ON MAKING A THEOREM FOR A CHILD

Seymour Papert
MIT

Part I: RHETORIC

I shall tell you about some theorems made for children. This does not mean that they are bad for adults--any more than Winnie-the-Pooh. The point is that most theorems (including the ones in the school books) are effectively X-rated. By this I mean that children can't get at them.

Before talking about my theorems I want to say something about what makes theorems good or bad. The most important factor is power. The best theorems have given men the power to think and do what no man could think or do before. But though all kids know how to rate cars by horsepower, they have never imagined there is such a thing as MATHPOWER. And it's not surprising: the stuff they call math at school gets you nowhere in doing or thinking anything. All you can do with long division is long divide...if that.

So good theorems for children should have a power punch. And the reason why I'm saying this to members of the Association for Computing Machinery is that you have the intellectual and material means to create contexts in which mathematics with a punch can be developed for use by children.

I do not mean by this merely that computers are powerful mathematical instruments and can make mathematics real for action-oriented kids. True enough; but you are scarcely scratching the surface when you use computers to teach, motivate or liven up the same old math. The concept of computation is beginning to spawn new mathematical topics of which I shall describe one in a moment. I maintain as obvious that those topics from computational mathematics are enormously better matched to, and enormously amplify, the use of the computer in learning math at any age. I also maintain, though this is less obvious, that these topics provide a better route into mathematical sophistication even in contexts where physical computers are not available. Finally, I maintain, though this is far from obvious, that the few examples we have of good theorems growing from computational roots are an indication that the soil is rich and inviting for people with a talent for research and a desire to contribute to the lives of children.

Part II: MATH

Computational geometry is an embryonic but growing branch of geometry concerned with the kinds of computation needed to generate, recognize or otherwise manipulate geometric figures. Turtle geometry is a piece of computational geometry. It is about the generation of line figures by programs which direct the motion of an abstract or real entity called a turtle. Basic turtle commands are FORWARD, which causes the turtle to move in a certain direction known as its HEADING; and RIGHT, which changes the turtle's heading by causing a clockwise rotation without change of position. Thus the STATE of a turtle is a heading and a position; the command FORWARD changes the position component, the command RIGHT changes the heading component. To indicate how much change is produced we write these commands as operators with a numerical input, measured (say) in millimeters for FORWARD and degrees for RIGHT. Thus the following program generates a square:

```
1  FORWARD 100
2  RIGHT 90
3  GO TO LINE 1
```

What input should RIGHT have in order to generate an equilateral triangle? Think before reading on! There is a fifty percent chance that you thought "RIGHT 60" before you corrected it to RIGHT 120". In any case, you used the theorem that the internal angles of a triangle add up to 180 degrees.

My theorem reflects a different way to think about this problem. It presupposes that we are familiar with the ideas of STATE and TURTLE-TRIP. So imagine a turtle taking a trip around a triangle, ending up in the same state as at the start. How much did it turn? Obviously 360 degrees! And it turned in three goes, so it must have turned 360/3 each time.

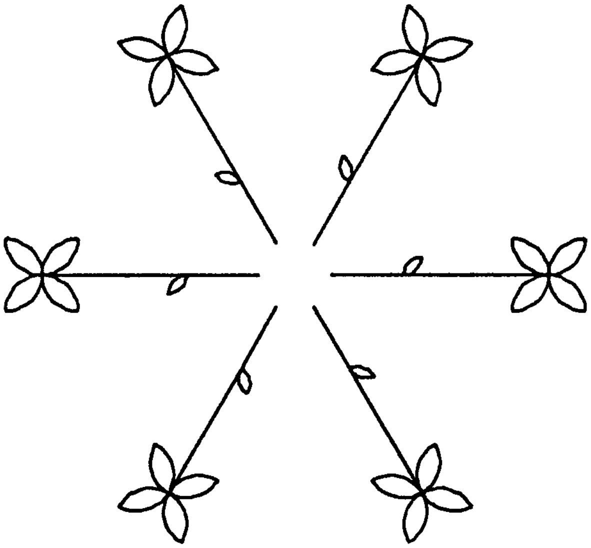
One could state this as: the sum of the

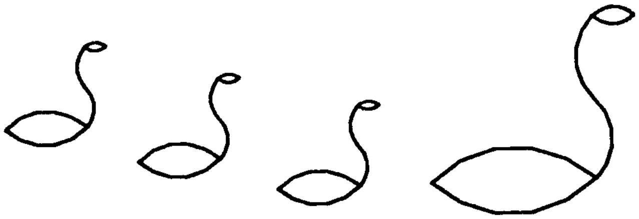
external angles of a triangee is 360 degrees. But there is nothing special about triangles: the sum of the external angles of any simple polygon is 360 degrees. There isn't even anything special about polygons. So we have the theorem in some such form as: if a turtle makes a simple round trip, its total rotation is 360 degrees.

This kind of geometric thinking has been extensively developed and tested at an appropriate level for elementary schools. (Variants have been used at pre-school and at college levels, but talking about that would take me too far afield.) The children use the turtle geometry to write programs for CRT displays (think of the turtle as a cursor generating the figure) and to drive "physical turtles". Let's look at some simple situations in which the theorem is used to overcome a geometric difficulty.

Suppose you know how to make a turtle draw a piece of arc, and now want to combine two pieces of 60 degree arc to make a petal or a swan's body. How much should the turtle turn between the arcs? Well, it must turn 360 degrees all the way around, and it turns 60 degrees + 60 degrees = 120 degrees while on the curving part of its trip. This leaves 240 degrees; so 240/2 = 120 degrees at each end. To make a "fatter" swan we could use 90-degree arcs,...and I leave it to the reader to figure out (or measure) the end angle.

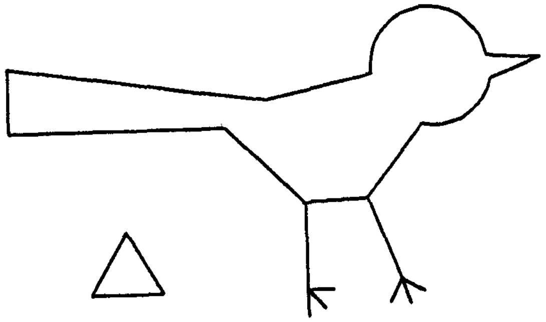
HEXAFLOWER 100



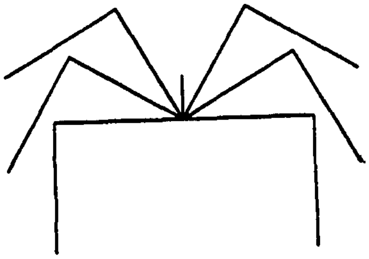


Similarly a shield (or curved triangle) made up of (say) 30-degree arcs can be seen to require a 90-degree turn at the vertices. And a little imagination will show how powerful the principle is in practical problems of generating interesting graphics:

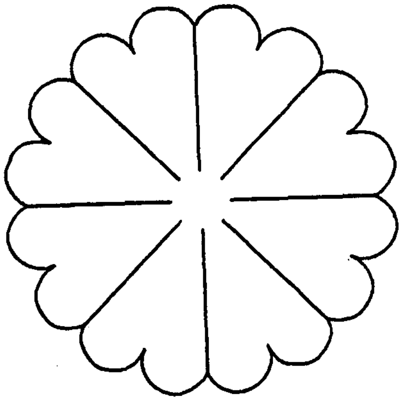
BIRD TIRD



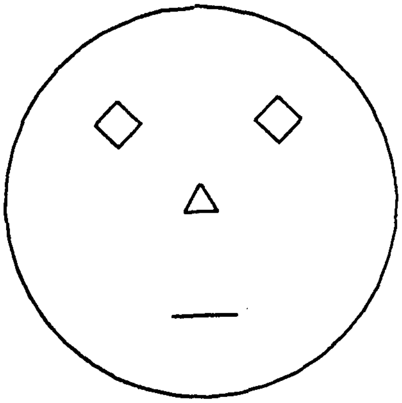
SPIDER



HEART PIE



POKER FACE

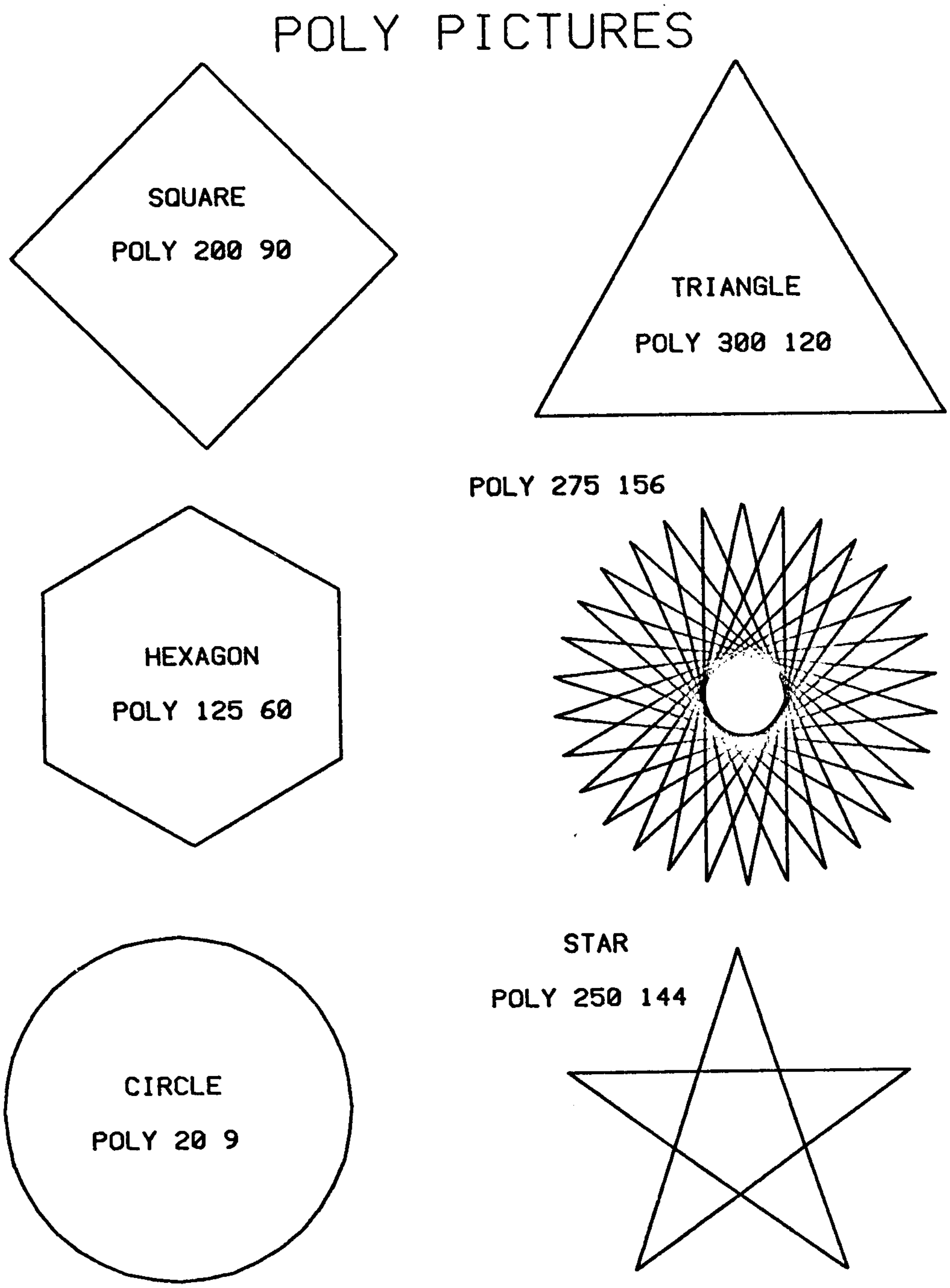


On a more abstract mathematical level, the day comes when the child gets involved with figures that cross themselves and so need an extension of the theorem. For example, consider the following turtle procedure written recursively in LOGO:

```
     TO POLY :ANGLE :STEP
     1.  FORWARD :STEP
     2.  RIGHT :ANGLE
     3.  POLY :ANGLE :STEP

     END
```

The figures show the action of this procedure with various inputs. It will be seen that in general the figure crosses itself.



Now suppose that we want to introduce a stop rule, so the turtle will eventually halt instead of running forever around the same orbit. An appropriate form of our theorem is:

> When a turtle takes a round trip, it turns a whole number of 360's.

I leave the reader to translate the idea into an effective stop-rule. I conclude by recapitulating some ways in which this theorem and its geometric setting seems better than the traditional theorems in the traditinnal Euclidean setting:

(1) The idea of angle and angular measure is messy (and difficult for many children) in a status setting. The dynamic command: "RIGHT GO" is more real and intuitively accessible than "Angle of 60 degrees".

(2) More generally the direct representation of process in the formal mathematics reduces its distance from the child's intuitive mathematics. This is shown in our theorem which really captures a geometric intuition and contrasts with the much less intuitive nature of the Euclidean proofs about the internal angles.

(3) We were able to take the theorem through several stages of purposeful, useful extensions.

(4) The kids use it.

(5) They also learn to use some very powerful general ideas of which the cluster of concepts associated with "state", "change of state", "return to initial state" is just one example.

Related aspects of these ideas are covered in the following articles, available from the A.I. Laboratory, MIT.

A.I. Memo No. 246:
A Computer Laboratory for Elementary Schools, Seymour Papert, October 1971 (LOGO Memo No. 1).

A.I. Memo No. 247:
Teaching Children Thinking, Seymour Papert, Octover 1971 (LOGO Memo No. 2).

A.I. Memo No. 248:
Twenty Things To Do With a Computer, Seymour Papert and Cynthia Solomon, June 1971 (LOGO Memo No. 3).

A.I. Memo No. 249:
Teaching Children to be Mathematicians VS. Teaching About Mathematics, Seymour Papert, July 1971 (LOGO Memo No. 4).