UNIVERSIDADE DA CORUÑA

facultade de informática da coruña

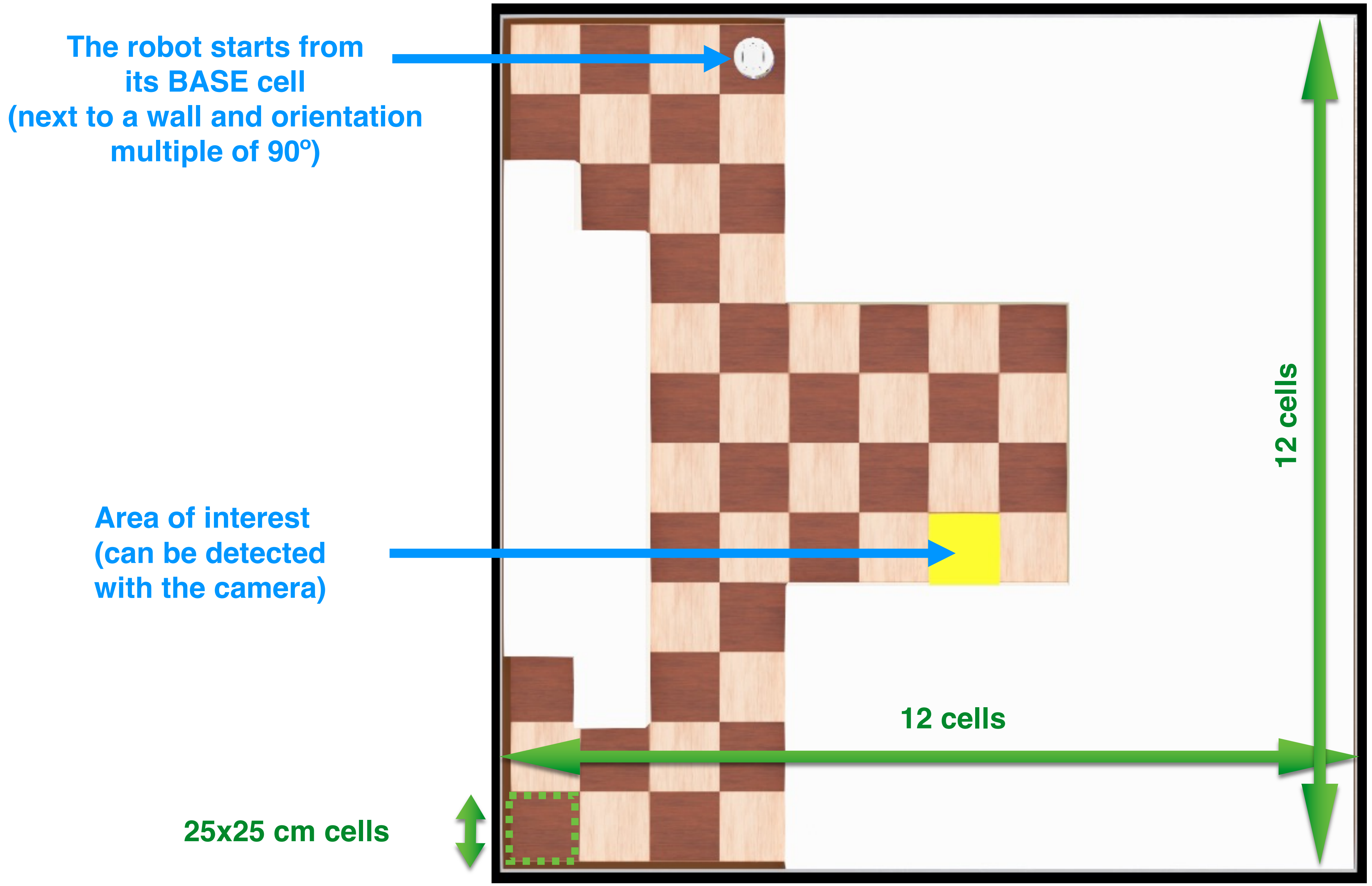
Practical work

Mapping and localization

Degree in Computer Science Education - 4th year
ROBOTICS

Statement

- This practice will consist of the implementation of a **hybrid architecture , where the robot will have reactive or deliberative** behavior depending on its internal state.



Statement

- The practice will be carried out **only in simulation and with real robot**, using the **Robobo robot**
- **Robot objective:** patrol an environment following walls with **odometry** and return to base immediately if an event of interest is detected with the camera (in this case it will be marked with a solid yellow area).
  - **The environment will be discrete** (divided into 25x25 cm cells) and of known dimensions (12x12 cells, equivalent to 3x3 m). Therefore, obstacles will never be halfway between two cells. You can use the maze worlds available in RoboboSim
  - The robot's behavior should be valid for any initial position of the base as long as it is next to a wall (the robot does not know the absolute position of its base).
- The execution must be carried out in 2 stages:
  1. An initial reactive stage of exploring the environment **by following walls** and at the same time **creating a map internal of the environment** .
  2. A second stage in which the robot takes advantage of the map of the environment it has created (along with deliberative behavior) to improve its performance:
     1. Patrol the environment following walls (you leave the base always leaving walls on the same side).
     2. If you detect an event of interest with the camera (yellow zone), **use the internal map** to immediately return to the robot station in a more efficient way (**deliberative planning of the return route**).

Statement

- **WHILE THE ROBOT DOES NOT HAVE A MAP OF THE ENVIRONMENT CREATED:** a functionality will be used that allows the map to be updated while the robot explores the environment following walls. Since there are no islands in the environment, you can create a complete map by going around completely until you return to the base.
  - The **base of the robot** will be next to a wall (the robot's starting orientation is unknown , one of north, south, east and west).
  - An area of interest (yellow) will always be next to a wall. While the map is being created, detecting a yellow zone does not interrupt the wall following behavior. Likewise, there could be no yellow zone at that moment without altering the behavior.
  - Since we have a discrete environment (divided into cells) and we need to create an occupancy map, **the movement will be carried out in discrete advances and turns using odometry** , so that it is known at all times in which cell the robot is.
  - In each cell the robot can use **infrared** to detect obstacles around it and **update the occupancy map** .
  - **Detect area of interest (yellow) with camera:** it will be considered detected only if the robot sees it from very close (approximately one or two cells away) . A basic detection can be done by the relative size of yellow pixels with respect to all pixels.
  - Infrared also detects yellow areas, so it is moved to the side as with any other wall.
- **WHEN THE ROBOT ALREADY HAS A MAP OF THE ENVIRONMENT:** a path planning behavior that uses the created map will be added so that the robot efficiently returns to the base when it detects an area of interest (yellow).
  - What is necessary to navigate to the base will be added.

Statement

MAP

- A binary occupancy grid will be defined (occupied cell / free cell).
  - The occupancy grid can be implemented as a matrix.
  - It will be updated based on infrared information, which will allow us to know which cells adjacent to the one occupied by the robot are free or occupied.
    - Tip: perform the infrared reading with the robot standing in the center of each cell (the robot has infrared around its entire body).

Statement

MAP

- It is necessary to use odometry to be able to locate the robot on the map.
  - The motor encoders are used to advance cell by cell on the map and thus be able to know the discrete position coordinates in the environment.
  - Likewise, the motor encoders can be used to make precise turns multiples of 90º, waiting for the movement to end.
    - This can be done more accurately in simulation.
    - In a real environment it would be preferable to use the gyroscope.
  - Fewer errors accumulate if discrete movements are made cell by cell (the robot advances or rotates, stops, sensors, advances or rotates again).

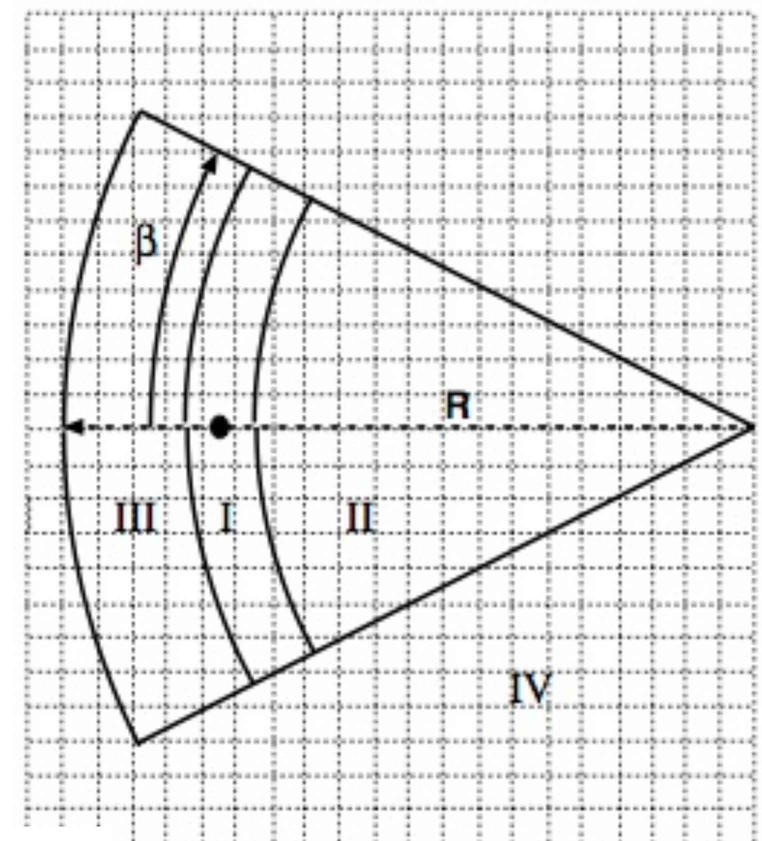
Statement

NAVIGATION

- A heuristic search method, such as A*, will be used to plan navigation back to base using the map.
- If you want to use another search method, the method must be proposed to the teacher for acceptance.

Example of creating an occupancy grid

- We start from the sonar sensor model seen in theory
- We use the probabilities that a grid[i][j] is Busy or Empty



- Region I:
$$P(Occupied) = \frac{(\frac{R-r}{R}) + (\frac{\beta-\alpha}{\beta})}{2} \times Max_{occupied}$$
$$P(Empty) = 1.0 - P(Occupied)$$
- Region II:
$$P(Occupied) = 1.0 - P(Empty)$$
$$P(Empty) = \frac{(\frac{R-r}{R}) + (\frac{\beta-\alpha}{\beta})}{2}$$

Max occupied = 0.98

Example of creating an occupancy grid

- Bayes rule:

$$P(Occupied|s) = \frac{P(s|Occupied)\boxed{\mathrm{P(Occupied)}}}{P(s|Occupied)\boxed{\mathrm{P(Occupied)}} + P(s|Empty)\boxed{\mathrm{P(Empty)}}}$$

$$\begin{aligned} P(Occupied) &= 0.5 \\ P(Empty) &= 0.5 \end{aligned}$$

- Bayes rule update:

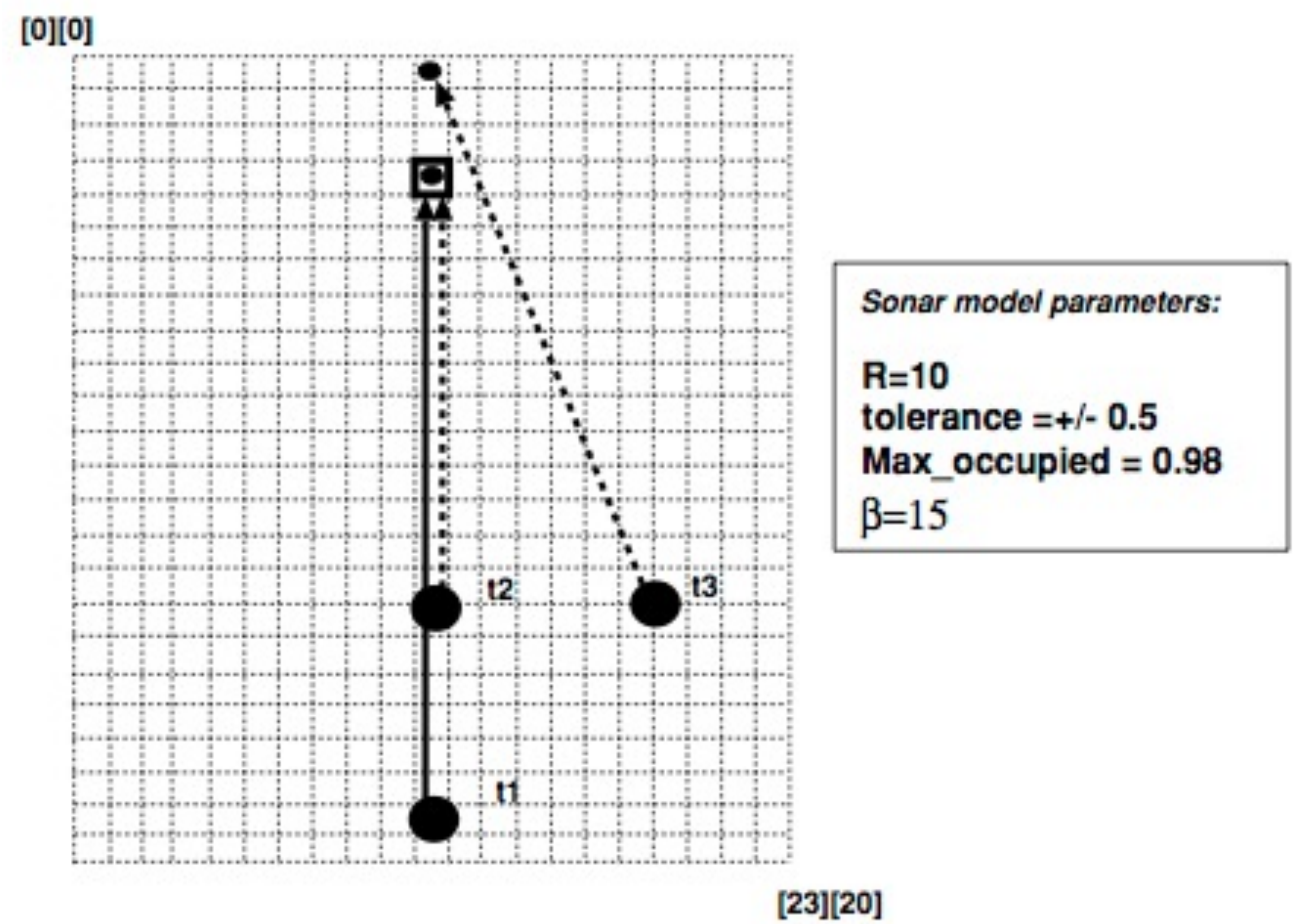
$$P(H|s_n) = \frac{P(s_n|H)P(H|s_{n-1})}{P(s_n|H)P(H|s_{n-1}) + P(s_n|\neg H)P(\neg H|s_{n-1})}$$

Example of creating an occupancy grid

- Each grid[i][j] element is a P structure with two fields, representing the probability of being busy or empty:
  - P(Busy) and P(Empty)
- It is assumed that initially each grid[i][j] has an equal probability of being busy or empty:
  - P(Busy) = P(Empty) = 0.5

Example of creating an occupancy grid

- Suppose a robot must map an unknown area
- The occupancy grid is shown in the figure:
    - 12 x 10 units
    - A 24 x 21 matrix is created



t1, t2 and t3 represent 3 sonar measurements for the grid element [3][10]

Example of creating an occupancy grid

- First measurement (t1):
  - The robot is in grid[22][10]
  - Sonar measurement for grid[3][10]=9.0
    - r=9, α=0º
  - For this sensor reading, the grid[3][10] element must comply:
    - α<=β in absolute value, so the element is within the angular field of the sensor
    - r<=s+tolerance, so the element is within the upper limit of the reading range

Example of creating an occupancy grid

- Next you need to check in which region the measurement falls:
  - Region I: *s-tolerance<=r<=s+tolerance*
    - *s=9.0, tolerance=0.5, r=9 so it falls in Region I*
  - You can calculate the probability that sensor s says that grid[3][10] is Busy if there really is something at s=9.0: *P(s|Busy)*

$$\begin{aligned} P(s|Occupied) &= \frac{(\frac{R-r}{R}) + (\frac{\beta-\alpha}{\beta})}{2} \times Max_{occupied} \\ &= \frac{(\frac{10-9}{10}) + (\frac{15-0}{15})}{2} \times 0.98 = 0.54 \\ P(s|Empty) &= 1.0 - P(s|Occupied) \\ &= 1.0 - 0.54 = 0.46 \end{aligned}$$

Example of creating an occupancy grid

- Next, the value of grid[3][10] must be updated:

$$P(H|s_n) = \frac{P(s_n|H)P(H|s_{n-1})}{P(s_n|H)P(H|s_{n-1}) + P(s_n|\neg H)P(\neg H|s_{n-1})}$$

$$\begin{aligned}
P(s_{t_1}|O) &= 0.54 \\
P(s_{t_1}|E) &= 0.46 \\
P(s_{t_0}|O) &= 0.50 \\
P(s_{t_0}|E) &= 0.50
\end{aligned}$$

$$\begin{aligned}
P(O|s_{t_1}) &= \frac{P(s_{t_1}|O)P(O|s_{t_0})}{P(s_{t_1}|O)P(O|s_{t_0}) + P(s_{t_1}|E)P(E|s_{t_0})} \\
&= \frac{(0.54)(0.50)}{(0.54)(0.50) + (0.46)(0.50)} \\
&= 0.54 \\
P(E|s_{t_1}) &= 1 - P(O|s_{t_1}) = 0.46
\end{aligned}$$

Example of creating an occupancy grid

- Second measure (t2):
  - The robot is in grid[16][10]
  - Sonar measurement for grid[3][10]=6.0
    - r=6, α=0°
  - The measurement falls within the sonar field of action and is in Region I
    - *P(s|Busy)=0.69, P(s|Empty)=0.31*

Example of creating an occupancy grid

- There is an increase in the probability of occupation:

$$
\begin{aligned}
P(O|s_{t_2}) &= \frac{P(s_{t_2}|O)P(O|s_{t_1})}{P(s_{t_2}|O)P(O|s_{t_1}) + P(s_{t_2}|E)P(E|s_{t_1})} \\
&= \frac{(0.69)(0.54)}{(0.69)(0.54) + (0.31)(0.46)} \\
&= 0.72 \\
P(E|s_{t_2}) &= 1 - P(O|s_{t_2}) = 0.28
\end{aligned}
$$

Example of creating an occupancy grid

- Third measure (t3):
  - Sonar measurement for grid[3][10]=8.5
    - r=6.7, α=5º
  - The measurement falls within the sonar range and is in Region II

$$
\begin{aligned}
P(s|Empty) &= \frac{(\frac{R-r}{R}) + (\frac{\beta-\alpha}{\beta})}{2} \\
&= \frac{(\frac{10-6.7}{10}) + (\frac{15-5}{15})}{2} = 0.50 \\
P(s|Occupied) &= 1.0 - P(s|Empty) = 1.0 - 0.50 = 0.50
\end{aligned}
$$

Example of creating an occupancy grid

- Updating the odds:

$$
\begin{aligned}
P(O|s_{t_3}) &= \frac{P(s_{t_3}|O)P(O|s_{t_0})}{P(s_{t_3}|O)P(O|s_{t_0}) + P(s_{t_1}|E)P(E|s_{t_0})} \\
&= \frac{(0.50)(0.72)}{(0.50)(0.72) + (0.50)(0.28)} \\
&= 0.72 \\
P(E|s_{t_3}) &= 1 - P(O|s_{t_3}) = 0.28
\end{aligned}
$$

Example of creating an occupancy grid

- Summarizing all the measures:

sonar certainty:	Bayesian $P(s\|O)$	Bayesian $P(s\|E)$
t_1	0.54	0.46
t_2	0.69	0.31
t_3	0.50	0.50

after update:	Bayesian $P(O\|s)$	Bayesian $P(E\|s)$
t_1	0.54	0.46
t_2	0.72	0.28
t_3	0.72	0.28