Quarts *(Continued from previous page)*

Polya suggests that I start off by "playing around" with the problem and hopefully, in the process, a solution will emerge. I took his advice and I did come up with a` solution though not an elegant one. Here is the solution that Polya offered. The goal is to make six quarts. Polya suggests that we think about what the containers should look like at the end and *work backwards*. In order to make container B hold 6 quarts, the previous setup (figure 2) must have container B filled and container A holding 1 quart of water. That's because if you empty container B when it is filled into container A when it is holding 1 quart, 3 quarts from B will fill up A and leave 6 quarts in container B. Working *forwards*, we can make one quart by filling container A twice from container B (See steps 1 to 4, fig. 2). So with 3 more steps (Steps 5 to 7) you will have the next to last condition. I found this to be a clever solution and not obvious at all. But I realized that with some experience, I could probably get good at solving this kind of problem. It helped to have a Logo context for solving this problem, because playing around with it was fun. I tried different container sizes and different target amounts. Patterns began to emerge. I noticed that there were conditions that were not possible to solve. For example, containers with even number sizes cannot produce an odd number of quarts. As I continued to play, questions came to mind. I wondered if it were possible to write a procedure that takes three inputs (size of containers a and b and the target amount) that would eventually solve the problem in a finite number of "moves" (exps: POUR a b,EMPTY a, FIL b, etc.)? And if so, could I also come up with a procedure that would solve the problem in a minimum number of moves. (In other words, is there a "formula" that relates container sizes, target amount, and minimum number of transactions?) I have not as yet come up with such a procedure, but I do have an example of what I have in mind.

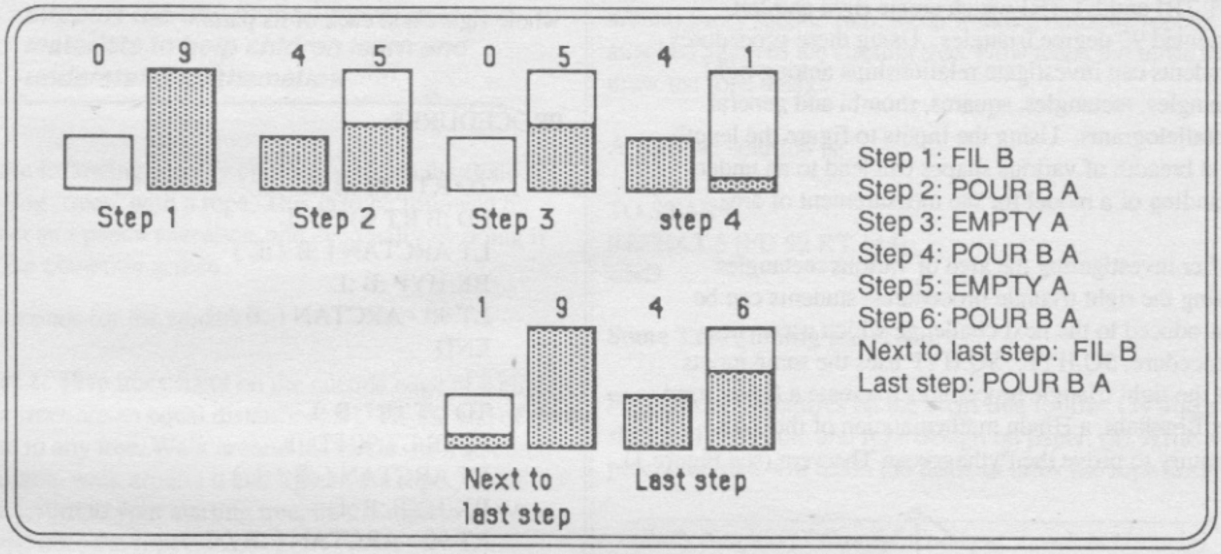

Fig. 2

If the size of container A (A) minus the size of container B (B) equals the target (T) (asumming A, B, and T are ≥ 2, A ≠ T, B ≠ T, A>B) then SOLVE will solve the problem in 2 "moves".

Example:

Container A: 7 quarts.
Container B: 4 quarts.
Target: 3 quarts.

SOLVE 7 4 3 will fill container A and then pour the liquid into container B leaving container A with 3 quarts of water.

```
TO SOLVE :A :B :T
SETUP :A :B
FIL :A
POUR :A :B
END

TO SETUP :A :B
CLEAN
CON A :A CON B :B
END
```

This example is trivial. I invite you to help me to come up with some more powerful SOLVE procedures. I look forward to hearing from you.

If you wish to have my procedures, send me a stamped, self addressed disk mailer to the CLIME address and I will put my QUARTS page on your disk and send it back to you. I also have a Terrapin and Apple Logo version If you wish these versions, send a formatted DOS 3.3 disk.. (Sorry, I don't have it on the IBM, but I can send you a printout of the procedures.)