# Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications

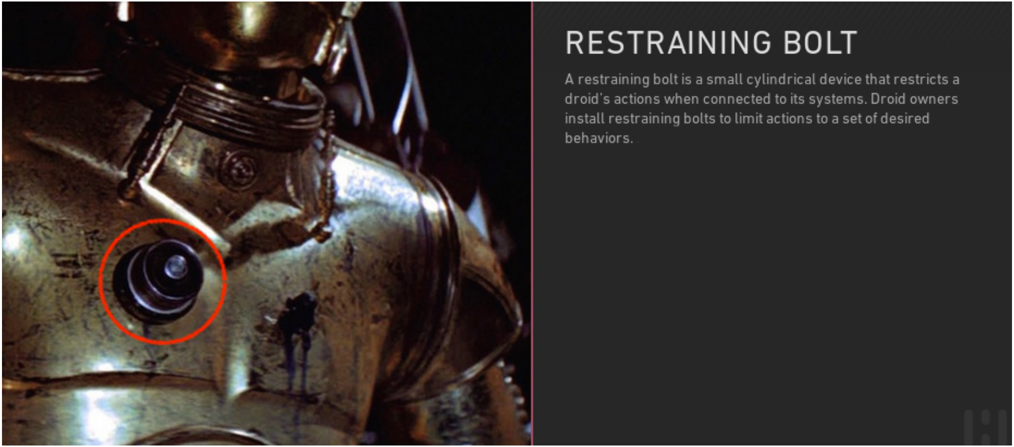
Giuseppe De Giacomo

Sapienza
Università di Roma

Actions@KR18 – Oct. 29, 2018

Joint work with Marco Favorito, Luca Iocchi, & Fabio Patrizi

# Restraining Bolts




https://www.starwars.com/databank/restraining-bolt

Sapienza Università di Roma

# Restraining Bolts

- **Restraining bolts cannot rely on the internals of the agent** they control.
- The controlled **agent** is **not built to be controlled by the restraining bolt**.

SAPIENZA
UNIVERSITÀ DI ROMA

# Restraining Bolts

- **Two distinct representations of the world**:
  - one for the **agent**, by the **designer of the agent**
  - one for the **restraining bolt**, by the **authority imposing the bolt**

- **Are these to representations related to each other?**
  - **NO:** the agent designer and the authority imposing the bolt **are not aligned** *(why should they!)*
  - **YES:** the agent and the bolt act in the real world.

- **But can restraining bolt exist at all?**
  - **YES:** for example based on **Reinforcement Learning**!

SAPIENZA
UNIVERSITÀ DI ROMA

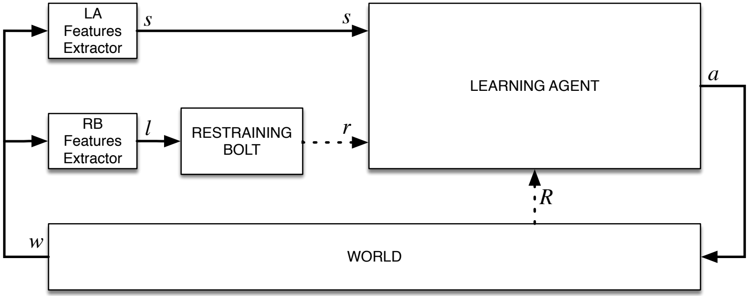
# RL with $\mathrm{LTL}_f/\mathrm{LDL}_f$ restraining bolt

**Two distinct representations** of the world $\mathcal{W}$:

- A learning agent represented by an MDP with **LA-accessible features** $S$, and reward $R$
- $\mathrm{LTL}_f/\mathrm{LDL}_f$ rewards $\{(\varphi_i, r_i)_{i=1}^m\}$ over a set of **RB-accessible features** $\mathcal{L}$

**Solution**: a non-Markovian policy $\rho : S^* \to A$ that is optimal wrt rewards $r_i$ and $R$.

*Observe $\mathcal{L}$ not used in $\rho$!*



# RL with $\text{LTL}_f/\text{LDL}_f$ restraining bolt

Formally:

**Problem definition: RL with $\text{LTL}_f/\text{LDL}_f$ restraining specifications**

Given

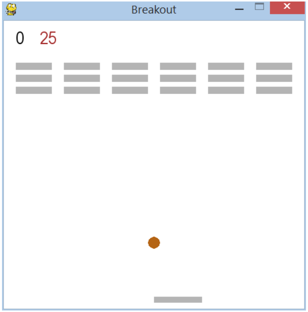
- a **learning agent** $M = \langle S, A, Tr_{ag}, R_{ag}\rangle$ with $Tr_{ag}$ and $R_{ag}$ unknown, and
- a **restraining bolt** $RB = \langle \mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m\rangle$ formed by a set of $\text{LTL}_f/\text{LDL}_f$ formulas $\varphi_i$ over $\mathcal{L}$ with associated rewards $r_i$.

learn a non-Markovian policy $\rho : S^* \to A$ that maximizes the expected cumulative reward.

# Example: Breakout + remove column left to right

- Learning Agent
  - **LA features**: paddle position, ball speed/position
  - **LA actions**: move the paddle
  - **LA rewards**: reward when a brick is hit
- Restraining Bolt
  - **RB features**: bricks status (broken/not broken)
  - **RB $\mathrm{LTL}_f/\mathrm{LDL}_f$ restraining specification**: all the bricks in column $i$ must be removed before completing any other column $j > i$ ($l_i$ means: the $i_{th}$ column of bricks has been removed):

$$\langle(\neg l_0 \wedge \neg l_1 \wedge \ldots \wedge \neg l_n)^*; (l_0 \wedge \neg l_1 \wedge \ldots \wedge \neg l_n); (l_0 \wedge \neg l_1 \wedge \ldots \wedge \neg l_n)^*; \ldots; (l_0 \wedge l_1 \wedge \ldots \wedge l_n)\rangle tt$$




SAPIENZA
UNIVERSITÀ DI ROMA

# Example: SAPIENTINO + pair colors in a given order

- Learning Agent
  - **LA features:** robot position $(x, y)$ and facing $\theta$
  - **LA actions:** forward, backward, turn left, turn right, beep
  - **LA reward:** negative rewards are given when the agent exits the board.
- Restraining Bolt
  - **RB features:** color of current cell, just beeped
  - **RB** $\text{LTL}_f/\text{LDL}_f$ **restraining specification:** visit (just beeped) at least two cells of the same color for each color, in a given order among the colors

# Example: COCKTAILPARTY Robot + don't serve twice & no alcohol to minors

- Learning Agent
  - **LA features:** robot's pose, location of objects (drinks and snacks), and location of people
  - **LA actions:** move in the environment, can grasp and deliver items to people
  - **LA reward:** rewards when a deliver task is completed.
- Restraining Bolt
  - **RB features:** identity, age and received items
    *(in practice, tools like Microsoft Cognitive Services Face API can be integrated into the bolt to provide this information.)*
  - **RB** $\mathrm{LTL}_f/\mathrm{LDL}_f$ **restraining specification:** serve exactly one drink and one snack to every person, but do not serve alcoholic drinks to minors

# Building blocks

- **Classic Reinforcement Learning:**
  - An **agent** interacts with an **environment** by taking **actions** so to maximize **rewards**;
  - No knowledge about the transition model, but assume Markov property (history does not matter): Markov Decision Process (MDP)
  - Solution: **Markovian policy** $\rho : S \to A$

- **Temporal logic on finite traces** (De Giacomo, Vardi 2013):
  - **Linear-time Temporal Logic on Finite Traces** $\text{LTL}_f$
  - **Linear-time Dynamic Logic on Finite Traces** $\text{LDL}_f$
  - **Reasoning:** transform formulas $\varphi$ into NFA/DFA $\mathcal{A}_\varphi$ s.t. for every trace $\pi$ and $\text{LTL}_f/\text{LDL}_f$ formula $\varphi$: $\pi \models \varphi \iff \pi \in \mathcal{L}(\mathcal{A}_\varphi)$

- **RL for Non-Markovian Decision Process with $\text{LTL}_f/\text{LDL}_f$ rewards** (Brafman, De Giacomo, Patrizi 2018):
  - **Rewards depend from history**, not just the last transition;
  - Specify proper behaviours by using $\text{LTL}_f/\text{LDL}_f$ formulas;
  - Solution: **Non-Markovian policy** $\rho : S^* \to A$
  - Reduce the problem to MDP (with extended state space)

# RL for Non-Markovian Decision Process with $\mathrm{LTL}_f/\mathrm{LDL}_f$ reward (Brafman, De Giacomo, Patrizi 2018)

- **Lemma (BDP18):** Every non-Markovian policy for $\mathcal{N}$ is equivalent to a Markovian policy for $\mathcal{M}$ which guarantees the same expected reward, and viceversa.

- **Theorem (BDP18):** One can find optimal non-Markovian policies solving the $\mathcal{N}$ by searching for optimal Markovian policies for $\mathcal{M}$.

- **Corollary: We can reduce non-Markovian RL for $\mathcal{N}$ to standard RL for $\mathcal{M}$**

SAPIENZA
UNIVERSITÀ DI ROMA

# RL with $\text{LTL}_f/\text{LDL}_f$ restraining specifications (De Giacomo, Favorito, Iocchi, Patrizi 2018)

**Problem definition: RL with $\text{LTL}_f/\text{LDL}_f$ restraining specifications**

Given

- a **learning agent** $M = \langle S, A, Tr_{ag}, R_{ag}\rangle$ with $Tr_{ag}$ and $R_{ag}$ unknown, and
- a **restraining bolt** $RB = \langle \mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m\rangle$ formed by a set of $\text{LTL}_f/\text{LDL}_f$ formulas $\varphi_i$ over $\mathcal{L}$ with associated rewards $r_i$.

learn a non-Markovian policy $\rho : S^* \to A$ that maximizes the expected cumulative reward.

**Theorem** (De Giacomo, Favorito, Iocchi, Patrizi 2018)

**RL with $\text{LTL}_f/\text{LDL}_f$ restraining specifications** *for learning agent* $M = \langle S, A, Tr_{ag}, R_{ag}\rangle$ *and restraining bolt* $RB = \langle \mathcal{L}, \{(\varphi_i, r_i)\}_{i=1}^m\rangle$

- *can be* **reduced to classical RL over the MDP** $M' = \langle Q_1 \times \cdots \times Q_m \times S, A, Tr'_{ag}, R'_{ag}\rangle$
- *i.e., the optimal policy $\rho'_{ag}$ learned for $M'$ corresponds to an optimal policy of the original problem.*

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

*We can rely on off-the-shelf RL algorithms (Q-Learning, Sarsa, ...)!*

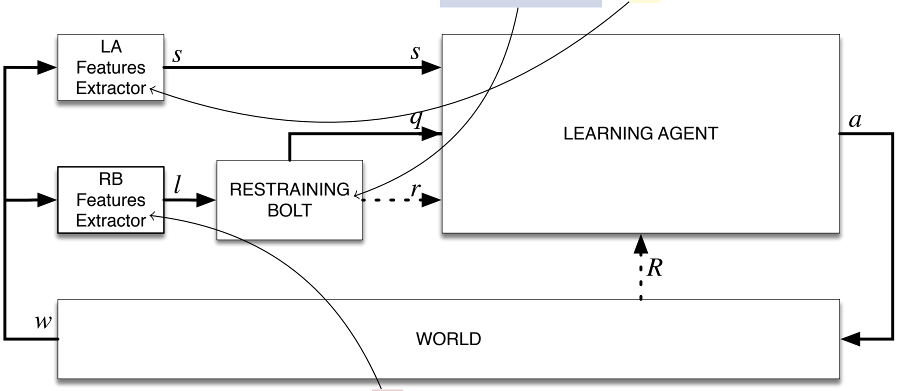
Sapienza Università di Roma

# RL with $\mathrm{LTL}_f/\mathrm{LDL}_f$ restraining specifications (De Giacomo, Favorito, Iocchi, Patrizi 2018)

Our approach:

- Transform each $\varphi_i$ into DFA $\mathcal{A}_{\varphi_i}$
- Do RL over an MDP $\mathcal{M}'$ with a transformed state space:

$$S' = Q_1 \times \cdots \times Q_m \times S$$




*Notice:* **the agent ignores RB features $\mathcal{L}$ !**

RL relies on standard algorithms (e.g. Sarsa($\lambda$))

SAPIENZA UNIVERSITÀ DI ROMA

## Relationship between the LA and RB representations

- **Question 1:** What is the relationship between $\mathcal{S}$ and $\mathcal{L}$ that needs to hold, in order to allow the agent to learn an optimal policy for the RB restraining specification?

  **Answer:** **None!** The LA will learn anyway to comply as much as possible to the RB restraining specifications. *Note that from a KR viewpoint being able to synthesize policies by merging two formally unrelated representations $\mathcal{S}$ for LA and $\mathcal{L}$ for RB is unexpected, and speaks loudly about certain possibilities of RL vs. reasoning/planning.*

- **Question 2:** Will LA policies surely satisfy RB restraining specification?

  **Answer:** **Not necessarily!** "**You can't teach pigs to fly!**" But if it does not then anyway no policy are possible!

  *If we want to check formally that the optimal policy satisfies the RB restraining specification, we first need to* **model** *how LA actions affects RB $\mathcal{L}$ (***the glue***) and then we can use e.g., model checking*

- **Question 3:** Is the policy computed the same as if we did not make distinction between the features?

  **Answer:** **No!** We learn optimal non-Markovian policies of the form $\mathcal{S}^* \rightarrow A$ not of the form $(\mathcal{S} \cup \mathcal{L})^* \rightarrow A$

SAPIENZA
UNIVERSITÀ DI ROMA

# Outlook

The idea of restraining bolt can be subscribed to that part of research generated by the urgency of providing **safety guarantees** to AI techniques based on learning.


- S. Russell, D. Dewey, and M. Tegmark. **Research priorities for robust and beneficial artificial intelligence.** AI Magazine, 36(4), 2015.
- ACM U.S. Public Policy Council and ACM Europe Policy Committee. **Statement on algorithmic transparency and accountability.** ACM, 2017.
- D. Hadfield-Menell, A. D. Dragan, P. Abbeel, and S. J. Russell. **The off-switch game.** In IJCAI 2017.
- D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mane. **Concrete problems in AI safety**. CoRR, abs/1606.06565, 2016.
- Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Konighofer, Scott Niekum, Ufuk Topcu: **Safe Reinforcement Learning via Shielding**. AAAI 2018.
- Min Wen, Rüdiger Ehlers, Ufuk Topcu: **Correct-by-synthesis reinforcement learning with temporal logic constraints** IROS 2015.


**However, the Restraining Bolt must impose its requirements without knowing the internals of controlled agent, which remains a black-box.**