Imagine that you are in a labirynth of caves connected with tunels. Each cave has two outgoing, one-way tunnels. Cross sections of one of them is always circular, and the second one is always quadratic. As there is deep darkness in the labirynth, you cannot tell which cave you are currently in. In each cave there is a hatch, but only one of them leads to the surface. If you open any other, a poisoning gas will be released and you will be instantly killed. You have a map of the caves and tunnels and with marked surface hatch. Unfortunately you do not know which cave you are currently in.

Are you able to get out from the cave labirynth and save your life? It turnes out, that in most cases the answer is yes! (It depends on some structural properties of the map). In such case there is a sequence of moves that consists of two types of symbols (squares and circles, denoting types of tunels) with the following property. If you follow the tunnels corresponding to the symbols in this sequence, you will find yourself in a cave with the surface hatch, no matter which cave you were at the beginning.

Sequences with this property are called synchronizing sequences. The problem described above may seem to be of theoreticla interest only, but in fact it has many real applications. For example, think about electronic circuits testing. You stimulate the circuit by some sequence of input signals and observe its behavior. In such process often there is a need to bring the circuit to a given, particular state. As the circuit under test may be faulty, we do not always have a full control over the current state of a circuit. Using a synchronizing sequence we are able to bring the circuit to one given state regardless of the initial state.

In our project we will examine such sequences. We want to design and implement effective algorithms for finding possibly short synchronizing sequences in different variants of synchronizing problems (including the ones related to the so-called Road Coloring Problem). We also want to use the synchronizing theory in other areas, for example in General Game Playing.

In game theory, when dealing with so-called games with incomplete information, each player has only a partial knowledge about the game environment. We may express this situation in terms of an algorithm exploring the game automaton: a player does not know what is his current state. The number of such possible states can be limited to some subset called information set or belief state. Making the player situation less ambiguous (by reducing the number of possible states) allows us to design more effective strategy for further play. This
problem may be interpreted as a task of synchronizing the game automaton, where input symbols correspond to possible moves of a given player.

As many synchronizing problems are computationally hard, we are not always able to design fast algorithms that give us the optimal solution. That is why we also plan to utilize different methods and tools from the field of machine learning and artificial intelligence (for example, genetic algorithms). Such approach will allow us to design algorithms working in reasonable (short) time and returning reasonable (maybe not optimal, but acceptable) results.