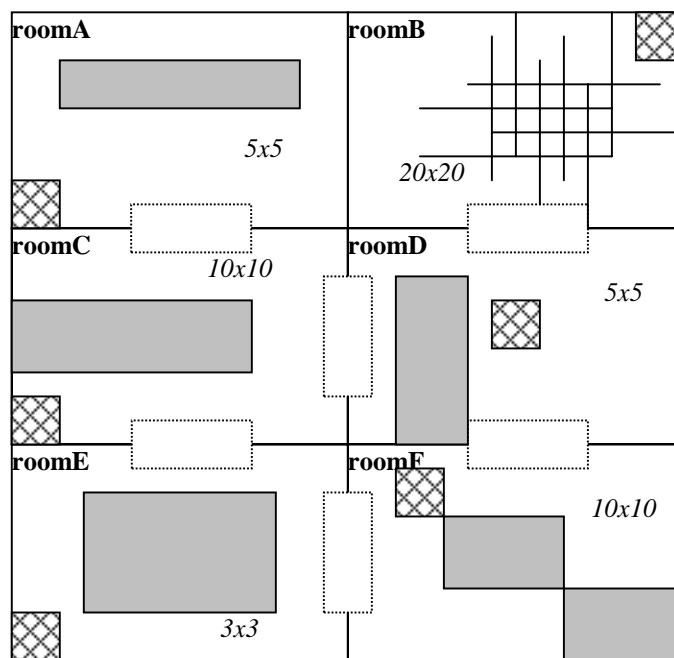


SCENARIO:

Consider the world of rooms in the figure below:

- There are six rooms
- Rooms are connected by doors
- Each room may have obstacles
- There is one agent in each room.
- An agent can only move in the room it is located.



The problem is to transfer an object from the location in one room into another location in another room. For example if the object is in room F and it should be transferred to room A the following routes should be follow:

- Agent F should move to the position of the object.
 - Agent F should move to the door that connects rooms F and E (to deliver object).
 - Agent E should move to the door that connects rooms E and F (to pick up object).
 - Agent E should move to the door that connects rooms E and C (to deliver object).
- and so on, until
- Agent A should move to the position that the object should be put.

The Prolog programs provided are (more comments are given in the code):

- `lab1.pl`: a representation of the world and one possible coordinated solution
- `search.pl`: various search algorithms.

The coordinated solution is as follows:

Given an initial position and a final position of the object, another agent (supervisor) forms a set of goals, which express problems to be solved by individual agents. This is done by applying a search algorithm on the connectivity graph of the rooms. Each agent uses a search algorithm to find its way around the room without falling into obstacles. This is done by applying a search algorithm in the connectivity graph of a grid, which is used to partition the room into equal squares (coordinates). The grid is of varying granularity, that is, every room is split into a grid of different size (see figure).

PREPARATION:

Read and understand the Prolog code so far. If you need to trace it, then try some of the following queries and try to figure out their meaning:

```
?- godfs(supervisor, roomB, roomA, Path).  
?- gobfs(supervisor, roomB, roomA, Path).  
?- godfs(agentA, (roomA,1,1), (roomA,5,5), Path).  
?- goastar(agentA, (roomA,1,1), (roomA,5,5), Path).  
?- solve((roomA,5,5), (roomF,10,5)).
```

PRACTICAL WORK:

- ☐ The Iterative Deepening algorithm is incomplete. Write the rest of the Prolog code to make the iterative deepening algorithm work.
- ☐ The Best First algorithm is not implemented. What changes would you do in A* code to obtain it?
- ☐ At the moment, all agents are able to move horizontally and vertically. Adjust the code so that agent A and F can move only diagonally, while the rest as they do now. Could agent D move diagonally as well?
- ☐ Adjust the Prolog code so that each agent knows, and therefore performs, a different search to find paths. To which agents you would assign which search algorithms and why?
- ☐ It is stated that A* is “naively” implemented. Why? How could it be more efficient?

QUESTIONS:

In a “real” implementation of this system, there might be a thread of computation for each subproblem.

- ☐ Could any Prolog system do that? If so, explain how. If not, why not?
- ☐ Since subproblems might be solved in any order, how would you arrange the sequence in the overall solution?

This prototype implies communication between agents which, however, it is implicitly implemented (one program, all agents have direct access, etc.). Think of ways, which could aid in moving a step towards a real implementation.