodeToLetter(34)" =

Append(

        Zip(Element(AvLi, 3), AvLi, "UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"),

        "UnicodeToLetter(34)"Name(endP) "UnicodeToLetter(34)")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,

{

AvTr"UnicodeToLetter(34)" =

Polyline("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,

{

AvRe"UnicodeToLetter(34)" =

Zip(Element(AvLi, 4), AvLi, "UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,
```

Comment [A32]: creates list hingestag

Comment [A33]: creates polylineTrtag

Comment [A34]: creates list Reactionstag

Comment [A35]: creates list VectorsSumtag

{

AvVeSum"UnicodeToLetter(34)" =

Zip(Translate("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)", Vector(AvLi)), Vector(AvLi), "UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

Comment [A36]: creates list VectorsSumTracetag

"SetValue(AvInst,

{

AvVeSumTr"UnicodeToLetter(34)" =

Zip(Vector(a, b), a, Append(" Name(forcediag) ", "UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)"), b, "UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

Comment [A37]: creates vector RAtag

"SetValue(AvInst,

{

AvRA"UnicodeToLetter(34)" =

Vector("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)"(1),Translate("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)"(1), -Vector("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)"," Name(forcediag) ")))

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

Comment [A38]: creates vector RBtag

"SetValue(AvInst,

{

AvRB"UnicodeToLetter(34)" =

Vector("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)"(-1),Translate("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)"(-1), "UnicodeToLetter(34)"AvRes"UnicodeToLetter(34)"- "UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)"))

```
"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,

{

AvOrPt"UnicodeToLetter(34)" =

Sequence("UnicodeToLetter(34)"AvPt"UnicodeToLetter(34)"("UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"(n,
1)), n, 2, Length("UnicodeToLetter(34)"AvFuni"UnicodeToLetter(34)"))

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,

{AvSeg"UnicodeToLetter(34)"=

Zip(Segment(a, b), a, Take("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",2), b,
"UnicodeToLetter(34)"AvOrPt"UnicodeToLetter(34)")

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{AvDirA"UnicodeToLetter(34)"=

Segment(" Name(startP) ","Name(dir) ")

"UnicodeToLetter(34)"

})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetPointStyle("Name(forcediag)", 2)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

Comment [A39]: creates list OrderPttag

Comment [A40]: creates list Segmentstag

Comment [A41]: creates segment DirAtag

```
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointSize("Name(forcediag)", 9)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointStyle("Name(startP)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointSize("Name(startP)", 9)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointStyle("Name(endP)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointSize("Name(endP)", 9)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointStyle("Name(dir)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointSize("Name(dir)", 9)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst, deletenames("Name(For1)"))",
"Execute(AvInst)",
"Delete(AvInst)"
}


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tool nicefuni(For1:list)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
{"Execute(createnames("Name(For1)"))",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetPointStyle("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",2)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetPointSize("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)",5)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",blue)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineStyle("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",4)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetLineThickness("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",7)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"ShowLabel("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)",false)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)",red)"UnicodeToLetter(34)"})","Execute(AvInst)",
"SetValue(AvInst,{"UnicodeToLetter(34)"SetColor("UnicodeToLetter(34)"AvRB"UnicodeToLetter(34)",red)"UnicodeToLetter(34)"})","Execute(AvInst)",
"Execute(deletenames("Name(For1)"))"}
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
tool looknice(For1:list)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
{

"Execute(createnames("Name(For1)"))",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)", blue)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointStyle("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointSize("UnicodeToLetter(34)"AvHi"UnicodeToLetter(34)", 5)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)", blue)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
```

Comment [A42]: creates all the names

```
SetLineStyle("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)", 4 )
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetLineThickness("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)", 7)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
ShowLabel("UnicodeToLetter(34)"AvTr"UnicodeToLetter(34)", false)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)", black)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetLineThickness("UnicodeToLetter(34)"AvRe"UnicodeToLetter(34)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetPointStyle("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)",0 )
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
```

```
SetPointSize("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)",2 )
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvVeSum"UnicodeToLetter(34)", black)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvVeSumTr"UnicodeToLetter(34)", red)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetLineThickness("UnicodeToLetter(34)"AvVeSumTr"UnicodeToLetter(34)", 2)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)", red)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
SetColor("UnicodeToLetter(34)"AvRB"UnicodeToLetter(34)", red)
"UnicodeToLetter(34)"})",
"Execute(AvInst)",
"SetValue(AvInst,
{"UnicodeToLetter(34)"
```

```
SetColor("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)", black)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetLineThickness("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)", 1)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetLineStyle("UnicodeToLetter(34)"AvSeg"UnicodeToLetter(34)", 1)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetColor("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)", black)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetLineThickness("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)", 1)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetLineStyle("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)", 1)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"
```

```
ShowLabel("UnicodeToLetter(34)"AvDirA"UnicodeToLetter(34)", false)

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"SetValue(AvInst,

{"UnicodeToLetter(34)"

SetValue(AvInst, deletenames("Name(For1)"))

"UnicodeToLetter(34)"})",

"Execute(AvInst)",

"Delete(AvInst)"

}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

tool createinst(nam:text)

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

{"AvInstTemp={}",

"SetValue(AvInstTemp,

        If(IsDefined(Object("UnicodeToLetter(34) nam UnicodeToLetter(34)")),

                {},

                {"UnicodeToLetter(34) nam"={}"UnicodeToLetter(34)"}

        ))",

"Execute(AvInstTemp)",

"Delete(AvInstTemp)"}


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


"addvector_{resultline}="UnicodeToLetter(34)" "UnicodeToLetter(34)"",

"SetValue(addvector_{resultline}, "UnicodeToLetter(34)"resultline"UnicodeToLetter(34)"Ctag)",
```

Comment [A43]: creates text resultline tag

Comment [A44]: creates line(ResultA tag, ResultB tag)

"SetValue(AvInst,

{AvRes"UnicodeToLetter(34)"= Line(

"UnicodeToLetter(34)"AvResA"UnicodeToLetter(34)",

 "UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)")

"UnicodeToLetter(34)"}

)",

"Execute(AvInst)",

+++

+++

tool deletenames(For1:list)

+++

{"SetValue(Ctag,"Name(For1)")",

Comment [A45]: dummy instruction to make the tool wizard believe that the output of the tool depends on some input

"Delete(Ctag)",

"Delete(AvVect)",

"Delete(AvLen)",

"Delete(AvPt)",

"Delete(AVxv)",

"Delete(AVyv)",

"Delete(AVxp)",

"Delete(AVyp)",

"Delete(AVxr)",

"Delete(AVyr)",

"Delete(AvXr)",

"Delete(AvYr)",

"Delete(AvResA)",

"Delete(AvResB)",

"Delete(AvRes)",

"Delete(AvLin)",

"Delete(AvPol)",

```
"Delete(AvPtA)",

"Delete(AvPtB)",

"Delete(AvDir)",

"Delete(AvDyn)",

"Delete(AvFuni)",

"Delete(AvHi)",

"Delete(AvTr)",

"Delete(AvRe)",

"Delete(AvVeSum)",

"Delete(AvVeSumTr)",

"Delete(AvRA)",

"Delete(AvRB)",

"Delete(AvOrPt)",

"Delete(AvSeg)",

"Delete(AvDirA)"}



+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

tool pole(startP,endP,dir,forcediag,AvPol,AvResB,AVxr,AVyr,AvResB)
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

{AvPol"=Intersect(Line(Intersect(Line("Name(forcediag)",Line("Name(startP)","AvResB")),Line(Translate("Name(force
diag) ",Vector(("AVxr"," AVyr"))),Line("Name(endP)
","AvResB"))),Line("Name(startP)","Name(endP)")),Line("Name(forcediag)",Line("Name(startP)","Name(dir)")))"}

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


"SetValue(AvInst,{

AvA"UnicodeToLetter(34)"="Name(startP) UnicodeToLetter(34)",

AvB"UnicodeToLetter(34)"="Name(endP) UnicodeToLetter(34)",

AvPol"UnicodeToLetter(34)"="Name(pole) UnicodeToLetter(34)",

AvDirA"UnicodeToLetter(34)"="Name(dir) UnicodeToLetter(34)"})",

"Execute(AvInst)",
```

++++++++++++++++++++

"SetValue(AvInst,{

AvRA"UnicodeToLetter(34)"=ReactionA("AvResA","AvResB","AvA","AvB","AvDirA")"UnicodeToLetter(34)",

AvRB"UnicodeToLetter(34)"=Vector((0,0))"UnicodeToLetter(34)",

"UnicodeToLetter(34)"SetValue("AvRB","AvRes"-"AvRA")"UnicodeToLetter(34)"

})",

"Execute(AvInst)"

tool pole (startP,endP,dir,forcediag: 4 points)

-- AvInst and names must exist before calling this tool -------

{"SetValue(AvInst,{AvPol"UnicodeToLetter(34)"=Intersect(Line(Intersect(Line("Name(forcediag)",Line("Name(startP)","UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)")),Line(Translate("Name(forcediag)",Vector(("UnicodeToLetter(34)"AVxr"UnicodeToLetter(34)","UnicodeToLetter(34)"AVyr"UnicodeToLetter(34)"))),Line("Name(endP)","UnicodeToLetter(34)"AvResB"UnicodeToLetter(34)"))),Line("Name(startP)","Name(endP)")),Line("Name(forcediag)",Line("Name(startP)","Name(dir)")))"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvRA"UnicodeToLetter(34)"=Vector("Name(startP)",Translate("Name(startP)",-Vector("UnicodeToLetter(34)"AvPol"UnicodeToLetter(34)","Name(forcediag)")))"UnicodeToLetter(34)"})","Execute(AvInst)",

"SetValue(AvInst,{AvRB"UnicodeToLetter(34)"=Vector("Name(endP)",Translate("Name(endP)","UnicodeToLetter(34)"AvRes"UnicodeToLetter(34)"-"UnicodeToLetter(34)"AvRA"UnicodeToLetter(34)"))"UnicodeToLetter(34)"})","Execute(AvInst)"}