An Application of Context-Learning in a Goal-Seeking Neural Network

Thomas E. Portegys,
Illinois State University
Normal, Illinois 61790
USA
portegys@ilstu.edu

ABSTRACT
An important function of many organisms is the ability to use contextual information in order to increase the probability of achieving goals. For example, a street address has a particular meaning only in the context of the city it is in. In this paper, predisposing conditions that influence future outcomes are learned by a goal-seeking neural network called Mona. A maze problem is used as a context-learning exercise. At the beginning of the maze, an initial door choice forms a context that must be remembered until the end of the maze, where the same door must be chosen again in order to reach a goal. Mona must learn these door associations and the intervening path through the maze. Movement is accomplished by expressing responses to the environment. The goal-seeking effectiveness of the neural network in a variety of maze complexities is measured.

KEY WORDS
Connectionism, context-learning, goal-seeking, neural networks.

1. Introduction

Context learning is an important function for many organisms, especially humans. Behavior that is socially accepted at a sporting event will not be welcome in a classroom setting. My cats consider their chances of getting fed far better after I arrive home. These are examples of the significance of context: utilizing knowledge of one's overarching situation and possibly making certain prior arrangements in it can have a great deal to do with a prospective outcome.

There are many references to general context learning in the literature [1,2,3]. Various specialized approaches also exist. For example, context learning has been described as hierarchical sequence learning by Sun and Giles [4]. Researchers have also proposed context models of brain and behavior such as Howard and Kahana's Temporal Context Model (TCM) of the recency and contiguity memory effects [5], and Hasselmo and McClellands' model of the hippocampus' role in memory formation [6]. In the robotics field, Brooks and Maes trained a robot operated by a hierarchy of control contexts to walk by using environmental feedback [7]. For non-symbolic learning, mathematical methods have been developed to optimize reinforcement produced by an environmenal context function [8].

The subject of context learning narrows considerably as an application of artificial neural networks. Perhaps some of the most related work is in the field of grammar learning [9,10] and text classification [11] using recurrent and cascading neural networks. In the grammar learning studies, neural networks are trained to recognize sequences of inputs produced by a grammar, and are later tested on their predictive performance given incomplete sequences. The neural network plays a passive recognition role in these experiments. In our study, the aim is to allow the neural network to take an active part in the learning process by producing responses that affect state-transition probabilities. The predisposing conditions that affect future outcomes are environmental contexts. Learning these contexts allows the neural network to navigate its environment to reach a goal state.

The purpose of this project is to develop and test a learning mechanism suitable for a goal-seeking neural network called Mona. Although a connectionist architecture, Mona is more of a state-based planning system that a conventional pattern classifying neural network. Planners [12] are typically symbolic, not connectionistic systems, necessitating a novel learning solution for Mona.

Mona has modeled complex behavior on a number of tasks, including foraging and cooperative nest-building [13,14]. For an exhibit of the nest-building task, see www.itk.ilstu.edu/faculty/portegys/programs/NestViewer/NestViewer.html Mona features an integrated motivation mechanism designed to produce responses that yield need-reducing outcomes.

For this project, a maze problem is used as a context-learning exercise. At the beginning of the maze, an initial door choice forms a context that must be remembered until the end of the maze, where the same door must be chosen again in order to reach a goal. Mona must learn these door associations and the intervening path through the mazes.

One way that animals can be taught is by a conditioning process known as behavior shaping [15].

Mona can be taught by conditioning as well, although the response-overriding technique used in this project affords an accelerated learning option.

1.1 A Review of Mona

This section describes the existing system that will incorporate the new learning capability. Mona is based on the rationale that brains are goal-seeking neural networks. It has a simple interface with the environment, shown in Figure 1. All knowledge of the state of the environment is absorbed through senses. Responses are expressed to the environment with the goal of eliciting sensory inputs which are internally associated with the reduction of needs.

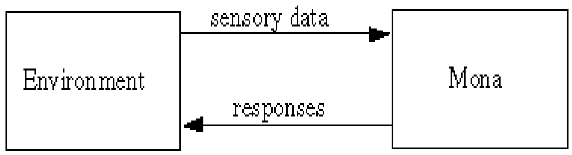

Figure 1 – Mona/Environment Interface

Events can be drawn from sensors, responses, or the states of component neurons, calling for three types of neurons. Neurons attuned to sensors are receptors, those associated with responses are motors, and those mediating other neurons are mediators. Mediators can be structured in hierarchies representing environmental contexts. A mediator neuron controls the transmission of need through and the enablement of its component neurons.

To elucidate by example, consider this somewhat whimsical task: let Mona be a mouse that has been out foraging in a house and now wishes to return back to her mouse-hole in a certain room. For the sake of keeping peace with her fellow mice, she must not make the mistake of going into a hole in another room. Figure 2 shows her neural network at this juncture.

The triangle-shaped object at the bottom is the receptor neuron that fires once she has reached her hole; the inverted triangles are motor neurons that accomplish the responses of going to the correct room (Go Room), and going into the hole (Go Hole). The ellipses are mediator neurons. Each is linked up to a *cause* and *effect* event neuron. The "Hole Ready" mediator is not *enabled*, reflecting the importance of not going into a hole in the wrong room. The "Room Ready" mediator is enabled, signifying an expectation that if its cause event fires, its effect will also fire.

The "Home!" receptor neuron has a high goal value, indicating that it is associated with a need. Because of this, *motive* influence propagates into the network, flowing into motor neurons whose firings will navigate to the goal. Since the "Hole Ready" neuron is not enabled, the motive bypasses the "Go Hole" motor neuron in search of a mediator whose firing will enable "Go Hole". Since "Hole Ready" is an effect of "Room Ready", it flows into the "Go Room" motor via the enabled "Room Ready" mediator and causes it to fire (double outline).

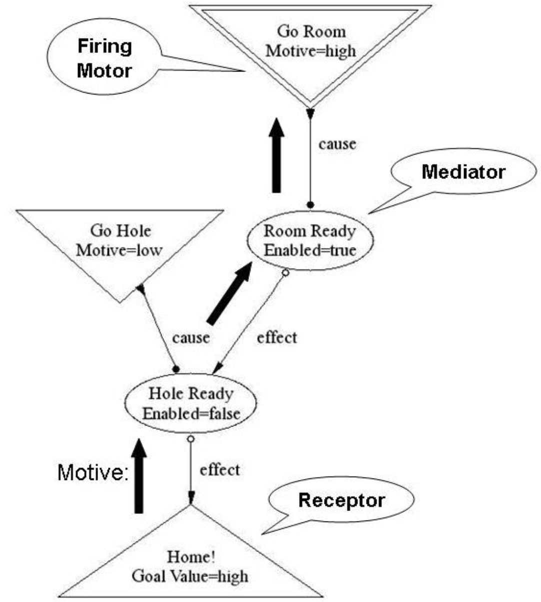

Figure 2 – Intitial Mouse Network

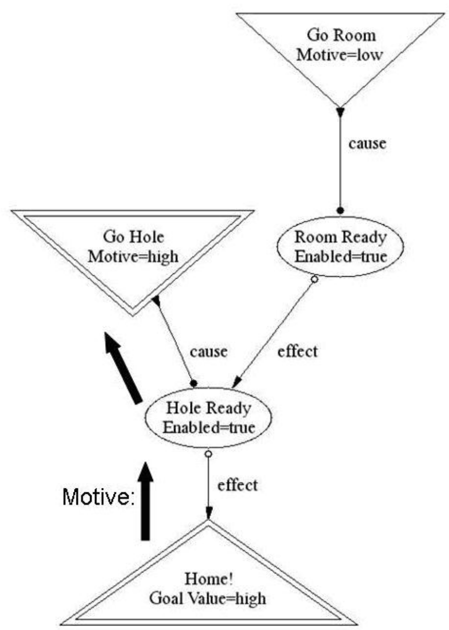

Figure 3 – Final Mouse Network

The flow of motive illustrates how mediators representing contexts work together. The appropriate context for "Hole Ready" is "Room Ready", which means that the latter should necessarily contribute something to the former in order to enable it. This something is called a *wager*. A wager temporarily modifies the enablement of a mediator that is the effect event of another mediator. It is called a wager because the base-level enablement of the wagering mediator will be evaluated based on subsequent firing of the effect neuron.

In Figure 3 the “Go Room” cause firing can be understood as a conditional probability event: given that Mona is in the correct room (“Room Ready”), she is quite certain that she can go into her own hole. This accomplished by a wager from “Room Ready”, triggered by “Go Room”, that boosts the enablement of “Hole Ready”. After this enablement occurs, motive flows into the “Go Hole” motor neuron, causing it to fire. Subsequently the Mona senses that she is home in her hole.

2. Maze Environment and Training

For this project mazes are generated that embody a context-learning problem. An example maze is shown in Figure 4.

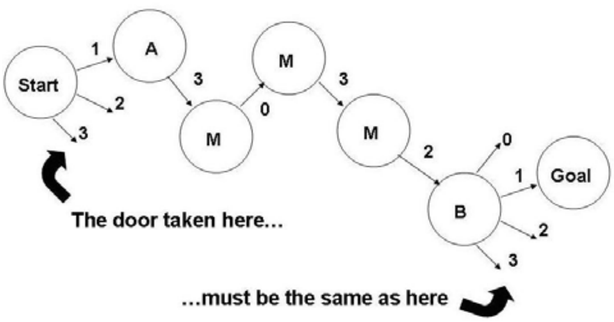

Figure 4 – An Example Maze

The Start room has a variable number of doors. Only one of these doors actually leads to room A. The middle portion of the maze consists of a randomly generated path of a variable number of rooms connected by doors. All the M rooms in the path appear to Mona exactly the same; however, only one door can be opened to reach the next room. At room B at the end of the path, the learner must choose the same door as that leading from the Start room to A in order to reach the Goal room. It can be seen that the context in this problem is to retain information about which door led from the Start room in order to repeat this choice at room B. In order to ensure that the context information varies, for each trial the door leading from the Start room is randomly determined. Since the learner has no a priori knowledge of which door this is, it is allowed several tries to determine the successful one. However, once past the Start room, any subsequent wrong door choice counts as a trial failure.

Mona was trained on each maze in two phases. For the first phase of 150 trials, its memory was allowed to retain 150 mediator neurons. Mona’s behavior in the first phase was shaped by overriding its responses with the correct ones. The door-choice correspondences between the Start room and room B were established by exposing Mona to various door configurations in a maze with no middle path (M rooms). The middle path from room A to B was taught by directing Mona along this path. A schedule of mixing end-maze and middle-maze learning proved effective. In the next training phase, Mona was allowed to run free for 100 trials to allow it to adjust the enablements of mediators to more accurately reflect maze probabilities. For example, if the Start room featured 3 possible doors, each door would be successful 33% of the time. It was also forced to cull its memory down to a final limit of 100 mediators in this phase. An example of a training run over a full maze is given in Appendix A.

3. Learning

A further refinement in the definition of a mediator is required here. A mediator consists of a cause event, an effect event, and a variable number of intermediate events representing a specific temporal sequence between cause and the effect. For mediators overseeing receptor and motor events, the cause and effect must be receptors, and the intermediate event a motor, thus embodying a stimulus-response-stimulus sequence. However, additional intermediate events are possible as long as they conform to an alternating receptor-motor pattern.

The generation of new mediators is conceptually straightforward. A history of firing neurons is retained that serves as a basis for hypothesizing new cause and effect relationship that are incarnated as new mediators. How far back in time the history is kept is a system parameter. Furthermore this can vary based on the “level” of mediators, allowing higher level mediators to associate events more distantly separated in time. For the maze problem, the history was set to allow the highest possible mediators to oversee events spanning the entire maze, thus allowing the correct door choice to be remembered.

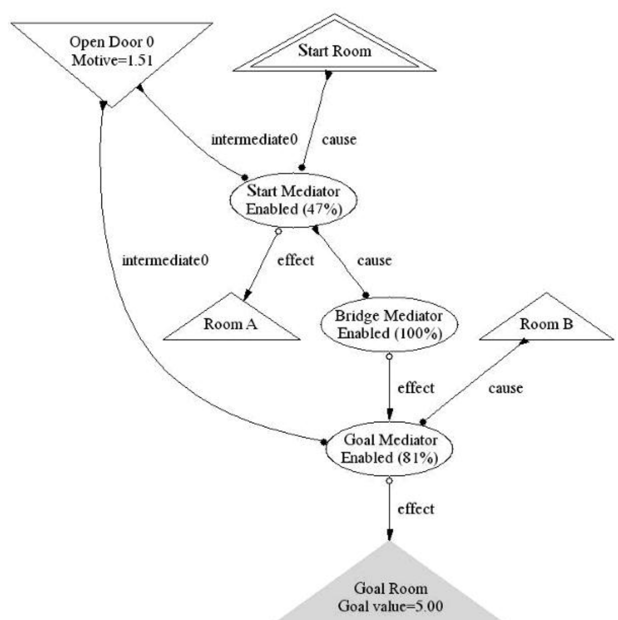

Figure 5 – “Bridge Mediator” at Start Room

Another essential learning activity besides the generation of new mediators is the evaluation of existing ones. This allows more reliable (enabled) mediators to be retained in memory. The "base" enablement of a mediator is updated as follows:

$$base\text{-}enablement = \sum wager_{successful} / \sum wagers$$

This means that the base-enablement of a mediator is roughly equivalent to the average number of successsful ones. However, since each wager can have a variable weight, this is a weighted average.

Figure 5 shows three learned mediators that oversee the association of taking door 0 at the beginning and end of the maze. This snapshot was taken while Mona was in the Start room, indicated by the firing receptor.

The "Bridge Mediator" mediates two otherwise unrelated mediators: "Start Mediator" and "Goal Mediator", each of which oversee taking Door 0 from the Start and B rooms, respectively. As can be seen, neither of these mediators are completely enabled, reflecting the maze's inherent indeterminacy. Another item of note is the motive value of 1.51 at the "Open Door 0" motor, a value propagated from the "Goal Room" receptor. Of course, at this junction the other doors that are not shown also receive equivalent motivation.

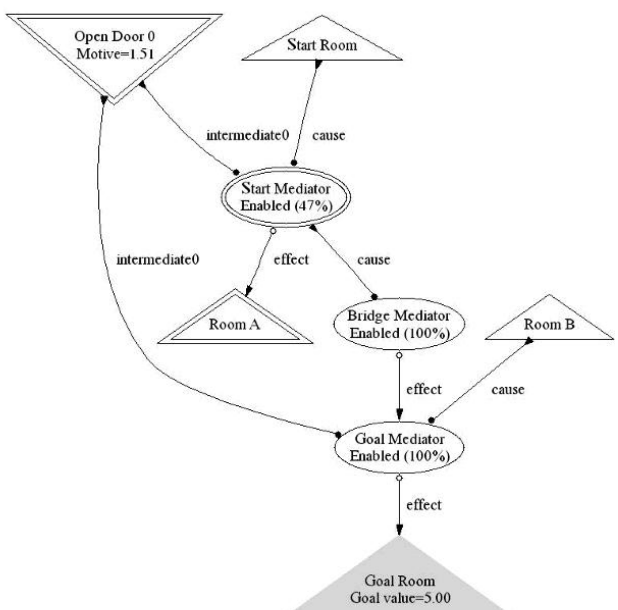

Figure 6 – "Bridge Mediator" at Room A

Figure 6 shows the same set of mediators after a successful attempt to reach room A via door 0, indicated by the firing of the "Open Door 0" motor and the "Room A" receptor neurons. Note that since the "Start Mediator" has also fired, being the cause event of "Bridge Mediator", a wager has boosted the "Goal Mediator" enablement to 100%. In effect, since conditional event of using door 0 worked at the beginning of the maze, it is now a certainty that door 0 will succeed at the end. Note that the state in Figure 6 will persist while the middle maze navigation proceeds independently, serving as a background context whose influence asserts itself later in time.

4. Results

Testing consisted of varying the number of doors appearing in each room and the length of the path in the maze, and measuring the success rate of the learner. Each data point represents the average performance of 10 randomly generated mazes (10 seemed sufficient as there was little variation in results between mazes). The results appear in Figure 7.

The general observation is that the number of doors per room has a significant effect on the success rate. This was not unexpected, since with more choices presented in each room, there are more learned mediators that possibly apply. Most of the errors in these cases arose from the influence of mediators being applied out of context. Many of these mediators were deemed "parasites" because they grow stronger under the guidance of more reliable mediators, until at last able to exert a wrong choice. This diminishes them for a time until a new growth cycle begins.

Generalized pruning techniques proved mostly effective against parasites. However, as a general philosophy, some of them can be thought of as representing possibilities of environmental variation, and as such should not be utterly exterminated. In sum, learning new mediators was quite easy, unlearning poor ones proved difficult.

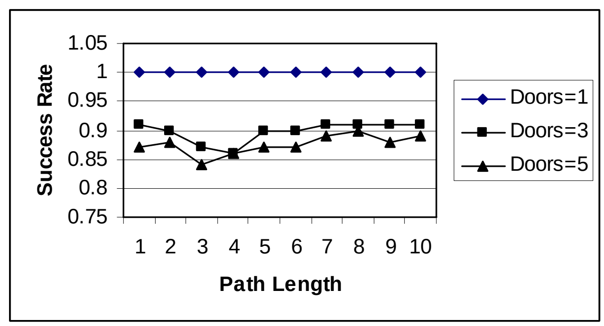

Figure 7 – Maze-learning performance

As an additional observation, the length of the middle maze proved to have little effect on the success rate. As it turned out, a single mediator could be reliably learned to impeccably navigate this portion of the maze.

5. Discussion

One way of looking at the method presented here is as an extension of reinforcement learning [16] to context-related problems. The purpose of reinforcement learning is to learn paths through a state space to goal states. Actions causing state transitions are scored with a utility value according to how well they contribute to goal-seeking. In this sense the utility of a mediator corresponds to its enablement. In prototypical reinforcement learning models, exemplified by Q-Learning [17] and Temporal Difference Learning [18], the state space is a flat Markovian space, necessitating the embedding of context information into state labels, which in turn can result in a proliferation of states. The use of hierarchies is a powerful means of avoiding this proliferation: they provide modularity and reusability.

For example, consider a state transition *S0* → *S1* may exist within context *C0* and *C1* wherein the two contexts affect the transition probability differently. In a flat space, *S0&C0* → *S1&C0* and *S0&C1* → *S1&C1* are needed to express this. Moreover, context hierarchies allow the dynamic linking of cause and effect chains that are not explicity encoded. For example, suppose context *C0* has *S0* → *S1*, and in *C1* has *S1* → *Goal*. The two contexts can be linked through the shared state *S1* to create a goal path. If *S1* were encoded in a flat space as *S1&C0* and *S1&C1* the linkage information would be lost.

As a verification, Q-Learning applied to the maze problem results in the plot shown in Figure 8. As expected, the lack of context information is reflected in the random responses made at the end of the maze. For example, if there are three possible doors the correct one will be chosen one third of the time. It should also be noted that the middle (M) rooms were uniquely marked for Q-Learning, otherwise the performance would have been much worse.

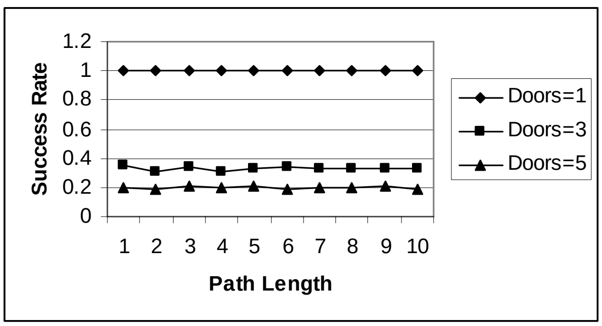

Figure 8 – Q-Learning Maze Performance

The question arises as to how Mona differs from more conventional, e.g. feedforward, artificial neural networks. On an architectural level, Mona also creates new neurons to represent learned relationships, rather than exclusively modifying existing connection weights. As previously noted, the destruction of ineffective neurons is also a necessary function. This raises interesting parallels with the development of animal and human brains which exhibit growth and consolidation phases.

It may be surmised that recurrent networks could be trained to recognize input patterns representing the existence of contexts and to associate these with response sequences. However, the use of recurrent networks to retain temporally distant information is a significant challenge for these networks. As an verification of this as applied to the maze problem, a set of maze trials involving Elman and auto-associative recurrent neural networks built with the Stuttgart Neural Network Simulator (SNNS) [19] proved disappointing. Neither were able to learn more than short two-door mazes.

The most important functional distinction from conventional neural networks is that Mona is a goal-seeker, more like a planner than a pattern classifier. For a goal-seeker, many state path variations may suffice to achieve success. Pattern classifiers, such as feedforward neural networks, can be used to recognize environmental states. In sum, goal-seeking and pattern classification are complementary techniques.

6. Conclusion and Future Work

The described technique allows Mona to successfully learn the given context-related maze task. This represents a significant accomplishment for a connectionist system. Conventional artificial neural networks have primarily focused on pattern classification tasks, yet this is only one of the functions of a brain. The ability to seek out goals in an environment in order to satisfy needs is also of fundamental importance. Furthermore, these two functions may not be as disparate as they seem; nature has apparently built them out of the same components, organized in similar fashions.

For future work, the representation of logical *and* causal conjunctions is scheduled for testing. As an example of a conjunction, a door might not open unless a code is entered and a key is used. Therefore a logical *and* relationship exists between the two causal events. Some other tasks involve the necessity to learn inhibiting influences, an essential aspect of many animal and human behaviors.

The C++ source code and other reference materials are available at:
www.itk.ilstu.edu/faculty/portegys/research/.

References:

[1] P. Bonzon, A Reflexive Proof System for Reasoning in Contexts, in *Proceedings 14th National Conference on Artificial Intelligence* (AAAI 97), Providence, RI, 1997.

[2] R.C. Schank and P.G. Childers, *The Cognitive Computer; On Language, Learning, and Artificial Intelligence*. Addison-Wesley Publishing Company, Inc. 1984.

[3] R. Turner, Context-Mediated Behavior for Intelligent Agents, *International Journal of Human-Computer Studies* special issue on "Using Context in Applications", 1998, vol. 48, no. 3, pp. 307-330.

[4] R. Sun and C.L. Giles, Sequence Learning: From Recognition and Prediction to Sequential Decision Making, *IEEE Intelligent Systems*, 2001.

[5] M. Howard and M. Kahana, A Distributed Representation of Temporal Context, *Journal of Mathematical Psychology*, 2002, 46, 269-299.

[6] M. Hasselmo and J. McClelland, Neural models of memory, *Current Opinion in Neurobiology*, 1999, 9:184–188.

[7] P. Maes and R. Brooks, Learning to Coordinate Behaviors, *AAAI-90*, Boston, MA. 1990, 796-802.

[8] P. Sabes and M. Jordan, Reinforcement Learning by Probability Matching, In *Advances in Neural Information Processing Systems*, 1996, 8.

[9] M. Bodén and J. Wiles, On learning context free and context sensitive languages, *IEEE Transactions on Neural Networks*. 2002, 13(2), pp. 491-493.

[10] M. Steijvers and P. Grunwald, A recurrent network that performs a contextsensitive prediction task. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*. Erlbaum. 1996.

[11] S. Wermter, G. Arevian, and C. Panchev, Recurrent neural network learning for text routing. In *Proceedings of the International Conference on Artificial Neural Networks*, 1999, pages 898-903, Edinburgh, UK.

[12] S. Benson and N. Nilsson, Reacting, Planning and Learning in an Autonomous Agent, *Machine Intelligence 14*, Edited by K.Furukawa, D. Michie, and S. Muggleton. Oxford: Clarendon Press. 1995.

[13] T. Portegys, A Connectionist Model of Motivation, *IJCNN'99 Proceedings.* 1999.

[14] T. Portegys, Goal-Seeking Behavior in a Connectionist Model, *Artificial Intelligence Review*, 2001, 16 (3):225-253.

[15] F. Carpenter, *The Skinner Primer: Behind Freedom and Dignity*. New York: The Free Press, a Division of Macmillan Publishing Company, Inc. 1974.

[16] L. Kaelbling, M. Littman, and A. Moore, Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research.* 1996.

[17] C. Watkins, Learning from Delayed Rewards, Thesis, University of Cambidge,England. 1989.

[18] R. Sutton, Learning to predict by the method of temporal differences. *Machine Learning*, 1988, 3(1):9-44.

[19] Stuttgart Neural Network Simulator, http://www-ra.informatik.uni-tuebingen.de/SNNS/

Appendix A – Sample Training Trial

Maze Dump:
Valid door = 2, Room id=0 mark=0 doors: 1 1 1 goals: 0
Valid door = 1, Room id=1 mark=1 doors: 1 1 1 goals: 0
Valid door = 0, Room id=2 mark=2 doors: 1 1 1 goals: 0
Valid door = 0, Room id=3 mark=2 doors: 1 1 1 goals: 0
Valid door = 1, Room id=4 mark=2 doors: 1 1 1 goals: 0
Valid door = 2, Room id=5 mark=2 doors: 1 1 1 goals: 0
Valid door = 2, Room id=6 mark=2 doors: 1 1 1 goals: 0
Valid door = 2, Room id=7 mark=3 doors: 1 1 1 goals: 0
Valid door = -1, Room id=8 mark=4 doors: 0 0 0 goals: 1

Cycle=0
Room id=0 mark=0 doors: 1 1 1 goals: 0
Response: Door 2

Cycle=1
Room id=1 mark=1 doors: 1 1 1 goals: 0
Response: Door 1

Cycle=2
Room id=2 mark=2 doors: 1 1 1 goals: 0
Response: Door 0

Cycle=3
Room id=3 mark=2 doors: 1 1 1 goals: 0
Response: Door 0

Cycle=4
Room id=4 mark=2 doors: 1 1 1 goals: 0
Response: Door 1

Cycle=5
Room id=5 mark=2 doors: 1 1 1 goals: 0
Response: Door 2

Cycle=6
Room id=6 mark=2 doors: 1 1 1 goals: 0
Response: Door 2

Cycle=7
Room id=7 mark=3 doors: 1 1 1 goals: 0
Response: Door 2

Cycle=8
Room id=8 mark=4 doors: 0 0 0 goals: 1
Response: Wait