Reinforcement Learning in Non markovian Environments

Zubair Baqai

DIAG, Universita di Roma “La Sapienza”, Italy

Figure 1 - Concept of Restraining Bolt

ABSTRACT

In this Research we will be talking about how restraining bolts (Constraints) can be applied to a learning Agent , by which the agent can learn to act in an environment as required without breaking any rules defined by the Restraining bolts . Restraining bolts constraints are specified using Linear temporal logic (LTL) , The idea behind Restraining bolt is to let Agent learn from Non Markovian Reward Decision processes unlike Regular Reinforcement learning where rewards are generated based on only current state and action taken on the state (MDP) .

I. INTRODUCTION

The Concept of Restraining bolt is inspired from Science Fiction , Figure 1 gives an example of what motivated the need of this research . Using restraining bolts on an agents limits its action at each instance given the environment configuration . The observations to Restraining bolt are different than the observations than are received by our Agent (RL agent) . The Model of RL agent is represented as Markov Decision Process (MDP) through Low-level World Features . By low level we mean the observations are Discrete or continous values , where each value represent a configuration of the environment , these set of features are visible and observable to the agent . Unlike regular RL agent , When working with restraining bolts , the

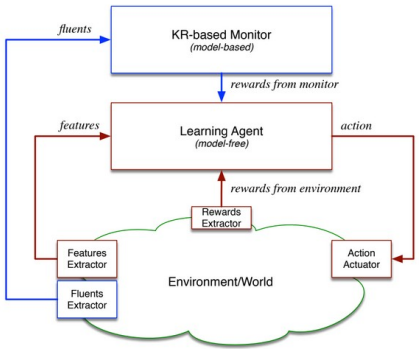

Figure 2 - Architecutre with Restraining bolts intregration

Reward and transition functions are hidden to Agent, this is done as a mean to bring the concept of reward generation by means of Non markovian configuration , by non markovian we mean that rewards are generated based on past transition of states (Traces of states) and not just based on current state and action . Figure 2 demonstrates , how RL algorithm receives the rewards from Restraining bolts (Rewards monitor) on which RL is acting .

II. Defining Restraining Bolts Using LTL formulas

Learning Agent modeled by MDP is defined as following
M = [S,A,Tr,R] , Where 'S' is denotes Current state , 'A' represents the Action taken at state 'S' . Tr is the new state agent arrives at by taking an action , and R represents the reward that is given to the Agent at that instance .

Now we shall introduce how Restraining bolt [RB] is defined .

$$RB = \langle \text{Ł}, \{(\varphi_i, r_i)\}_{i=1}^{m} \rangle$$

L = is the Set of possible fluents configuration (Observations) . These fluents are different than the State 'S' in Learning Agent .

φ_i Represents each restraining specifciation defined with Ltl formula . Sequence of fluents configuration from 'L' are taken and based on satisfiability of the formula a Reward 'Ri' is given to that fluent(state) .

The Agent receives the rewards from Tr from MDP and 'Ri" from Restraining bolts,The RL algorithm is trained as addition of these 2 rewards for each step .

III. Defining Dfa from LTL formula

To define Constraints we use LTL formulas , and following is the syntax to define LTL formulas which is taken from [1] .

$\varphi ::= A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathit{next}\,\varphi \mid \mathit{eventually}\,\varphi \mid \mathit{always}\,\varphi \mid \varphi_1\ \mathit{until}\ \varphi_2$

For example if we need to Define a constraint "Every time the robot opens the door d it closes it immediately after ": , we can write this constraint as following LTL forumla **always**(openDoor → **next** closeDoor) . Now that we have the LTL formula , we need to track of the states if it has to be implemented in Reinforcement learning , to handle that , we introduce the integration of DFA (deterministic finite automaton) , From the Research in [5] , we know that each Ltl formula can be converted into DFA . And based on current state and past traces , we can know the satisfiability of the formula , to continue our previous example , Figure 3 is the converted DFA of the defined LTL formula , which is generated from flloat package which is developed by whiteMech team[3] .

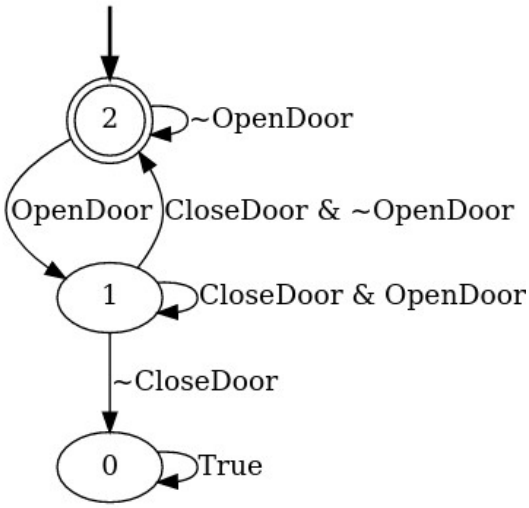

Figure 3 - DFA of defined Ltl formula

IV. Demonstration with example

In this section we present the above theory with an Example by implementing Restraining Constraint in Breakout environment .
A constraint is introduced where apart from Agent learning to break the bricks , we require the agent to learn to break the bricks starting from left to right row . Description of Learning agent and Restraining bolt for this specifications are define below

Learning Agent (LA)
LA features : Paddle Position , ball speed , ball position
LA action : move the paddle [Left ,Right , Nothing]
LA Rewards : reward when a brick is hit

Restraining Bolt (RB) :
RB Fluents / Features : Status of each brick [0 ,1]
RB Ltl Restraining Specification : all the bricks x must be removed before completing any other column y > x

definition of the above Constraint in Ltl formula can be expessed as follow
"!d U(g)" which means Dont Die(Break wrong brick) until all bricks are broken .To implement this constraint in RL , we need to convert it to its corresponding DFA as show in figure 4 . now that restraining bolt is set , we need to store all the past traces as we progress during the episode(States it landed on previous states) . To do that we convert the fluents on current step as a truth values of our Ltl formula . These truth values are saves in our list along with previous once that we got on each step . A reward is given to agent for the action taken by agent at Environment state based on the DFA state it has landed on with our updated trace . In Our example a reward of -0.01 is given when on State 0 [no constraints broken so far] , -10 when we land on state 2 [Constraint is broken] , +10 if current state is 1 [All bricks broken in correct order] .
As discussed earlier , apart from getting rewards from just Constraining bolt , we also get rewards from environment configuration , in our case +5 reward is given each time a brick is broken regardless of the order of it. With this Configuration of our Learning agent and Restraining bolt our Reinforcement learning algorithm will be able to find a policy to break all the bricks in our required specification using the Non Markovian Reward Decision processes

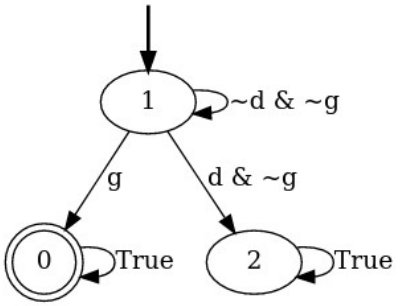

Figure 4 – DFA of Breakout

(NMRDPs) we have defined while acting in MDP environment itself .

V. Experiments and Observations

As an experiment , we defined the Breakout environment with a 3 by 3 grid (3 rows and 3 columns of bricks) , with an option to break the bricks using the Fire option instead of breaking the brick with the ball . Furthermore we used DQN as the reinforcement learning Algorithm with default hyperparameters . The agent was able to converge the model in 500 iterations while able to achieve an average of 35 rewards per 100 episodes which means it successfully broke all the bricks in correct . The results of trained model can be viewed on following link [4] . Another interesting observation that was found is , defining similar constraint without the use of Non markovian rewards were not converging as the problem statement were not solvable using MDP , nonetheless after self engineering the State space and adding observation of previous timesteps manually were able to solve the problem , but it was quite complicated and convergence was not as fast .

VI. Conclusions

In this Research , We discussed how restraining bolts can be defined by writing Linear time logic(LTL) formulas defining the constraints that we need to introduce to the agent , also this research discusses about how Rewards are generated from Reward Monitor by evaluating the landing state

of the complete trace of that episode given the current action on the current state . Furthermore we explained how Restraining bolts specifications are independent of Learning agent , and how agent without the underlying knowledge of restraining bolt can leverage by its encoded state and the rewards and solve the problems which are not possible in Markovian decision process in its true nature . Apart from the basic example of breakout there are many real world applications that will benefit from the research happening in this domain , for example developing an Agent(robot) which can serve drinks to customer successfully while restraining itself to not serve underage or minors

VII. References

1. https://drive.google.com/drive/folders/1ayjo5KWJJhZLD0s0VLV8xZJljMQqmwG5
2. https://www.ijcai.org/Proceedings/15/Papers/223.pdf
3. https://flloat.herokuapp.com/
4. https://drive.google.com/file/d/1QJ2RuX2wlWKgj1sfnHkwn5DJA6WA4P5_/view?usp=sharing
5. *Giacomo, Giuseppe De, Luca Iocchi, Marco Favorito and Fabio Patrizi. "Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications." ICAPS (2019).*
6. *Brafman, R.I., Giacomo, G.D., & Patrizi, F. (2018). LTLf/LDLf Non-Markovian Rewards. AAAI.*