Problem: Develop all of the processes necessary to solve the TopSpin game and write up an algorithm for solving it. Include a justification for how many of the states are reachable and, if possible, which permutations in S_{20} the represent.

Proof:

This algorithm has three distinct parts. The first is correctly positioning the first 16 'numbers' - which we will call beads from now on. Each bead can be positioned with the same sub-algorithm in this step. The second main part of the algorithm is placing the 17th bead in the correct position, and the last part of the algorithm places the last three beads. It should be noted now the naming conventions used in this algorithm. There can only be four beads manipulated by the circle in the topspin at any time, and we will label the beads in those positions from left to right as b, c, d, and e, respectively. It should be noted that the letters refer to the position of the bead and not the numbered beads themselves. The bead position to the left of the circle we will call a and will we label the rest of the positions in the 'expected' way. That is, going from the right of the circle the positions are labeled f, g, and so on so that the t position is in between the s and a positions.

There are three basic operations in the topspin game. The beads can all be shifted in a counter-clockwise direction, they can all be shifted in a clockwise direction, or the circle can be used to manipulate 4 beads. We will label the operation that shifts all the beads in the counter-clockwise direction L, the clockwise operation R and the move with the circle T. In cycle notation the operations are as follows:

$$L = (a\ t\ s\ r\ q \ldots c\ b) \tag{1}$$

$$R = (a\ b\ c\ d\ e \ldots\ s\ t) \tag{2}$$

$$T = (b\ e)(c\ d) \tag{3}$$

There is also the identity operation, and it should be easy to see that $L = R^{-1}$ and $T = T^{-1}$ The bead that is trying to be placed correctly will be called the target bead - so at the very start the target bead is the '1' bead. We will call the place we are trying to put that bead the target position. So after we have placed the '1' bead correctly, the target position for the '2' bead is directly adjacent to the '1' bead in a clockwise manner.

The Algorithm

Phase 1: Correctly position beads 1 through 16. We will assume that the beads are positioned like the numbering on a clock and not in reverse.

a. Let's say that 1 is already positioned.

b. If target bead has more than two beads between it and the target position, then move the target bead, using the L and R moves, into the e position and perform the T move.

c. Repeat step b as many times as necessary to get the target bead within three places of target position.

d. If there are two beads in between target bead and target position, then use L and R where appropriate to position the target bead in the e position and perform T.

e. If there is one bead between target bead and target position, move (from here on out move can be read as "use L and R to correctly position") the target bead into the c position and perform T. Now see step d.

f. If the target bead is directly adjacent to the target position, move the target bead to position c and perform T. Now see step e.

g. Now the target bead should be placed correctly. Repeat steps b through f until the first 16 beads are placed correctly.

Phase 2: Correctly position the 17th bead.

a. Use L and R to position the four out-of-place beads in positions b, c, d, and e.

b. If bead '17' in in position b, then it is in the right position; go on to phase 3.

c. If bead '17' is in position e, then apply T, and go to phase 3.

d. If bead '17' is in either position c or d, then either one or two applications of the following process will move the bead into the target position:

$$RT(LL)TR(TRTL)^2TLTRT = (b\ d\ c)$$

We will call this sequence of moves A from now on. The move A^{-1} can also be useful, and for convenience is as follows:

$$TLTRT(RTLT)^2LT(RR)TL = (b\ c\ d)$$

Phase 3: Correctly position the final three elements.

a. Move the beads so that the three out of place beads are in positions b, c, and d.

b. Use A as necessary to place the '18' bead into the correct position.

c. Either the last two remaining beads are both in the correct position or they need to be swapped. If they're in the correct position, then we're done.

d. If the two remaining beads need to be swapped, then unfortunately we have to move all of the beads, but this can still be done. The following sequence swaps beads c and d:

$$(TL)^{18}TA^{-1}$$

The idea for this sequence came from http://www.org2.com/jaap/puzzles/topspin.htm Although I didn't take any processes directly from this site, after a couple of days of trying to figure out the sequence that swapped two adjacent beads, I glanced at it to get a hint, and found exactly the hint I needed. The site has a great little Java-script Topspin game to play around with, by the way.

Notes: Since we can swap any two adjacent beads, we can move any bead to any position desired without disturbing the other beads by swapping the bead until it gets into the desired position. So, all permutations in S_{20} are possible (although only one out of every 20 of those is interesting because the other 19 can be generated by a series of simple R or L processes from one of the interesting permutations).