Intelligent Systems
Assignment 1 (2025)

Exercise 1 "Scotland Yard" (up to 14P)

In the first exercise, we are modeling a (simple) variant of the board game "Scotland Yard". In this variant of the game there is a "Person X" (shortened to X from now on) and two detectives "Jane" and "Hercule". Intuitively, the goal of the detectives is to catch X, while X attempts to escape capture. Every character can move through specified directions, while X is restricted in their movement. If X is surrounded by detectives, X is caught.

The Board We will use a board defined below and a single means of transportation. There are nine locations, and the locations can be changed according to the edges in Figure 1. Going from a location to another location uses a Taxi.

Starting Positions Jane starts at location 6, Hercule at 8, and X at 1.

Rules We consider two rounds. Each round a character must move one step (uses one Taxi). The detectives are free to move (according to the directions given), while X must not go to a location where a detective was or will be in a given round. For instance, if Jane is at location 2 and X at location 1, then Jane may move to 1 or 6, but X must not move to 2. X may go only to 3 or 4. If, additionally, Hercule is at 7 and would go to 3, X can only go to 4 (X can not go to 3 since Hercule goes there). If X at some point is surrounded by detectives, X is caught. For instance, if X is at 5 and Jane at 4, then X is caught. X may move to a location where a detective was in a previous round.

Tasks We will model the scenario and some questions in first-order logic (FoL) and let Vampire solve these tasks. For modeling, we refer here to modeling in TPTP format. Make sure that all your encodings are accepted by Vampire. That is, the conjectures must be provable or not in Vampire, depending on the task.

Consider the following three tasks.

1. Model the board and additional formulas that you need for the subsequent tasks in FoL. (1P)

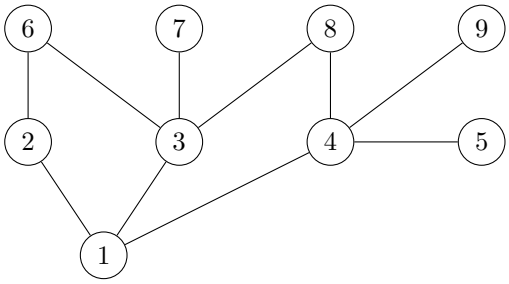

Figure 1: The board for the simplified Scotland Yard game.

2. Write one conjecture formula that shows that there is a way for the two detectives to move two steps each (ignore X here). (3P)

3. Write another conjecture showing that there is a way for the two detectives to move two steps each such that regardless how X moves (i) X is caught (surrounded by the detectives after two moves) or (ii) X cannot make two legal moves. (4P)

4. You get an additional point if, for the two tasks preceding this one (i.e., 1.2 and 1.3), you express starting positions of characters with a specific binary predicate *start* in a single (not conjecture) formula (that is, not as "hard coded" starting positions in the conjecture). (1P)

5. Expand the board, by adding a single location, such that the two detectives cannot capture X in two moves. The conjecture of 3. must not be provable anymore. (2P)

6. Write a conjecture that there are at least two locations that are not adjacent to each other. (1P)

7. Write a conjecture that X can reach every location in two steps. Ignore the detectives in this sub task. You can choose to prove/disprove whether X can reach every location with (a) exactly two steps or (b) at most two steps. Write down what you chose. (2P)

Remark Note that it is not necessary to represent time explicitly. Moreover, it is not allowed to give specific moves of characters before-hand (for instance, do not specify in TPTP that Jane has to move from 6 to 2).

Exercise 2 "Cat World" (up to 15P)

We model here a simple scenario where we have to find (narrow down) the location of a cat. Imagine a room ($n \times n$ dimension of squares) and you find

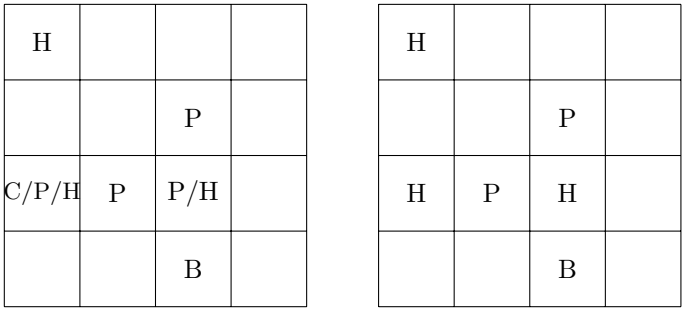

Figure 2: A room with a broken pot (B), four pawprints (P), three hiding spots (H), and the cat (C). The pawprints go "first" north, then back south, then west twice. Actual locations on the left and input on the right (hiding spots obfuscate cats and pawprints). The cat can be in either of the hiding spots on the bottom, but not in the top left corner (not reachable).

a broken pot (in one square). You suspect the cat tipped the pot over, since you also see pawprints leaving from the broken pot. However, in your room, there are many hiding spots (e.g., furniture) and the pawprints sometimes are obfuscated (hidden).

The Room consists of $n \times n$ squares. There is a broken pot in one square. There are hiding spots across the room (at least one and can be several squares). There are pawprints from the broken pot to the location where the cat actually is. The cat is in a hiding spot and pawprints in hiding spots are not visible. The pawprints (hidden or not) must be reachable from the broken pot (except for a subtask below). The cat may have gone back and forth (as cats sometimes do). See Figure 2 for an example. For adjacency, cells are connected horizontally, vertically, and diagonally.

Tasks We will model this scenario and some questions in answer set programming (ASP) and let an ASP solver (e.g., clingo) solve these tasks. For modeling, we refer here to modeling in ASP format. Make sure that all your encodings are accepted by an ASP solver (preferably clingo).

Consider the following three tasks.

1. Model the scenario above as the "basic" scenario such that brave reasoning in clingo narrows down to all positions the cat can be. The dimension n, the locations of the broken pot, pawprints, and the hiding spots are given as input (ASP facts). Write an ASP program that has as its models the possible scenarios where the cat can be. Give three example inputs. E.g., calling clingo with cat-basic.lp and cat-basic-example1.lp with the brave reasoning mode should result in all positions the cat can be. (3P)

H		P/H	C/P/H
P/H			
B			H

H		H	H
H			
B			H

Figure 3: Another room with a broken pot (B), three pawprints (P), five hiding spots (H), and the cat (C). The pawprints go "first" north, then jump north-east and go then to the east. Actual locations again on the left and input on the right. It is also possible that the cat is in the top left corner or in the hiding spot by the broken pot (just north of the pot), but the cat cannot be in the bottom right corner (not reachable via a jump).

2. Expand the scenario with the following. For each modify the reasoning where the cat can be accordingly.

   (a) You hear a noise from a single square. This square is adjacent to the cat. That is, you additionally get as input a square with noise. Give an example input. (2P)

   (b) The cat can jump exactly once. That is, the cat can skip a square and there is no paw print (the cat cannot jump back). That is, the path of paw prints can be interrupted once. Give an example input. See Figure 3 for an example. (5P)

3. Write a generator of instances using ASP for the basic scenario (2P). You get overall (3P) if you also incorporate the subtasks. Here an answer set of the generator corresponds to one possible input (with a specified dimension n).

4. Consider the properties of environments from the lecture. Argue for each what is appropriate for the Cat World? (2P)

   - fully or partially observable?
   - deterministic or stochastic?
   - episodic or sequential?
   - static or dynamic?
   - discrete or continuous?
   - single agent or multi-agent?

Exercise 3 (up to 2P)

We consider two questions for non-monotonic reasoning.

1. Give an example that shows that ASP is non-monotonic. That is, give a program P and an expansion P' s.t. inference is non-monotonic. (1P)
2. Argue whether you did (did not) use predicate completion in Exercise 1. What would change if you modify your encoding? (1P)

Submission Files

Submit your solutions in TeachCenter (deadline see slides and TeachCenter). Submit the following files. For each code file, add (in comments) a description of what is in the file. Name the files as stated below.

- submission.pdf: this **must** contain your name and matriculation number. Moreover, answer here all questions not associated with code.
- scotland-yard-board.p: Exercise 1.1
- scotland-yard-detectives-move.p: Exercise 1.2
- scotland-yard-catch.p: Exercise 1.3
- scotland-yard-escape.p: Exercise 1.5
- scotland-yard-unconnected.p: Exercise 1.6
- scotland-yard-x-reaches.p: Exercise 1.7
- cat-basic.lp: Exercise 2.1
- cat-basic-exampleN.lp: example N for Exercise 2.1
- cat-noise.lp: Exercise 2.2a
- cat-noise-example.lp: example for Exercise 2.2a
- cat-jump.lp: Exercise 2.2b
- cat-jump-example.lp: example for Exercise 2.2b
- cat-generator.lp: example for Exercise 2.3

Due to technical reasons, archive the .p and .lp files into one archive.