Translating Bolts: LTL_f and STRIPS Based Restraining Bolts on the Taxi-v3 Environment

- Claudia Melis Tonti: 1888489
- Lorenzo Papa: 1699806
- Nicolas Zaccaria: 1904366
- Roberto Gallotta: 1890251

I

Temporal Logic & Deterministic Finite Automaton

Temporal Logic (TL)

Logic language that offers some time-related rules and operators
to deal with propositions in which truth varies with time

Semantic

- Graph representation: Kripke Model
- Facts can be true or false in more situations called possible worlds
- The set of possible worlds lies on a timeline

Syntax

- Not, And, Or connectives $\neg, \wedge, \vee$
- Necessarily operator $\Box\varphi$ states that φ is true in all the possible worlds
- Possibly operator $\Diamond\varphi$ states that φ is true or will be true in all the possible worlds

Linear Temporal Logic (LTL)

An extension of TL with:

Semantic

- Describes the evolution over time as a sequence of time-points: trace
- Formulas are evaluated over a trace
- As standard, LTL traces are infinite
- LTL can be also used over finite traces (LTL_f) with a lower expressive power

Syntax

- Globally $\mathcal{G}\varphi$: states that φ is true now and in all the future time points
- Finally $\mathcal{F}\varphi$: states that φ is true now or it will at some time point in the future
- Next $\mathcal{X}\varphi$: states that φ will be true in the next time point
- Until $\varphi_1\mathcal{U}\varphi_2$: states that φ_1 is true until φ_2 is true

Deterministic Finite Automaton (DFA)

Finite state machine that takes a string of symbols as input and, running through a sequence of states, may accept or reject it.

May be represented as a graph:

$$\mathcal{DFA} = (\Sigma, S, s^0, \rho, F)$$

with:

- $\sum$ a finite and non-empty alphabet, where a word is an element of $\sum^*$
- S a finite and non-empty set of states
- $s^0 \in S$ the initial state
- $F \subseteq S$ the set of accepting states
- $\rho : S \times \Sigma \to S$ a transition function

II

Reinforcement Learning

Reinforcement Learning

improve the ability of performing some task with experience: **learning**

Given a set of States, Actions and Rewards:

$$D = \{(\langle s_0, a_1, r_1, s_1, ..., a_n, r_n, s_n \rangle_i)_{i=1}^{n}\}$$

Learn an optimal behavior function:

$$\pi(a, s) = Pr(a_i = a | s_i = s)$$

Reinforcement Learning

modeled as a time-discrete system: Markov Decision Process

$$\mathcal{MDP} = \langle S, A, \delta, r \rangle$$

with:

- S a finite set of states
- A a finite set of actions
- $\delta : S \times A \rightarrow S$ a transition function
- $r : S \times A \times S \rightarrow \mathbb{R}$ a reward function

Reinforcement Learning

Aims to maximize the expected cumulative discounted reward:

$$V^{\pi}(s_1) = E[\overline{r}_1 + \gamma\overline{r}_2 + \gamma^2\overline{r}_3 + ...]$$

where $\gamma \in [0, 1]$ is the discount factor

To obtain the optimal policy:

$$\pi^* \equiv argmax_{\pi} V^{\pi}(s), \forall s \in S$$

Taxi-v3

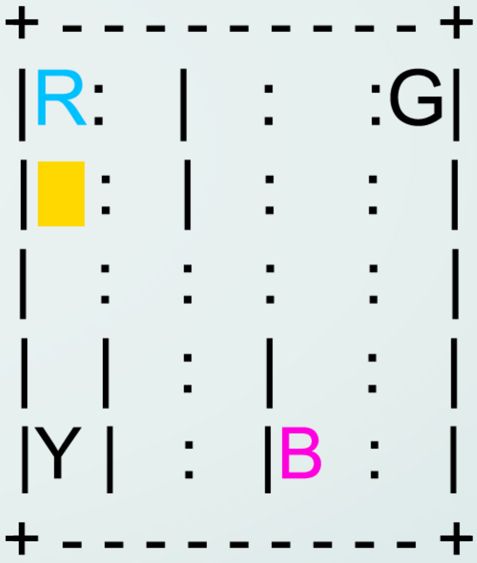
Goal: pick up the passenger and drop him off at the destination

Actions:

- Move North
- Move South
- Pick up
- Move West
- Move East
- Drop off

Rewards:

- Default per-step -1
- Illegal Pick up and Drop off -10
- Successful Drop off +20



III

Restraining Bolts & Automata as Reward Shaping

Temporal Goals

IDEA:
Make the agent learn to reach a set of temporal goals until it reaches the final goal.

THE PROBLEM IT FIXES:
The RL algorithm behavior may degenerate into a random walk

Restraining Bolts

WHAT IS IT:
A restraining bolt restricts the agent's actions limiting them to a set of wanted behaviors.

Agent's world representation

Restraining Bolt's world representation

Reinforcement Learning

$$M_{ag} = \langle S, A, T_{r_{ag}}, R_{ag} \rangle$$

$$RB = \langle L,\ \{(\phi_i,\ r_i)\}_{i=1}^{m} \rangle$$

Restraining Bolts

THE RL PROBLEM:

$$M_{ag}^{rb} = \langle M_{ag},\ RB \rangle$$

THE SOLUTION:

$$\bar{\rho}:\ (Q_1 \times \cdots \times Q_m \times S)^* \to A$$

- For each Φi compute the DFA AΦi
- Do RL on

$$M_{ag}^{q} = \langle Q_1 \times \cdots \times Q_m \times S,\ A,\ Tr'_{ag},\ R'_{ag} \rangle$$

$$R'_{ag}(q_1, \ldots,\ q_m,\ s,\ a,\ q'_1, \ldots,\ q'_m,\ s') = \sum_{i:q'_i \in F_i} r_i + R_{ag}(s,\ a,\ s')$$

Automata as Reward Shaping

Reward Machine <S,A,P,L> **= Mealy Machine** <Q, q_0, Σ, R, δ, ρ> where:

$$\Sigma = 2^P$$

$$L : S \times A \times S \to 2^P$$

The idea is to approximate the cumulative discounted reward in any RM state by using the RM itself as a MDP.

The resulting reward shaping is: $\alpha(r(s,\ a,\ s') + \gamma \cdot \max \tilde{q}(s',\ a'))$

IV

Reward Shaping & STRIPS

Reward Shaping

In Reinforcement Learning: Consisting of supplying additional rewards to a learning agent

Pros:

- Helps the agent in achieving the optimal policy faster
- Guide the learning process more efficiently

Final Reward = MDP Reward + Reward Shaping

Reward Shaping

IDEA:

Engineering a Reward Function to provide more frequent feedback on appropriate behaviours

RELATED WORKS:

Policy invariance under reward transformations: Theory and application to reward shaping [1]

↓

Potential-Based Reward Shaping
(PBRS)

[1] Andrew Y Ng, D.H., Russell, S.: Policy invariance under reward transformations:Theory and application to reward shaping (1999)

Reward Shaping

Markov Decision Process: M = (S, A, δ, R)

R = R(s, a, s′)

Reward Shaping function: F(s, a, s′)

$$R'(s, a, s') = R(s, a, s') + F(s, a, s')$$

[1] Andrew Y Ng, D.H., Russell, S.: Policy invariance under reward transformations:Theory and application to reward shaping (1999)

STRIPS

STanford Research Institute Problem Solver:

A problem solver that aims to find the optimal sequence of operators

Initial State $\longrightarrow$ **Final State (goal)**

Problem Space:

$$\boldsymbol{\Pi} = \langle P, O, I, G \rangle \longrightarrow \mathrm{cost}(\boldsymbol{\pi}_{opt}) < \mathrm{cost}(\boldsymbol{\pi}')$$

STRIPS to LTLf

Translate from STRIPS formalism to an equivalent LTLf formula:

1. Define the domain with a set of fluents
2. From STRIPS to LTLf [2]
   2.1. Define each action with a precondition
   2.2. Encode the Initial set of fluents, as TRUE states
   2.3. Encode the Final set of fluents that eventually hold

STRIPS → **LTLf** → **DFA** → **Automata**

[2] De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13, AAAI Press (2013) 854–860

V

Our experiments

Overview of experiments and discussion of results

LTL_f Temporal Goals

Increasingly complex temporal goals described by LTL_f formulas:

- Base environment goal: *a U (b U c)*
- Pass through center: *a U (b U (c U d))*
- Pass through 1 corner: *a U (b U (c U d))*
- Pass through 2 corners: *a U (b U (c U (d U e)))*
- Pass through 3 corners: *a U (b U (c U (d U (e U f))))*

Generate DFA from LTL_f formulas to track Temporal Goal.

Meaning of fluents changes according to specific environment initialization.

We test both with and without Reward Shaping

STRIPS Temporal Goals

Define the base environment goal as STRIPS problem.

Convert STRIPS to LTL_f and then to DFA.

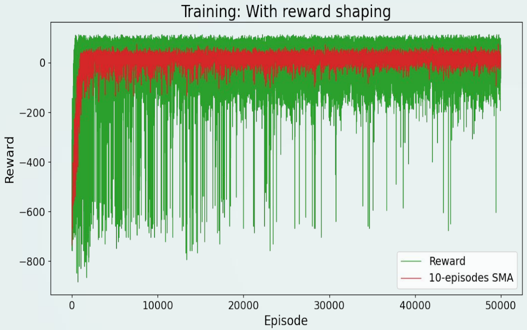
Ensure no *don't care* and terminal state in formula.

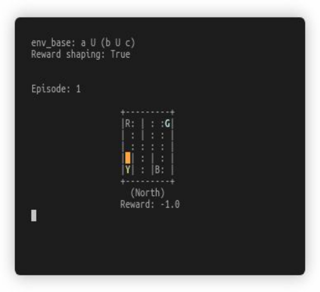
Introduction of Step Controller in the training and testing loop to compensate loopless DFA states.

Results

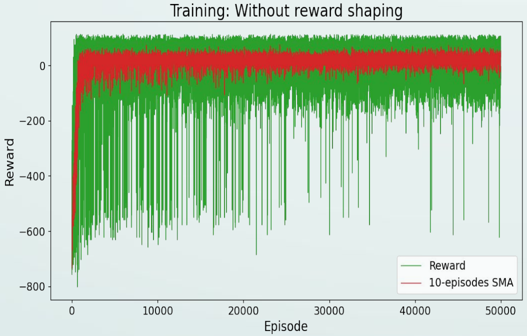
Base environment goal

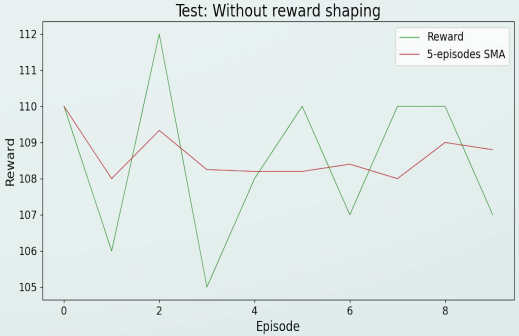
With Reward Shaping

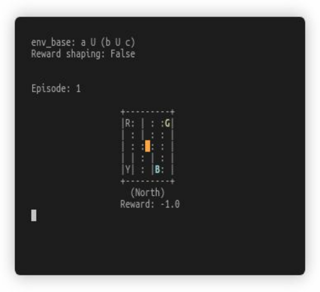




Without Reward Shaping



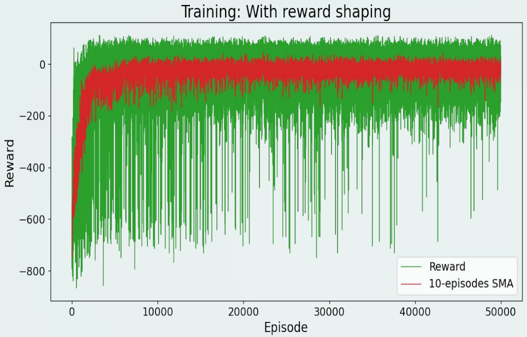


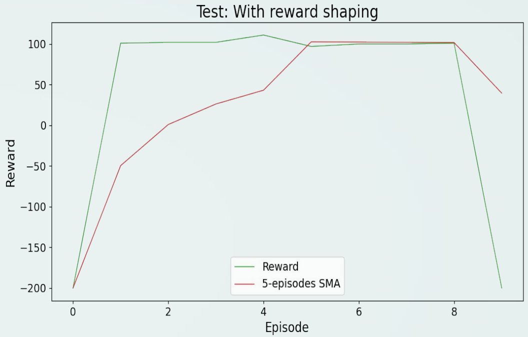


Results

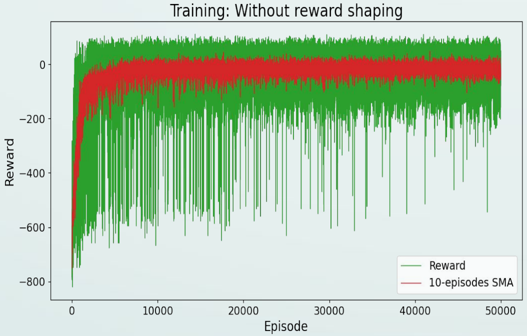
Through 1 checkpoint

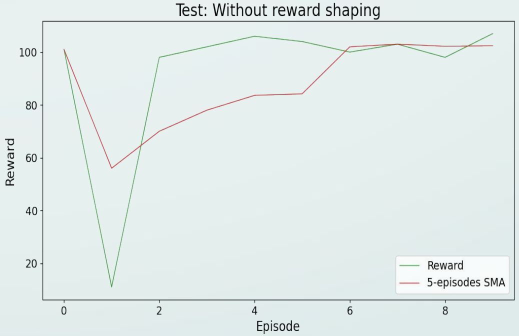
With Reward Shaping





Without Reward Shaping

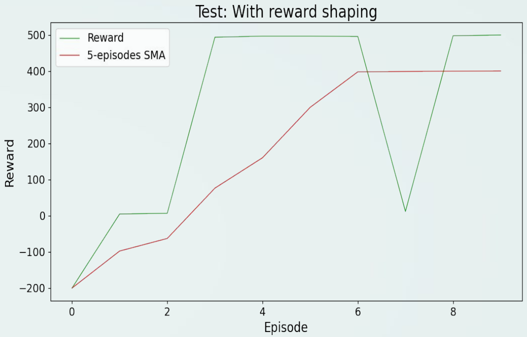


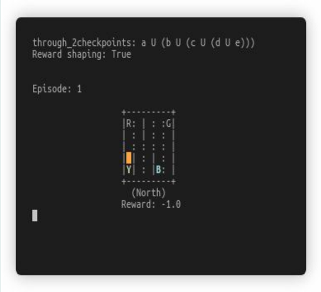


Results

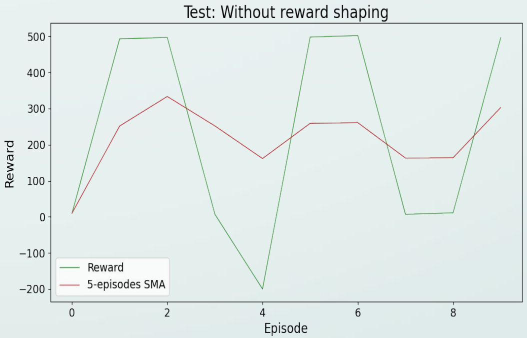
Through 2 checkpoints

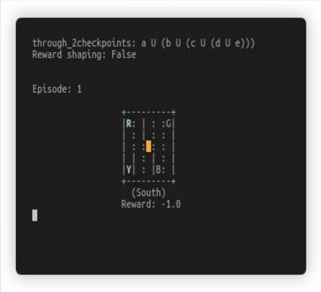
With Reward Shaping





Without Reward Shaping

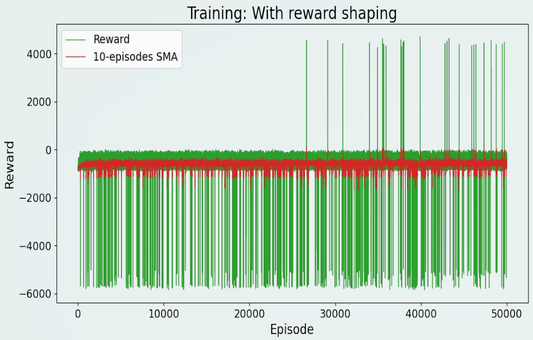


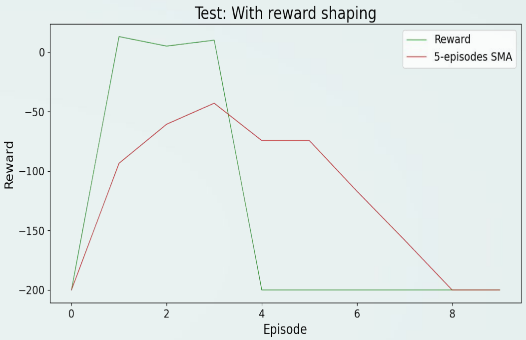


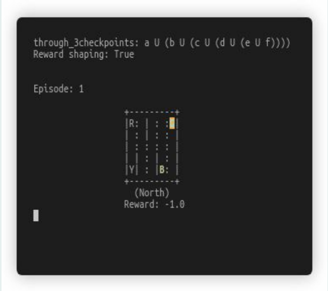
Results

Through 3 checkpoints

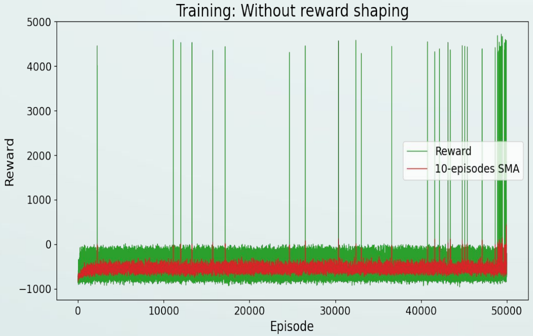
With Reward Shaping

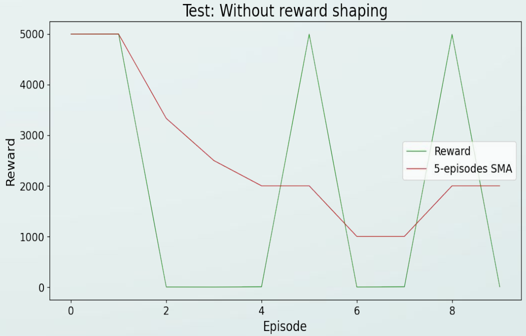


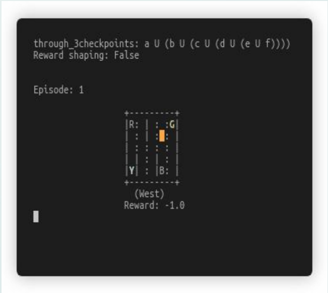




Without Reward Shaping



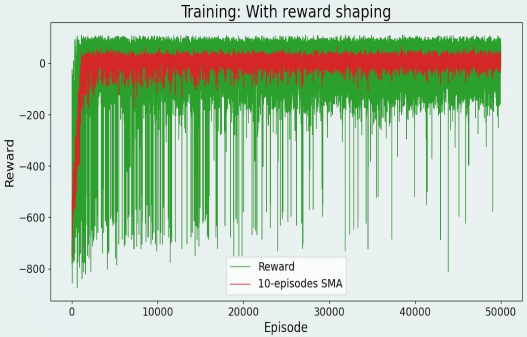


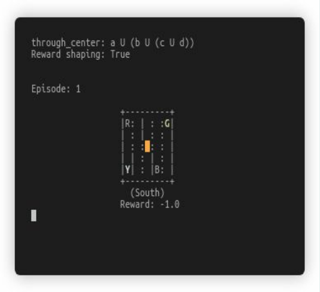


Results

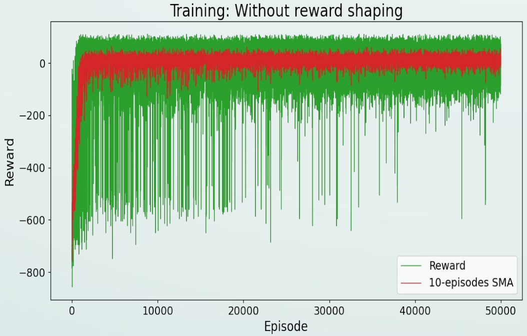
Through center

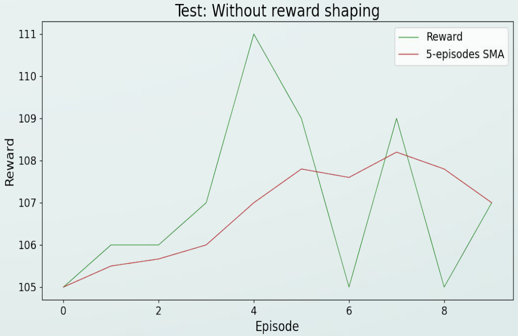
With Reward Shaping

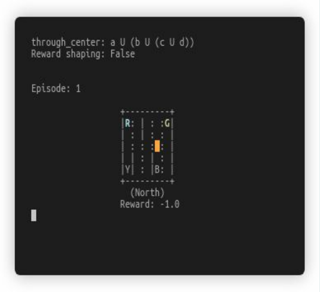




Without Reward Shaping

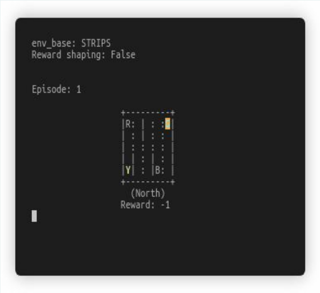


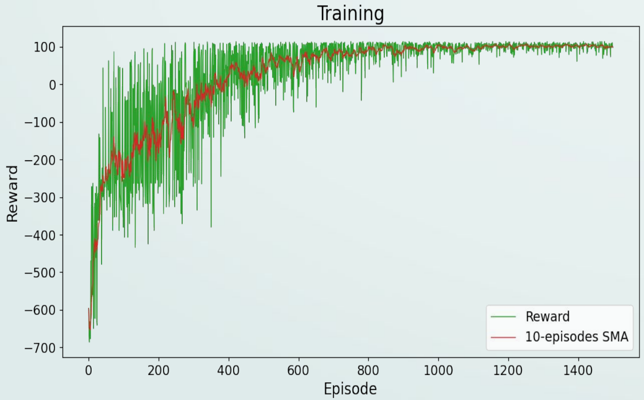




Results

STRIPS

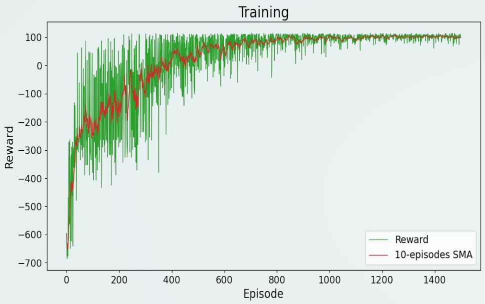




Results

Overhead with STRIPS?

STRIPS



No Temporal Goal