

5. Finite state machine

We used a finite state machine to represent the behavior of the copters. This defines the different state of the copters. It is important for the later swarm behavior that although everyone uses the same definition of machine, but the machine runs independently on each copter. All possible states and the corresponding transitions were modeled.

As a result, three states were defined:

- State 0: no copter found
- State 1: one copter found
- State 2: two copters found

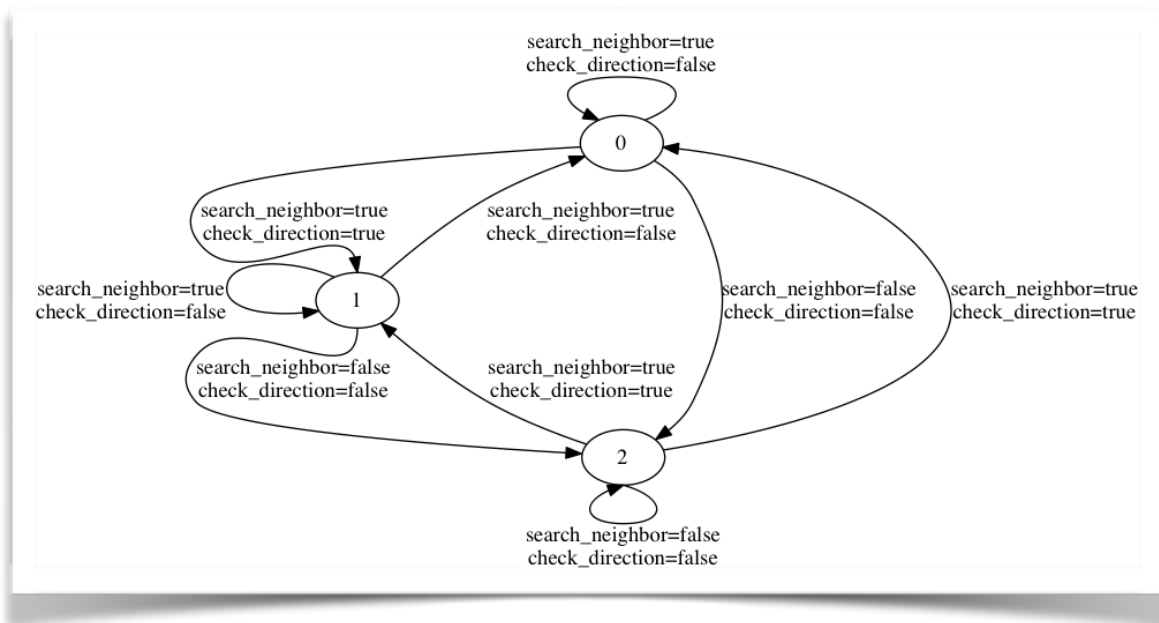
Due to the given environment, each copter can only find two other copter. At each state two variables are considered:

- `search_neighbor`
- `check_direction`

The variable `search_neighbor` indicates whether the copter continue to search for neighbors (`search_neighbor` equal to "true"), or whether to keep the current position (`search_neighbor` equal to "false").

Furthermore, for a joint search with no communication the flight direction must be determined. If the variable `check_direction` equal to "true", then the current position will be determined and adjusted the flight direction. The boarder is the center of the search space. Is the copter above the limit, it flies in the next step down. If it is below the limit, so he flies upwards. This ensures that two copters who have found themselves, will fly in the same direction.

With this assumption, there are thus nine state transitions. The following graph shows the finite state machine with its three states and nine state transitions.



The finite state machine is initially designed only for the default environment and the described behavior. In theory the swarm can be extended to any number of copter, but so far only the behavior of four copters has been tested.

The variables `search_neighbor` and `check_direction` are both independent in the number of individuals, as well as the dimensions of the search space. But the implementation has to be adjusted. Since the Copter in our case have fixed positions, we prevent interference by selecting explicitly the lateral distance sensors in the implementation. In an extension or modification of the test environment, therefore the selection of the sensors must be adjusted.