457 (co-ordinates) In a graphical program, a pixel might be identified by its Cartesian co-ordinates x and y, or by its polar co-ordinates r (radius, or distance from the origin) and a (angle in radians counter-clockwise from the x axis). An operation written using one kind of co-ordinates may need to be transformed into the other kind of co-ordinates.

(a) What is the data transformer to transform from Cartesian to polar co-ordinates?

(b) In Cartesian co-ordinates, one of the operations on a pixel is *translate*, which moves a pixel from position x and y to position $x+u$ and $y+v$.

$$translate \;=\; x := x+u.\ y := y+v$$

Use the data transformer from (a) to transform operation *translate* from Cartesian to polar co-ordinates.

(c) What is the data transformer to transform from polar to Cartesian co-ordinates?

(d) In polar co-ordinates, one of the operations on a pixel is *rotate* by d radians.

$$rotate \;=\; a := a+d$$

Use the data transformer from (c) to transform operation *rotate*.

After trying the question, scroll down to the solution.

(a) What is the data transformer to transform from Cartesian to polar co-ordinates?

§ $x^2+y^2 = r^2 \ \wedge\ sin\ a = y/r \ \wedge\ cos\ a = x/r \ \wedge\ tan\ a = y/x$

This transform has some redundancy; any two of those conjuncts imply the other two. We can already see a constraint on its use: $r \neq 0 \ \wedge\ x \neq 0$. This constraint has some redundancy: if $x \neq 0$ then $r \neq 0$. To transform from Cartesian to polar, this transform is more conveniently written

$$x = r \times cos\ a \ \wedge\ y = r \times sin\ a$$

to use one-point laws to get rid of quantifications $\forall x, y\cdot$ and $\exists x', y'\cdot$.

(b) In Cartesian co-ordinates, one of the operations on a pixel is *translate* , which moves a pixel from position x and y to position $x+u$ and $y+v$.

$$translate \ =\ x:= x+u.\ y:= y+v$$

Use the data transformer from (a) to transform operation *translate* from Cartesian to polar co-ordinates.

§
$$\begin{aligned}
& \forall x, y\cdot \quad x = r \times cos\ a \ \wedge\ y = r \times sin\ a \\
& \qquad \Rightarrow \exists x', y'\cdot\ x' = r' \times cos\ a' \ \wedge\ y' = r' \times sin\ a' \ \wedge\ translate \\
= \ & \forall x, y\cdot \quad x = r \times cos\ a \ \wedge\ y = r \times sin\ a \\
& \qquad \Rightarrow \exists x', y'\cdot\ x' = r' \times cos\ a' \ \wedge\ y' = r' \times sin\ a' \ \wedge\ (x:= x+u.\ y:= y+v) \\
= \ & \forall x, y\cdot \quad x = r \times cos\ a \ \wedge\ y = r \times sin\ a \\
& \qquad \Rightarrow \exists x', y'\cdot\ x' = r' \times cos\ a' \ \wedge\ y' = r' \times sin\ a' \ \wedge\ x' = x+u \ \wedge\ y' = y+v \\
& \hfill \text{one-point } x'\ y' \\
= \ & \forall x, y\cdot \quad x = r \times cos\ a \ \wedge\ y = r \times sin\ a \\
& \qquad \Rightarrow r' \times cos\ a' = x+u \ \wedge\ r' \times sin\ a' = y+v \hfill \text{one-point } x\ y \\
= \ & r' \times cos\ a' = (r \times cos\ a)+u \ \wedge\ r' \times sin\ a' = (r \times sin\ a)+v
\end{aligned}$$

This is not yet a program. It appears that the way to get a *translate* program in polar co-ordinates is to transform to Cartesian, *translate* in Cartesian, then transform back to polar.

$$\begin{aligned}
= \ & \mathbf{new}\ x, y{:}\ real\cdot\ x:= (r \times cos\ a)+u.\ y:= (r \times sin\ a)+v. \\
& r:= (x^2+y^2)^{1/2}.\ a:= arctan\ (y/x)
\end{aligned}$$

(c) What is the data transformer to transform from polar to Cartesian co-ordinates?

§ The same transformer from part (a) works, but for this direction, it is more conveniently rewritten, using trigonometric identities, as

$$r = (x^2+y^2)^{1/2} \ \wedge\ a = arctan\ (y/x)$$

to use one-point laws to get rid of quantifications $\forall r, a\cdot$ and $\exists r', a'\cdot$.

(d) In polar co-ordinates, one of the operations on a pixel is *rotate* by d radians.

$$rotate \ =\ a:= a+d$$

Use the data transformer from (c) to transform operation *rotate* .

§
$$\begin{aligned}
& \forall r, a\cdot \quad r = (x^2+y^2)^{1/2} \ \wedge\ a = arctan\ (y/x) \\
& \qquad \Rightarrow \exists r', a'\cdot\ r' = (x'^2+y'^2)^{1/2} \ \wedge\ a' = arctan\ (y'/x') \ \wedge\ rotate \\
= \ & \forall r, a\cdot \quad r = (x^2+y^2)^{1/2} \ \wedge\ a = arctan\ (y/x) \\
& \qquad \Rightarrow \exists r', a'\cdot\ r' = (x'^2+y'^2)^{1/2} \ \wedge\ a' = arctan\ (y'/x') \ \wedge\ (a:= a+d) \\
= \ & \forall r, a\cdot \quad r = (x^2+y^2)^{1/2} \ \wedge\ a = arctan\ (y/x) \\
& \qquad \Rightarrow \exists r', a'\cdot\ r' = (x'^2+y'^2)^{1/2} \ \wedge\ a' = arctan\ (y'/x') \ \wedge\ (r'=r \ \wedge\ a' = a+d) \\
& \hfill \text{one-point } r'\ a' \\
= \ & \forall r, a\cdot \quad r = (x^2+y^2)^{1/2} \ \wedge\ a = arctan\ (y/x) \\
& \qquad \Rightarrow (x'^2+y'^2)^{1/2} = r \ \wedge\ arctan\ (y'/x') = a+d \hfill \text{one-point } r\ d \\
= \ & (x'^2+y'^2)^{1/2} = (x^2+y^2)^{1/2} \ \wedge\ arctan\ (y'/x') = arctan\ (y/x) + d \\
= \ & x'^2+y'^2 = x^2+y^2 \ \wedge\ arctan\ (y'/x') = arctan\ (y/x) + d
\end{aligned}$$

This is not yet a program. It appears that the way to get a *rotate* program in Cartesian co-ordinates is to transform to polar, *rotate* in polar, then transform back to Cartesian.

$=$ **new** r, a: *real*· $r := (x^2+y^2)^{1/2}$. $a := arctan\ (y/x) + d$.
$x := r \times cos\ a$. $y := r \times sin\ a$