Automatic Reasoning and Learning - Homework Assignment 1: Knowledge Based Agents

Ramón Béjar

20 March 2017

Goal

The goal of this work is to develop an intelligent agent for the Barcenas World we have seen at the classroom **but with the modification presented at problem 3 from the problem set about intelligent agents**. That is, you have to develop a program that will receive as input:

1. The dimension of the world (the value of n for the $n \times n$ Barcenas World)

2. A sequence of l steps with their smell, Mariano and Cospedal readings of this form:

$$[x_1, y_1, s_1, ma_1, ca_1], [x_2, y_2, s_2, ma_2, ca_2], ..., [x_l, y_l, s_l, ma_l, ca_l]$$

   where $[x_t, y_t, s_t, ma_t, ca_t, ra_t]$ indicates that at time step t:

   (a) the agent moves to position (x, y)

   (b) the smell sensor reads the value s_t, that can be either 1 (it smells like hell in x_t, y_t) or 0 (it does not smell in x_t, y_t).

   (c) if the agent finds Mariano at (x, y), his answer to the question *Where is Barcenas ?* will be in ma_t and will be either 1 (is on my left) or 0 (is on my right). If Mariano is not found at that time t ma_t will be -1.

   (d) if the agents finds Cospedal at (x, y), her answer to the question *Is Mariano lying ?* will be in ca_t and will be either 1 (true) or 0 (false). If Cospedal is not found at that time t ca_t will be -1.

   Your program should be able to accept input sequences with any length l (although you can also ask the value of l as part of the input for your program). [1].

With that input, your program should print at the standard output (the screen) the knowledge state for possible locations of Barcenas that the agent has **after it processes each step of the agent**. This knowledge state will be presented as the $n \times n$ matrix with 1s and 0s, where 1 indicates a possible location and 0 a not possible location.

[1] You can assume that any input sequence for your program will be a **valid and consistent sequence of readings**, so it will never happen that from a sequence of readings you can derive contradictory conclusions for a same location. So, when you test your program use **only** input sequences with this property

Requirements

You have two possible ways for developing your intelligent agent:

- Using **propositional logic**. In this case, the architecture of your agent will follow the one we have seen at the classroom (check the slides about knowledge based systems with propositional logic) for intelligent agents based on propositional logic, so the main process that you have to implement is:

  1. When no inputs have been processed, the only knowledge of the agent is the original formula Γ you have created for the $n \times n$ world.
  2. Each time t that a new step $[x_t, y_t, s_t, ma_t, ca_t]$ of the input sequence is processed, your agent must:
     (a) For any location (x', y') of the world, ask whether it is not possible that Barcenas is in that location, that is, the agent checks if:
     $$\Gamma \cup \{s_t, ma_t, ca_t\} \models \neg b^{t+1}_{x',y'}$$
     holds for its current knowledge formula Γ. [2] As you know, you can perform the inference questions using a SAT solver.
     (b) Update the knowledge Γ of the agent that is true so far incorporating all the clauses corresponding to the positions that have been inferred as **not possible locations**. That is, add all the clauses of the set:
     $$\{\ (\neg b^{t-1}_{x',y'}) \mid \Gamma \cup \{s_t, ma_t, ca_t\} \models \neg b^{t+1}_{x',y'} \}$$
     So at the end of the iteration the knowledge formula Γ is updated with new information. Observe that any location (x', y') that was previously not possible for Barcenas (so $\neg b^{t-1}_{x',y'}$ was already a clause in Γ at the beginning of the iteration), will be also not possible at time step $t+1$.

- Using **first order logic**. In this case, the whole reasoning process (processing the whole sequence of agent steps and smell sensor, Rajoy, Cospedal and Rita readings) has to be performed through a logic program written in `prolog`. This logic program will have as main top goal this predicate:

  ```
  updateSequenceOfSteps( S0, SequenceOfSteps, SF )
  ```

  where:

  – `S0` denotes the initial knowledge state (the set of possible Barcenas locations before processing the sequence of steps).

[2] Obviously, ma_t or ca_t must only be included if they are different from -1, as when they are -1 they do not provide any information that can be used to derive anything new. Also, when performing the encoding to the propositional formula remember that the information provided by s_t and ma_t is linked with the actual position (x_t, y_t) where that information was obtained (check the solution to problem 1). So, strictily speaking your variables would be $s_t^{x,y}$ and $ma_t^{x,y}$. Although remember that for the variables ma_t it is enough to remember the column (y) where the answer was obtained.

- `SequenceOfSteps`. Denotes a list with the sequence of steps to process. Each step is a list $[X, Y, SmellsXY, MaXY, CaXY]$, that denotes that at position (X, Y) the smell sensor reads the value $SmellsXY$ (0 or 1), the answer to the question for Mariano is $MaXY$ and the answer to the question for Cospedal is $CaXY$.
- `SF` denotes the final knowledge state resulting after processing **all** the steps of the sequence.

For implementing this predicate, you can follow a recursive scheme that is based on using the predicate:

```
updatePosBarcenasLocs( PrevS, X, Y,  SmellsXY, NextS )
```

that you have available in the logic program `testSimpleBarcenasLocation.pl` in the virtual campus. [3]

[3] But observe that you have to modify it in order to include also the information from the readings of $MaXY$ and $CaXY$.

You have also to implement the main algorithm of your agent, that will be the one that receives the input for the agent we have mentioned at the first section (with an imperative language like java or python), and that will have to:

1. Generate the `prolog` logic program corresponding to the $n \times n$ Barcenas World.
2. Call the SWI-prolog interpreter with your logic program and the appropriate query to answer (written using the sequence of steps given as input for your program and as the initial state `S0` the state where all the locations are possible for Barcenas, except the $(1, 1)$). Your logic program should also be the one that prints the resulting knowledge state **after processing each step of the agent**. Use the built-in SWI-prolog predicate `write(Message)` to print information on the terminal.

What you Have to Deliver

You must deliver:

- **Documentation**. A document where you explain the design of your program and how to run it. You must also provide a set of test inputs and show the resulting output of your program for these test inputs. The documentation must also contain an explanation of the logical model (with CP0 or CP1 logic), that you have used to encode the inference rules of the agent. In case of the program with CP1 logic (with Prolog), this is simply to explain the relation of the Prolog rules with the original rules of the problem.
- **Code**. Give all the needed code for running your program. **Do not give a program** that needs the installation of special python or java

libraries that cannot be installed through the standard repositories of a Fedora 23 distribution (or for python, libraries that can be installed with pip). So, check that you use only standard java or python libraries. All your code must contain enough comments so that you can convince me that you really understand how your program works !