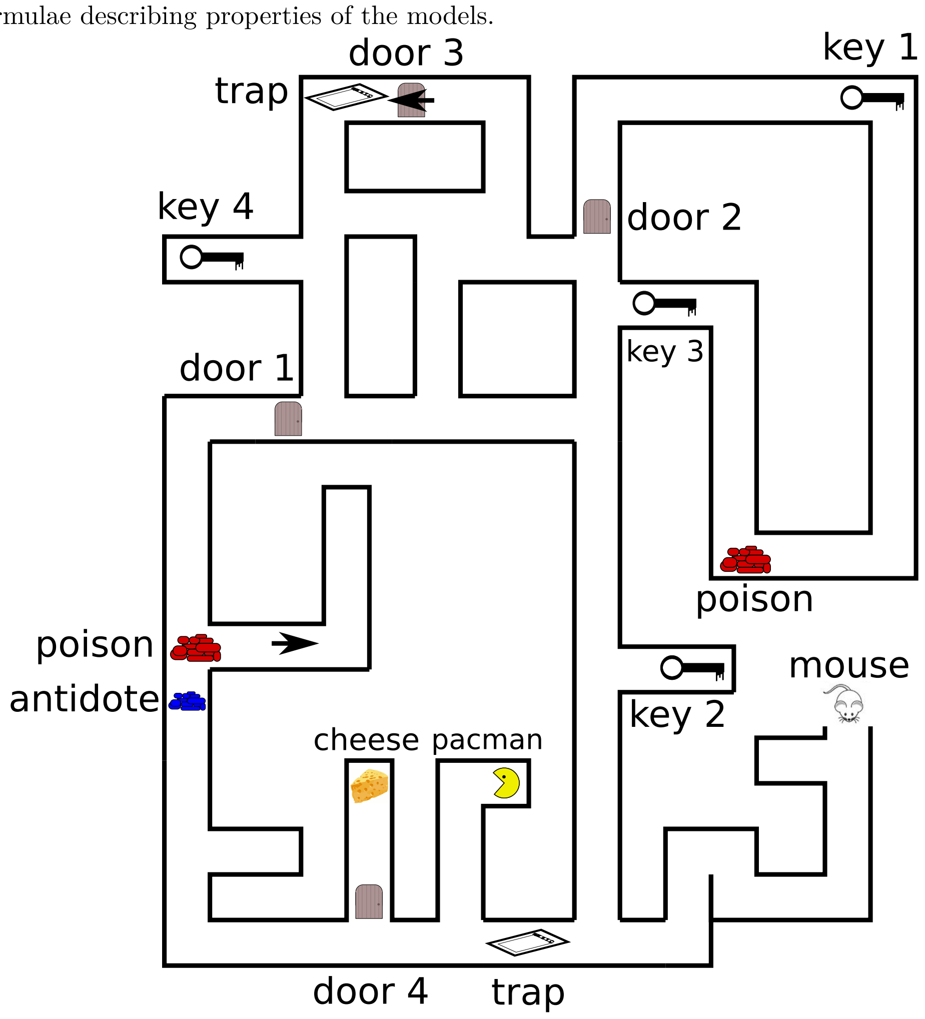
Your task is to create two models of the following maze and to create some formulae describing properties of the models.



In the first model, there is a mouse which wants to get the cheese. Every time the mouse must do at least one step (it can not stand) in any ward (respecting the walls).

If the mouse steps into a trap or if it meets pacman, it looses (the pacman stands still on its place). If it steps on poison, then it must step on the antidote immediately (the very next step). The mouse can not walk through (step on) any doors unless it found a corresponding key before (door 1 must be opened with the key 1, etc.). If there's an arrow in the maze, it means that the mouse must make the next step in that direction once it gets on the arrow.

Create a model of this maze in the *nuXmv*. Formulate the following properties as CTL or LTL formulae and verify them using *nuXmv*. Keep on mind all the restrictions about obstacles on the path, like that the door must be unlocked with a key, etc.).

1. It may happen that mouse won't run into any item (key, door, posion, cheese, antidote, trap)
2. Whenever the mouse eats a poison, it will survive.
3. Whenever the mouse eats a poison, it can survive.
4. It can happen that the mouse eats a poison and survives.
5. Whenever the mouse walks through door 3, it will be trapped.
6. Whenever the mouse walks through door 4, then it has a free path (without any obstacles like door, trap or poison) to the cheese.
7. There exists a path on which the mouse gets the cheese (the path must fulfill all the restriction on doors and keys, poisons and traps, etc.).

The formulae must be generic, which means that if the placement of atomic proposition changes, the formulae still may be used to decide whether the given property holds.

The second model is the same but pacman can move. It moves the same way as the mouse, with these exceptions: it can not change ward until it runs into a wall (that is, it goes straight until a wall, then it can change the ward and again goes straight until a wall, ...). If pacman eats a poison, nothing happens to him, the poison just disappears (the same with the antidote). Pacman can eat also the mouse, keys and cheese. However, for doors and traps the rules are the same as for the mouse – trap kills pacman and for walking through doors a key is needed. Create CTL or LTL formulae which you use to decide whether the following properties hold:

1. There exists a path on which the mouse gets the cheese.
2. There is a possibility that pacman eats the mouse.
3. Pacman will always get trapped.
4. Pacman will never eat the cheese.
5. Whenever pacman eats a poison, then the poison will never be eaten by the mouse later.

Hint: Instead of checking whether the model satisfies a formula, you may also create a formula that makes the model checker generate a counter-example witnessing that there exist a such path – that is, you can create also formula that is *false* if and only if the given property holds. In this case, explicitly mention that these formulae are counter-generating.