Current Results and Open Problems on Reconfiguration of Modular Robots
A few solutions to the exercises

1. Geometric model on a triangular grid

Pivoting

A pivoting edge-connected triangle may collide with a module in two different ways: by and edge-edge contact or by a vertex-vertex contact. The edge-edge case is illustrated in Figure 1.

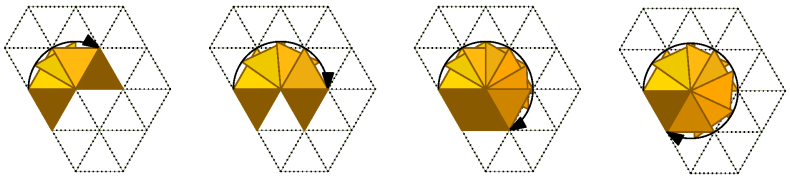

Figure 1: Edge-edge contact of pivoting triangles.

The vertex-vertex contacts give rise to an extended set of moves, by combining consecutive rotations about the vertices in contact, as shown in Figure 2. In two of the three cases, there exist (in principle) two options for the second rotation: clockwise and counterclockwise, while in the third case the second rotation can only be clockwise.

Figure 2: Vertex-vertex contact of pivoting triangles. The first rotation is indicated with a continuous arrow, the second rotation is shown with a dotted arrow.

The second rotation, though, can be obstructed, as can bee seen in Figure 3. In such cases, pivoting is impossible, as if would produce a configuration that is not edge-connected.

Figure 3: Possible obstructions to the second rotation in vertex-vertex collisions of pivoting triangles.

If no obstruction exists to the second rotation, how are the final contacts? Can they be vertex-vertex again? Yes, and the number of iterations is not bounded by a constant, as Figure 4 shows.

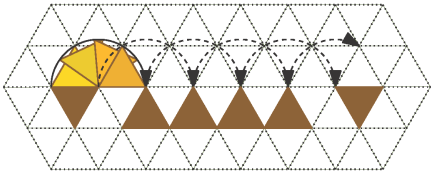

Figure 4: Example of a linear number of concatenated vertex-vertex rotations.

In this sense, triangles behave differently from squares and hexagons, since for the latter the maximum number of vertex rotations is two (as can be easily checked).

All together, then, we have obtained three sets of possible pivot moves, namely:

1. **Restricted.** A module can only pivot from edge to edge of the same neighboring module.

2. **Edge-edge.** In addition to the previous move, a module can also pivot from an edge of a neighboring module, to an edge of a different module.

3. **Vertex-vertex.** In addition to the previous moves, a module can also concatenate rotations about vertices of modules that become incident to it.

To finish with the pivoting case, let us consider circular robots pivoting in a triangular grid. Figure 5 shows the possible rotations and the empty space required to perform them without collisions. In the figure, we have marked with a cross the cells that would interfere with a pivoting triangle. Obviously, they do not interfere with a pivoting disc. Furthermore, the cell marked with a dot would have stopped a pivoting triangle, producing an edge-edge move, while it does not interfere with the rotation of a disc.

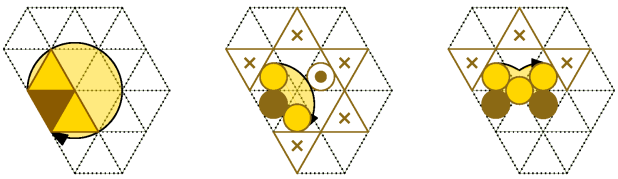

Figure 5: Pivoting circular robots in a triangular grid. See the text for an explanation of the symbols.

Moreover, looking at Figure 6 one can realize that triangular modules can be trapped and unable to pivot in positions that a circular module can easily leave.

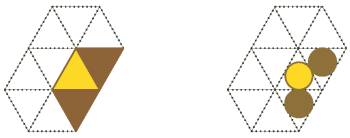

Figure 6: Left: the light-colored triangle cannot pivot. Right: the light-colored disc can easily pivot.

As a conclusion, we can say that the shape of the modules, in a triangular grid, severely condition the possibilities of the pivoting move.

Sliding

Similarly to the square and hexagonal cases, the options for the sliding move (see Figure 7) are somehow simpler than those for pivoting.

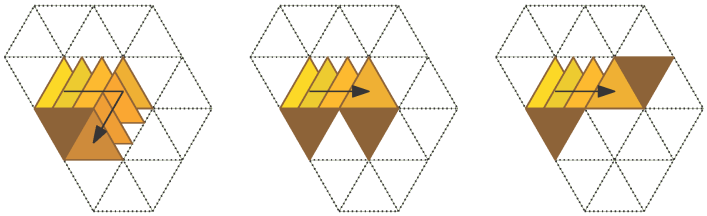

Figure 7: The sliding options.

Notice that when the modules of the robot are circular, sliding and pivoting end up being the same move (recall Figure 5). Therefore, sliding a disk requires less empty space than sliding a triangle. More precisely, the grid cell marked with a dot in Figure 5 needs to be empty in order to slide a triangle, while it can be occupied and still allow the move in the circular case.

Squeezing

The squeezing case is even simpler, as the only possible move is illustrated in Figure 8, and there is no difference between squeezing triangles and discs.

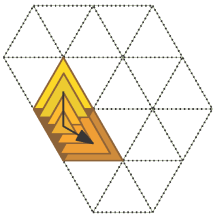

Figure 8: The only squeezing option.

2. The reconfiguration graph for edge-connected sliding triangles

The first thing to be noticed is that the sliding move has a chessboard restriction: the triangular cells of the lattice have two different orientations and each triangular robot can only slide between cells with the same orientation (see Figure 9). This makes a big difference with the square and hexagonal settings.

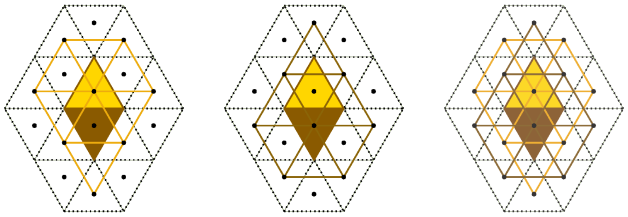

Figure 9: The chessboard restriction of sliding triangles.

If we consider the reconfiguration graph G_n whose nodes are all connected configurations with n triangles, and whose edges indicate that one slide move transforms one such

configuration into another, it is obvious that G_n is not connected and that no slide move can connect it.

Therefore, the graphs to be considered are G_{n_1,n_2}, whose nodes are connected configurations of n_1 triangles with the first orientation and n_2 triangles with the second orientation. Such graphs are also disconnected, as we will see.

In the first place, it is easy to see that they have connected components that are singletons, since some configurations –like the one shown in Figure 10– are completely rigid.

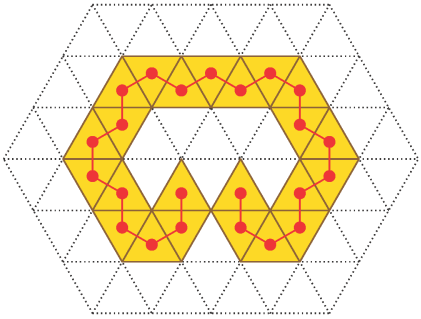

Figure 10: A rigid configuration of sliding triangles.

These graphs also have connected components of constant size. See, for example, the situation depicted in Figure 11, where a slide move is possible together with its reverse, but no other slide move can be produced without disconnecting the robot.

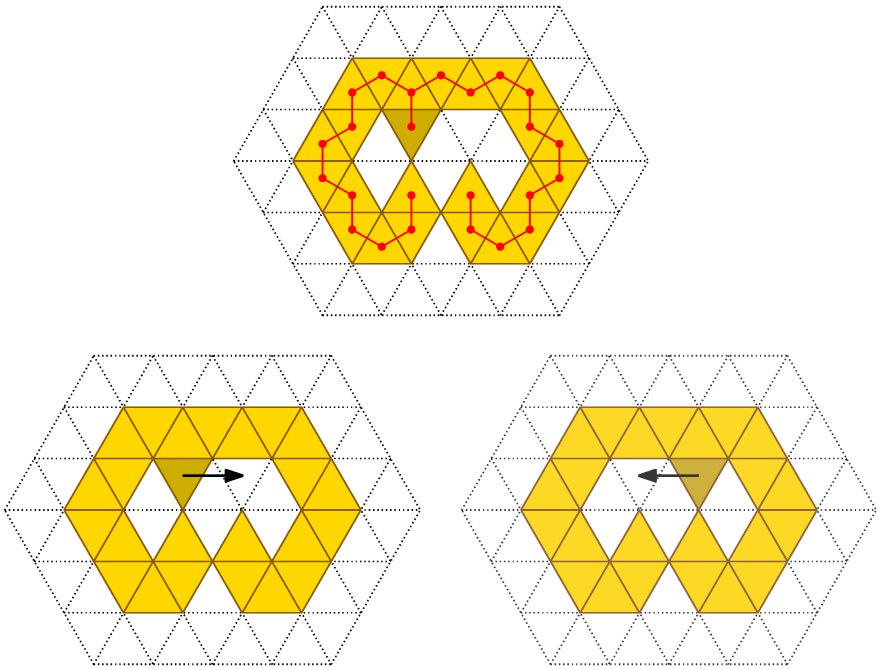

Figure 11: The dark module can slide back and forth, and these are the only slide moves possible.

Moreover, G_{n_1,n_2} has an exponential number of connected components of exponential size. By connecting several configurations like the one in Figure 11, one proves that there exists connected components of exponential size. Using a constant fraction of the modules to form a path between two of these configurations, one can prove that there exists an

exponential number of such connected components because the path can be configured in an exponential number of ways that cannot be reconfigured. Figure 12 illustrates these ideas. Observe that this last result would not be true if the modules had linear strength, i.e., if one module could carry in its slide move all the modules attached to it.

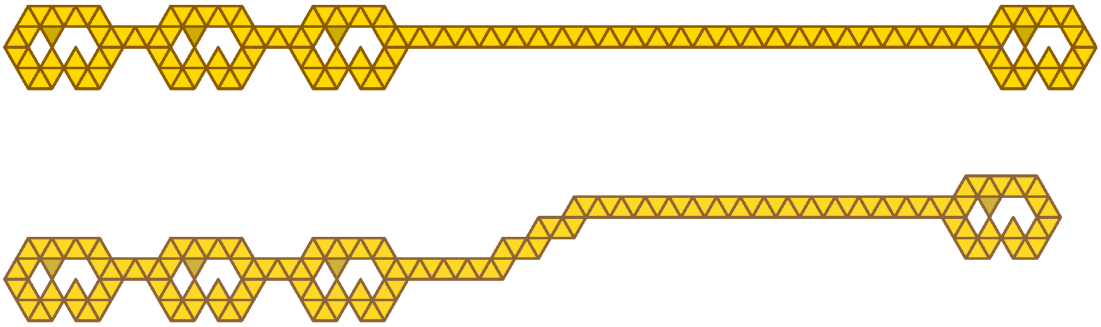

Figure 12: Above: a configuration belonging to a connected component of exponential size. Below: how to produce an exponential number of connected components similar to the above one.

Notice that most of the above observations are not true when the shape of the robot modules is circular. What happens in such case?

3. A reconfiguration algorithm for edge-connected sliding triangles

This is an open problem. As an inspiration to solve it, you can find the details on the helpers strategy for pivoting squares in [2].

4. The reconfiguration graph for edge-connected pivoting triangles

This questions should be easy to solve following the ides from the solution to Question 2.

5. A reconfiguration algorithm for edge-connected pivoting triangles

Question a) is partially answered in [4]. But the answer is not complete, since it shows necessary conditions, but it is unclear whether they are sufficient.

6. Sliding squares

This is an open problem. The current algorithm can be found in [5]. It reconfigures using an intermediate shape, namely a strip. This means that the reconfiguration can be carried out within the **(almost) disjoint** union of I' and G', if we make coincide the topmost of the rightmost modules of I with the bottommost of the leftmost modules of G. The question is, then, to be able to reconfigure also when I and G overlap, without getting stuck at deadlocks.

7. Sliding cubes

This is an open problem. The $O(n^3)$-moves algorithm can be found in [1].

8. Sliding modules

This is an open problem.

9. Pivoting cubes

This is an open problem. The known algorithm for reconfiguring a class of admissible shapes can be found in [6].

10. Squeezing modular robots

This is an open problem. The algorithm it refers to can be found in [3].

References

[1] Z. Abel, S. D. Kominers, Universal Reconfiguration of (Hyper-)cubic Robots, arXiv:0802.3414v3 [cs.CG], 2011. Available at https://arxiv.org/abs/0802.3414

[2] H. Akitaya, E. Arkin, M.Damian, E. Demaine, V. Dujmovic, R. Flatland, M. Korman, B. Palop, I. Parada, A. Renssen, V. Sacristán, Universal Reconfiguration of Facet-Connected Modular Robots by Pivots: The O(1) Musketeers, *27th Annual European Symposium on Algorithms (ESA)*, pp: 3:1–3:14, 2019. Available at https://drops.dagstuhl.de/opus/volltexte/2019/11124/

[3] G. Aloupis, S. Collette, M. Damian, E. D. Demaine, R. Flatland, S. Langerman, J. O'Rourke, V. Pinciu, S. Ramaswami, V. Sacristán, S. Wuhrer, Efficient Constant-Velocity Reconfiguration of Crystalline Robots, *Robotica*, 29(1):59-71, 2011. DOI: 10.1017/S026357471000072X

[4] Nadia M. Benbernou, *Geometric algorithms for reconfigurable structures*, Thesis (Ph. D.)–Massachusetts Institute of Technology, Dept. of Mathematics, 2011. Availabe at http://hdl.handle.net/1721.1/68480

[5] A. Dumitrescu, J. Pach, Pushing squares around, *Graphs and Combinatorics*, 22:37-50, 2006. DOI: 10.1007/s00373-005-0640-1

[6] C. Sung, J. Bern, J. Romanishin, D. Rus, Reconfiguration Planning for Pivoting Cube Modular Robots, *Proc. IEEE Int. Conf. Rob. Autom. (ICRA)*, pp. 1933-1940, 2015. DOI: 10.1109/ICRA.2015.7139451